



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και ανάπτυξη διαδραστικού παιχνιδιού
εκπαιδευτικού σκοπού για παιδιά,
με χρήση της πλατφόρμας Unity 3D,
για περιβάλλοντα σταθερών υπολογιστών
και έξυπνων κινητών συσκευών**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

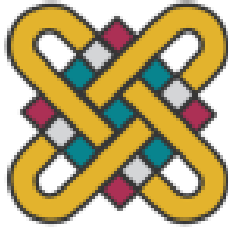
ΤΟΥ

Ρόμπου Ευστράτιου

(ΑΕΜ: 2910)

Επιβλέπων: κ. Νικολάου Σπυρίδων, Λέκτορας

Καστοριά Μάϊος - 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και ανάπτυξη διαδραστικού παιχνιδιού
εκπαιδευτικού σκοπού για παιδιά,
με χρήση της πλατφόρμας Unity 3D,
για περιβάλλοντα σταθερών υπολογιστών
και έξυπνων κινητών συσκευών**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Ρόμπου Ευστράτιου

(ΑΕΜ: 2910)

Επιβλέπων: κ. Νικολάου Σπυρίδων, Λέκτορας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29^η Μαΐου 2023

Δημήτριος Ι. Βέργαδος
Επίκουρος Καθηγητής

Γεώργιος Σίσιας
Διδάσκων με Σύμβαση

Σπυρίδων Νικολάου
Λέκτορας

Καστοριά Μάϊος – 2023

Copyright © 2023 – ΡΟΜΠΟΣ ΕΥΣΤΡΑΤΙΟΣ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Είμαι ευγνώμων για την υπομονή και συμπαράσταση που έδειξε η γυναίκα μου κατά την δημιουργία αυτής της πτυχιακής εργασίας. Επίσης θέλω να ευχαριστήσω των επιβλέπων καθηγητή της παρούσας πτυχιακής εργασίας κ. Νικολάου Σπυρίδων ως προς την καθοδήγηση για την υλοποίηση της. Τέλος θέλω να ευχαριστήσω όλους τους καθηγητές και καθηγήτριες του τμήματος πληροφορικής του Πανεπιστημίου Δυτικής Μακεδονίας για το δημιουργικό χρόνο που πέρασα μαζί τους γιατί με έκαναν να νοιώθω πως ήταν το δεύτερό μου σπίτι.

Περίληψη

Η παρούσα πτυχιακή επικεντρώνεται στο σχεδιασμό και στην ανάπτυξη ενός διαδραστικού , εκπαιδευτικού ηλεκτρονικού παιχνιδιού για παιδιά με χρήση της μηχανής παιχνιδιών Unity 3D. Στόχος είναι να δημιουργηθεί ένα παιχνίδι που παρέχει μια διασκεδαστική και εκπαιδευτική εμπειρία για παιδιά σε σταθερούς υπολογιστές και έξυπνες κινητές συσκευές, χρησιμοποιώντας σύγχρονες μεθόδους μάθησης όπως το gamification. Η ανάπτυξη αυτού του παιχνιδιού έγινε για να διδάξει στα παιδιά βασικές γνώσεις , να οξύνει τις αισθήσεις τους και να τα κάνει να διασκεδάσουν , χρησιμοποιώντας μια ποικιλία διαδραστικών δραστηριοτήτων, παζλ και προκλήσεων. Το παιχνίδι περιλαμβάνει μια εύκολη στη χρήση διεπαφή ([UI](#)) , πολύχρωμα γραφικά με έντονα χρώματα και ηχητικά εφέ για να δημιουργήσει μια καθηλωτική και διασκεδαστική εμπειρία ([UE](#)). Η διαδικασία ανάπτυξης περιλαμβάνει τη χρήση του visual studio [IDE](#) για την δημιουργία του κώδικα στην γλώσσα προγραμματισμού C# και το photoshop για τη δημιουργία των ψηφιακών στοιχείων του παιχνιδιού. Η τελική εφαρμογή είναι ένα διασκεδαστικό και επιμορφωτικό παιχνίδι, σχεδιασμένο για να κατακτήσει και να διεγείρει το μυαλό των παιδιών. Η διατριβή παρέχει επίσης μια επισκόπηση στις μηχανές παιχνιδιών, μια πιο σχολαστική ανάλυση στην Unity 3D, την διαδικασία σχεδιασμού του διαδραστικού εκπαιδευτικού παιχνιδιού και του δυναμικού αντίκτυπου τέτοιων παιχνιδιών στη μάθηση και την ανάπτυξη των παιδιών.

Λέξεις Κλειδιά: Σχεδίαση Εφαρμογής, Ανάπτυξη εφαρμογής, Διαδραστικό εκπαιδευτικό παιχνίδι, Ηλεκτρονικό παιχνίδι, Παιχνίδι για παιδιά, Unity 3D, Μηχανή παιχνιδιών, Σταθεροί υπολογιστές, Έξυπνες κινητές συσκευές

Abstract

This thesis focuses on designing and developing an interactive, educational video game for children using the Unity 3D game engine. The goal is to create a game that provides a fun and educational experience for children on desktop computers and smart mobile devices using modern learning methods like gamification. This game's development was to teach children basic knowledge, sharpen their senses, and make them have fun, using a variety of interactive activities, puzzles, and challenges. The game includes an easy-to-use interface (UI), colorful graphics with bright colors, and sound effects to create an engaging and entertaining experience (UE). The development process includes the use of visual studio IDE to create the code in the C# programming language and Photoshop to create the game's digital elements (assets). The final application is an entertaining and educational game designed to conquer and stimulate children's minds. The thesis also provides an overview of game engines, a more thorough analysis of Unity 3D, the process of designing an interactive educational game, and the potential impact of such games on children's learning and development.

Key Words: App Design, App Development, Interactive educational game, Video game, Game for children, Unity 3D, Game engine, Desktop computers, Smart mobile devices

Πίνακας Περιεχομένων

| | |
|--|-----|
| Περίληψη | i |
| Abstract | ii |
| Πίνακας Περιεχομένων | iii |
| Λίστα Εικόνων | vi |
| Λίστα Πινάκων | x |
| 1. Εισαγωγή | 1 |
| 2. Τύποι εκπαιδευτικών εφαρμογών | 3 |
| 2.1 Διάφοροι τύποι εκπαιδευτικών εφαρμογών[18] | 5 |
| 2.1.1 Εκπαιδευτικές εφαρμογές πολλαπλών θεμάτων | 5 |
| 2.1.2 Εκπαιδευτικές εφαρμογές συγκεκριμένου θέματος | 6 |
| 2.1.3 Υποστηρικτικές εφαρμογές (Supportive Apps)..... | 6 |
| 2.1.4 Εκπαιδευτικά παιχνίδια (Educational Games)..... | 6 |
| 2.1.5 Εφαρμογές διαχείρισης μάθησης (Learning Management Apps)..... | 6 |
| 2.1.6 Εκπαιδευτικές εφαρμογές για παιδιά προσχολικής ηλικίας και νήπια | 7 |
| 2.2 Εκπαιδευτικές εφαρμογές στο εμπόριο | 7 |
| 3. Μηχανές Παιχνιδιών | 10 |
| 3.1 Unity [4][22]..... | 11 |
| 3.1.1 Unity Features..... | 12 |
| 3.2 Unreal [9][22]..... | 13 |
| 3.2.1 Unreal Pipeline Integration..... | 14 |
| 3.2.2 Κατασκευή Κόσμων | 15 |
| 3.2.3 Characters and Animation | 18 |
| 3.2.4 Rendering, Lighting and Materials | 21 |
| 3.2.5 Εξομοίωση και εφέ | 25 |
| 3.2.6 Gameplay and interactivity authoring | 26 |
| 3.3 Godot [2]..... | 28 |
| 3.3.1 Σχεδιασμός εμπνευσμένος με γνώμονα την σκηνή | 29 |
| 3.3.2 Εργαλεία κωδικοποίησης που ταιριάζουν στις ανάγκες του χρήστη..... | 29 |
| 3.3.3 Απλή αλλά πολύ ικανή 3D μηχανή | 30 |
| 3.3.4 Πολύ δυνατή μηχανή για 2D γραφικά σε παιχνίδια και εφαρμογές | 30 |
| 3.3.5 Μια εφαρμογή μπορεί να υποστηρίξει πολλές πλατφόρμες | 31 |
| 3.3.6 Είναι δωρεάν και ανοικτού κώδικα (Free and Open Source)..... | 31 |

| | | |
|-------|---|-----|
| 3.4 | CryEngine [7] | 31 |
| 3.4.1 | Γραφικά (Visuals) | 32 |
| 3.4.2 | Sandbox | 37 |
| 3.4.3 | AI & Animation | 39 |
| 3.4.4 | Audio | 41 |
| 3.4.5 | Physics | 43 |
| 3.4.6 | Performance | 44 |
| 4 | Μεθοδολογία | 48 |
| 4.1 | Εγκατάσταση της μηχανής Unity | 48 |
| 4.1.1 | Unity Hub | 53 |
| 4.2 | Version Control | 62 |
| 4.2.1 | Centralized Version Control Systems | 62 |
| 4.2.2 | Distributed Version Control Systems | 63 |
| 4.3 | Best Version Control App for Unity | 64 |
| 4.3.1 | Plastic | 64 |
| 4.3.2 | Git [3] | 66 |
| 4.3.3 | Perforce Helix Core [6] | 73 |
| 4.4 | Unity's Interface Version Control | 74 |
| 4.5 | Unity Learn | 80 |
| 4.6 | Διαχείριση κώδικα στο Unity | 83 |
| 4.7 | Unity editor interface [14][16][17][19][23] | 86 |
| 4.7.1 | Scene View | 89 |
| 4.7.2 | Hierarchy window | 97 |
| 4.7.3 | Project window | 105 |
| 4.7.4 | Status bar | 107 |
| 4.7.5 | Game view | 108 |
| 4.7.6 | Inspector Window | 109 |
| 4.7.7 | Συνοτμεύσεις Πληκτρολογίου | 115 |
| 4.8 | Unity Project Settings | 117 |
| 4.8.1 | Ρύθμιση Adaptive Performance | 117 |
| 4.8.2 | Ρυθμίσεις Ήχου | 121 |
| 4.8.3 | Ρυθμίσεις Burst Compiler | 123 |
| 4.8.4 | Ρυθμίσεις Editor | 123 |
| 4.8.5 | Ρυθμίσεις Γραφικών | 125 |
| 4.8.6 | Ρυθμίσεις Χειρισμού | 130 |
| 4.8.7 | Package Manager | 132 |

| | | |
|----------|--|-----|
| 4.8.8 | Ρυθμίσεις Μηχανής Φυσικής 3D | 133 |
| 4.8.9 | Ρυθμίσεις Μηχανής Φυσικής 2D | 137 |
| 4.8.10 | Ρυθμίσεις παίκτη | 139 |
| 4.8.10.1 | Icon..... | 141 |
| 4.8.10.2 | Resolution And Presentation | 142 |
| 4.8.10.3 | Splash Image | 146 |
| 4.8.10.4 | Other Settings | 152 |
| 4.8.10.5 | Publishing Settings | 163 |
| 4.8.11 | Quality | 167 |
| 4.9 | Android and Windows Build Settings..... | 175 |
| 4.10 | C# και Unity [23] [4] [16] | 182 |
| 4.10.1 | Object..... | 182 |
| 4.10.2 | GameObject | 183 |
| 4.10.3 | MonoBehaviour | 186 |
| 4.10.4 | Component | 195 |
| 4.10.5 | Canvas | 197 |
| 5 | Παρουσίαση Εφαρμογής..... | 203 |
| 5.1 | Καταστάσεις μετάβασης της εφαρμογής | 203 |
| 5.2 | Διαγράμματα UML..... | 211 |
| 6 | Συμπεράσματα και Μελλοντική Εξέλιξη | 220 |
| | Βιβλιογραφία | 223 |
| | Παράρτημα Α. | 225 |

Λίστα Εικόνων

| | |
|---|----|
| Εικόνα 1. Οι 11 πιο δημοφιλείς εφαρμογές για παιδιά 25/3/2022 [21]..... | 4 |
| Εικόνα 2. Χαρακτηριστικά Εκπαιδευτικών Εφαρμογών | 7 |
| Εικόνα 3. Unity’s Percentage Usage on Mobile devices | 10 |
| Εικόνα 4. Unity..... | 10 |
| Εικόνα 5. Υποστηριζόμενες Πλατφόρμες της Μηχανής Unity..... | 11 |
| Εικόνα 6. Unreal Engine | 14 |
| Εικόνα 7. Godot..... | 29 |
| Εικόνα 8. CryEngine | 32 |
| Εικόνα 9. Κεντρικό Interface του Unity Hub | 54 |
| Εικόνα 10. Ρυθμίσεις του Unity Hub - Projects | 54 |
| Εικόνα 11. Ρυθμίσεις του Unity Hub - Installs | 55 |
| Εικόνα 12. Ρυθμίσεις του Unity Hub – Install Editor | 56 |
| Εικόνα 13. Ρυθμίσεις του Unity Hub – Install modules | 58 |
| Εικόνα 14. Ρυθμίσεις του Unity Hub – Install Process..... | 59 |
| Εικόνα 15. Ρυθμίσεις του Unity Hub – Install Settings | 60 |
| Εικόνα 16 Project Templates in Unity hub version 3.4.2 | 61 |
| Εικόνα 17. GIT | 67 |
| Εικόνα 18. Perforce Helix Core | 74 |
| Εικόνα 19. Setting up version control in Unity | 76 |
| Εικόνα 20. Ρυθμίσεις του Version Control στο Unity | 77 |
| Εικόνα 21. The Version Control window docked in the Editor | 79 |
| Εικόνα 22. Version Control and Project Organization | 79 |
| Εικόνα 23. Unity learn στο Διαδίκτυο | 81 |
| Εικόνα 24. Unity Learn στο Unity Hub | 82 |
| Εικόνα 25. JetBrains Rider for Unity | 86 |
| Εικόνα 26. Microsoft Visual Studio and Visual Studio Code[10]..... | 86 |
| Εικόνα 27. Unity editor interface..... | 87 |
| Εικόνα 28. Unity version 2021.3.8f1 editor layout | 88 |
| Εικόνα 29. Main Interface Unity Toolbar | 88 |

| | |
|--|-----|
| Εικόνα 30. Unity Tools and Shortcuts | 90 |
| Εικόνα 31. Transform Tool Set..... | 90 |
| Εικόνα 32. Grid and snap toolbar overlay..... | 93 |
| Εικόνα 33. Scene view (View Options toolbar)..... | 94 |
| Εικόνα 34. 2D, Lighting and Audio switches | 95 |
| Εικόνα 35. Scene visibility switch , Camera settings menu and Gizmo menu tools | 96 |
| Εικόνα 36. Scene Gizmo Perspective and Isometric | 97 |
| Εικόνα 37. The Hierarchy window | 98 |
| Εικόνα 38. Right Mouse Button Hierarchy GameObject Choices | 99 |
| Εικόνα 39. Toggle Scene pickability | 100 |
| Εικόνα 40. Toggle Scene Visibility | 101 |
| Εικόνα 41. Search Engine in Hierarchy..... | 102 |
| Εικόνα 42. Unsaved scene indication..... | 105 |
| Εικόνα 43. Project Window options..... | 106 |
| Εικόνα 44. Project window Toolbar | 106 |
| Εικόνα 45. The Status Bar | 107 |
| Εικόνα 46. Game View Control Bar..... | 108 |
| Εικόνα 47. Inspector window layout..... | 110 |
| Εικόνα 48. Adding Extra Inspector Window | 111 |
| Εικόνα 49. Comparing GameObjects with 2 Inspectors..... | 112 |
| Εικόνα 50. Αναδιάταξη των Component του GameObject..... | 113 |
| Εικόνα 51. Ανάθεση custom εικονιδίων σε GameObjects, scripts και Prefabs..... | 114 |
| Εικόνα 52. Toggle Debug Mode | 115 |
| Εικόνα 53. Popular keyboard shortcuts for visual studio | 116 |
| Εικόνα 54. Adaptive Performance | 119 |
| Εικόνα 55. Ρυθμίσεις Ήχου | 121 |
| Εικόνα 56. Ρυθμίσεις Burst Compiler για Android | 123 |
| Εικόνα 57. Εφαρμογή Unity Remote στο Google Play Store | 124 |
| Εικόνα 58. Ρυθμίσεις του Editor | 124 |
| Εικόνα 59. Graphics Settings..... | 129 |
| Εικόνα 60. Input Manager..... | 131 |

| | |
|---|-----|
| Εικόνα 61. Input System | 132 |
| Εικόνα 62. Διαχειριστής Πακέτων..... | 132 |
| Εικόνα 63. Ρυθμίσεις Φυσικής 3D | 134 |
| Εικόνα 64. Ρυθμίσεις Φυσικής 2D | 137 |
| Εικόνα 65. Player Settings..... | 140 |
| Εικόνα 66. Player Settings for Android and Windows | 141 |
| Εικόνα 67. Ρυθμίσεις Icon Android και Windows..... | 141 |
| Εικόνα 68. Ρυθμίσεις Resolution and Presentation Android και Windows..... | 142 |
| Εικόνα 69. Ρυθμίσεις Splash Image Android και Windows | 148 |
| Εικόνα 70. Other Settings | 151 |
| Εικόνα 71. Quality settings | 168 |
| Εικόνα 72. Ρυθμίσεις Tags and Layers | 173 |
| Εικόνα 73. Inspector Tags Attribute..... | 174 |
| Εικόνα 74. Build Settings..... | 176 |
| Εικόνα 75. Windows Clean Build and Incremental..... | 180 |
| Εικόνα 76. Checkbox for enabling or disabling in editor | 188 |
| Εικόνα 77. Order of Execution for event functions | 194 |
| Εικόνα 78. Παράδειγμα Screen Space Overlay | 198 |
| Εικόνα 79. Screen Space Camera..... | 198 |
| Εικόνα 80. World Space | 199 |
| Εικόνα 81. Canvas | 199 |
| Εικόνα 82. User Interface Implementation in Unity | 203 |
| Εικόνα 83. Καταστάσεις που μπορεί να βρεθεί η εφαρμογή..... | 204 |
| Εικόνα 84. Logo Unity during App Start..... | 204 |
| Εικόνα 85. Κεντρικό μενού εφαρμογής | 205 |
| Εικόνα 86. Πίνακας Βαθμολογίας..... | 205 |
| Εικόνα 87. Πληροφορίες σχετικά με την εφαρμογή | 206 |
| Εικόνα 88. Κεντρική Σκηνή παιχνιδιού | 206 |
| Εικόνα 89. Διαθέσιμα Μίνι Παιχνίδια | 207 |
| Εικόνα 90. Εκκίνηση Μίνι Παιχνιδιού | 207 |
| Εικόνα 91. Αποτέλεσμα Μίνι Παιχνιδιού | 208 |

| | |
|--|-----|
| Εικόνα 92. Επάνω Μπάρα Εικόνων Κεντρικής Σκηνής | 208 |
| Εικόνα 93. Κεντρικό Μενού Παιχνιδιού | 208 |
| Εικόνα 94. Κεντρική Σκηνή παιχνιδιού Νύχτα | 209 |
| Εικόνα 95. Ξεκλείδωμα Επόμενης Πίστας | 210 |
| Εικόνα 96. Κεντρική Σκηνή Δεύτερη Πίστα..... | 210 |
| Εικόνα 97. Game Over | 210 |
| Εικόνα 98. The End..... | 211 |
| Εικόνα 99. UML case Diagram | 211 |
| Εικόνα 100. AudioManager GameObject | 212 |
| Εικόνα 101. GameManager GameObject | 213 |
| Εικόνα 102. Local Volume GameObject..... | 214 |
| Εικόνα 103. HighScoreManager GameObject..... | 215 |
| Εικόνα 104. AudioManager GameObject Εισαγωγική Σκηνή | 215 |
| Εικόνα 105. MainMenu GameObject Εισαγωγική Σκηνή | 215 |
| Εικόνα 106. GridPanel και ImageHidden GameObjects | 216 |
| Εικόνα 107. HighScores GameObject Εισαγωγική Σκηνή | 216 |
| Εικόνα 108. Star Shine GameObject Εισαγωγική Σκηνή | 217 |
| Εικόνα 109. BirdFlying GameObject | 217 |
| Εικόνα 110. Moving Clouds GameObject | 218 |
| Εικόνα 111. CheckPoint GameObjects..... | 218 |

Λίστα Πινάκων

| | |
|--|-----|
| Πίνακας 1. Γνωστές Εκπαιδευτικές Εφαρμογές ανά κατηγορία | 8 |
| Πίνακας 2. Σύγκριση Μηχανών Παιχνιδιών[1][11][13][15][24] | 45 |
| Πίνακας 3. Σύγκριση θετικών και αρνητικών στις Μηχανές Παιχνιδιών[13][24] | 47 |
| Πίνακας 4. Unity Editor System Requirements..... | 48 |
| Πίνακας 5. Ελάχιστες απαιτήσεις του Unity Player σε Mobile Platforms | 49 |
| Πίνακας 6. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System..... | 51 |
| Πίνακας 7. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System..... | 52 |
| Πίνακας 8. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System | 52 |
| Πίνακας 9. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System..... | 52 |
| Πίνακας 10. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System | 52 |
| Πίνακας 11. Versioning | 57 |
| Πίνακας 12. Κόστος Plastic SCM | 65 |
| Πίνακας 13. Αρχείο .gitignore | 70 |
| Πίνακας 14. Αρχείο .gitattributes | 71 |
| Πίνακας 15. Δωρεάν Πλάνω Αποθετηρίων..... | 73 |
| Πίνακας 16. Ενημέρωση Κατάστασης του Version Control..... | 75 |
| Πίνακας 17. Ρυθμίσεις Version Control ανάλογα με την έκδοση | 78 |
| Πίνακας 18. Toolset of Unity Toolbar | 89 |
| Πίνακας 19. Hand Tool Controls | 90 |
| Πίνακας 20. Transform tools set Move, Rotate, Scale, RectTransform and Transform | 91 |
| Πίνακας 21. Pivot Points | 92 |
| Πίνακας 22. Grid and Snap Toolbar Overlay..... | 93 |
| Πίνακας 23. Draw Mode Menu..... | 94 |
| Πίνακας 24. Child GameObject Visibility..... | 100 |
| Πίνακας 25. Toggle pickability icons and their meaning..... | 101 |
| Πίνακας 26. Toggle visibility icons and their meaning..... | 102 |
| Πίνακας 27. Regular searches..... | 103 |
| Πίνακας 28. Special searches..... | 103 |
| Πίνακας 29. Combining search tokens..... | 104 |

| | |
|--|-----|
| Πίνακας 30. Project window Toolbar..... | 107 |
| Πίνακας 31. Game View Control Bar Function | 108 |
| Πίνακας 32. Icon Type..... | 114 |
| Πίνακας 33. Unity Popular Keyboard Shortcuts | 115 |
| Πίνακας 34. Various thermal and bottleneck conditions | 119 |
| Πίνακας 35. Scaler..... | 120 |
| Πίνακας 36. The target of a Scaler | 120 |
| Πίνακας 37. Adaptive Framerate | 120 |
| Πίνακας 38. Adaptive Resolution..... | 121 |
| Πίνακας 39. Ιδιότητες ρυθμίσεων ήχου | 122 |
| Πίνακας 40. Camera Settings | 126 |
| Πίνακας 41. Ενσωματωμένα Shader Settings..... | 126 |
| Πίνακας 42. Shader Stripping properties and functions..... | 127 |
| Πίνακας 43. Shader Loading properties and functions..... | 127 |
| Πίνακας 44. Tier Settings properties and functions..... | 128 |
| Πίνακας 45. Input Axis Properties and Functions | 131 |
| Πίνακας 46. Settings for the Package Manager | 133 |
| Πίνακας 47. Physics 3D properties and functions..... | 135 |
| Πίνακας 48. Physics 2D properties and functions..... | 138 |
| Πίνακας 49. Γενικές Ρυθμίσεις Παίκτη που μοιράζονται όλες οι πλατφόρμες..... | 140 |
| Πίνακας 50. Συγκεκριμένες Ρυθμίσεις Παίκτη ανάλογα την πλατφόρμα..... | 140 |
| Πίνακας 51. Icon (Android) | 141 |
| Πίνακας 52. Resolution And Presentation (Android)..... | 142 |
| Πίνακας 53. Resolution Scaling (Android)..... | 143 |
| Πίνακας 54. Supported Aspect Ratio (Android)..... | 144 |
| Πίνακας 55. Orientation (Android) | 144 |
| Πίνακας 56. Allowed Orientations for Auto Rotation (Android) | 145 |
| Πίνακας 57. General Settings (Android) | 145 |
| Πίνακας 58. Resolution And Presentation (Windows)..... | 145 |
| Πίνακας 59. Standalone Player Options (Windows) | 146 |
| Πίνακας 60. Splash Image | 149 |

| | |
|--|-----|
| Πίνακας 61. Static Splash Image And Scaling..... | 150 |
| Πίνακας 62. Rendering (Android) | 152 |
| Πίνακας 63. Vulcan Settings (Android) | 153 |
| Πίνακας 64. Identification (Android) | 154 |
| Πίνακας 65. Configuration (Android)..... | 154 |
| Πίνακας 66. Shader Variant Loading (Android) | 157 |
| Πίνακας 67. Script Compilation (Android) | 157 |
| Πίνακας 68. Optimization (Android) | 158 |
| Πίνακας 69. Rendering (Windows) | 159 |
| Πίνακας 70. Vulcan Settings (Windows) | 160 |
| Πίνακας 71. Configuration (Windows)..... | 161 |
| Πίνακας 72. Shader Variant Loading (Windows) | 162 |
| Πίνακας 73. Optimization (Windows)..... | 162 |
| Πίνακας 74. Project Keystore properties and description | 163 |
| Πίνακας 75. Project key properties and descriptions | 164 |
| Πίνακας 76. The manifest files, Gradle templates and Proguard files..... | 164 |
| Πίνακας 77. Rendering Player settings for desktop platforms (Windows)..... | 165 |
| Πίνακας 78. Minify properties and descriptions..... | 167 |
| Πίνακας 79. Περιγραφή τμημάτων Quality settings..... | 169 |
| Πίνακας 80. Rendering Properties and Description..... | 169 |
| Πίνακας 81. Textures Properties and descriptions | 169 |
| Πίνακας 82. Particles Properties and Descriptions..... | 170 |
| Πίνακας 83. Terrain Properties and Descriptions | 171 |
| Πίνακας 84. Shadows Properties and Descriptions | 171 |
| Πίνακας 85. Async Asset Upload Properties and Descriptions | 172 |
| Πίνακας 86. Level of Detail Properties and Descriptions..... | 172 |
| Πίνακας 87. Meshes Properties and Descriptions | 172 |
| Πίνακας 88. Build Settings (Android) | 177 |
| Πίνακας 89. Build Settings (Windows)..... | 178 |
| Πίνακας 90. Asset Import Overrides Options | 179 |
| Πίνακας 91. Ιδιότητες Object..... | 182 |

| | |
|---|-----|
| Πίνακας 92. Δημόσιες Μέθοδοι | 182 |
| Πίνακας 93. Στατικές Μέθοδοι | 183 |
| Πίνακας 94. Τελεστές..... | 183 |
| Πίνακας 95. Ιδιότητες GameObject | 184 |
| Πίνακας 96. Constructors..... | 184 |
| Πίνακας 97. Δημόσιες Μέθοδοι | 184 |
| Πίνακας 98. Στατικές Μέθοδοι | 185 |
| Πίνακας 99. Κληρονομημένες Ιδιότητες..... | 185 |
| Πίνακας 100. Κληρονομημένες Δημόσιες Μέθοδοι..... | 185 |
| Πίνακας 101. Κληρονομημένες Στατικές Μέθοδοι..... | 185 |
| Πίνακας 102. Τελεστές..... | 186 |
| Πίνακας 103. Ιδιότητες MonoBehaviour | 189 |
| Πίνακας 104. Δημόσιες Μέθοδοι | 189 |
| Πίνακας 105. Στατικές Μέθοδοι..... | 189 |
| Πίνακας 106. Messages | 189 |
| Πίνακας 107. Κληρονομημένες Ιδιότητες..... | 192 |
| Πίνακας 108. Κληρονομημένες Δημόσιες Μέθοδοι..... | 192 |
| Πίνακας 109. Κληρονομημένες Στατικές Μέθοδοι..... | 193 |
| Πίνακας 110. Κληρονομημένοι Τελεστές..... | 193 |
| Πίνακας 111. Ιδιότητες | 195 |
| Πίνακας 112. Δημόσιες Μέθοδοι | 195 |
| Πίνακας 113. Κληρονομημένες Ιδιότητες..... | 196 |
| Πίνακας 114. Κληρονομημένες Δημόσιες Μέθοδοι..... | 196 |
| Πίνακας 115. Κληρονομημένες Στατικές Μέθοδοι..... | 196 |
| Πίνακας 116. Κληρονομημένοι Τελεστές..... | 196 |
| Πίνακας 117. Canvas Scaler Properties..... | 201 |
| Πίνακας 118. Constant Pixel Size | 201 |
| Πίνακας 119. Scale With Screen Size | 201 |
| Πίνακας 120. Constant Physical Size..... | 202 |
| Πίνακας 121. World Space Canvas..... | 202 |
| Πίνακας 122. Graphics Raycaster..... | 202 |

Λίστα Συντομογραφίες

| | |
|--|-----------------------|
| 2D : Two-Dimensional | 11, 13, 31 |
| 3D : Three-Dimensional | i, ii, 11, 12, 30, 36 |
| ABAB : Area-based Ambience Blending | 43 |
| ABRDF : Angular Bidirectional Reflectance Distribution Function..... | 37 |
| ACE : Audio Controls Editor | 42, 43 |
| AI : Artificial Intelligence | 35, 40 |
| AOT : Ahead-Of-Time | 124 |
| AR : Augmented Reality | 11 |
| ARM : Advanced RISC Machines | 50 |
| BSD : Berkeley Software Distribution | 29 |
| C# : C Sharp | i, 12, 29, 30 |
| C++ : C Plus Plus | 13, 30, 32 |
| CC : Cyclic Coordinate Descent | 42 |
| CPU : Central Processing Unit | 45 |
| CRI ADX2 : Computer Research Institute Advanced eXtended..... | 43 |
| D-IK : Directional Inverse Kinematics | 42 |
| DPI : Dots Per Inch | 203 |
| DRS : Dynamic Response System | 44 |
| eol : End-of-Line | 74 |
| FBX : Filmbox..... | 14, 38 |
| FMOD : Functional Music Object Development | 43 |
| GPU : Graphics Processing Unit | 22, 37, 45 |
| GUI : Graphical User Interface | 31, 191 |
| HDR : High Dynamic Range | 24, 34 |
| HRTF : Head-Related Transfer Function..... | 43 |
| IBL : Image-Based Lighting | 37 |
| IDE : Integrated Development Environment | ii |
| IK : Inverse Kinematics | 42 |

| | |
|--|-----------------|
| IL : Intermediate Language | 124 |
| If : line feed | 74 |
| LLVM : Low-Level Virtual Machine..... | 124 |
| LOD : Level Of Detail | 17 |
| MIT : Massachusetts Institute of Technology..... | 29 |
| MNM : Multi-Layer Navigation Mesh | 41 |
| MR : Mixed Reality | 11 |
| OFPA : One File Per Actor | 17 |
| OpenGL : Open Graphics Library..... | 30 |
| OS : Operating System | 11 |
| PBR : Physical Based Rendering | 33, 37 |
| PC : Personal Computer | 13 |
| PCG : 6. Procedural Content Generation | 17 |
| PN triangles : Phong–Ngo triangles | 34 |
| POM : Parallax Occlusion Mapping..... | 38 |
| PSA : Parametric Skeletal Animation | 40 |
| RTPCs : Real-Time Parameter Controls | 42 |
| SB IK : Sampling-based Inverse Kinematics..... | 42 |
| SCM : Source Code Management | 65 |
| SDL : Simple DirectMedia Layer | 43 |
| SSAO : Screen Space Ambient Occlusion | 37 |
| SSDO : Screen Space Directional Occlusion..... | 37 |
| SSR : Screen Space Reflections..... | 35 |
| SSSS : Screen-Space Sub-surface Scattering | 40 |
| UE : User Experience | ii |
| UI : User Interface | ii, 12, 28, 203 |
| UML : Unified Modeling Language..... | 206, 214 |
| USD : Universal Scene Description..... | 14 |
| UV : “U” and “V” denote the axes of the 2D texture | 15, 16, 23 |
| VR : Virtual Reality | 11, 13 |
| WYSIWYG : What You See Is What You Get..... | 16, 38 |

| | |
|---|----|
| Indie Developers : Independent Developers | 11 |
| κτλ : και τα λοιπά | 3 |
| OBM : Offset Bump Mapping..... | 38 |
| π.χ. : παραδείγματος χάρη | 3 |

1. Εισαγωγή

Τα τελευταία χρόνια, η τεχνολογία έχει επηρεάσει σημαντικά τον τρόπο που ζούμε και μαθαίνουμε. Το εκπαιδευτικό σύστημα δεν θα μπορούσε να μείνει ανεπηρέαστο και για αυτό υπάρχει μια αυξανόμενη υιοθέτηση καινούργιων μεθόδων μάθησης. Μια από τις μεθόδους που χρησιμοποιείται είναι των διαδραστικών εκπαιδευτικών παιχνιδιών. Αυτά τα παιχνίδια έχουν τη δυνατότητα να προσφέρουν μια ελκυστική και διαδραστική εμπειρία μάθησης στα παιδιά, βοηθώντας τα να αποκτήσουν νέες γνώσεις και δεξιότητες με διασκεδαστικό και επιμορφωτικό τρόπο. Η χρήση εκπαιδευτικών παιχνιδιών είναι όλο και πιο διαδεδομένη και η μάθηση με βάση τα παιχνίδια έχει αποδειχθεί ότι είναι ένας αποτελεσματικός τρόπος διδασκαλίας των παιδιών.

Στόχος αυτής της διπλωματικής εργασίας είναι να σχεδιαστεί και να αναπτυχθεί ένα διαδραστικό εκπαιδευτικό παιχνίδι για παιδιά χρησιμοποιώντας τη μηχανή παιχνιδιών Unity 3D για σταθερούς υπολογιστές και έξυπνες κινητές συσκευές. Το παιχνίδι έχει σχεδιαστεί για να διδάξει στα παιδιά ένα ή περισσότερα θέματα χρησιμοποιώντας μια ποικιλία διαδραστικών δραστηριοτήτων, παζλ και προκλήσεων. Τα χαρακτηριστικά του παιχνιδιού περιλαμβάνουν ένα εύχρηστο περιβάλλον διάδρασης, πολύχρωμα γραφικά και ηχητικά εφέ για να δημιουργηθεί μια συναρπαστική και διασκεδαστική εμπειρία για τα παιδιά.

Η διαδικασία ανάπτυξης της εφαρμογής περιλαμβάνει τη δημιουργία των αντικειμένων του παιχνιδιού, του κώδικα και τη δοκιμή για να διασφαλιστεί ότι το παιχνίδι είναι κατάλληλο για επιτραπέζιους υπολογιστές και για φορητές πλατφόρμες. Το τελικό προϊόν είναι ένα πλήρως λειτουργικό παιχνίδι που είναι ταυτόχρονα διασκεδαστικό και εκπαιδευτικό, σχεδιασμένο να αιχμαλωτίζει και να διεγείρει το μυαλό των παιδιών.

Αυτή η διατριβή παρέχει μια επισκόπηση της μηχανής παιχνιδιών Unity 3D, της διαδικασίας σχεδιασμού ενός διαδραστικού εκπαιδευτικού παιχνιδιού και της πιθανής επίδρασης τέτοιων παιχνιδιών στη μάθηση και την ανάπτυξη των παιδιών. Μέχρι το τέλος αυτής της διατριβής, ο αναγνώστης θα έχει μια πλήρη κατανόηση της διαδικασίας ανάπτυξης ενός εκπαιδευτικού παιχνιδιού με την χρήση της μηχανής Unity και των πλεονεκτημάτων της μάθησης με βάση το παιχνίδι ως εργαλείο διδασκαλίας.

Διάρθρωση κεφαλαίων :

1. Το Κεφάλαιο I παρέχει μια εισαγωγή στο θέμα, συμπεριλαμβανομένου του ιστορικού και του πλαισίου της μελέτης, του σκοπού και των στόχων της μελέτης και μια επισκόπηση της δομής της διατριβής.
2. Στο Κεφάλαιο II γίνεται ανασκόπηση των εφαρμογών σχετικά με τα εκπαιδευτικά παιχνίδια και τη μάθηση που βασίζεται σε παιχνίδια και τους διάφορους τύπους παιχνιδιών.
3. Στο Κεφάλαιο III γίνεται μια παρουσίαση των πιο γνωστών μηχανών παιχνιδιών.
4. Το Κεφάλαιο IV περιγράφει τη μεθοδολογία που χρησιμοποιείται για το σχεδιασμό και την ανάπτυξη του εκπαιδευτικού παιχνιδιού χρησιμοποιώντας την μηχανή Unity.
5. Το Κεφάλαιο V περιγράφει το τελικό προϊόν και τις προσθήκες για την βελτιστοποίηση της εφαρμογής .
6. Το Κεφάλαιο VI περιγράφει τα συμπεράσματα και τις προτάσεις για μελλοντική εξέλιξη του παιχνιδιού .
7. Η Βιβλιογραφία
8. Τέλος, το Παράρτημα Α περιλαμβάνει παραρτήματα που περιέχουν πρόσθετο υλικό όπως στιγμιότυπα οθόνης, αποσπάσματα κώδικα, εγχειρίδια χρήση κ.λπ.».

2. Τύποι εκπαιδευτικών εφαρμογών

Τα εκπαιδευτικά παιχνίδια και η μάθηση με βάση τα παιχνίδια έχουν κερδίσει δημοτικότητα τα τελευταία χρόνια λόγω των δυνατοτήτων τους να προσελκύουν και να παρακινούν τους μαθητές.

- Τα εκπαιδευτικά παιχνίδια αναφέρονται σε όλα τα παιχνίδια που έχουν σχεδιαστεί με πρωταρχικό σκοπό τη διδασκαλία ή τη μάθηση.
- Η μάθηση με βάση το παιχνίδι (gamification), από την άλλη πλευρά, αναφέρεται στη χρήση παιχνιδιών για τη βελτίωση της επίδοσης των μαθητών.

Σε εκπαιδευτικά περιβάλλοντα, τα παιχνίδια χρησιμοποιούνται για τη διδασκαλία ποικίλων θεμάτων και δεξιοτήτων, από βασικά μαθηματικά και προγραμματισμό έως την επίλυση σύνθετων προβλημάτων και τη λήψη αποφάσεων. Υπάρχουν διάφοροι τύποι παιχνιδιών που χρησιμοποιούνται για εκπαίδευση, συμπεριλαμβανομένων παιχνιδιών προσομοίωσης, παιχνιδιών ρόλων και της παιχνιδοποίησης. Τα παιχνίδια προσομοίωσης προσομοιώνουν πραγματικές καταστάσεις, όπως προσομοιώσεις οδήγησης, ενώ τα παιχνίδια ρόλων αντιμετωπίζουν κοινωνικά ζητήματα και εκπαιδεύουν τους παίκτες σε προβλήματα του πραγματικού κόσμου. Η παιχνιδοποίηση, από την άλλη πλευρά, αναφέρεται στη εισαγωγή μηχανισμών του παιχνιδιού, όπως πόντοι, σήματα και πίνακες κατάταξης, για να παρακινηθούν και να εμπλακούν οι μαθητές σε περιβάλλον εκτός παιχνιδιού.

Θα γίνει μια προσπάθεια να αναφερθούν οι πιο γνωστές εφαρμογές που υπάρχουν στην αγορά οι οποίες καλύπτουν διάφορα εκπαιδευτικά θέματα και δεν περιορίζονται στα βασικά όπως μαθηματικά, φυσική κτλ. Αυτές οι εφαρμογές θα μπορούσαν να κατηγοριοποιηθούν ανά εκπαιδευτικό αντικείμενο, ηλικιακή ομάδα, επίπεδο δυσκολίας κατανόησης εκπαιδευτικού αντικειμένου, τρόπο παιχνιδιού δηλαδή πολλαπλών παικτών ή μόνο ενός και τι τύπου είναι όπως π.χ. παζλ, εξομοιωτής, περιπέτειας κτλ., για να πάρουμε μια εικόνα από την δυναμική που υπάρχει αυτή την στιγμή στην εκπαίδευση.

Με βάση το Google App μια από της πιο διαδεδομένες κατηγορίες εφαρμογών που δέχεται μεγάλη επισκεψιμότητα είναι τα εκπαιδευτικά προγράμματα. Εδώ να τονίσουμε πως μια εκπαιδευτική εφαρμογή δεν χρειάζεται απαραίτητα να είναι για τα

μαθηματικά, τη φυσική ή τη χημεία και ούτω καθεξής αλλά οτιδήποτε έχει να κάνει με την εκπαίδευση του εκάστοτε χρήστη της εφαρμογής. Για παράδειγμα θα μπορούσε να είναι ένα πρόγραμμα για να μάθει κάποιος να ψαρεύει ή ένα πρόγραμμα για να μάθει να παίζει κιθάρα. Εδώ η λίστα με τις επιλογές συνεχώς εμπλουτίζεται με αποτέλεσμα ο χρήστης να έχει μια πληθώρα επιλογών που το μόνο μειονέκτημα που θα μπορούσε κάποιος να ισχυριστεί ότι υπάρχει, είναι το ότι κάποιες από αυτές τις εφαρμογές είναι επί πληρωμή.



Εικόνα 1. Οι 11 πιο δημοφιλείς εφαρμογές για παιδιά 25/3/2022 [21]

Πηγή: <https://www.goodhousekeeping.com/electronics/g28212386/best-apps-for-kids/>

Η κατανόηση των διαφορετικών τύπων παιχνιδιών που χρησιμοποιούνται για την εκπαίδευση είναι σημαντική για το σχεδιασμό αποτελεσματικών εκπαιδευτικών παιχνιδιών και την ενσωμάτωσή τους στη μαθησιακή διαδικασία. Ακολουθούν μερικά παραδείγματα των διαφορετικών τύπων παιχνιδιών που χρησιμοποιούνται για εκπαίδευση:

- ❖ **Παιχνίδια προσομοίωσης:** Αυτά τα παιχνίδια προσομοιώνουν πραγματικές καταστάσεις και επιτρέπουν στους παίκτες να πειραματιστούν και να μάθουν από τις ενέργειές τους σε ένα ασφαλές και ελεγχόμενο περιβάλλον. Τα παραδείγματα περιλαμβάνουν προσομοιωτές πτήσης, ιατρικές προσομοιώσεις και προσομοιώσεις επιχειρήσεων.
- ❖ **Σοβαρά παιχνίδια:** Αυτά τα παιχνίδια έχουν σχεδιαστεί με πρωταρχικό σκοπό την εκπαίδευση των παικτών σε προβλήματα του πραγματικού κόσμου και κοινωνικά

ζητήματα. Συχνά χρησιμοποιούνται για την αντιμετώπιση σύνθετων ζητημάτων όπως η κλιματική αλλαγή, η υγεία και η κοινωνική δικαιοσύνη.

- ❖ **Παιχνιδοποίηση:** Αυτή η προσέγγιση περιλαμβάνει την ενσωμάτωση στοιχείων παιχνιδιού, όπως πόντους, σήματα και πίνακες κατάταξης, σε περιβάλλοντα εκτός παιχνιδιού για να παρακινηθούν και να εμπλακούν οι μαθητές. Παραδείγματα περιλαμβάνουν εφαρμογές εκμάθησης γλωσσών που απονέμουν πόντους για την ολοκλήρωση μαθημάτων και εφαρμογές γυμναστικής που χρησιμοποιούν πίνακες κατάταξης για να ενθαρρύνουν τον ανταγωνισμό.
- ❖ **Παιχνίδια παζλ:** Αυτά τα παιχνίδια προκαλούν τους παίκτες να λύσουν προβλήματα και γρίφους, συχνά μέσω δοκιμής και λάθους. Χρησιμοποιούνται συνήθως για τη διδασκαλία δεξιοτήτων επίλυσης προβλημάτων, κριτικής σκέψης και λογικής.
- ❖ **Παιχνίδια ρόλων:** Αυτά τα παιχνίδια επιτρέπουν στους παίκτες να αναλάβουν το ρόλο ενός χαρακτήρα σε έναν φανταστικό κόσμο και να κάνουν επιλογές που επηρεάζουν το αποτέλεσμα του παιχνιδιού. Συχνά χρησιμοποιούνται για διδασκαλία ιστορίας, λογοτεχνίας και κοινωνικών σπουδών.

2.1 Διάφοροι τύποι εκπαιδευτικών εφαρμογών[18]

2.1.1 Εκπαιδευτικές εφαρμογές πολλαπλών θεμάτων

Οι τύποι των εκπαιδευτικών εφαρμογών ποικίλλουν και έχουν να κάνουν συνήθως με το κοινό στο οποίο αναφέρονται. Κάποιος μπορεί να βρει από μια απλή εφαρμογή ενός απλού ερωτηματολογίου μέχρι και ολόκληρο ακαδημαϊκό πεδίο. Περιορισμοί μόνο σε ένα ακαδημαϊκό πεδίο δεν υπάρχουν και ο χρήστης έχει την δυνατότητα να παρακολουθήσει ακόμα και ξενόγλωσσα μαθήματα. Εδώ μπορούμε να δούμε ένα παράδειγμα της πλατφόρμας Coursera η οποία ξεκίνησε σαν μια απλή online πλατφόρμα του πανεπιστημίου Stanford και έχει φτάσει τώρα να καλύπτει εκπαιδευτικά πεδία από διάφορα πανεπιστήμια και διαδικτυακές εταιρίες από όλο τον κόσμο , οπότε καταλήγουμε στο συμπέρασμα ότι εάν σε μια καλή ιδέα αφιερώσεις χρήμα και χρόνο αυτή μπορεί να πάει πολύ μακριά στο ψηφιακό κόσμο.

2.1.2 Εκπαιδευτικές εφαρμογές συγκεκριμένου θέματος

Εδώ όπως λέει και ο τίτλος υπάρχουν εφαρμογές οι οποίες καλύπτουν συγκεκριμένα εκπαιδευτικά προγράμματα όπως π.χ. το Duolingo και το Yusician. Πολλές εταιρίες αποφασίζουν πως καλό θα ήταν από το να είναι μέτριες σε διάφορα εκπαιδευτικά προγράμματα καλύτερα να είναι καλές σε μόνο ένα οπότε και αφιερώνουν όλο τους το δυναμικό προς μια κατεύθυνση. Αυτό προϋποθέτει πως υπάρχει και το ανάλογο βάθος στον τομέα που έχει επιλέξει ο προγραμματιστής ή η εταιρία αυτή.

2.1.3 Υποστηρικτικές εφαρμογές (Supportive Apps)

Υποστηρικτικές εφαρμογές είναι αυτές που βοηθάνε τον χρήστη με το να του δίνουν εργαλεία που τα χρησιμοποιεί στην καθημερινότητά του, είτε είναι στην δουλειά, είτε είναι σε κάποια μορφή εκπαίδευσης. Για παράδειγμα υπάρχουν εφαρμογές που μετράνε το εμβαδόν ενός χώρου και είναι πολύ χρήσιμες για μηχανικούς ενώ ένας απλός αριθμητικός υπολογιστής το λεγόμενο κομπιουτεράκι είναι χρήσιμο για μαθηματικούς υπολογισμούς για όλους.

2.1.4 Εκπαιδευτικά παιχνίδια (Educational Games)

Κανείς δεν έχει πρόβλημα με την μελέτη αλλά όταν αυτή μπορεί να γίνει με κάποιο τρόπο που θα είναι πιο ελκυστική τότε η εκπαιδευτική κοινότητα πρέπει να το λάβει σοβαρά σημασία. Η εκπαίδευση μέσα σε ένα περιβάλλον παιχνιδιού δίνει το έναυσμα στους μαθητές να προσηλωθούν στη μάθηση επειδή ταυτόχρονα παίζουν ένα παιχνίδι. Το μόνο που μένει να αποφασιστεί είναι πόσα στοιχεία Gamification θα ενσωματωθούν στην εφαρμογή.

2.1.5 Εφαρμογές διαχείρισης μάθησης (Learning Management Apps)

Εδώ έχουμε εφαρμογές που όπως και στις υποστηρικτικές εφαρμογές (Supportive apps) έτσι και εδώ βοηθούν τον χρήστη με τις καθημερινές δραστηριότητές του αναλαμβάνοντας να διαχειρίζονται επαναλαμβανόμενες και προγραμματισμένες δραστηριότητες που έχει να πραγματοποιήσει ο ίδιος. Τέτοιες επαναλαμβανόμενες διαδικασίες είναι το πρόγραμμα σπουδών σε μια σχολή, ο έλεγχος απουσιών, η διορία εργασιών, τα διαδικτυακά τεστ, οι βαθμολογίες και πολλά άλλα. Παραδείγματος χάριν τέτοιες εφαρμογές είναι το Schoology, Canvas Student κτλ. Τέτοιες εφαρμογές μπορούν να

χρησιμοποιηθούν από ένα χρήστη ή μια ομάδα χρηστών όπως π.χ. από φοιτητές ενός πανεπιστημίου.

2.1.6 Εκπαιδευτικές εφαρμογές για παιδιά προσχολικής ηλικίας και νήπια

Τα παιδιά σε μικρή ηλικία ίσως είναι και ο πιο δύσκολος πελάτης που πρέπει να ικανοποιήσει ένας προγραμματιστής μιας εκπαιδευτικής εφαρμογής. Σε αυτήν την ηλικία οι μικροί μας ήρωες δεν έχουν την υπομονή για να ασχοληθούν με μια εφαρμογή που δεν θα τους κεντρίσει το ενδιαφέρον με την πρώτη διάδραση. Οι ικανότητες να κατανοούν πολύπλοκες έννοιες είναι ακόμα σε πρώιμη φάση. Εφαρμογές που απευθύνονται σε παιδιά ηλικίας 0-5 ετών είναι φωτεινές με πολλά έντονα χρώματα και διαδραστικές για να τραβήξουν το ενδιαφέρον του παιδιού και να βεβαιωθούν ότι μέσω του παιχνιδιού μαθαίνουν.



Εικόνα 2. Χαρακτηριστικά Εκπαιδευτικών Εφαρμογών

Πηγή: <https://www.netsolutions.com/insights/effective-features-of-best-educational-apps/>

Πηγή: <https://www.iconarchive.com/show/office-icons-by-thalita-torres.html>

2.2 Εκπαιδευτικές εφαρμογές στο εμπόριο

Οι πιο γνωστές εκπαιδευτικές εφαρμογές ανά κατηγορία με βάση ηλεκτρονικό άρθρο είναι :

Πίνακας 1. Γνωστές Εκπαιδευτικές Εφαρμογές ανά κατηγορία

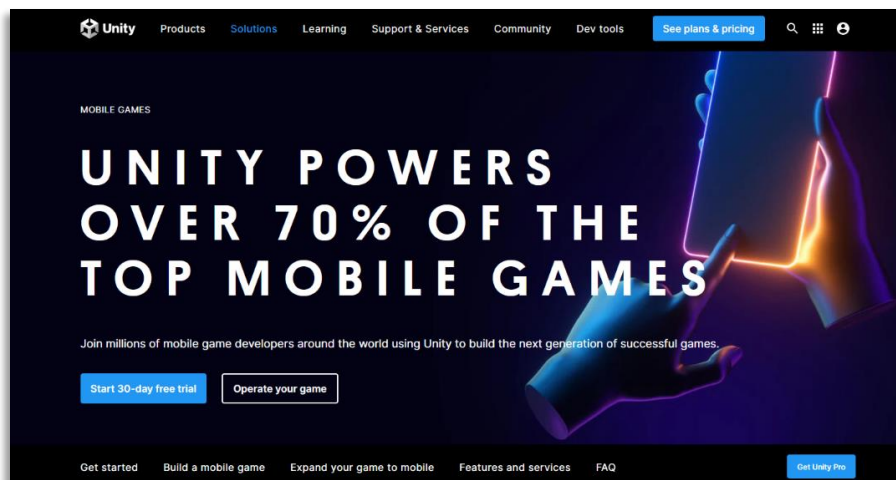
| Κατηγορία Εφαρμογής | Όνομα Εφαρμογής |
|---|--|
| Multi-subject Apps: | <ol style="list-style-type: none"> 1. Khan Academy 2. Khan Academy Kids 3. BrainPOP 4. BrainPOP Jr. 5. Quizlet 6. Fact Monster 7. Poptropica 8. Coursera (Not Free) 9. Udemy (Not Free) 10. Starfall (Not Free) 11. ABCmouse (Not Free) 12. ABCya! 13. Elevate – Brain Training Games 14. Lumosity |
| Free Educational Math Apps: | <ol style="list-style-type: none"> 1. Prodigy Math Game 2. Coolmath Games 3. Bedtime Math 4. Formulia 5. Endless Numbers 6. Math Training for Kids 7. Photomath 8. DragonBox |
| Science: | <ol style="list-style-type: none"> 1. Science 360 2. NASA 3. NSF Science Zone 4. Plum’s Creaturizer 5. Play and Learn Science 6. MarcoPolo Ocean 7. Periodic Table 8. True or False Chemistry 9. AUTODESK Tinkercad 10. Minecraft: Education Edition (Not Free) 11. Star Walk 2 12. Google Earth |
| Free Educational Apps for Reading: | <ol style="list-style-type: none"> 1. Kid Mode: Play + Learn: Multiple Reading Skills 2. Libby 3. Starfall ABCs 4. Kindle 5. Bamboo Learning 6. LitCharts 7. Epic 8. SpellingCity 9. Vocabulary.com |

| | |
|--|---|
| Free Coding Apps: | <ol style="list-style-type: none"> 1. PBS Kids ScratchJr 2. Bee-Bot 3. Codecademy Go 4. Hopster Coding Safari 5. Encode: Learn to Code 6. CodeSpark Academy 7. Sololearn 8. Swift Playgrounds 9. Code.org 10. Scratch |
| Free Educational Art Apps: | <ol style="list-style-type: none"> 1. Toca Life World: Build Stories 2. DailyArt 3. SuperHero Comic Book Maker 4. Draw and Tell 5. Google Arts & Culture 6. Joy Doodle: Movie Color and Draw |
| Free Educational Apps for Learning Languages: | <ol style="list-style-type: none"> 1. Duolingo 2. Memrise 3. EngVarta 4. Babbel 5. Rosetta Stone (Not free) 6. Lingodeer 7. Drops by Kahoot |
| Free Educational Apps for Getting Active: | <ol style="list-style-type: none"> 1. GoNoodle 2. Super Stretch Yoga 3. Exercise: At Home Workout App 4. Cosmic Kids |
| Supportive Apps or Learning Management Apps: | <ol style="list-style-type: none"> 1. Quizizz 2. Kahoot! 3. Socrative 4. Book Creator 5. Classcraft 6. Brainly 7. Seesaw 8. Mathway 9. StudyBlue 10. AnkiDroid |

Πηγή : <https://lifefamilyfun.com/free-educational-apps/>

3. Μηχανές Παιχνιδιών

Οι μηχανές παιχνιδιών (game engines) είναι λογισμικά σχεδιασμένα για τη δημιουργία βιντεοπαιχνιδιών. Υπάρχουν πολλές διαφορετικές game engines διαθέσιμες, για να επιλέξει ο προγραμματιστής. Ανάλογα με τις ανάγκες και τις προτιμήσεις ο προγραμματιστής, επιλέγει την καταλληλότερη μηχανή παιχνιδιών (game engine) για τη δημιουργία της εφαρμογής του. Μερικές από τις δημοφιλέστερες game engines που χρησιμοποιούνται σήμερα είναι οι εξής: Unity, Unreal Engine, Godot και CryEngine. Όλες οι προαναφερόμενες μηχανές έχουν τα δικά τους πλεονεκτήματα και μειονεκτήματα αλλά σε γενικές γραμμές είναι πάρα πολύ καλές σε αυτό που κάνουν και συνεχίζουν να αναβαθμίζονται με περισσότερα χαρακτηριστικά. Στις κινητές συσκευές το Unity κατέχει το μεγαλύτερο μερίδιο της αγοράς με ποσοστό περίπου 70% στα κορυφαία παιχνίδια σε φορητές πλατφόρμες επειδή οι προγραμματιστές εκμεταλλεύονται τις δυνατότητες που προσφέρει η μηχανή σε σχέση με τους περιορισμούς των κινητών συσκευών.



Εικόνα 3. Unity's Percentage Usage on Mobile devices

Πηγή : <https://unity.com/solutions/mobile>

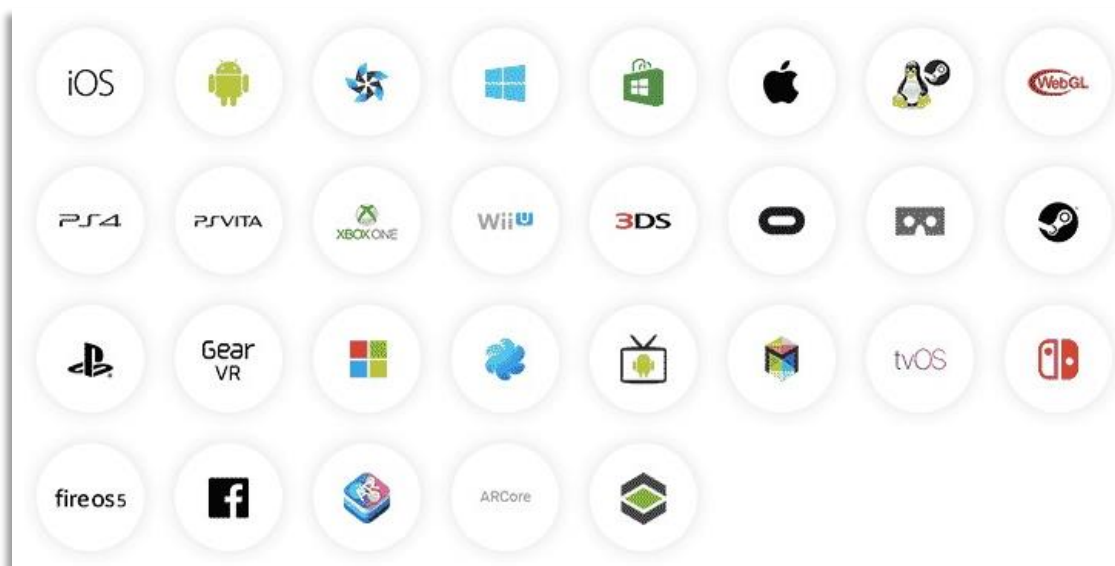


Εικόνα 4. Unity

Πηγή: <https://unity.com/>

3.1 Unity [4][22]

Η **Unity** είναι μια πολύ δημοφιλής μηχανή παιχνιδιών και μπορεί να σχεδιάσει και υλοποιήσει μια εφαρμογή για πολλές πλατφόρμες. Η εταιρεία που είναι υπεύθυνη για την μηχανή είναι η Unity Technologies και την δημοσίευσε το 2005. Στην αρχή είχε κατασκευαστεί για την δημιουργία παιχνιδιών στην πλατφόρμα της Apple OS X. Αργότερα όμως και με πολλές αναβαθμίσεις και ανανεώσεις έχει φτάσει στην παρούσα μορφή της και είναι ικανή να παράγει υψηλής ποιότητας παιχνίδια για πολλές πλατφόρμες όπως σταθερούς υπολογιστές, κινητές συσκευές, κονσόλες, πλατφόρμες εικονικής πραγματικότητας κτλ. Είναι εύκολη στην χρήση ακόμα και για αρχάριους χρήστες και παρέχει ένα ολοκληρωμένο πακέτο εργαλείων για διαχείριση ψηφιακών στοιχείων (assets), όπως textures, sounds και 3D models όπως επίσης μια πολύ καλή μηχανή εξομοίωσης φυσικής (physics engine), εργαλείων για την δημιουργία κινουμένων σχεδίων (animation tools) και τη δημιουργία ρεαλιστικών γραφικών. Αυτό έχει ως αποτέλεσμα να είναι η πρώτη επιλογή για μικρές ομάδες προγραμματιστών (από 2 έως 10 άτομα που ονομάζονται indie Developers). Η μηχανή δίνει την δυνατότητα για κατασκευή 2D παιχνιδιών , 3D παιχνιδιών , AR, VR, MR παιχνιδιών και πολλά αλλά. Η μηχανή έχει τόσες πολλές δυνατότητες που πλέον χρησιμοποιείται από βιομηχανίες άλλων κλάδων όπως οι αυτοκινητοβιομηχανίες, η αρχιτεκτονική, οι κατασκευές, ο κινηματογράφος, ο στρατός, κτλ.



Εικόνα 5. Υποστηριζόμενες Πλατφόρμες της Μηχανής Unity

Πηγή : <https://www.nanalyze.com/2017/10/unity-technologies-leading-game-engine/>

Η μηχανή έχει δωρεάν έκδοση αλλά ο χρήστης της μηχανής πρέπει να πληρώσει 5% εάν ξεπεράσει της 100 χιλιάδες δολάρια είτε σε διαφημίσεις, είτε σε χρηματοδότηση, είτε σε ακαθάριστα έσοδα (royalties fee) και να αναβαθμίσει το πλάνο του από Unity Personal σε Unity Plus.

3.1.1 Unity Features

Μερικά από τα βασικά χαρακτηριστικά του Unity περιλαμβάνουν:

- ❖ Υποστήριξη για πολλαπλές πλατφόρμες: Το Unity επιτρέπει στους προγραμματιστές να δημιουργούν παιχνίδια σε ένα μεγάλο φάσμα από πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS, Linux, iOS, Android και πολλών άλλων.
- ❖ Οπτικός επεξεργαστής: Ο οπτικός επεξεργαστής του Unity παρέχει μια διεπαφή drag-and-drop για τη δημιουργία σκηνών παιχνιδιών, χαρακτήρων και αντικειμένων. Επιτρέπει επίσης τον εύκολο χειρισμό τρισδιάστατων μοντέλων και κινουμένων σχεδίων.
- ❖ Scripting: Το Unity υποστηρίζει τη C# γλώσσα προγραμματισμού C#. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν προσαρμοσμένους μηχανισμούς παιχνιδιού, στοιχεία UI και άλλα χαρακτηριστικά.
- ❖ Κατάστημα περιουσιακών στοιχείων: Το κατάστημα περιουσιακών στοιχείων της Unity παρέχει ένα μεγάλο φάσμα προκατασκευασμένων στοιχείων, όπως 3D μοντέλα, κινούμενα σχέδια, σενάρια και πολλά άλλα. Αυτό μπορεί να εξοικονομήσει χρόνο και προσπάθεια στους προγραμματιστές κατά τη δημιουργία παιχνιδιών.
- ❖ Φυσική: Το Unity περιλαμβάνει μια ισχυρή μηχανή φυσικής που επιτρέπει ρεαλιστικές συγκρούσεις, βαρύτητα και άλλες φυσικές αλληλεπιδράσεις.
- ❖ Ήχος: Το Unity περιλαμβάνει μια σειρά εργαλείων ήχου, συμπεριλαμβανομένων της τρισδιάστατης χωροθέτησης, των ηχητικών εφέ και της αναπαραγωγής μουσικής.
- ❖ Δικτύωση: Το Unity παρέχει ενσωματωμένα χαρακτηριστικά δικτύωσης που επιτρέπουν παιχνίδια για πολλούς παίκτες μέσω του διαδικτύου.

- ❖ Υποστήριξη εικονικής πραγματικότητας (VR): Το Unity περιλαμβάνει υποστήριξη για μια σειρά συσκευών VR, συμπεριλαμβανομένων των Oculus Rift, HTC Vive και Google Cardboard.
- ❖ Υποστήριξη 2D: Το Unity περιλαμβάνει χαρακτηριστικά ειδικά σχεδιασμένα για την ανάπτυξη παιχνιδιών 2D, όπως η διαχείριση sprite και η φυσική 2D.
- ❖ Ανάλυση: Το Unity περιλαμβάνει ενσωματωμένα εργαλεία ανάλυσης που επιτρέπουν στους προγραμματιστές να παρακολουθούν τη συμπεριφορά των παικτών, την απόδοση του παιχνιδιού και άλλες μετρήσεις.

Συνολικά, το Unity παρέχει ένα ολοκληρωμένο σύνολο εργαλείων και χαρακτηριστικών που το καθιστούν μια δημοφιλή επιλογή για προγραμματιστές παιχνιδιών όλων των επιπέδων εμπειρίας.

3.2 Unreal [9][22]

Η **Unreal Engine** είναι μια μηχανή παιχνιδιών που είναι κατασκευασμένη για υλοποίηση παιχνιδιών 3D και η εταιρεία που την ανέπτυξε είναι η Epic Games. Πρωτοεμφανίστηκε το 1998 σαν παιχνίδι πρώτου προσώπου με την ονομασία Unreal και στην αρχή χρησιμοποιήθηκε για την κατασκευή shooters πρώτου προσώπου για προσωπικούς υπολογιστές(PC) αλλά από τότε έχει αλλάξει δραματικά και με τις αναβαθμίσεις που έχει υποστεί και πλέον χρησιμοποιείται από βιομηχανίες εκτός της βιομηχανίας παιχνιδιών όπως του κινηματογράφου και της τηλεόρασης. Είναι γραμμένο στην γλώσσα προγραμματισμού C++ και έχει την δυνατότητα να σχεδιάσει και υλοποιήσει μια εφαρμογή για πολλές πλατφόρμες όπως σταθερούς υπολογιστές, κινητά, κονσόλες και πλατφόρμες εικονικής πραγματικότητας προσφέροντας έτσι μια ευελιξία ως προς την επιλογή μηχανής παιχνιδιών. Παρόλο που είναι γραμμένη σε C++ παρέχει την δυνατότητα σε ένα αρχάριο προγραμματιστή να γράψει σε visual scripting τα λεγόμενα Blueprints και έτσι να εισάγει κάποιον στον προγραμματισμό πιο εύκολα. Πρόσφατα βγήκε στην αγορά η έκδοση 5 (Απρίλιος 2022) προσθέτοντας πολλές ακόμα δυνατότητες για ακόμα πιο ρεαλιστικά γραφικά. Η χρήση της μηχανής είναι δωρεάν μέχρι ο δημιουργός της εφαρμογής να ξεπεράσει σε έσοδα κάποιο ποσό που ορίζει η εταιρία. Μετά από έσοδα της τάξεως του 1 εκατομμυρίου δολαρίων το φυσικό ή νομικό πρόσωπο πρέπει να πληρώσει

στην εταιρία 5% των ακαθάριστων εσόδων(royalties fee) για έσοδα που υπερβαίνουν το προαναφερόμενο ποσό.



Εικόνα 6. Unreal Engine

Πηγή : <https://www.unrealengine.com/en-US/unreal-engine-5>

Ακολουθεί μια λεπτομερής περιγραφή των χαρακτηριστικών της Unreal:

3.2.1 Unreal Pipeline Integration

- ❖ **Υποστήριξη FBX, USD και Alembic:** Η ενσωμάτωση αγωγών στην Unreal Engine επιτρέπει την απρόσκοπτη σύνδεση με την γραμμή παραγωγής πολυμέσων, υποστηρίζοντας βιομηχανικά πρότυπα όπως FBX, USD και Alembic. Η υποστήριξη USD ενισχύει τη συνεργασία μεταξύ των μελών της ομάδας, επιτρέποντας την ταυτόχρονη εργασία. Η Unreal Engine μπορεί να διαβάσει ένα αρχείο USD από οπουδήποτε στο δίσκο χωρίς χρονοβόρα πλήρη εισαγωγή και να γράψει πίσω τις αλλαγές ως επικαλύψεις. Η επαναφόρτωση του ωφέλιμου φορτίου USD ενημερώνει αμέσως τις αλλαγές που έγιναν από άλλους χρήστες στο προηγούμενο στάδιο.
- ❖ **Python scripting:** Ο χρήστης ενσωματώνοντας την Unreal Engine στο pipeline του και αξιοποιώντας πλήρως την υποστήριξη του βιομηχανικού προτύπου Python scripting στον Unreal Editor, μπορεί να αυτοματοποιήσει τις ροές εργασίας. Μπορεί να δημιουργήσει διαδικασίες διαχείρισης περιεχομένου, να αυτοματοποιήσει τις προετοιμασίες δεδομένων, να οργανώσει το περιεχόμενο σε ένα επίπεδο και να δημιουργήσει προσαρμοσμένα περιβάλλοντα εργασίας για τον έλεγχο του Unreal Editor.
- ❖ **Datasmith: Απρόσκοπτη Μετατροπή Δεδομένων:** Με τη χρήση του DataSmith, ο χρήστης μπορεί να μετατρέψει ολόκληρες σκηνές -συμπεριλαμβανομένων των κινούμενων σχεδίων και των μεταδεδομένων- από προγράμματα όπως το 3ds Max, Revit, Sketch-Up Pro, Cinema 4D, Rhino, SolidWorks, Catia και άλλες μορφές DCC,

CAD και BIM, με υψηλή ακρίβεια. Η διαδικασία μετατροπής είναι μη καταστροφική, επιτρέποντάς αυτόν να συνεχίσει την επεξεργασία του αρχικού πακέτου χωρίς να απωλέσει τις μελλοντικές αλλαγές. Η πρόσβαση στα μεταδεδομένα ανοίγει τον δρόμο για την αυτοματοποίηση της προετοιμασίας δεδομένων μέσω σεναρίων Python ή του Visual Dataprep.

- ❖ **Visual Dataprep:** Η αυτοματοποίηση των ροών εργασίας προετοιμασίας δεδομένων μπορεί να επιτευχθεί εύκολα, ακόμα και όταν δεν είναι κάποιος προγραμματιστής, με τη χρήση ενός απλοποιημένου οπτικού εργαλείου που επιτρέπει τη δημιουργία μιας <<συνταγής>> φίλτρων και τελεστών. Αυτή η <<συνταγή>> μπορεί να αποθηκευτεί και να επαναχρησιμοποιηθεί σε άλλες σκηνές ή έργα. Με αυτό το εργαλείο, μπορούν να δημιουργηθούν επίπεδα λεπτομέρειας (LODs), να ρυθμιστούν τα Lightmap UVs, να αντικατασταθούν υλικά και να διαγραφούν ή να συγχωνευθούν αντικείμενα με βάση κριτήρια όπως η κλάση, το όνομα, οι ετικέτες μεταδεδομένων ή το μέγεθος.
- ❖ **Ενσωμάτωση ShotGrid:** Η απλοποιημένη σύνδεση μεταξύ των 3D δεδομένων ψηφιακών στοιχείων που δημιουργήθηκαν από καλλιτέχνες σε άλλες εφαρμογές, όπως η Maya, και των δεδομένων εικόνας που πρέπει να αξιολογηθούν από επόπτες και σκηνοθέτες στο ShotGrid, παρέχεται μέσω της ενσωμάτωσης του ShotGrid στην Unreal Engine.
- ❖ **Υποστήριξη LiDAR point cloud:** Τεράστια σύνολα δεδομένων που έχουν συλλεχθεί από τον πραγματικό κόσμο μπορούν να συγκεντρωθούν και να χρησιμοποιηθούν μέσα στην Unreal Engine, με τη δυνατότητα εισαγωγής, οπτικοποίησης, επεξεργασίας και αλληλεπίδρασης με νέφη σημείων που έχουν αποκτηθεί από συσκευές σάρωσης λέιζερ. Αυτά τα νέφη σημείων μπορούν να χρησιμοποιηθούν για την οπτικοποίηση τοποθεσιών και για να παρέχουν ακριβές πλαίσιο σε πρόσφατα σχεδιασμένα στοιχεία.

3.2.2 Κατασκευή Κόσμων

- ❖ **Ο Unreal Editor:** Η μηχανή Unreal ενσωματώνει τον Unreal Editor, ένα ολοκληρωμένο περιβάλλον ανάπτυξης που είναι διαθέσιμο σε Linux, macOS και Windows για τη σύνταξη περιεχομένου και την ανάπτυξη παιχνιδιών. Με την

υποστήριξη της επεξεργασίας από πολλαπλούς χρήστες, καλλιτέχνες, σχεδιαστές και προγραμματιστές μπορούν να πραγματοποιούν ταυτόχρονα αλλαγές στο ίδιο έργο της Unreal Engine με ασφάλεια και αξιοπιστία. Επιπλέον, η δυνατότητα εκτέλεσης του πλήρους Unreal Editor σε λειτουργία VR σημαίνει ότι ο χρήστης μπορεί να δημιουργεί με περιβάλλον εξομοίωσης WYSIWYG.

- ❖ **Μοντελοποίηση, UVs και Bake:** Ένα εκτεταμένο σύνολο εργαλείων δημιουργίας και επεξεργασίας πλεγμάτων είναι διαθέσιμο στην Unreal Engine, το οποίο περιλαμβάνει λειτουργίες υποδιαίρεσης, δυναμικά εργαλεία γλυπτικής και κώδικα γεωμετρίας. Επιπλέον, παρέχεται μια ισχυρή σουίτα εργαλείων δημιουργίας και επεξεργασίας UV, καθώς και μια χρήσιμη επιλογή εργαλείων για το ψήσιμο υφών και τη μεταφορά χαρακτηριστικών πλέγματος. Αυτά τα εργαλεία επιτρέπουν τον χρήστη να βελτιώνει και να επαναλαμβάνει τα ψηφιακά στοιχεία στο τελικό τους πλαίσιο, χωρίς την ανάγκη να μεταβεί σε ένα πακέτο DCC.
- ❖ **Εργαλεία τοπίου και εδάφους:** Με το σύστημα Landscape, ο χρήστης μπορεί να δημιουργήσει περιβάλλοντα και εδάφη ανοιχτού κόσμου τεράστιας κλίμακας, που περιλαμβάνουν βουνά, κοιλάδες και ακόμη και σπηλιές. Μπορεί να προσθέσει πολλαπλά επίπεδα χαρτών ύψους και χρωμάτων και να τα διαμορφώσει και να τα βάψει ανεξάρτητα το ένα από το άλλο. Ο χρήστης έχει επίσης τη δυνατότητα να επεξεργαστεί μη καταστροφικά το τοπίο του, χρησιμοποιώντας ένα επίπεδο που είναι σχεδιασμένο για splines, και να δημιουργήσει μοναδικά προσαρμοσμένα πινέλα μέσω του Blueprint, τα οποία μπορεί να χρησιμοποιήσει για να προσαρμόσει τα τοπία του βάσει άλλων στοιχείων.
- ❖ **Κλιμακούμενη βλάστηση:** Τα τεράστια εξωτερικά περιβάλλοντα καλύπτονται αυτόματα με διάφορους τύπους γρασιδιού, λουλουδιών, μικρών βράχων ή το πλέγμα της επιλογής του χρήστη, χρησιμοποιώντας το Grass Tool, ενώ με το Procedural Foliage Tool δημιουργούνται τεράστια δάση γεμάτα με πολλά διαφορετικά είδη δέντρων και θάμνων, προσομοιώνοντας τον τρόπο με τον οποίο ένα δάσος αναπτύσσεται με την πάροδο του χρόνου.
- ❖ **Κατακερματισμός κόσμου:** Η δημιουργία ανοιχτών κόσμων γίνεται πιο γρήγορη και εύκολη με το World Partition, καθώς ο κόσμος διαιρείται αυτόματα σε ένα πλέγμα και μόνο τα απαραίτητα κελιά μεταφέρονται. Η ομάδα μπορεί να εργάζεται

ταυτόχρονα στην ίδια περιοχή του κόσμου χωρίς να επηρεάζεται η μία από την παρουσία της άλλης, χάρη στο σύστημα << Ένα αρχείο ανά ηθοποιό >>(OFPA). Επιπλέον, με τα επίπεδα δεδομένων είναι δυνατή η δημιουργία διάφορων παραλλαγών του ίδιου κόσμου ως επίπεδα που υπάρχουν στον ίδιο χώρο.

- ❖ **Procedural Content Generation framework:** Οι κανόνες και οι παράμετροι μπορούν να οριστούν μέσω του πλαισίου παραγωγής διαδικαστικού περιεχομένου (PCG) για να γεμίσουν μεγάλες σκηνές με στοιχεία της Unreal Engine της επιλογής του χρήστη. Τα εργαλεία εντός του επεξεργαστή καθιστούν τη δημιουργία μεγάλων κόσμων μια γρήγορη και εύκολη υπόθεση. Επιπλέον, με την προσθήκη ενός στοιχείου χρόνου εκτέλεσης, τα στοιχεία μπορούν να αντιδρούν σε αλλαγές που συμβαίνουν στο παιχνίδι ή στη γεωμετρία.
- ❖ **Asset optimization:** Οι διαδικασίες προετοιμασίας και βελτιστοποίησης ενός πολύπλοκου μοντέλου για βέλτιστη απόδοση σε πραγματικό χρόνο μπορούν να αποτελούν κουραστική και χρονοβόρα εμπειρία, η οποία συχνά περιλαμβάνει πολλαπλά βήματα. Στην Unreal Engine, παρέχονται εργαλεία που διευκολύνουν αυτές τις διαδικασίες. Για παράδειγμα, το εργαλείο αυτόματης δημιουργίας επιπέδων λεπτομέρειας (LOD) επιτρέπει την αυτόματη δημιουργία ποιοτικών επιπέδων λεπτομέρειας για τα μοντέλα ανάλογα την απόσταση που έχει ο θεατής από αυτά. Το εργαλείο Proxy Geometry συνδυάζει πολλαπλά πλέγματα και τα υλικά τους σε ένα μόνο πλέγμα και υλικό, βοηθώντας στη μείωση του αριθμού των αντικειμένων και των υλικών για βελτιωμένη απόδοση. Επιπλέον, υπάρχουν εργαλεία μοντελοποίησης που επιτρέπουν την απλοποίηση των πλεγμάτων και την αφαίρεση των κρυφών πολυγώνων για βελτιωμένη απόδοση.
- ❖ **Φωτισμός ουρανού, σύννεφων και περιβάλλοντος:** Με την Unreal Engine, ο προγραμματιστής μπορεί να δημιουργήσει ρεαλιστικούς ή στιλιζαρισμένους ουραμούς, σύννεφα και άλλα ατμοσφαιρικά εφέ, με πλήρη καλλιτεχνική ελευθερία. Χρησιμοποιώντας το στοιχείο Volumetric Cloud, μπορεί να δημιουργήσει εφέ σύννεφων που αλληλεπιδρούν με το Sky Atmosphere, το Sky Light και έως και δύο κατευθυντικά φώτα. Αυτά τα στοιχεία είναι δυναμικά φωτιζόμενα και σκιασμένα, και μπορούν να ανταποκρίνονται στις ενημερώσεις της ώρας της ημέρας σε πραγματικό χρόνο. Μπορεί να προσαρμόσει τον χρωματικό τους τόνο, την

πυκνότητά τους, τον ρεαλισμό τους και άλλες παραμέτρους για να δημιουργήσει το επιθυμητό ατμοσφαιρικό αποτέλεσμα. Μπορεί να εξομοιώσει διάφορες καιρικές συνθήκες, όπως καλοκαιρινός ή χειμερινός ουρανός, με αντίστοιχη ατμόσφαιρα, σύννεφα και φωτισμό.

- ❖ **Σύστημα νερού:** Με το σύστημα Water της Unreal Engine, ο καλλιτέχνης μπορεί να δημιουργήσει αληθοφανείς υδάτινες μάζες στα τοπία του. Αυτό το σύστημα του επιτρέπει να δημιουργήσει ωκεανούς, λίμνες, ποτάμια και νησιά χρησιμοποιώντας splines, το οποίο του προσφέρει πλήρη καλλιτεχνική ελευθερία. Ένα από τα σημαντικότερα χαρακτηριστικά του συστήματος είναι η ενσωματωμένη προσομοίωση ρευστών. Αυτό σημαίνει ότι οι χαρακτήρες και τα αντικείμενα μπορούν να αλληλεπιδρούν με το νερό ρεαλιστικά. Το νερό αντιδρά στις κινήσεις των χαρακτήρων και των αντικειμένων και απεικονίζει αποτυπώματα όπως οι αντανακλάσεις των κυμάτων στην ακτή και οι αλληλεπιδράσεις με τις ροές των ποταμών.

3.2.3 Characters and Animation

- ❖ **Animation Blueprints:** Με τα Animation Blueprints της Unreal Engine, ο animator μπορεί να δημιουργήσει και να ελέγξει σύνθετες συμπεριφορές για τα κινούμενα σχέδιά του. Ένα Animation Blueprint είναι ένα Blueprint που είναι υπεύθυνο για το πώς θα κινηθεί στο χώρο ένα mannequin. Ο χρήστης επεξεργάζεται τα γραφήματα μέσα στον Animation Blueprint Editor, όπου μπορεί να εκτελέσει blend tree movement, να ελέγξει τα κομμάτια που απαρτίζουν τον εκάστοτε σκελετό ή να ορίσει το πώς θα συμπεριφέρεται η τελική πόζα της κίνησης σε ένα Skeletal Mesh σε κάθε frame.
- ❖ **Machine Learning Deformer:** Το Machine Learning (ML) Deformer είναι ένα πρόσθετο για το Maya που παράγει υψηλής πιστότητας μη γραμμικές προσεγγίσεις παραμορφωτών. Με αυτό το πρόσθετο, ο χρήστης μπορεί να δημιουργήσει σύνθετα προσαρμοσμένα εξαρτήματα ή οποιαδήποτε αυθαίρετη παραμόρφωση χρησιμοποιώντας μοντέλα μηχανικής μάθησης που έχουν εκπαιδευτεί με τη χρήση του πρόσθετου. Αυτά τα μοντέλα μηχανικής μάθησης μπορούν στη συνέχεια να εκτελεστούν σε πραγματικό χρόνο στην Unreal Engine. Αυτό επιτρέπει να

προσομοιώσει παραμορφώσεις υψηλής ποιότητας, όπως μύες που κινούνται, διογκωμένες φλέβες και δέρμα που γλιστράει.

- ❖ **Character animation authoring:** Το Control Rig είναι ένα εργαλείο που επιτρέπει στον animator να δημιουργήσει γρήγορα και εύκολα rigs για χαρακτήρες στην Unreal Engine. Μπορεί να δημιουργήσει rigs που να μπορούν να μοιραστούν σε πολλαπλούς χαρακτήρες και να προσαρμοστούν πλήρως στις ανάγκες του. Χρησιμοποιώντας το Control Rig, μπορεί να ποζάρει και να εμψυχώσει τους χαρακτήρες του στο Sequencer, το οποίο είναι το εργαλείο της Unreal Engine για τη δημιουργία κινούμενων σκηνών και επεξεργασία της χρονολογίας. Μπορεί να δημιουργήσει αληθοφανείς κινήσεις χρησιμοποιώντας ισχυρά χαρακτηριστικά όπως μηχανές καταστάσεων, χώρους ανάμειξης και πλήρους σώματος αντίστροφη κινηματική(FBIK).
- ❖ **Ροή δεδομένων Live Link:** Το πρόσθετο Live Link στην Unreal Engine πράγματι επιτρέπει τη σύνδεση μιας ροής δεδομένων σε πραγματικό χρόνο από εξωτερικές πηγές. Αυτό επιτρέπει στον χρήστη να λάβει δεδομένα κίνησης, όπως κινούμενα σχέδια χαρακτήρων, κάμερες, φώτα και άλλα, από εργαλεία DCC (Digital Content Creation), όπως το Maya ή το MotionBuilder, ή από συστήματα καταγραφής κίνησης ή απόδοσης. Μπορεί να χρησιμοποιήσει το Live Link για να καταγράψει παραστάσεις κίνησης ζωντανά από διάφορες πηγές, συμπεριλαμβανομένης της ανίχνευσης προσώπου ARKit της Apple για την καταγραφή προσώπου από ένα iPhone. Το Live Link είναι επεκτάσιμο μέσω των Unreal Plugins, επιτρέποντας σε τρίτους να προσθέσουν υποστήριξη για νέες πηγές δεδομένων. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να δημιουργήσουν πρόσθετα που υποστηρίζουν άλλα εργαλεία DCC ή συστήματα καταγραφής κίνησης, επεκτείνοντας έτσι τις δυνατότητες του Live Link.
- ❖ **Take Recorder:** Το Take Recorder είναι ένα εργαλείο στην Unreal Engine που επιτρέπει στο χρήστη να καταγράψει κινούμενα σχέδια από διάφορες πηγές κίνησης και να αποθηκεύει αυτές τις εγγραφές για μελλοντική αναπαραγωγή. Μπορεί να καταγράψει κίνηση από την καταγραφή κίνησης που συνδέεται με χαρακτήρες στη σκηνή, δημιουργώντας έτσι μια εγγραφή επιδόσεων. Επίσης, μπορεί να καταγράψει κίνηση από δεδομένα Live Link που προέρχονται από

εξωτερικές πηγές κίνησης, όπως το MotionBuilder ή άλλα συστήματα καταγραφής κίνησης. Οι καταγραφές που δημιουργούνται μπορούν να αποθηκευτούν και να αναπαραχθούν στο μέλλον, επιτρέποντας να επαναλάβει γρήγορα τις εγγραφές επιδόσεων ή να εξετάσει εύκολα προηγούμενες λήψεις. Ένα πλεονέκτημα του Take Recorder είναι ότι μπορεί να οργανώσει τις καταγραφές σε υποσειρές και να τις διαχειριστεί με βάση τα μεταδεδομένα των λήψεων. Αυτό επιτρέπει να διαχειρίζεται ευκολότερα πολύπλοκες παραγωγές

- ❖ **Επαναπροσδιορισμός χαρακτήρων:** Η Unreal Engine παρέχει ένα φιλικό προς τους καλλιτέχνες σύνολο εργαλείων επαναπροσδιορισμού που επιτρέπει εύκολη επαναχρησιμοποίηση υφιστάμενων κινούμενων σχεδίων ακόμη και σε χαρακτήρες με διαφορετικούς σκελετούς και αναλογίες. Για παράδειγμα, ο animator μπορεί να πάρει ένα κινούμενο σχέδιο ανθρώπου και να το επαναχρησιμοποιήσει σε ένα χαρακτήρα λύκου, προσαρμόζοντάς το στο νέο σκελετό και την αναλογία του λύκου. Επιπλέον, μπορεί να δημιουργήσει διαδραστικές λύσεις και στόχους που εκτελούν επεξεργασία πόζας για τα σκελετικά πλέγματα των χαρακτήρων του. Αυτό επιτρέπει να προσαρμόσει τον χαρακτήρα προσθέτοντας επιπλέον κίνηση. Για παράδειγμα, μπορεί να κάνει έναν χαρακτήρα να κοιτάζει πάντα έναν στόχο, προσαρμόζοντας την πόζα του κεφαλιού του χαρακτήρα με βάση τη θέση του στόχου.
- ❖ **Sequencer nonlinear editing and animation:** Το Sequencer είναι ένα πλήρως μη γραμμικό, κινηματογραφικό εργαλείο μοντάζ και animation σε πραγματικό χρόνο σχεδιασμένο από επαγγελματίες του κινηματογράφου και της τηλεόρασης. Επιτρέπει τον animator να ξεκλειδώσει τις δημιουργικές του δυνατότητες και να δημιουργήσει εντυπωσιακά μοντάζ και animation. Με το Sequencer, μπορεί να καθορίσει και να τροποποιήσει το φωτισμό, το κλείδωμα της κάμερας, τους χαρακτήρες και τη διαμόρφωση του σκηνικού ανά πλάνο. Αυτό επιτρέπει να δημιουργήσει εντυπωσιακές σκηνές και να εκφράσει την καλλιτεχνική του όραση. Ένα από τα βασικά πλεονεκτήματα του Sequencer είναι η δυνατότητα συνεργασίας. Ομάδες καλλιτεχνών μπορούν να εργάζονται ταυτόχρονα σε ολόκληρες ακολουθίες, επιτρέποντας την αποτελεσματική συνεργασία και συντονισμό των διαδικασιών δημιουργίας. Επιπλέον, με το Sequencer μπορεί να δημιουργήσει νέα κινούμενα σχέδια αναμειγνύοντας κλιπ κινούμενων σχεδίων.

❖ **Runtime animation tools:** Η Unreal Engine παρέχει διάφορα εργαλεία για τη βελτίωση της συγγραφής κινούμενων σχεδίων κατά την εκτέλεση και την προσαρμογή τους σε διαφορετικά σενάρια παιχνιδιού. Μερικές από αυτές τις δυνατότητες περιλαμβάνουν:

- **Motion Warping:** Το Motion Warping σας επιτρέπει να προσαρμόσετε δυναμικά τη βασική κίνηση ενός χαρακτήρα για να ευθυγραμμιστεί με διαφορετικούς στόχους. Αυτό μπορεί να συμπεριλαμβάνει υπερπήδηση τοίχων διαφορετικού ύψους ή προσαρμογή της κίνησης του χαρακτήρα για να αντιμετωπίσει διάφορα εμπόδια.
- **Distance Matching:** Το Distance Matching σας επιτρέπει να ελέγξετε τον ρυθμό αναπαραγωγής ενός animation κατά την εκτέλεση. Μπορείτε να προσαρμόσετε την ταχύτητα αναπαραγωγής του animation με βάση την απόσταση ανάμεσα στον χαρακτήρα και σε άλλα αντικείμενα ή χαρακτήρες στο παιχνίδι.
- **Pose Warming:** Το Pose Warming σας επιτρέπει να προσαρμόσετε δυναμικά τη στάση του χαρακτήρα κατά την εκτέλεση. Μπορείτε να τροποποιήσετε τη στάση του χαρακτήρα ώστε να συμβαδίζει καλύτερα με την κίνηση του χαρακτήρα στο παιχνίδι. Αυτό μπορεί να χρησιμοποιηθεί για να προσαρμόσετε δυναμικά τη στάση του χαρακτήρα για να φαίνεται πιο φυσική και ρεαλιστική κατά τη διάρκεια της κίνησης.

3.2.4 Rendering, Lighting and Materials

❖ **Nanite & Virtual Shadow Maps:** Το Nanite είναι ένα εικονικοποιημένο σύστημα γεωμετρίας μικροπολυγώνων που ενσωματώνεται στην Unreal Engine. Αυτό το σύστημα μαζί με τους εικονικούς χάρτες σκιών (VSM) επιτρέπει τον σχεδιαστή να δημιουργήσει παιχνίδια και εμπειρίες με τεράστιες ποσότητες γεωμετρικής λεπτομέρειας, γιατί επεξεργάζεται μόνο την πληροφορία που μπορεί να αντιληφθεί ο παίκτης, προσφέροντας εξαιρετική ποιότητα γραφικών. Με το Nanite, μπορεί να εισαγάγει καλλιτεχνικό υλικό κινηματογραφικής ποιότητας που περιλαμβάνει εκατομμύρια πολύγωνα. Αυτό το υλικό μπορεί να τοποθετηθεί εκατομμύρια φορές στον εικονικό κόσμο. Αυτό έχει ως αποτέλεσμα

εντυπωσιακά γραφικά με λεπτομέρεια και πιστότητα σε κάθε πολύγωνο, παρέχοντας μια εμπειρία κινηματογραφικής ποιότητας στους παίκτες ενώ ταυτοχρόνως το frame rate παραμένει σε πολύ υψηλό επίπεδο .

- ❖ **Virtual Texturing:** Η Unreal Engine παρέχει δύο μεθόδους για την υποστήριξη πολύ μεγάλων υφών: το Streaming Virtual Texturing και το Runtime Virtual Texturing.
 - Το **Streaming Virtual Texturing** χρησιμοποιεί μετασχηματισμένα δεδομένα texel από υφές που αποθηκεύονται στον δίσκο. Αυτό επιτρέπει τη μείωση της απαιτούμενης μνήμης για υφές, ειδικά για χάρτες φωτός και λεπτομερείς υφές UDIM UV που έχουν δημιουργηθεί από καλλιτέχνες. Με αυτήν τη μέθοδο, φορτώνονται μόνο τα ορατά πλακίδια υφής κατά τη διάρκεια της εκτέλεσης, εξοικονομώντας μνήμη και βελτιώνοντας την απόδοση.
 - Το **Runtime Virtual Texturing** από την άλλη πλευρά, δημιουργεί δεδομένα texel απευθείας από τη GPU κατά τη διάρκεια της εκτέλεσης. Αυτό βελτιώνει την απόδοση για διαδικαστικά και πολυεπίπεδα υλικά, καθώς διαχειρίζεται δυναμικά τα δεδομένα υφής κατά τη διάρκεια της εκτέλεσης του παιχνιδιού.

- ❖ **Δυναμικός παγκόσμιος φωτισμός Lumen:** Με το Lumen, επιτυγχάνεται πλήρης δυναμικός παγκόσμιος φωτισμός και αντανακλάσεις, επιτρέποντας τη δημιουργία αληθοφανών σκηνών όπου ο έμμεσος φωτισμός προσαρμόζεται δυναμικά στις αλλαγές του άμεσου φωτισμού ή της γεωμετρίας. Αυτό σημαίνει ότι μπορεί να αλλάξει η γωνία του ήλιου ανάλογα με την ώρα της ημέρας, να ανάψει ένας φακός ή να ανοίξει μια εξωτερική πόρτα. Με τη χρήση του Lumen, δεν χρειάζεται πλέον η σύνταξη του UVs lightmap, η αναμονή για την επεξεργασία των lightmaps ή η τοποθέτηση των ανακλαστικών λήψεων.

- ❖ **Photoreal ray tracing in real time:** Δυνατότητα οπτικών στοιχείων ποιότητας Χόλυγουντ out of the box με τον υβριδικό ανιχνευτή ακτίνων που βασίζεται σε φυσική βάση της Unreal Engine. Επιλέγεται επιλεκτικά η ανίχνευση ανακλάσεων, σκιών, ημιδιαφάνειας, απόκρυψης περιβάλλοντος, φωτισμού με βάση την εικόνα και του παγκόσμιου φωτισμού, ενώ συνεχίζεται η

ραστεροποίηση άλλων περασμάτων για την επίτευξη λεπτών, ακριβών εφέ με την απαιτούμενη απόδοση. Τα εφέ περιλαμβάνουν δυναμικές απαλές σκιές από φωτισμούς περιοχής και φως με ιχνηλάτηση ακτίνων από φεγγίτες HDR.

- ❖ **Temporal Super Resolution:** Αναμένεται ότι οι παίκτες των κονσόλων επόμενης γενιάς θα ζητούν ρυθμούς καρέ των 60 fps ή ακόμα υψηλότερους σε οθόνες υψηλής ανάλυσης, γεγονός που θα επιβαρύνει σημαντικά τους πόρους απόδοσης. Το Temporal Super Resolution (TSR), ένα ενσωματωμένο, ανεξάρτητο από την πλατφόρμα, σύστημα upsampling υψηλής ποιότητας, επιτρέπει στη μηχανή να κάνει rendering σε πολύ χαμηλότερη ανάλυση, αλλά με παρόμοια πιστότητα εικονοστοιχείων εξόδου με τα καρέ που δημιουργούνται σε υψηλότερη ανάλυση. Το TSR είναι ενεργοποιημένο από προεπιλογή στην Unreal Engine 5.
- ❖ **Flexible Material Editor:** Ικανότητα για έναν άνευ προηγουμένου έλεγχο της εμφάνισης και της ατμόσφαιρας των χαρακτήρων και των αντικειμένων με τον επεξεργαστή υλικών της Unreal Engine που βασίζεται στη φυσική δομή. Δημιουργήθηκε με ταχύτητα ένα ευρύ φάσμα επιφανειών που αντέχουν οπτικά σε προσεκτική εξέταση χρησιμοποιώντας τη διαισθητική ροή εργασίας που βασίζεται σε κόμβους. Τα υλικά τοποθετήθηκαν σε στρώσεις και οι τιμές προσαρμόστηκαν σε επίπεδο pixel για να επιτευχθεί η επιθυμητή εμφάνιση.
- ❖ **Path Tracer:** Ο ανιχνευτής διαδρομής ενεργοποιείται απευθείας χωρίς την ανάγκη για επιπλέον ρυθμίσεις και αποτελεί μια επιταχυνόμενη μέθοδο προοδευτικής απεικόνισης με τη χρήση του DXR. Παράγει έξοδο τελικού pixel που συνήθως αντιστοιχεί σε αποτελέσματα offline rendering και παρέχει φυσικά ακριβή χαρακτηριστικά, όπως παγκόσμιο φωτισμό που είναι φυσικά σωστός και απαλλαγμένος από συμβιβασμούς, φυσικά σωστές διαθλάσεις, υλικά με πλήρη χαρακτηριστικά που αποτυπώνονται στις ανακλάσεις και τις διαθλάσεις, καθώς και anti-aliasing με υπερδειγματοληψία.
- ❖ **Εκλεπτυσμένος φωτισμός:** Τα ρεαλιστικά εφέ εσωτερικού και εξωτερικού φωτισμού δημιουργούνται διατηρώντας παράλληλα την απόδοση σε πραγματικό χρόνο με ένα ευρύ φάσμα προηγμένων εργαλείων φωτισμού. Αυτά τα εργαλεία περιλαμβάνουν ένα ατμοσφαιρικό περιβάλλον Ήλιου και Ουρανού,

ογκομετρική ομίχλη, ογκομετρικά lightmaps, προ-υπολογισμένα σενάρια φωτισμού και πεδία απόστασης πλέγματος. Αυτά τα εργαλεία επιτρέπουν τη δημιουργία εντυπωσιακών εφέ φωτισμού που ενισχύουν το ρεαλισμό των σκηνών εσωτερικού και εξωτερικού χώρου. Τα εν λόγω εφέ παράγουν ατμοσφαιρικό φωτισμό, προσομοιώνουν την ομίχλη, χρησιμοποιούν προ-υπολογισμένα σενάρια φωτισμού και πεδία απόστασης πλέγματος για να προσφέρουν ρεαλιστικές απεικονίσεις με εξαιρετική απόδοση.

- ❖ **Post-process and screen-space effects:** Επιλέγονται εφέ μετεπεξεργασίας κινηματογραφικής ποιότητας για να ρυθμιστεί η συνολική εμφάνιση και αίσθηση της σκηνής, όπως το HDR bloom, η απεικόνιση τόνου, το lens flare, το βάθος πεδίου, η χρωματική εκτροπή, η βινιέτα και η αυτόματη έκθεση. Οι αντανακλάσεις στο χώρο της οθόνης, η απόκρυψη περιβάλλοντος και ο παγκόσμιος φωτισμός χρησιμοποιούνται για την επίτευξη ρεαλιστικών εφέ με ταυτόχρονη ελαχιστοποίηση του κόστους. Αυτά τα εφέ προσφέρουν προηγμένες δυνατότητες μετεπεξεργασίας που επηρεάζουν την όλη εμφάνιση της σκηνής και παρέχουν κινηματογραφική ποιότητα.
- ❖ **Color-accurate final output:** Η σύνθεση σε πραγματικό χρόνο στον Unreal Editor είναι διευκολυνόμενη από το Composure, τον ενσωματωμένο συνθέτη της Unreal Engine, επιτρέποντας την απόδοση τελικού pixel στην κάμερα. Επιπλέον, μπορούν να εξαχθούν μεμονωμένα περάσματα για σύνθεση εκτός σύνδεσης, ενώ η υποστήριξη του OpenColorIO στο Composure, το viewport, η ουρά απόδοσης ταινιών και το nDisplay, μαζί με τη δυνατότητα εξόδου σε οθόνες HDR με σεβασμό στο πρότυπο ACES, διασφαλίζουν τη συνοχή των χρωμάτων καθ' όλη τη διαδικασία παραγωγής.
- ❖ **Movie Render Queue:** Η Unreal Engine δίνει τη δυνατότητα δημιουργίας υλικού υψηλής ποιότητας για κινηματογραφικές ταινίες, υλικό μάρκετινγκ και γραμμική ψυχαγωγία χωρίς την ανάγκη Post Editing. Το Unreal Engine διαθέτει τη δυνατότητα απόδοσης ταινιών και φωτογραφιών με συσσωρευμένο anti-aliasing και motion blur. Η απόδοση σε κλίμακα υποστηρίζεται, επιτρέποντάς σας να δημιουργήσετε εικόνες υψηλής ανάλυσης, όπως αυτές που προορίζονται για

έξοδο σε έντυπα μέσα. Επιπλέον, μπορούν να εξαχθούν περάσματα απόδοσης για μελλοντική σύνθεση.

- ❖ **Forward rendering:** Ο Forward Shading Renderer παρέχει μια ταχύτερη βασική λειτουργία, με περάσματα απόδοσης υψηλής ταχύτητας, με αποτέλεσμα τη βελτίωση της απόδοσης σε πλατφόρμες VR και ορισμένες κονσόλες όπως το Nintendo Switch. Η υποστήριξη του Multisample Anti-Aliasing είναι επίσης χρήσιμη για εφαρμογές VR, όπου η συνεχής κίνηση κεφαλής εισάγει ανεπιθύμητη θολούρα σε επίπεδο pixel με το Temporal Anti-Aliasing.
- ❖ **Προηγμένα μοντέλα σκίασης:** Τα προηγμένα μοντέλα σκίασης της Unreal Engine περιλαμβάνουν τα μοντέλα Lit, Unlit, Clear Coat, Subsurface Scattering, Skin, Hair, Two-Sided Foliage και Thin Transparency, επιτρέποντας την παραγωγή πιο ρεαλιστικών αποτελεσμάτων σε ένα ευρύ φάσμα αντικειμένων και επιφανειών.

3.2.5 Εξομίωση και εφέ

- ❖ **Σωματίδια Niagara και οπτικά εφέ:** Επίπεδα γυαλίσματος VFX κινηματογραφικής ποιότητας δημιουργούνται σε πραγματικό χρόνο για εφέ όπως φωτιά, καπνός, σκόνη και νερό με τη χρήση πλήρως παραμετροποιήσιμων συστημάτων σωματιδίων στον ενσωματωμένο επεξεργαστή οπτικών εφέ Niagara. Στη σκηνή ασκείται αντίκτυπος με φώτα σωματιδίων, δημιουργείται πολύπλοκη κίνηση σωματιδίων με χρήση διανυσματικών πεδίων, δημιουργούνται εφέ όπως φλοκάτες και αλυσίδες με αλληλεπίδραση μεταξύ των σωματιδίων, και χρησιμοποιείται διεπαφή δεδομένων κυματομορφής ήχου για να αντιδρούν τα σωματίδια στη μουσική ή άλλες πηγές ήχου.

- ❖ **Εργαλεία ένδυσης:** Ο ρουχισμός και άλλα υφάσματα προσομοιώνονται με τη χρήση του solver φυσικής Chaos. Οι παράμετροι του ρουχισμού ορίζονται απευθείας μέσα από τον Unreal Editor, και τα αποτελέσματα μπορούν να παρακολουθηθούν άμεσα για γρήγορη και εύκολη επανάληψη. Το εργαλείο Paint Cloth Tool χρησιμοποιείται για να επιλεγθούν διαισθητικά οι περιοχές του πλέγματος που θα συμπεριφέρονται ως ύφασμα και να καθοριστεί ο βαθμός που θα επηρεάζονται από τη φυσική.
- ❖ **Chaos physics and destruction system:** Το Chaos αναπτύχθηκε ως το επόμενη γενιάς σύστημα φυσικής υψηλής απόδοσης της Unreal Engine. Με τη χρήση της λειτουργίας καταστροφής του Chaos, η δυνατότητα σπασίματος, θρυμματισμού και κατεδάφισης σκηνών μεγάλης κλίμακας σε κινηματογραφική ποιότητα επιτυγχάνεται με εξαιρετικό επίπεδο καλλιτεχνικού ελέγχου. Επιπλέον, το Chaos υποστηρίζει τη στατική δυναμική πλέγματος, υφάσματα, μαλλιά, οχήματα και ακίνητα σώματα για animation, ενώ έχει ενσωματωθεί με το Niagara για την προσθήκη δευτερευόντων εφέ, όπως σκόνη και καπνός.
- ❖ **Strand-based hair and fur:** Με το Chaos, μπορεί να γίνει προσομοίωση και απόδοση εκατοντάδων χιλιάδων φωτορεαλιστικών τριχών σε ταχύτητες έως και πραγματικού χρόνου από τις τρίχες που έχουν δημιουργηθεί σε πακέτα DCC. Αυτό επιτρέπει τη δημιουργία πιο πειστικών ανθρώπινων χαρακτήρων και τριχωτών ή φτερωτών πλασμάτων. Οι κλώνοι μπορούν να παρακολουθούν τις παραμορφώσεις του δέρματος για ρεαλιστική αναπαράσταση της γούνας και των τριχών στο πρόσωπο. Το σύστημα διαθέτει ένα προηγμένο σύστημα shader και απόδοσης μαλλιών, ενώ έχει ενσωματωθεί επίσης η προσομοίωση φυσικής Niagara μέσω του Chaos.

3.2.6 Gameplay and interactivity authoring

- ❖ **Robust multiplayer support:** Με εμπειρία δύο δεκαετιών, έχουν γίνει δοκιμές του τμήματος multiplayer της Unreal Engine σε πολλές πλατφόρμες και τύπους παιχνιδιών, με σκοπό να παράγει μερικές από τις πιο συναρπαστικές εμπειρίες multiplayer στην βιομηχανία. Η Unreal Engine διαθέτει μια αρχιτεκτονική πελάτη/διακομιστή out of the box, η οποία είναι επεκτάσιμη και έχει

δοκιμαστεί, προσφέροντας άμεση βιωσιμότητα στο στοιχείο multiplayer κάθε έργου.

- ❖ **Διαχειριστής παραλλαγών:** Με τη Διαχείριση παραλλαγών, μπορούν να δημιουργηθούν και να επεξεργαστούν παραλλαγές ενός ψηφιακού στοιχείου με επιλογές για την ορατότητα, τους μετασχηματισμούς σχήματος και τις αναθέσεις υλικών, και μπορούν να ενεργοποιούνται ή απενεργοποιούνται στον Unreal Editor ή κατά την εκτέλεσή του. Αυτό είναι ιδανικό για τις αναθεωρήσεις σχεδιασμού και το εμπόριο προσαρμόσιμων προϊόντων, όπως ένα εμπορικό αεροσκάφος που μπορεί να έχει διαφορετικές διατάξεις καμπίνας, επίπλωση και εξαρτήματα. Η αυτόματη δημιουργία παραλλαγών είναι δυνατή μέσω ενός API της Python.
- ❖ **Blueprint visual scripting system:** Με το φιλικό προς τον καλλιτέχνη οπτικό σύστημα σεναρίων Blueprint, μπορούν να δημιουργηθούν γρήγορα πρωτότυπα και να αποσταλεί διαδραστικό περιεχόμενο χωρίς να αγγίξει κανείς ούτε μια γραμμή κώδικα. Τα Blueprints χρησιμοποιούνται για τη δημιουργία συμπεριφορών και αλληλεπιδράσεων αντικειμένων, την τροποποίηση των διεπαφών χρήστη, την προσαρμογή των στοιχείων ελέγχου εισόδου και πολλών άλλων. Οπτικοποιείται η ροή του παιχνιδιού και επιθεωρούνται οι ιδιότητες, ενώ δοκιμάζεται η εργασία χρησιμοποιώντας ισχυρή αποσφαλμάτωση.
- ❖ **Unreal Motion Graphics UI Designer (UMG):** Τα στοιχεία της διεπαφής χρήστη, όπως τα HUDs μέσα στο παιχνίδι/στην εφαρμογή, τα μενού και άλλα γραφικά που σχετίζονται με τη διεπαφή, μπορούν να δημιουργηθούν με το εργαλείο δημιουργίας οπτικού UI UMG και να παρουσιαστούν στους χρήστες. Η διεπαφή κατασκευάζεται με τη χρήση επεξεργάσιμων Blueprint Widgets, τα οποία προσφέρουν προκατασκευασμένες λειτουργίες όπως κουμπιά, πλαίσια ελέγχου, ρυθμιστικά και μπάρες προόδου.
- ❖ **Gameplay frameworks:** Το MassEntity της Unreal Engine επιτρέπει να δημιουργηθούν μεγάλα συστήματα με τεράστιο αριθμό στοιχείων, όπως πλήθη με δεκάδες χιλιάδες αληθοφανείς πράκτορες τεχνητής νοημοσύνης. Το State Tree είναι μια ιεραρχική μηχανή καταστάσεων γενικής χρήσης, ενώ το Enhanced Input παρέχει πολύπλοκο χειρισμό εισόδου, ανακατασκευή ελέγχου κατά τη

διάρκεια εκτέλεσης και άλλες λειτουργίες. Επιπλέον, το Gameplay Ability System επιτρέπει τη γρήγορη δημιουργία συστημάτων παιχνιδιού με βάση τα δεδομένα και με φιλικό τρόπο για τον σχεδιαστή. Αυτά τα ισχυρά πλαίσια παιχνιδιού περιλαμβάνονται στην Unreal Engine και παρέχουν επεκτασιμότητα και ευελιξία στη δημιουργία πολύπλοκων και πλούσιων παιχνιδιών.

Εδώ να σημειωθεί πως η μηχανή παρέχει και άλλα εξίσου σημαντικά χαρακτηριστικά αλλά θα αναφερθούν μόνο ονομαστικά και είναι τα εξής :

- Integrated media support
- Virtual production
- Content
- Developer tools
- Platform support

3.3 Godot [2]

Η **Godot** είναι μια μηχανή παιχνιδιών που έχει την δυνατότητα να σχεδιάσει και να υλοποιήσει παιχνίδια για πολλές πλατφόρμες . Είναι ανοικτού κώδικα , δωρεάν και κυκλοφορεί με άδεια χρήσης MIT δηλαδή ο χρήστης μπορεί να κάνει οτιδήποτε τον κώδικα με την μόνη υποχρέωση απλώς να αναφέρει τον αρχικό δημιουργό ο οποίος δεν θα φέρει καμία ευθύνη αν συμβεί ζημιά από την χρήση της εφαρμογής .Υπεύθυνοι για την δημιουργία της μηχανής είναι οι Αργεντινοί προγραμματιστές Juan Linietsky και Ariel Manzur και στην αρχή κυκλοφόρησε μόνο για εταιρίες στην Λατινική Αμερική πριν από την δημόσια κυκλοφορία της, τον Φεβρουάριο του 2014. Μπορεί να τρέξει σε πολλά λειτουργικά συστήματα όπως macOS, Linux, Windows και BSD .Είναι ικανή να δημιουργήσει εφαρμογές 2D και 3D σε σταθερούς υπολογιστές , κινητές και διαδικτυακές πλατφόρμες και μπορεί επίσης να χρησιμοποιηθεί για την κατασκευή λογισμικού που δεν έχει να κάνει με παιχνίδια. Προσφέρει μια ενσωματωμένη γλώσσα προγραμματισμού την JScript καθώς και υποστήριξη για άλλες γλώσσες όπως C# και Visual Scripting .



Εικόνα 7. Godot

Πηγή: <https://godotengine.org/>

Τα κύρια χαρακτηριστικά της μηχανής είναι :

3.3.1 Σχεδιασμός εμπνευσμένος με γνώμονα την σκηνή

Παρέχει στο χρήστη τη δυνατότητα κατασκευής ενός παιχνιδιού με απλά κομμάτια τα λεγόμενα nodes για να μπορεί να χτίσει επαναχρησιμοποιούμενες και πολύπλοκες σκηνές. Ο χρήστης μπορεί να επεκτείνει την εφαρμογή πέρα από τις δυνατότητες που παρέχονται από τη μηχανή ενσωματώνοντας επεκτάσεις και πρότυπα άλλων προγραμματιστών. Από την 4η έκδοση ο χρήστης μπορεί να δημιουργήσει στοιχεία που βασίζονται σε δεδομένα με προσαρμοσμένους πόρους όπως να ορίσει ένα κωδικοποιημένο αντικείμενο που ονομάζεται πόρος για να περιγράψει χαρακτήρες, οντότητες και δομές δεδομένων στο παιχνίδι που έχει κατασκευάσει και στην συνέχεια να τα χρησιμοποιήσει απευθείας στη μηχανή αντιστοιχίζοντάς τα σε κόμβους(nodes) .

3.3.2 Εργαλεία κωδικοποίησης που ταιριάζουν στις ανάγκες του χρήστη

Με ενσωματωμένη γλώσσα προγραμματισμού την GDScript που είναι εμπνευσμένη από την Python και άλλες γλώσσες είναι εύκολο ακόμα και για αρχάριους να μάθουν να δημιουργούν. Η μηχανή μπορεί να παρέχει υποστήριξη και σε γλώσσες προγραμματισμού που δεν παρέχονται από την ομάδα GODOT χάρη στην μεγάλη κοινότητα που την χρησιμοποιεί. Εάν ο χρήστης γνωρίζει C# τότε μπορεί να αξιοποιήσει την πρότερη εμπειρία που κατέχει καθώς η πλατφόρμα προσφέρει πλήρη υποστήριξη της .NET. Χάρη στην τμηματική δομή και την απλή διαδικασία κατασκευής της Godot ένας προγραμματιστής μπορεί να κατασκευάσει τα δικά του node ή να ενσωματώσει nodes από βιβλιοθήκες τρίτων με χρήση της C++ γλώσσας προγραμματισμού.

3.3.3 Απλή αλλά πολύ ικανή 3D μηχανή

Υποστηρίζει πολλές συσκευές από τις πιο δυνατές όπως σταθερούς υπολογιστές μέχρι τις πιο απλές όπως μεσαίας κατηγορίας κινητά Android. Με τη χρήση της διεπαφής απεικόνισης γραφικών OpenGL ή της Vulkan, δίνει την δυνατότητα ακόμα και στις πιο σύγχρονες κάρτες γραφικών ή στις ενσωματωμένες στον επεξεργαστή να τρέχουν με πολύ αξιόλογα αποτελέσματα την εφαρμογή.

Η μεταφορά των 3D Asset γίνεται πολύ εύκολα και γρήγορα καθώς μπορεί να εισάγει ολόκληρες σκηνές που μπορεί να αποτελούνται από κινούμενα μοντέλα ,φωτισμό χώρου, κάμερες και φυσικά αντικείμενα και παραμετροποιώντας τα να δει αμέσως τις αλλαγές στην μηχανή. Αναγνωρίζει αρχεία τύπου fbx και μπορεί να πάρει κατευθείαν αρχεία από το blender και να τα επεξεργαστεί. (Το blender είναι μια ελεύθερη και ανοικτού κώδικα σουίτα δημιουργίας 3D)

Έχει την ικανότητα να δημιουργήσει κινούμενα βίντεο και προ καταγραφμένες σκηνές. Στην έκδοση 4 του Godot ένας χρήστης έχει την δυνατότητα με την λειτουργία δημιουργίας ταινιών να καταγράψει σκηνές από το έργο ή το παιχνίδι του σε πολύ καλή ποιότητα προσομοίωσης και σε συνδυασμό με την ικανότητα κινουμένων σχεδίων του Godot να κάνει χρήση της μηχανής στο έπακρο.

3.3.4 Πολύ δυνατή μηχανή για 2D γραφικά σε παιχνίδια και εφαρμογές

Το πιο δυνατό σημείο της μηχανής είναι τα πολύ καλά εργαλεία που έχει για την διαχείριση δυσδιάστατων γραφικών όπως συστήματα μονάδων μέτρησης που βασίζονται πάνω σε εικονοστοιχεία (pixel), την απουσία του Z άξονα δηλαδή του βάθους οπότε ο σχεδιαστής επικεντρώνεται καθαρά σε 2 άξονες ώστε να είναι απλή η λογική υλοποίηση του παιχνιδιού παίρνοντας μόνο ως δεδομένα τις συντεταγμένες οθόνης και τα εικονοστοιχεία. Με τον ενσωματωμένο επεξεργαστή διαχείρισης γραφικών καταμερισμένων σε 2D κομματιών (tile map editor) η διαδικασία δημιουργίας ενός επιπέδου μπορεί να γίνει πολύ εύκολη. Ο προγραμματιστής μπορεί να δημιουργήσει μια επέκταση για δημιουργία και χρήση κομματιών με βάση προδιαγραφές ή κάτι τελείως χειροποίητο. Από την μηχανή δεν θα μπορούσε να λείπει ένα αξιόλογο σύστημα διαχείρισης για την δημιουργία του γραφικού περιβάλλοντος χρήστη (GUI). Είναι ένα

ευέλικτο σύστημα που παρέχει την δυνατότητα στον διαχειριστή να το επεκτείνει ανάλογα με τις ανάγκες της εφαρμογής.

3.3.5 Μια εφαρμογή μπορεί να υποστηρίξει πολλές πλατφόρμες

Η μηχανή μπορεί να αναπτύξει και δημοσιεύσει το έργο σε οποιαδήποτε σύγχρονη πλατφόρμα συμπεριλαμβανομένου του σταθερού υπολογιστή, κινητών και διαδικτύου. Ανάλογα με την συσκευή η εφαρμογή μπορεί να έχει διάφορους τρόπους χειρισμού. Ο προγραμματιστής μπορεί να δοκιμάσει την εφαρμογή του κατευθείαν στην συσκευή που θέλει να γίνει η υλοποίηση. Στην έκδοση 4 μπορεί να κάνει κατευθείαν εξομοίωση της λειτουργίας μια εφαρμογής στην αντίστοιχη πλατφόρμα που στοχεύει ή να το εγκαταστήσει και να το τρέξει αμέσως στη εκάστοτε πλατφόρμα. Για να δημιουργηθεί ένα παιχνίδι για κονσόλες θα πρέπει να συνεργαστεί ο εκάστοτε ιδιοκτήτης της εφαρμογής με έναν εκδότη από μια τρίτη εταιρία που ειδικεύεται σε αυτό.

3.3.6 Είναι δωρεάν και ανοικτού κώδικα (Free and Open Source)

Η λήψη μηχανής και η δημιουργία μιας εφαρμογής δεν έχει συμβόλαια και κρυφές χρεώσεις και με βάση την άδεια MIT που έχει η μηχανή για το μόνο πράγμα που είναι υποχρεωμένος ο χρήστης είναι να κάνει μια φιλική αναφορά της χρήσης της εφαρμογής.

Χάρη στην ανοικτού κώδικα βάση δεδομένων μπορεί ο οποιοσδήποτε να ψάξει για πληροφορίες σχετικά με την λειτουργία διάφορων μεθόδων/συναρτήσεων της μηχανής ή ακόμα και για την ίδια την μηχανή χωρίς να περιμένει από κάποια ομάδα υποστήριξη για βοήθεια.

3.4 CryEngine [7]

Η μηχανή παιχνιδιών **CryEngine** αναπτύχθηκε από την εταιρία ηλεκτρονικών παιχνιδιών CRYTEK και πρωτοκυκλοφόρησε το 2004. Έχει χρησιμοποιηθεί στην ανάπτυξη πολλών τίτλων παιχνιδιών όπως το FARCRY, το CRYISIS και το Kingdom Come. Η εταιρεία συνεχίζει να ενημερώνει την μηχανή ώστε να υποστηρίζει νέες κονσόλες και υλικό για τα παιχνίδια τους. Άλλες εταιρίες χρησιμοποιούν αδειές της CRYTEK για να εκδώσουν τους

δικούς τους τίτλους όπως το Sniper : Ghost Warrior 2 και το Snow. Η μηχανή δεν είναι εύκολη για αρχάριους χρήστες γιατί χρειάζεται κάποιες γνώσεις προγραμματισμού και γνώση από έννοιες ανάπτυξης παιχνιδιών. Ο προγραμματιστής πρέπει να είναι κάλος γνωστής της C++ γλώσσας προγραμματισμού. Ένα από τα δυνατά σημεία της μηχανής είναι τα εξαιρετικά ρεαλιστικά γραφικά που μπορεί να περιλαμβάνουν φωτορεαλιστικά τοπία, ανθρώπινους χαρακτήρες με πολύ μεγάλη λεπτομέρεια στα χαρακτηριστικά τους όπως και αντικείμενα. Με τις πιο πρόσφατες αναβαθμίσεις μπορεί να υποστηρίξει σε μεγάλο βαθμό εφαρμογές εικονικής πραγματικότητας και να παράγει πολύ ρεαλιστικές και υψηλής ποιότητας VR εμπειρίες. Παρόλο που η μηχανή χρησιμοποιεί την C++ γλώσσα προγραμματισμού για να κάνει πιο εύκολη την χρήση της παρέχει στους χρήστες έναν sandbox editor που τους επιτρέπει να δημιουργούν εφαρμογές με μια φιλική για αυτούς διεπαφή.



Εικόνα 8. CryEngine

Πηγή: <https://www.cryengine.com/>

Τα κύρια χαρακτηριστικά της μηχανής είναι :

3.4.1 Γραφικά (Visuals)

Οι δυνατότητες των γραφικών που παρέχει η μηχανή:

- ❖ Ένα από τα δυνατά σημεία της μηχανής είναι τα φώτα περιοχής (Area Lights) τα οποία βελτιώνουν αισθητά τον φωτισμό του χώρου με ρεαλιστικές πηγές φωτός. Αυτές οι πηγές φωτός παίρνουν συγκεκριμένα σχήματα σε μια περιοχή και αυτό τις κάνει πιο ρεαλιστικές γιατί παρέχουν μια καλύτερη αναπαράσταση στο πώς συμπεριφέρεται το φως.
- ❖ Η μηχανή υποστηρίζει πλήρως την DirectX 12 της Microsoft στην έκδοση CRYENGINE V με αποτέλεσμα οι επιδόσεις και τα γραφικά να έχουν αναβαθμιστεί πάρα πολύ σε σχέση με προηγούμενες εκδόσεις.

- ❖ Για να έχει μια πιο ρεαλιστική απεικόνιση πάνω στις επιφάνειες αντικειμένων και φωτισμού η CryEngine χρησιμοποιεί ένα PBR σύστημα που προσομοιώνει φυσικές συνθήκες. Αυτό το μοντέλο προσομοιώνει την αλληλεπίδραση μεταξύ του φωτός και των υλικών όπως γίνεται στον πραγματικό κόσμο. Το αποτέλεσμα είναι τα αντικείμενα να μοιάζουν πιο αληθοφανή ανεξάρτητα από τις συνθήκες φωτισμού. Αυτό σημαίνει πως δεν χρειάζεται η χρήση ιδιαίτερων παραμέτρων για ειδικά εφέ και η ανάπτυξη και δημιουργία γίνεται πιο εύκολη και απλή.
- ❖ Η CryEngine έχει την δυνατότητα να δημιουργεί πολύ εντυπωσιακές υδάτινες επιφάνειες εύκολα καταγράφοντας 1:1 τις διακυμάνσεις του νερού όταν έρχονται σε επαφή με κάποιο αντικείμενο το οποίο δημιουργεί κυματισμούς πάνω στην επιφάνεια του νερού με αποτέλεσμα να έχουμε πολύ ρεαλιστικά αποτελέσματα.
- ❖ Η δυνατότητα επιλογής του επιπέδου tessellation εξασφαλίζει ότι το περιβάλλον της σκηνής θα έχει πολύ μεγάλη λεπτομέρεια. Με την υποστήριξη της επιτάχυνσης υλικού του tessellation σε όλα τα meshes συμπεριλαμβανομένων των κινουμένων χαρακτήρων αυτό γίνεται πολύ αποδοτικό και φαίνεται στις επιδόσεις του παιχνιδιού. Υποστηρίζει 3 τύπους tessellation το Phong, PN triangles και το Displacement Mapping. Η πρώτη και η δεύτερη χρησιμοποιούνται σε απλά μοντέλα χαμηλής πολυπλοκότητας και με την χρήση του tessellation αυτά τα μοντέλα αποκτούν υψηλότερη ακρίβεια και ρεαλιστική εμφάνιση. Ο τρίτος τρόπος που είναι ο πιο σημαντικός γιατί δίνει ακόμα πιο πολύ έλεγχο είναι η χαρτογραφημένη μετατόπιση (Displacement Mapping) όπου η μετατόπιση πραγματοποιείται με την χρήση ενός χάρτη ύψους. Όλες αυτές οι μέθοδοι μπορούν να επεξεργαστούν ξεχωριστά κάθε ένα αντικείμενο μέσα στην σκηνή και ο προγραμματιστής αποφασίζει ποια θα επιλέξει και που για να έχει την καλύτερη δυνατή απόδοση αναμεσα στην οπτική πιστότητα και την ομαλή λειτουργία του παιχνιδιού.
- ❖ Με τις 3D HDR φωτοβολίδες φακού (Lens Flare) ο σχεδιαστής μπορεί να παράγει ακτίνες φωτός που πηγάζουν από μια πηγή φωτός παρόμοια με αυτές που χρησιμοποιούνται στη βιομηχανία του κινηματογράφου εύκολα και γρήγορα. Το σύστημα αυτό ονομάζεται composite 3D Lens Flare. Με την χρήση του Flare Editor ο χρήστης μπορεί εύκολα και γρήγορα να αφαιρέσει ή να προσθέσει εφέ για την όσο τον δυνατόν πιο όμορφη εμφάνιση. Ο διαχειριστής ακτίνων φωτός παρέχει άμεση

ανατροφοδότηση (feedback) για τις επιλογές των προγραμματιστών. Ο χρήστης μπορεί να συνδυάσει πολλές επιλογές και να δημιουργήσει ένα πρότυπο το οποίο μπορεί να εφαρμοστεί σε οποιαδήποτε πηγή φωτός συμπεριλαμβανομένου του ηλίου.

- ❖ Με μια αποδοτική διαδικασία Anti-Aliasing εξαλείφει τις ακατέργαστες άκρες στις εικόνες και πλέον ο προγραμματιστής μπορεί να επικεντρωθεί σε άλλα στοιχεία του παιχνιδιού. Με τις καινούργιες τεχνικές Anti-Aliasing οι οποίες έχουν δοκιμαστεί σε διαφορά παιχνίδια εισάγει μια εξελιγμένη λύση η οποία είναι αρκετά αποδοτική και την καταστεί ικανή να τρέχει σε όλες τις πλατφόρμες. Για να επιτύχουμε την καλύτερη απόδοση και ποιότητα στα γραφικά γίνεται μια προεπεξεργασία στους χάρτες γεωμετρίας ώστε να μειωθεί το specular aliasing ενώ προαιρετικά υπάρχει ειδικό φίλτρο για να διορθώνει τα τρεμοπαίγματα στον φωτισμό περιγράμματος στη λεπτή γεωμετρία όπως σχοινιά, φύλλα, συρματοπλέγματα κτλ. Τέλος μέσω της υπερδειγματοληψίας μπορεί να επιτευχθεί ακόμα υψηλότερη πιστότητα εικόνας.
- ❖ Μια ακόμα δυνατότητα που παρέχει η μηχανή είναι το θόλωμα κίνησης (Motion Blur) και βάθος πεδίου (Depth of Field). Με αυτά τα ειδικά εφέ ένας παίχτης βυθίζεται περισσότερο στο περιβάλλον του παιχνιδιού. Το Motion Blur μπορεί να εφαρμοστεί είτε σε μεμονωμένα αντικείμενα (με βάση το αντικείμενο) είτε σε ολόκληρη την σκηνή (με βάση την οθόνη) ενώ τα εφέ βάθους πεδίου μπορούν να οριστούν με βάση την οθόνη εύκολα μέσω του sandbox editor. Η CryEngine πλέον αυτά τα εφέ τα υλοποιεί με πολύ μικρό κόστος επεξεργαστικής ισχύος σε όλες τις υποστηριζόμενες πλατφόρμες.
- ❖ Ένα ακόμα δυνατό σημείο της μηχανής είναι οι τοπικές αντανάκλασεις σε πραγματικό χρόνο δηλαδή η μηχανή μπορεί να αναπαριστά με ακρίβεια τις αντανάκλασεις που δημιουργούν και δέχονται τα αντικείμενα σε πραγματικό χρόνο. Οι αντανάκλασεις έχουν πολύ μεγάλο κόστος στις επιδόσεις των παιχνιδιών ιδίως σε μηχανές που χρησιμοποιούν την αναβαλλόμενη τεχνική απεικόνισης γραφικών (deferred rendering). Η CryEngine παρέχει μια εκπληκτική τεχνολογία αντανάκλασεων την λεγόμενη αντανάκλαση του χώρου της εικόνας. Η τεχνολογία SSR προσεγγίζει την HDR ανιχνεύοντας με ακρίβεια τις αντανάκλασεις τοπικά στα

αντικείμενα πράγμα που επιτρέπει σε κάθε είδους επιφάνεια αντικειμένου να αντανακλά το κοντινό περιβάλλον σε πραγματικό χρόνο.

- ❖ Από την μηχανή δεν θα μπορούσε να λείπει η δυνατότητα δημιουργίας ρεαλιστικής βλάστησης που κινείται και αντιδρά σε κάθε αλληλεπίδραση. Παιχνίδια όπως το Ryse και το Crysis 3 είναι ένα παράδειγμα των δυνατοτήτων αυτής της λειτουργίας. Το σύστημα βλάστησης επιτρέπει τη δημιουργία ενός ζωντανού οικοσυστήματος που προσδίδει ζωντάνια στις στατικές σκηνές. Αυτή η τεχνική μπορεί να προσομοιώσει από τα πιο μικρά χορταράκια μέχρι ολόκληρα δάση ώστε να αλληλοεπιδρά με τον παίκτη, την AI, τα δυναμικά αντικείμενα, τις εκρήξεις και τον άνεμο. Η διαλειτουργικότητα μεταξύ της εξομοίωσης φυσικής και της απεικόνισης των γραφικών επιτυγχάνεται επειδή η CryEngine είναι υλοποιημένη ώστε να διαχειρίζεται όλα μαζί στο ίδιο πακέτο.
- ❖ Από τις δυνατότητες της μηχανής δεν θα μπορούσε να λείπει η διαχείριση του παγκοσμίου φωτισμού (Global Illumination). Το σύστημα παγκοσμίου φωτισμού έχει να κάνει με το πώς συμπεριφέρεται το έμμεσο φως μέσα στην σκηνή το οποίο δημιουργείται με την ανάκλαση και την σκέδαση του φωτός από τις επιφάνειες του περιβάλλοντος. Η μηχανή παρέχει ένα σύστημα βασισμένο πάνω στα Voxel το οποίο δημιουργεί ένα πλέγμα από τετράγωνα κομμάτια για την αναπαράσταση της σκηνής και των πληροφοριών του φωτισμού σε ένα 3D χώρο. Ο αλγόριθμος υπολογίζει τον φωτισμό σε κάθε τετράγωνο ακολουθώντας την πορεία των ακτίνων του φωτός μέσα στην σκηνή έχοντας υπόψιν τις ιδιότητες των υλικών κάθε αντικειμένου και τις αλληλεπιδράσεις μεταξύ των επιφανειών. Αυτό έχει ως αποτέλεσμα ρεαλιστικά και ακριβή εφέ φωτισμού όπως ο έμμεσος φωτισμός, η χρωματική αιμορραγία και οι απαλές σκιές. Σε σύγκριση με προ ψηφμένες καταστάσεις φωτισμού το Voxel δεν απαιτεί πολύ χρόνο για να υπολογιστεί και δεν καταλαμβάνει πολύ χώρο στον σκληρό δίσκο. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για μεγάλα περιβάλλοντα ανοικτού κόσμου με δυναμικό σύστημα μέρας και νύχτας. Αυτό έχει ως αποτέλεσμα να διευκολύνει πολύ την δουλειά του προγραμματιστή επιτρέποντάς του να δοκιμάζει άμεσα τον φωτισμό σε πραγματικό χρόνο χωρίς να περιμένει ώρες ποτέ η μηχανή θα τελειώσει το ψήσιμο του φωτός.

- ❖ Ένα πολύ σημαντικό και δύσκολο στοιχείο στην διαχείριση γραφικών είναι οι σκιές. Η μηχανή μπορεί να διαχειριστεί τις σκιές κάθε αντικειμένου με την χρήση χαρτών (per object shadow map). Έτσι μπορούν να δημιουργηθούν προσαρμοσμένοι χάρτες σκιών που μπορούν να ανατεθούν σε οποιαδήποτε οντότητα στην σκηνή. Αυτό οδηγεί σε πολύ ρεαλιστικές σκιές λόγω της μεγαλύτερης πληροφορίας σκιών σε πιο κοντινή απόσταση. Χρησιμοποιείται πάρα πολύ σε ενδιάμεσων σκηνών βίντεο (cut scene videos) ή για να συμπεριληφθεί περισσότερη πληροφορία σκιών σε κάποιο αντικείμενο του χώρου. Αυτή την δυνατότητα την διαχειρίζεται ο σχεδιαστής μέσω ενός γραφήματος ροής για εύκολη πρόσβαση στην καλύτερη δυνατή ποιότητα δυναμικών σκιών σε πραγματικό χρόνο.
- ❖ Για να έχει πιο ρεαλιστική απόδοση σκιών στο παιχνίδι συνήθως χρησιμοποιείται η μέθοδος Screen Space Ambient Occlusion (SSAO) η οποία είναι υπεύθυνη για την δημιουργία σκοταδιού στις γωνίες και στις σχισμές που συμβαίνει όταν το φως παρεμποδίζεται από κοντινά αντικείμενα. Η CryEngine για να το πάει ένα βήμα παρακάτω εισάγει το Screen Space Directional Occlusion (SSDO) το οποίο συμπεριλαμβάνει οποιαδήποτε αριθμό φωτών στον υπολογισμό του Ambient Occlusion και την συνεκτίμηση του χρώματος όλων των φωτών. Με αυτό τον τρόπο βελτιώνει την ακρίβεια του φυσικού φωτισμού και επιτρέπει την δημιουργία σκιών από κάθε πηγή φωτός στο περιβάλλον.
- ❖ Η κινηματογραφική χαρτογράφηση τόνων είναι μια τεχνική για την προσαρμογή της εικόνας ή της σκηνής με φωτισμό υψηλού δυναμικού εύρους. Ο σχεδιαστής φωτισμού έχει την δυνατότητα να προσαρμόσει την εικόνα ή σκηνή μέσω μιας ημιτονοειδούς καμπύλης για την δημιουργία μιας πιο ρεαλιστικής και οπτικά ευχάριστης εικόνας μιμούμενη τον τρόπο με τον οποίο οι κινηματογραφικές κάμερες συλλαμβάνουν το φως και το χρώμα.
- ❖ Ο μηχανισμός φωτισμού (IBL) με βάση την εικόνα είναι υπεύθυνος για να αποδίδει ρεαλιστικό φως στα αντικείμενα χρησιμοποιώντας προηγμένες τιμές της συνάρτησης κατανομής στιγματίωτων (Angular Bidirectional Reflectance Distribution Function values ή ABRDF ή PBR) και τις κανονικοποίησης των φωτεινών περιοχών στις επιφάνειες των υλικών που αντανακλάνε το φως, όπως το γυαλί ή το μέταλλο για την απόδοση πολύπλοκων καταστάσεων φωτισμού. Το IBL κάνοντας χρήση της

κατοπτρικής ανάκλασης και της τραχύτητας των υλικών αποδίδει μια πιο φυσική εμφάνιση στα αντικείμενα της σκηνής.

- ❖ Ένα από τα τελευταία δυνατά στοιχεία της μηχανής είναι το σύστημα για τα ειδικά εφέ το οποίο χρησιμοποιεί την κάρτα γραφικών (GPU) για την επεξεργασία και την διαχείριση των εφέ και έτσι οι προγραμματιστές να μπορούν να επιτύχουν σημαντικά υψηλότερη απόδοση με πολύ καλύτερη ποιότητα στα ειδικά εφέ.
- ❖ Οι ογκομετρικές σκιές ομίχλης (volumetric fog shadows) επιτρέπουν την δημιουργία σκιών από αντικείμενα και σύννεφα πάνω στην ίδια την ογκομετρική ομίχλη προσδίδοντας ακόμα ένα επίπεδο ατμοσφαιρικής αυθεντικότητας στο περιβάλλον της σκηνής.
- ❖ Τέλος η CryEngine έχει την δυνατότητα να δίνει στα αντικείμενα εφέ τρισδιάστατης επιφάνειας και σε κάποιο βαθμό να επηρεάζει και την όψη της γεωμετρίας στην επιφάνεια στην οποία εφαρμόζεται μέσω χαρτογράφησης της απόκρυψης παράλλαξης (POM) και χαρτογράφηση μετατόπισης εξογκωμάτων και ρυτίδων (OBM) τα οποία είναι αντίστοιχα του Tessellation και του Displacement αλλά είναι πολύ λιγότερο κοστοβόρες.

3.4.2 Sandbox

Οι δυνατότητες που παρέχει το Sandbox της μηχανής:

- ❖ Το sandbox της CryEngine είναι ένα ολοκληρωμένο εργαλείο περιεχομένου που παρέχει στους προγραμματιστές τον πλήρη έλεγχο των δημιουργιών τους. Αυτό δεν περιορίζεται μόνο σε μια πλατφόρμα και ο χρήστης μπορεί σε πραγματικό χρόνο να δει τις αλλαγές που έχει κάνει έχοντας στην διάθεση του μια σειρά από εργαλεία τα οποία επιτρέπουν την άμεση εξομοίωση παιχνιδιού κατά την δημιουργία του. Η σουίτα εργαλείων <<Αυτό που βλέπεις αυτό είναι>> (WYSIWYG) επιτρέπει την δημιουργία, την επεξεργασία και την προεπισκόπηση σε πραγματικό χρόνο της κάθε πτυχής και χαρακτηριστικού ενός παιχνιδιού.
- ❖ Ο διαχειριστής υλικών είναι υπεύθυνος για την βελτιστοποιημένη παλέτα σκιάσεων η οποία προσφέρει καλύτερη απόδοση στα μοντέλα του παιχνιδιού με αποτέλεσμα αυτά να είναι πιο ζωντανά και η διαχείρισή τους να είναι πιο ευέλικτη.

- ❖ Υποστήριξη των αρχείων με επέκταση FBX. Αυτή η μορφή αρχείων υποστηρίζεται ευρέως στη βιομηχανία και χρησιμοποιείται σε διάφορους τομείς όπως τα παιχνίδια, ο κινηματογράφος και η διαφήμιση. Μπορεί να εισάγει από διάφορες εφαρμογές λογισμικού 3D μοντελοποίησης και κινουμένων σχεδίων. Επιτρέπει την μεταφορά πληροφοριών όπως γεωμετρίας, υφές, φωτισμός, κινούμενα σχέδια ακόμη και χαρακτηριστικά που καθορίζονται από τον χρήστη μεταξύ διαφορετικών εφαρμογών.
- ❖ Ανάλογα με το επίπεδο του σχεδιαστή η μηχανή επιτρέπει πολύ γρήγορα την δημιουργία εντυπωσιακών επιπέδων και με την βοήθεια του sandbox παρέχει τα απαραίτητα εργαλεία για να διαμορφώσει το επίπεδο όπως ακριβώς θέλει. Είτε κάνοντας χρήση των πρωτοτύπων (prototyping) με το εργαλείο σχεδιασμού που παρέχει η πλατφόρμα, είτε γεμίζοντας σταδιακά τον χώρο με το εργαλείο της χλωρίδας (Vegetation tool) ή οποιοδήποτε άλλο εργαλείο ο σχεδιαστής έχει τον πλήρη έλεγχο.
- ❖ Ο διαχειριστής κινηματογραφικών σκηνών (TrackView Cinematic Editor) είναι ένα εργαλείο που επιτρέπει στους προγραμματιστές να δημιουργούν και να διαχειρίζονται συνθέτες κινηματογραφικές ακολουθίες. Είναι ένα ενσωματωμένο εργαλείο επεξεργασίας σκηνών για την δημιουργία δια δραστικών ακολουθιών οι οποίες είναι χρονικά εξαρτημένες από αντικείμενα και γεγονότα. Η δημιουργία κινηματογραφικών σκηνών ή γεγονότων από κώδικα γίνεται εύκολη επιτρέποντας στον χρήστη να δημιουργήσει ακολουθίες από αντικείμενα, κινούμενα σχέδια και ήχους μέσα στην σκηνή.
- ❖ Ένα ακόμα δυνατό εργαλείο είναι το γράφημα ροής το οποίο επιτρέπει την κατασκευή πολύπλοκων επιπέδων χωρίς να χρειάζεται να γραφεί ούτε μια γραμμή κώδικα. Αυτό είναι ένα σύστημα οπτικού προγραμματισμού (visual scripting) που είναι απλό αλλά ταυτόχρονα ισχυρό παρέχοντας στους χρήστες ένα περιβάλλον για τον έλεγχο και την δημιουργία όλων των σταδίων υλοποίησης του παιχνιδιού. Με τις νέες προσθήκες που παρέχουν βελτιωμένη διαχείριση ροής εργασίας όπως η αποσφαλμάτωση σε πραγματικό χρόνο (debugging), τα σημεία διακοπής (breakpoints) και οι ενότητες δίνουν την δυνατότητα στον χρήστη να επικεντρωθεί στην βελτίωση του παιχνιδιού.

- ❖ Με την ενσωμάτωση της εφαρμογής substance που πλέον είναι μέρος της σουίτας της adobe ο χρήστης έχει στα χέρια του ακόμα ένα ισχυρό εργαλείο για την διαχείριση των materials.

3.4.3 AI & Animation

Οι δυνατότητες των Κινουμένων Σχεδίων και της Τεχνητής Νοημοσύνης της Μηχανής είναι οι εξής:

- ❖ Η τεχνολογία διαχείρισης χαρακτήρων δίνει την δυνατότητα στον σχεδιαστή να μπορεί να δημιουργήσει τους πιο λεπτομερείς χαρακτήρες. Από τις ρυτίδες στο δέρμα και τις τρίχες στο κεφάλι η CryEngine παρέχει ένα κλιμακούμενο και τεχνικά προηγμένο σύστημα animation και rendering για να παρέχει εκπληκτικά ρεαλιστικούς χαρακτήρες σε παιχνίδια πραγματικού χρόνου. Για τη δημιουργία αυτών των λεπτομερών χαρακτήρων χρειάζονται ειδικά εφέ όπως :
 - Υποεπιφανειακή σκέδαση στο χώρο της οθόνης για όλες τις υποστηριζόμενες πλατφόρμες (Screen-space sub-surface scattering ή SSSSη οποία χρησιμοποιείται στις μηχανές παιχνιδιών για την προσομοίωση της σκέδασης του φωτός κάτω από την επιφάνεια ημιδιάφανων υλικών, όπως το δέρμα, το κερί η το μάρμαρο. Είναι ένας αποδοτικός τρόπος προσέγγισης της εμφάνισης των προαναφερόμενων χωρίς την χρήση ακριβών τεχνικών όπως η ογκομετρική απόδοση σε πραγματικό χρόνο.
 - Η απεικόνιση παράλλαξης ίριδας είναι μια μέθοδος κατά την χρήση γραφικών υπολογιστή για τη δημιουργία μιας πιο ρεαλιστικής αναπαράστασης των ματιών σε τρισδιάστατα μοντέλα. Περιλαμβάνει την προσομοίωση του βάθους της ίριδας με τη χρήση μιας τεχνικής χαρτογράφησης παράλλαξης που υπολογίζει τον τρόπο με τον οποίο η ίριδα κινείται ανταποκρινόμενη στις διαφορετικές γωνίες θέασης. Αυτό δημιουργεί μια πιο αληθοφανή εμφάνιση για τα μάτια, καθώς φαίνεται να έχουν μεγαλύτερο βάθος και ρεαλισμό.
 - Νέοι ενοποιημένοι shaders για σκιές και μαλλιά χαρακτήρων. Με την ενοποίηση των shaders που χρησιμοποιούνται για την απόδοση των σκιών και των

μαλλιών, η CryEngine είναι σε θέση να παράγει πιο ακριβή και οπτικά ευχάριστα αποτελέσματα, μειώνοντας παράλληλα τον φόρτο εργασίας της μηχανής.

- ❖ Για την κίνηση των μοντέλων η μηχανή κάνει χρήση Παραμετρικού σκελετικού animation (Parametric Skeletal Animation η αλλιώς PSA) το οποίο αναμειγνύει τη χρήση ενός συνόλου παραμέτρων για τον καθορισμό της κίνησης ενός χαρακτήρα, αντί της χειροκίνητης εμπύχωσης κάθε οστού και άρθρωσης. Οι χαρακτήρες περιηγούνται και αντιδρούν στο περιβάλλον εν κινήσει σε διάφορες αλλαγές όπως την κίνηση, την κατεύθυνση ή την ταχύτητα κάνοντας ανάμειξη μεταξύ των animations. Αυτό επιτρέπει μεγαλύτερη ευελιξία και προσαρμογή, καθώς και μείωση του όγκου των δεδομένων κινούμενων σχεδίων που χρειάζεται να αποθηκευτούν και να υποστούν επεξεργασία.
- ❖ Η τεχνολογία αποθήκευσης γεωμετρίας (Geometry Cache) στην CRYENGINE επιτρέπει την αποθήκευση και αναπαραγωγή πολύπλοκων γεωμετρικών δομών με χρήση τεχνικών αποθήκευσης σημείων (point caching), οι οποίες αρχικά αναπτύχθηκαν για τη βιομηχανία των κινηματογράφων. Οι τεχνικές αυτές επιτρέπουν τη μείωση του μεγέθους των δεδομένων σε ποσοστό μόλις 10% του αρχικού μεγέθους του αρχείου, διασφαλίζοντας ταυτόχρονα την αποτελεσματική απεικόνιση της γεωμετρίας. Ο μηχανισμός απεικόνισης, γνωστός ως 'GeomCache', λειτουργεί με την ίδια αποδοτικότητα με τη στατική γεωμετρία, με δυνατότητα συμπίεσης Lossy-to-Lossless, υποστήριξη για την εγκατάσταση πολλαπλών αντικειμένων, streaming και αναπαραγωγή με αποθήκευση στη μνήμη.
- ❖ Η Πολυεπίπεδη Δομή Πλοήγησης (Multi-Layer Navigation Mesh - MNM) είναι μια εξελιγμένη δυναμική δομή δεδομένων πλοήγησης, που χρησιμοποιείται από την τεχνητή νοημοσύνη για να βρίσκει την βέλτιστη διαδρομή για τους NPC μέσα στα επίπεδα του παιχνιδιού. Αυτό το σύστημα έχει ολοκληρωθεί ομαλά στο CRYENGINE Sandbox, επιτρέποντας στους σχεδιαστές να καθορίσουν μια περιοχή όπου θα πρέπει να δημιουργηθεί η δομή πλοήγησης, καθώς και να ανανεώνεται δυναμικά κατά την εξέλιξη του παιχνιδιού. Η MNM υποστηρίζει επίσης πραγματικού χρόνου ενημερώσεις, επιτρέποντας την αυτόματη προσαρμογή της δομής πλοήγησης σε αλλαγές στο περιβάλλον του παιχνιδιού.

- ❖ Η ρεαλιστική αναπαράσταση των κινούμενων χαρακτήρων απαιτεί τη χρήση προηγμένων συστημάτων τεχνητής νοημοσύνης που θα ανταποκρίνονται έξυπνα στο περιβάλλον του παιχνιδιού και θα διατηρούν την ψευδαίσθηση του ρεαλισμού. Η CRYENGINE διαθέτει ισχυρή τεχνολογία τεχνητής νοημοσύνης που διαχειρίζεται τις συμπεριφορές των χαρακτήρων μέσω modular sensory systems, όπως η ακοή και η όραση, και υποστηρίζει τις απαιτήσεις του συστήματος κίνησης. Το σύστημα τεχνητής νοημοσύνης ενσωματώνεται άριστα στη CRYENGINE και δεν απαιτεί επιπλέον λογισμικό, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στο δημιουργικό κομμάτι και όχι στον τρόπο ενσωμάτωσης στην εφαρμογή.
- ❖ Έχει απλοποιήσει τη διαδικασία ενισχυμένης προσθήκης φυσικών προσαρμογών στα μοντέλα χαρακτήρων, προσφέροντας περισσότερη δυναμική και λεπτομέρειες στους εικονικούς σας χαρακτήρες. Αυτό επιτυγχάνεται μέσω της προσομοίωσης φυσικών στοιχείων, όπως σχοινιά, υφάσματα που προσαρτώνται στους χαρακτήρες σας με βάση τη φυσική τους συμπεριφορά. Οι νέοι αλγόριθμοι προσομοίωσης έχουν σχεδιαστεί ειδικά για αποτελεσματική, σταθερή και γρήγορη επεξεργασία των χαρακτήρων, προσφέροντας ρεαλιστική απεικόνιση σε υψηλές ταχύτητες καρτέ. Οι προηγμένες τεχνικές αντίδρασης συγκρούσεων επιτρέπουν την ακριβή αλληλεπίδραση μεταξύ των μοντέλων και των χαρακτήρων, προσφέροντας πολύ ρεαλιστικές σκηνές και πολύ καλό ρυθμό μετάδοσης.
- ❖ Προσαρμοστικοί αλγόριθμοι όπως CC D-IK, αναλυτική IK, SB IK ή φυσικές προσομοιώσεις χρησιμοποιούνται για να ενισχύσουν προ-δημιουργημένες κινήσεις, αποφεύγοντας την τυπική υπολογιστική όψη. Η κλιμακούμενη τεχνική της παραμόρφωσης κίνησης της CRYENGINE διατηρεί το στυλ και το περιεχόμενο της βασικής κίνησης, λαμβάνοντας πλήρως υπόψη τους περιορισμούς που επιβάλλονται από ένα υψηλά διαδραστικό περιβάλλον.

3.4.4 Audio

Οι δυνατότητες του ήχου που παρέχει η μηχανή είναι:

- ❖ Ο ήχος παίζει καθοριστικό ρόλο στη δημιουργία καθηλωτικών και ελκυστικών εμπειριών. Για τον εξορθολογισμό της διαδικασίας παραγωγής ήχου, ο CRYENGINE Sandbox Editor εισάγει τον Audio Controls Editor (ACE). Αυτό το προηγμένο

εργαλείο προσφέρει πρωτοφανή αποτελεσματικότητα και ευελιξία στη δημιουργία και σύνδεση διαφόρων στοιχείων ήχου, όπως γεγονότα(Events), διακόπτες(Switches), καταστάσεις (States) και RTPCs (Real-Time Parameter Controls). Επιπλέον, παρέχει σχολαστικό έλεγχο του pre-loading των Soundbanks, εξασφαλίζοντας τη βέλτιστη διαχείριση των πόρων. Ένα από τα αξιοσημείωτα χαρακτηριστικά του ACE είναι η απόκριση σε πραγματικό χρόνο. Οι αλλαγές που πραγματοποιούνται στον επεξεργαστή αντικατοπτρίζονται αμέσως, εξαλείφοντας την ανάγκη για επανεκκίνηση της εφαρμογής και επιτρέποντας την απρόσκοπτη αναπαραγωγή του ήχου. Επιπλέον, το ACE ενσωματώνει μια ολοκληρωμένη λειτουργία αναζήτησης και φιλτραρίσματος, επιτρέποντας στους χρήστες να εντοπίζουν, να αναθεωρούν και να επεξεργάζονται γρήγορα ακόμη και εκτεταμένες συλλογές ηχητικών συμβάντων.

- ❖ Η μηχανή επιτρέπει τη χρήση οποιουδήποτε middleware ήχου, όπως το CRI ADX2, το FMOD Studio, το Miles Sound System ή το Wwise . Επιπλέον, η μηχανή παρέχει μια δωρεάν εναλλακτική λύση στη μορφή της υλοποίησης SDL_mixer. Τα παραπάνω middleware υποστηρίζονται από το πρόγραμμα και μπορούν να ενεργοποιηθούν με μια απλή επιλογή.
- ❖ Προσομοιώστε τις ακουστικές υπογραφές πραγματικών ήχων . Η υλοποίηση της μεταφοράς κεφαλής-σώματος (HRTF) σημαίνει ότι μπορεί να αξιοποιηθεί πλήρως η επεξεργασία ήχου 3D σε πραγματικό χρόνο για να παρέχει ακριβείς, πραγματικές ακουστικές ιδιότητες της διάδοσης του ήχου απευθείας στα αυτιά του ακροατή. Είναι σαν ο ακροατής να ήταν πραγματικά εκεί.
- ❖ Ο χρήστης έχει πρόσβαση σε όλα τα ηχητικά χαρακτηριστικά με ευκολία μέσω των ηχητικών στοιχείων (components). Οι χρήστες μπορούν να εκμεταλλευτούν την πλήρη ισχύ του ηχητικού αγωγού της CRYENGINE, συμπεριλαμβανομένης της επανάληψης(area-based reverb) και ανάμειξης της ατμόσφαιρας με βάση την περιοχή(area-based ambiance blending ή αλλιώς ABAB), των εφέ σε συγκεκριμένα σημεία και της ρύθμισης του ηχητικού ελέγχου, χρησιμοποιώντας ηχητικά στοιχεία.
- ❖ Ο χρήστης μέσω του ισχυρού συστήματος ηχητικής απόκρυψης (sound occlusion system) της CRYENGINE μπορεί να υπολογίζει αυτόματα την άμεση ηχητική απόκρυψη και να την ενσωματώνει ομαλά με το υποστηριζόμενο λογισμικό ήχου.

- ❖ Ο προγραμματιστής μπορεί να δημιουργήσει απλές ή περίπλοκες συμπεριφορές χρησιμοποιώντας ένα εξελιγμένο σύστημα λήψης αποφάσεων. Το Δυναμικό Σύστημα Απόκρισης (DRS) εισάγει έναν νέο τρόπο υλοποίησης των διαλόγων σε ένα παιχνίδι. Αποσυνδέει τη σχέση ανάδρασης-διαλόγου και λογικής παιχνιδιού. Το DRS το επιτυγχάνει αυτό με την προσθήκη ενός αφηρημένου επιπέδου ανάμεσα σε κάτι που συμβαίνει και σε μια ανατροφοδότηση διαλόγου που εκτελείται. Το DRS μπορεί επίσης να χρησιμοποιηθεί για να εκμεταλλευτεί οποιοδήποτε άλλο σύστημα, όπως για παράδειγμα την Animation, τα Particles ή την Φυσική.

3.4.5 Physics

Οι βασικές δυνατότητες της εξομοίωσης Φυσικής της μηχανής:

- ❖ Από άκαμπτες οντότητες έως οντότητες που βασίζονται σε αντικείμενα και σε οντότητες που μπορούν να καταστραφούν, η CRYENGINE περιλαμβάνει μια πλήρη λύση φυσικής out-of-the-box για να ζωντανέψει τον κόσμο και το gameplay. Η μηχανή παρέχει υποστήριξη για ένα ευρύ φάσμα χαρακτηριστικών, όπως αυθαίρετα πλέγματα σε δυναμικά αντικείμενα και μια ειδική λειτουργία προσομοίωσης για αρθρωτές δομές.
- ❖ Η μηχανή αυτή μετατρέπει το περιβάλλον σε έναν ζωντανό και καθηλωτικό κόσμο με τη δύναμη της ρεαλιστικής και ελεγχόμενης κάμψης της βλάστησης. Παρέχει τη δυναμική αλληλεπίδραση, καθώς τα μεμονωμένα στοιχεία της βλάστησης ανταποκρίνονται ρεαλιστικά σε εξωτερικούς παράγοντες, όπως αντικείμενα, παίκτες και το απαλό λίκνισμα του ανέμου. Το περιβάλλον ζωντανεύει με τη σαγηνευτική ομορφιά και τη φυσική κίνηση της βλάστησης, δημιουργώντας μια πραγματικά συναρπαστική και αληθοφανή εμπειρία.
- ❖ Η CryEngine δίνει μια αίσθηση ρεαλισμού στον εικονικό κόσμο με το ενσωματωμένο σύστημα προσομοίωσης πλευστότητας και νερού. Κάνει ακριβείς υπολογισμούς για την άνωση αντικειμένων σε διάφορα περιβάλλοντα, συμπεριλαμβανομένου του νερού και του αέρα. Το σύστημα υποστηρίζει προσαρμοσμένους όγκους νερού, ποτάμια και ωκεάνια κύματα, επιτρέποντάς να δημιουργεί ο χρήστης δυναμικά και καθηλωτικά υδάτινα σώματα. Επιπλέον, το προσομοιωμένο νερό προσαρμόζεται και αναδιανέμεται αυτόματα με βάση την

υποκείμενη γεωμετρία του σκάφους, δημιουργώντας μια ρεαλιστική και συναρπαστική οπτική εμπειρία με επιφανειακά κύματα που προσθέτουν ένα επιπλέον επίπεδο λεπτομέρειας και αυθεντικότητας.

- ❖ Η μηχανή υποστηρίζει διάφορα μοντέλα καταστροφής και ο χρήστης επιλέγει πώς θα καταστραφούν τα αντικείμενα. Μπορεί να συνδυάσει προ-σπασμένα μέρη, συνδεδεμένα με αρθρώσεις βασισμένες στις ενώσεις τάσεων, με άμεση τροποποίηση πλέγματος (χρησιμοποιώντας δυναμική αφαίρεση Boolean) για σύνθετες ρυθμίσεις που ταιριάζουν στις ανάγκες του παιχνιδιού.
- ❖ Το εξειδικευμένο εργαλείο Rope Tool διευκολύνει την υλοποίηση σχοινιών με διάφορους τρόπους σύγκρουσης και προσάρτησης, δίνοντάς τη δυνατότητα δημιουργίας σχοινιών χωρίς κόπο που αποτελούνται από επαναλαμβανόμενα τμήματα.

3.4.6 Performance

Τα βασικά χαρακτηριστικά της απόδοσης της μηχανής:

- ❖ Η CRYENGINE ενσωματώνει ετικέτες προφίλ σε μεγάλο βαθμό σε όλο το πλαίσιο της. Ο προγραμματιστής έχει τη δυνατότητα δημιουργίας προφίλ εντός του παιχνιδιού για να εντοπίζει άμεσα τα σημεία συμφόρησης της απόδοσης καθώς προκύπτουν και όπου εμφανίζονται. Οι ετικέτες προφίλ είναι οργανωμένες σε μια ιεραρχική δομή, παρέχοντας άμεση εικόνα των εμπλεκόμενων ενοτήτων και λειτουργιών. Επιπλέον, οι λεπτομερείς πληροφορίες σχετικά με τη διάρκεια και τη συχνότητα βοηθούν στην αξιολόγηση των επιπτώσεών τους. Με μια γρήγορη ματιά, μπορεί να διακρίνει αν ο περιορισμός των επιδόσεων προέρχεται από την CPU ή την GPU και να εντοπίσει εύκολα το υπεύθυνο στοιχείο σε υψηλότερο επίπεδο.
- ❖ Η μηχανή CRYENGINE ενσωματώνει ένα ανεξάρτητο από την πλατφόρμα σύστημα διαχείρισης νημάτων που κατανέμει αποτελεσματικά τις εργασίες σε νήματα λογισμικού και υλικού. Με μια συγκεντρωτική προσέγγιση για τη δημιουργία νημάτων ο προγραμματιστής, μπορεί να προγραμματίσει και να παρακολουθήσει παράλληλους υπολογισμούς σε πραγματικό χρόνο. Το threading παίζει καθοριστικό ρόλο στη βελτιστοποίηση εφαρμογών πραγματικού χρόνου για χρήση υλικού πολλαπλών πυρήνων. Με την CRYENGINE, ο προγραμματιστής έχει πλήρη έλεγχο σε

κάθε νήμα που γεννιέται από τη διεργασία και προσαρμόζει τα χαρακτηριστικά των μεμονωμένων νημάτων ώστε να ανταποκρίνονται στις ειδικές απαιτήσεις του έργου, μεγιστοποιώντας έτσι την απόδοση.

- ❖ Το ισχυρό διαγνωστικό εργαλείο της CRYENGINE, το Statoscope, χρησιμοποιείται για την ανάλυση της κατανομής των επιδόσεων της μηχανής. Με την ενσωμάτωση ομάδων δεδομένων στο Statoscope, οι χρήστες μπορούν να απεικονίζουν αποτελεσματικά τα χαρακτηριστικά του CRYENGINE και να εμφανίζουν στατιστικά στοιχεία ανά πλαίσιο σε ένα εξωτερικό εργαλείο. Τα δεδομένα μπορούν να συλλέγονται σε πραγματικό χρόνο από την CRYENGINE στο Statoscope μέσω μιας ζωντανής τροφοδοσίας ή εναλλακτικά, να καταγράφονται σε ένα αρχείο για μεταγενέστερη ανάλυση. Επιπλέον, οι χρήστες έχουν τη δυνατότητα να επικαλύψουν δύο συνεδρίες λήψης, επιτρέποντας τη συγκριτική αξιολόγηση τυχόν τροποποιήσεων που έγιναν μεταξύ των δύο εκτελέσεων.

Πίνακας 2. Σύγκριση Μηχανών Παιχνιδιών[1][11][13][15][24]

| GAME ENGINE | PLATFORM SUPPORT | PROGRAMMING LANGUAGES | COST/ROYALTIES/SUBSCRIPTION | GAME TYPE |
|-------------|---|-----------------------------|--|---|
| Unity | <ul style="list-style-type: none"> ● Desktop: <ul style="list-style-type: none"> ○ Windows (PC) ○ Mac ○ Universal Windows Platform (UWP) ○ Linux Standalone ● Mobile: <ul style="list-style-type: none"> ○ iOS ○ Android ○ Appetize.io ○ itch.io ○ simmer.io ● Extended Reality (XR): <ul style="list-style-type: none"> ○ ARKit ○ ARCore ○ Microsoft HoloLens ○ Windows Mixed Reality ○ Magic Leap (Lumin) ○ Magic Leap 2(Lumin) ○ Oculus ○ PlayStation VR VR2 ● Consoles: <ul style="list-style-type: none"> ○ PS5 ○ PS4 ○ Xbox One ○ Xbox X S ○ Nintendo Switch ○ Google Stadia ● WebGL ● Embedded: <ul style="list-style-type: none"> ○ Embedded Linux ○ QNX <p>Πηγή : Unity Platform Support</p> | 3* (c#, javascript, python) | Starting Licence: free, Subscription options, Royalties 5%: >1.000.000\$ | Action Adventure Role-Playing (RPG) First-Person Shooter (FPS) Third-Person Shooter Platformer Puzzle Strategy Real-Time Strategy (RTS) Turn-Based Strategy Tower Defense Simulation Sports Racing Fighting Beat 'em Up Stealth Survival Horror Point-and-Click Educational Casual Party Music/Rhythm Visual Novel Sandbox Open World Virtual Reality (VR) Augmented Reality (AR) Massively Multiplayer Online (MMO) Card and Board Games Arcade Exergames (Fitness Games) Trivia and Quiz Games Historical Sci-Fi Fantasy Western Crime and Detective Comedy Drama |

| | | | | |
|---------------|--|--|--|---|
| | | | | War Puzzle-Platformer Roguelike Metroidvania Interactive Fiction Nonlinear Exploration Life Simulation Art Games |
| Unreal | <ul style="list-style-type: none"> ● UNREAL ENGINE 5 <ul style="list-style-type: none"> ○ WINDOWS PC, ○ PLAYSTATION 4/5, ○ XBOX SERIES X/S/ ONE, ○ NINTENDO SWITCH, ○ MACOS, ○ IOS, ○ ANDROID, ○ ARKIT / ARCORE, ○ OPENXR, ○ STEAMVR, ○ OCULUS, ○ LINUX, ○ STEAMDECK. <p>Πηγή : Unreal Platform Support</p> | 5* (C++, Blueprint visual scripting, python, javascript, typescript) | Starting Licence: free, Subscription options, Royalties 5%: >1.000.000\$ | το ίδιο με Unity |
| Godot | <ul style="list-style-type: none"> ● Windows (PC) ● macOS ● Linux ● Android ● iOS (iPhone, iPad) ● HTML5 (Web) ● UWP (Universal Windows Platform) ● PlayStation 4 ● PlayStation Vita ● Xbox One ● Nintendo Switch ● WebGL (Web) ● Raspberry Pi <p>Πηγή : Godot Platform Support</p> | 6* (GDscript, c#, visual scripting, c++, rust, nim) | Free Open Source | 2D Platformer 3D Action Role-Playing Game (RPG) First-Person Shooter (FPS) Third-Person Shooter Strategy Real-Time Strategy (RTS) Turn-Based Strategy Tower Defense Simulation Sports Racing Fighting Beat 'em Up Stealth Survival Horror Point-and-Click Adventure Puzzle Casual Games Educational Games Multiplayer Online Games (MOG) Card and Board Games Visual Novels Virtual Reality (VR) Augmented Reality (AR) Text-based Games Retro Games Exploration Nonlinear Narrative Games Art Games Experimental Games Open World Games Music/Rhythm Games Exergames (Fitness Games) Trivia and Quiz Games Interactive Fiction |

| | | | | |
|-----------|---|-------------------|--|--|
| CryEngine | <ul style="list-style-type: none"> • Windows, • Linux, • PlayStation 4, • Xbox One, • Oculus Rift, • OSVR, • PSVR • HTC Vive. • Mobile support is in development <p>Πηγή : CryEngine Platform Support.</p> | 3* (C++, Lua, C#) | Starting Licence: Free, Royalties 5%: >5.000\$ | 3D Action/Adventure First-Person Shooter (FPS) Third-Person Shooter Open World Games Role-Playing Games (RPG) Survival Games Racing Games Simulation Games Platformer Games (both 2D and 3D) Puzzle Games Strategy Games Virtual Reality (VR) Games Augmented Reality (AR) Games Horror Games Stealth Games Sports Games Sandbox Games Educational Games Exploration Games Historical Games Sci-Fi Games Fantasy Games Multiplayer Online Games (MOG) Massively Multiplayer Online Games (MMOG) Adventure Games Narrative-Driven Games Artistic and Experimental Games War Games Flight and Space Simulation Games Casual Games Music and Rhythm Games |
|-----------|---|-------------------|--|--|

Πηγή: Analyzing strengths and weaknesses of Modern Game Engines [4][5]

Πίνακας 3. Σύγκριση θετικών και αρνητικών στις Μηχανές Παιχνιδιών[13][24]

| Όνομα Μηχανής | Θετικά | Αρνητικά |
|------------------|--|---|
| Unity | Δωρεάν για αρχάριους, Σπουδαίο για 2D και 3D παιχνίδια, Πολύ καλή υποστήριξη για παιχνίδια σε κινητές πλατφόρμες, VR και AR SDK, Asset Store με πολλά δωρεάν assets | Πολυδάπανες άδειες για Επαγγελματίες, Higher-end τεχνολογικές απαιτούν καλύτερους υπολογιστές, Πολλές UI αλλαγές |
| Unreal | Μια κορυφαία επιλογή για VR, Οπτική σχεδίαση για μη προγραμματιστές, Μεγάλη αγορά με δωρεάν assets | Δεν είναι ότι καλύτερο για απλά project, Πολύ καλά γραφικά απαιτούν πολύ δυνατούς υπολογιστές, Είναι καλύτερο για 3D και όχι για 2D παιχνίδια |
| Godot | Είναι πολύ αξιόλογο για 2D και 3D παιχνίδια, εντελώς δωρεάν και ανοιχτού κώδικα ακόμη και για εμπορική χρήση, παθιασμένη κοινότητα, Μοναδική αρχιτεκτονική για την ανάπτυξη παιχνιδιών | Για τους εμπείρους προγραμματιστές το GDScript δεν είναι και το καλύτερο , Δεν έχει τόσο μεγάλη κοινότητα για δημιουργία ψηφιακών asset όπως οι άλλες μηχανές |
| CryEngine | Εύκολο στην εκμάθηση και προσφέρει τον πλήρη πηγαίο κώδικα, Το CryEngine είναι εξοπλισμένο με προσανατολισμό στην ποιότητα ήχο | Δεν είναι η καλύτερη για κινητά, αλλά η υποστήριξη για κινητά βρίσκεται σε εξέλιξη |

Πηγή : Analyzing strengths and weaknesses of Modern Game Engines [4][5]

4 Μεθοδολογία

4.1 Εγκατάσταση της μηχανής Unity

Τα ελάχιστα απαιτούμενα χαρακτηριστικά που πρέπει να έχει ο χρήστης σε υπολογιστικό σύστημα για να χρησιμοποιήσει τον Unity Editor είναι :

Πίνακας 4. Unity Editor System Requirements

| Minimum requirements | Windows | macOS | Linux |
|---------------------------------|---|--|--|
| Operating system version | Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only. | Mojave 10.14+ (Intel editor) Big Sur 11.0 (Apple silicon Editor) | Ubuntu 20.04, Ubuntu 18.04, and CentOS 7 |
| CPU | X64 architecture with SSE2 instruction set support | X64 architecture with SSE2 instruction set support (Intel processors) Apple M1 or above (Apple silicon-based processors) | X64 architecture with SSE2 instruction set support |
| Graphics API | DX10, DX11, and DX12-capable GPUs | Metal-capable Intel and AMD GPUs | OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs. |
| Additional requirements | Hardware vendor officially supported drivers | Apple officially supported drivers (Intel processor) Rosetta 2 is required for Apple silicon devices running on either Apple silicon or Intel versions of the Unity Editor. | Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.) |
| | For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer. | | |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Κατά περίπτωση υπάρχουν ορισμένοι περιορισμοί όσο αναφορά το λειτουργικό σύστημα macOS και το Linux.

Οι ελάχιστες απαιτήσεις για να φτιάξει και να τρέξει μια εφαρμογή με το Unity Player είναι:

Πίνακας 5. Ελάχιστες απαιτήσεις του Unity Player σε Mobile Platforms

| Mobile | | | |
|--------------------------------|---|--|--------------------------|
| Operating systems | Android | iOS | tvOS |
| Version | 5.1 (API 22+) Customized versions of Android must include all supported Google standard APIs. | 13+ | 13+ |
| CPU | ARMv7 with Neon Support (32-bit) or ARM64 | A7 SoC+ | A8 SoC+ |
| Graphics API | OpenGL ES 2.0+, 3.0+, Vulkan Note: OpenGL ES 2.0 support is deprecated. | Metal | Metal |
| Additional requirements | <ul style="list-style-type: none"> - 1GB+ RAM - Supported hardware devices must meet or exceed Google's Android Compatibility Definition (Version 9.0) limited to the following Device Types: <ol style="list-style-type: none"> 1. Handheld (Section 2.2) 2. Television (Section 2.3) 3. Tablets (Section 2.6) - Hardware must be running Android OS natively. With the exception of Android for Chrome OS, Android within a container or emulator is not supported. - For development: Android SDK (12/API 31), Android NDK (r23b) and OpenJDK (11), which are installed by default with Unity Hub. | For development and debugging: see the Apple documentation on XCode support . For App Store submission: see Apple's submission guidelines for the required Xcode version. | Apple TV 4th generation+ |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Για πληροφορίες σχετικά με κονσόλες όπως τα PlayStation 4 (συμπεριλαμβανομένου του PS VR), PlayStation 5, Xbox Series S|X και Nintendo Switch κτλ ο χρήστης θα πρέπει να ανατρέξει στην τεκμηρίωση για προγραμματιστές στους ιστότοπους των κατόχων της εκάστοτε πλατφόρμας ή να επικοινωνήσει απευθείας με τον εκπρόσωπο της πλατφόρμας που τον ενδιαφέρει για περισσότερες πληροφορίες.

Για υλοποίηση η οποία προορίζεται να τρέξει σε κάποιο φυλλομετρητή οι απαιτήσεις είναι οι εξής :

Πίνακας 6. Ελάχιστες απαιτήσεις του Unity Player σε Browser Platforms

| WebGL | |
|----------------------------------|--|
| Operating system running browser | Windows, macOS, and Linux |
| Hardware | Workstation and laptop form factors. |
| Additional requirements | Versions of Chrome, Firefox, Safari or Edge (Chromium-based) that are: <ul style="list-style-type: none"> - WebGL 1.0 or 2.0 capable - HTML 5 standards compliant - 64-bit - WebAssembly capable Note: WebGL 1.0 support is deprecated. |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Πίνακας 7. Ελάχιστες απαιτήσεις του Unity Player σε Desktop Platforms

| Desktop | | | | |
|---|---|---|---|---|
| Operating system | Windows | Universal Windows Platform | macOS | Linux |
| Operating System Version | Windows 7 (SP1+), Windows 10 and Windows 11 | Windows 10+, Xbox Series X S, HoloLens | Mojave 10.14+ | Ubuntu 20.04, Ubuntu 18.04, and CentOS 7 |
| CPU | x86, x64 architecture with SSE2 instruction set support. | x86, x64 architecture with SSE2 instruction set support, ARM, ARM64. | Apple Silicon, x64 architecture with SSE2. | x64 architecture with SSE2 instruction set support. |
| Graphics API | DX10, DX11, DX12 capable. | DX10, DX11, DX12 capable GPUs. | Metal capable Intel and AMD GPUs | OpenGL 3.2+, Vulkan capable. |
| Additional requirements | Hardware vendor officially supported drivers. For development: IL2CPP scripting backend requires Visual Studio 2015 with C++ Tools component or later and Windows 10+ SDK. | Hardware vendor officially supported drivers. For development: Windows 10+ (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10+ SDK. | Apple officially supported drivers. For development: IL2CPP scripting backend requires Xcode. Targeting Apple Silicon with IL2CPP scripting backend requires macOS Catalina 10.15.4 and Xcode 12.2 or newer. | Gnome desktop environment running on top of X11 windowing system Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor) Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver. |
| For all operating systems, the Unity Player is supported on workstations, laptop or tablet form factors, running without emulation, container or compatibility layer. | | | | |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Για πλατφόρμες που θα έχουν το ρόλο server οι απαιτήσεις είναι λιγότερες γιατί τις περισσότερες φορές δεν υπάρχει γραφικό περιβάλλον οπότε η υλοποίηση είναι πιο <<ελαφριά>>.

Πίνακας 7. Ελάχιστες απαιτήσεις του Unity Player σε Server Platforms

| Server Platforms | | | |
|---------------------------------|--|---|---|
| Operating systems | Windows | macOS | Linux |
| Operating System Version | Windows 7 (SP1+), Windows 10 and Windows 11, running on workstation and rack form factors, without emulation or compatibility layer. | Mojave 10.14+ running on workstation and rack form factors, without emulation or compatibility layer. | Ubuntu 20.04, Ubuntu 18.04, and CentOS 7, running on workstation and rack form factors, without emulation or compatibility layer. |
| CPU | x86, x64 architecture with SSE2 instruction set support. | x64 architecture with SSE2 instruction set support. | x64 architecture with SSE2 instruction set support. |
| GPU | No explicit GPU support. | | |
| Additional requirements | Hardware vendor officially supported drivers. | | |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Για πλατφόρμες εικονικής πραγματικότητας οι απαιτήσεις είναι οι εξής :

Πίνακας 6. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System

| Oculus | | | |
|---------------------------------|---|--|-----------------------------|
| Hardware | Oculus Rift/Rift S | Oculus Quest | Oculus Quest 2 |
| Operating system version | Windows 10+ | Android | |
| CPU | <u>See Oculus recommended specifications.</u> | ARM 64-bit (Snapdragon 835) | ARM 64-bit (Snapdragon XR2) |
| GPU | <u>See Oculus recommended specifications.</u> | NVIDIA GTX 1050Ti/AMD Radeon RX 470 or greater | |
| Graphics API | DX11 | OpenGL ES 3.0, Vulkan(Experimental) | |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Πίνακας 7. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System

| Windows Mixed Reality | | | |
|--------------------------|--------------------|----------------------|-----------------------------|
| Hardware | WMR Immersive (VR) | HoloLens | HoloLens 2 |
| Operating system version | Windows 10 RS4+ | Embedded Windows 10+ | |
| CPU | Intel 64-bit | Intel Atom | ARM 64-bit (Snapdragon 850) |
| Graphics API | DX11 | | |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Πίνακας 8. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System

| Magic Leap | |
|--------------------------|--|
| Operating system | Magic Leap 2 |
| Operating system version | Magic Leap 2 is built on top of the Android Open Source Project (AOSP), Android 10 (Q) API level 29, |
| CPU | See Magic Leap 2 specifications |
| GPU | See Magic Leap 2 specifications |
| Graphics API | OpenGL ES 3.2+ |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Πίνακας 9. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System

| ARCore | |
|------------------------------|--|
| Operating system | Android phones |
| Operating system version | See list of ARCore-supported devices . |
| CPU | ARM 32-bit and 64-bit |
| Graphics API | OpenGL ES 3.0+ |
| Latest supported SDK version | ARCore 1.37 |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

Πίνακας 10. Ελάχιστες απαιτήσεις του Unity Player σε XR Platform System

| ARKit | |
|------------------------------|--|
| Operating system | iPhone and iPad |
| Operating system version | iOS 12+ |
| CPU | A9+ |
| Graphics API | Metal |
| Latest supported SDK version | ARKit 6 |
| Additional requirements | You must use Xcode 12.5 or later when compiling an iOS Player that includes the ARKit XR Plugin. |

Πηγή : <https://docs.unity.cn/2023.1/Documentation/Manual/system-requirements.html>

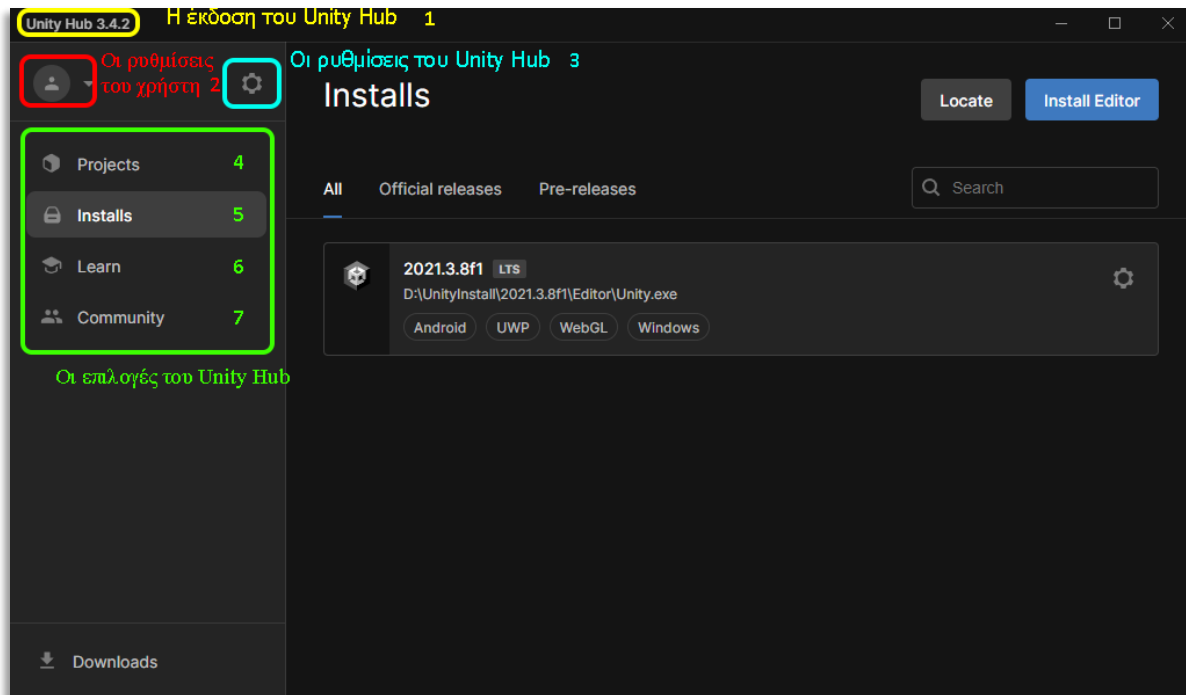
4.1.1 Unity Hub

Η εγκατάσταση στα windows είναι μια απλή διαδικασία . Ο χρήστης μπορεί να κατεβάσει το Unity hub που είναι ο κεντρικός διαχειριστής της πλατφόρμας από την ηλεκτρονική διεύθυνση <https://unity.com/download> ή να κατεβάσει κατευθείαν τον editor από την διεύθυνση <https://unity.com/releases/editor/archive>.


Το Unity hub παρέχει και μας ενημερώνει με τις εξής πληροφορίες μετά την εγκατάσταση :

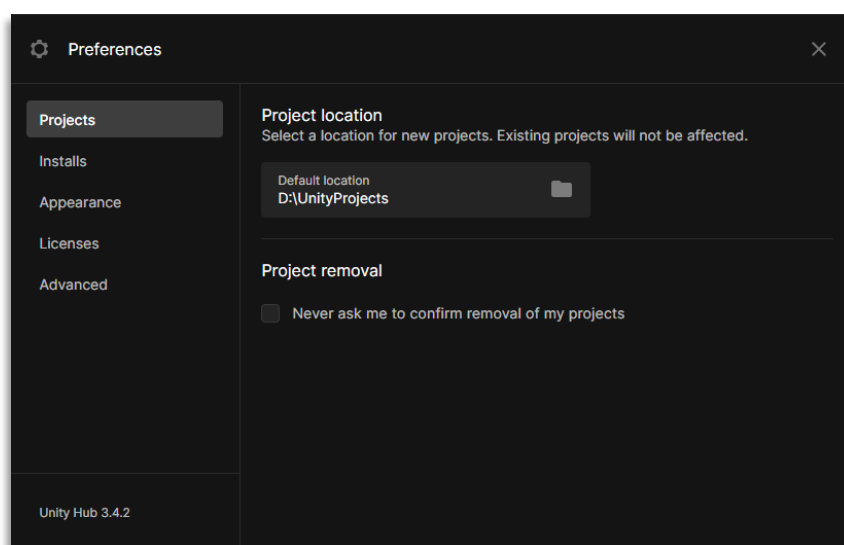
- ❖ Την έκδοση του Unity Hub ,
- ❖ Τις ρυθμίσεις του εκάστοτε χρήστη ,
- ❖ Τις ρυθμίσεις του Unity Hub ,
- ❖ Να δει τα project που έχει ενεργά στην πλατφόρμα ,
- ❖ Να κάνει εγκατάσταση μιας συγκεκριμένης έκδοσης της πλατφόρμας ,
- ❖ Να τρέξει τα εκπαιδευτικά μαθήματα που παρέχει η πλατφόρμα για τον τρόπο λειτουργίας της ,
- ❖ Να πουλήσει στο ηλεκτρονικό κατάστημα τα asset του ή να αγοράσει αυτά που θέλει μέσω του Unity asset store , να ενημερωθεί μέσω του Blog , να βοηθήσει ή να βοηθηθεί μέσω του QA chat , να αλληλοεπιδράσει με την κοινότητα στα Forums , να ανεβάσει την υλοποίηση της εφαρμογής σε WebGL μέσω του Unity Play ώστε να την δοκιμάσουνε και άλλοι παίχτες και τέλος το Unity pulse που δίνει την δυνατότητα σε κάποιον να έχει άμεση επικοινωνία με διάφορες ομάδες ανάπτυξης της Unity .

Πριν την επιλογή για ποια έκδοση της μηχανής θα χρησιμοποιηθεί πρέπει να ρυθμιστούν κάποιες συγκεκριμένες παράμετροι οι οποίες βρίσκονται μέσα στα settings του Unity Hub. Η πρώτη επιλογή είναι να ορίσουμε που θα αποθηκεύονται τα διάφορα projects που υλοποιούμε δηλαδή το path στο οποίο θα εγκατασταθεί το κάθε ένα παιχνίδι που θα δημιουργείται.



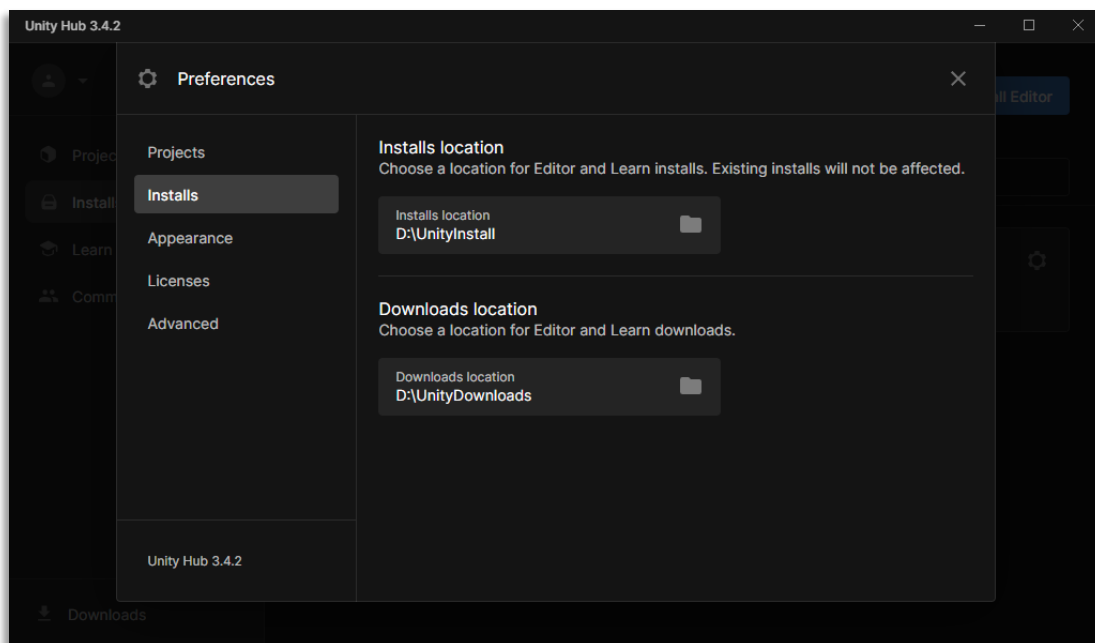
Εικόνα 9. Κεντρικό Interface του Unity Hub

| | |
|---|--|
|  | <p>Μια καλή πρακτική είναι να αποθηκεύονται τα αρχεία των project, της εγκατάστασης των editor της κάθε έκδοσης της μηχανής και των εκπαιδευτικών μαθημάτων σε ξεχωριστό path το οποίο δεν θα είναι το ίδιο με την εγκατάσταση των windows .</p> |
|---|--|




Εικόνα 10. Ρυθμίσεις του Unity Hub - Projects

Μετά έχει σειρά το που θα γίνει η εγκατάσταση του κάθε editor της μηχανής. Ο κάθε editor μπορεί να είναι διαφορετική έκδοση ανάλογα με τις απαιτήσεις του παιχνιδιού που πρέπει να υλοποιηθεί ή να έχει διαφορετικές ιδιότητες. Παραδείγματος χάριν για μια εφαρμογή η οποία έχει στόχο μόνο τις κινητές πλατφόρμες δεν είναι απαραίτητο να συμπεριληφθεί και επέκταση των windows, του Linux, του Mac και διάφορων άλλων. Μαζί με την εγκατάσταση πρέπει να οριστεί και το που θα αποθηκεύονται τα διάφορα downloads των editor και των εκπαιδευτικών μαθημάτων που παρέχει η μηχανή μέσω του site της εταιρείας. Τέλος στις ρυθμίσεις υπάρχει η επιλογή πάνω σε τι είδους άδεια θα δουλέψει ο χρήστης. Εδώ υπάρχουν διάφορες άδειες αλλά η δωρεάν είναι η Personal χωρίς αυτό να εμποδίζει κάποιον να αναβαθμίσει την άδειά του σε δεύτερο χρόνο με ό,τι αυτό συνεπάγεται.

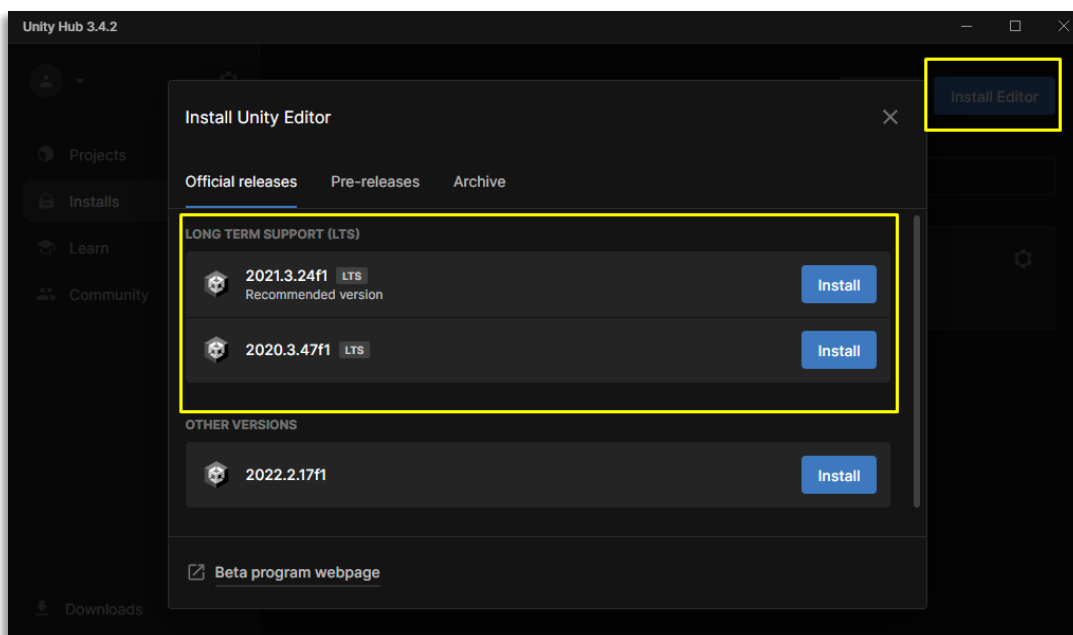


Εικόνα 11. Ρυθμίσεις του Unity Hub - Installs

Αφού γίνει η υλοποίηση της καλής πρακτικής που προαναφέρθηκε, ο χρήστης θα πρέπει πλέον να επιλέξει σε ποια έκδοση της μηχανής θέλει να δουλέψει. Αν θέλει να δοκιμάσει τις πιο πρόσφατες επεκτάσεις της εφαρμογής έχει την δυνατότητα να εγκαταστήσει την άλφα έκδοση της μηχανής η οποία έχει τα πιο εξελιγμένα εργαλεία και παρέχει στον προγραμματιστή μια ευρεία επιλογή λειτουργιών. Συνήθως αυτό έρχεται με κόστος την σταθερότητα στην ανάπτυξη της εφαρμογής και θα πρέπει να αποφεύγεται.

| | |
|---|---|
|  | <p>Πρέπει να αποφεύγεται η χρήση της πειραματικής έκδοσης Alpha (και της Beta) για project τα οποία θα είναι για επαγγελματική χρήση και προορίζονται για να γίνουν διαθέσιμα στην αγορά , για λόγους ασφάλειας διότι τα εργαλεία που παρέχει το Unity σε αυτήν την έκδοση δεν έχουν εκλεχθεί ενδελεχώς (είναι πειραματικά) .</p> |
|---|---|

Η σωστή πρακτική είναι ο προγραμματιστής να διαλέγει την έκδοση που η εταιρία Unity σκοπεύει να υποστηρίξει για ένα μεγάλο χρονικό διάστημα με ενημερώσεις , αναβαθμίσεις και διορθώσεις όπου και αν χρειάζεται και δηλώνεται ως LTS δηλαδή long term support.



Εικόνα 12. Ρυθμίσεις του Unity Hub – Install Editor

Εάν για κάποιο λόγο ο προγραμματιστής πρέπει να δουλέψει σε κάποιο παλιό project η Unity δίνει την δυνατότητα μέσω του Archive να πάει και να κατεβάσει την έκδοση που επιθυμεί. Τέτοιες περιπτώσεις είναι όταν ένα project έχει υλοποιηθεί σε μια προηγούμενη Major έκδοση του Unity και πλέον είναι legacy δηλαδή δεν υποστηρίζεται πλήρως ή υποστηρίζεται κάτω από κάποιες προϋποθέσεις. Αυτό συμβαίνει γιατί η μηχανή υπόκειται σε μεγάλες αλλαγές που επηρεάζουν την αρχική της δομή από Major release σε Major release.

Πίνακας 11. Versioning

| Increment this value | Under these conditions | Example |
|---|--|---|
| <p>MAJOR</p> <p>(20xx.1.1f1)</p> | <p>There is at least one breaking change and neither version of the package can be substituted for the other. A breaking change includes:</p> <ul style="list-style-type: none"> • Changing the API surface (the exposed part of the API) or features in a way that risks compilation or runtime errors. • Removing non-API features, including removing Assets or changing an Asset's GUID. • Removing or renaming assemblies and Assets (because) the compiler might fail to locate them). <p>Note: When incrementing the major version, always reset the PATCH and MINOR values to 0.</p> | <p>Versions 1.2.3 and 2.0.0 are not compatible and cannot be used interchangeably without risk.</p> |
| <p>MINOR</p> <p>(Same MAJOR value)</p> <p>(2021.xx.1f1)</p> | <p>The highest MINOR introduces functionality in a backward-compatible way. A backward-compatible (or non-breaking) API change includes:</p> <ul style="list-style-type: none"> • Changing the API surface or features without risking compilation or runtime errors. • Adding non-API features. • Adding assemblies and Assets (because new items don't have pre-existing references). <p>Note: When incrementing the minor version, always reset the PATCH version to 0.</p> | <p>You can use Version 1.3.0 to fulfill a dependency on 1.2.0 because 1.3.0 is backward-compatible.</p> <p>You cannot use 1.2.0 to fulfill a dependency on 1.3.0.</p> |
| <p>PATCH</p> <p>(Same MAJOR.MINOR values)</p> <p>(2021.1.xx1f1)</p> <p>(2023.2.0axx)</p> <p>(2023.1.0bxx)</p> | <p>The highest PATCH introduces bug fixes without changing the API at all, in a backward-compatible way. The API doesn't change if:</p> <ul style="list-style-type: none"> • The API surface is identical and the features remain unchanged. • The changes don't alter the public API. | <p>Versions 1.3.0 and 1.3.1 should be interchangeable because they have the same API, even though 1.3.1 contains a bug fix not present in 1.3.0.</p> |

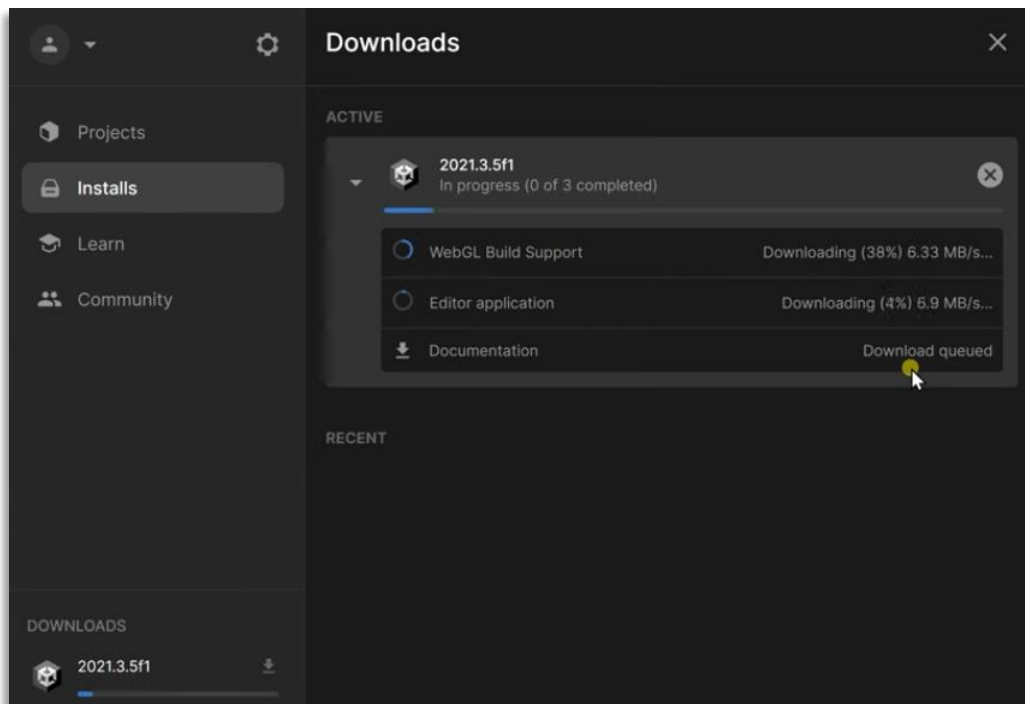
Πηγή : <https://docs.unity3d.com/Manual/upm-semver.html>

Αφού αποφασίσει ποια έκδοση να εγκαταστήσει και έχοντας υπόψιν τα προαναφερόμενα τότε διαλέγει σε πια πλατφόρμα θέλει να δουλέψει .

| Add modules | | Required: 6.47 GB | |
|-------------------------------------|--|-------------------|--------------|
| DEV TOOLS | | DOWNLOAD SIZE | SIZE ON DISK |
| <input checked="" type="checkbox"/> | Microsoft Visual Studio Community 2022 | 1.58 GB | 1.59 GB |
| PLATFORMS | | DOWNLOAD SIZE | SIZE ON DISK |
| <input type="checkbox"/> | Android Build Support | 438.19 MB | 2.07 GB |
| <input type="checkbox"/> | OpenJDK | 114.82 MB | 222.86 MB |
| <input type="checkbox"/> | Android SDK & NDK Tools | 1.12 GB | 3.03 GB |
| <input type="checkbox"/> | iOS Build Support | 478.11 MB | 2.01 GB |
| <input type="checkbox"/> | tvOS Build Support | 473.81 MB | 1.99 GB |
| <input type="checkbox"/> | Linux Build Support (IL2CPP) | 52.99 MB | 218.91 MB |
| <input type="checkbox"/> | Linux Build Support (Mono) | 52.53 MB | 215.82 MB |
| <input type="checkbox"/> | Linux Dedicated Server Build Support | 99.12 MB | 398.45 MB |
| <input type="checkbox"/> | Mac Build Support (Mono) | 339.8 MB | 1.85 GB |
| <input type="checkbox"/> | Mac Dedicated Server Build Support | 337.79 MB | 1.84 GB |
| <input type="checkbox"/> | Universal Windows Platform Build Support | 286.07 MB | 1.97 GB |
| <input type="checkbox"/> | WebGL Build Support | 558.68 MB | 2.53 GB |
| <input type="checkbox"/> | Windows Build Support (IL2CPP) | 87.73 MB | 377.49 MB |
| <input type="checkbox"/> | Windows Dedicated Server Build Support | 170.61 MB | 764.19 MB |
| LANGUAGE PACKS (PREVIEW) | | DOWNLOAD SIZE | SIZE ON DISK |
| <input type="checkbox"/> | 日本語 | 2.78 MB | 2.78 MB |
| <input type="checkbox"/> | 한국어 | 2.56 MB | 2.56 MB |
| <input type="checkbox"/> | 繁體中文 | 2.36 MB | 2.36 MB |
| <input type="checkbox"/> | 简体中文 | 2.36 MB | 2.36 MB |
| DOCUMENTATION | | DOWNLOAD SIZE | SIZE ON DISK |
| <input type="checkbox"/> | Documentation | 333.53 MB | 680.4 MB |

Εικόνα 13. Ρυθμίσεις του Unity Hub – Install modules

Ακολουθεί η εγκατάσταση της μηχανής με τις τελικές επιλογές

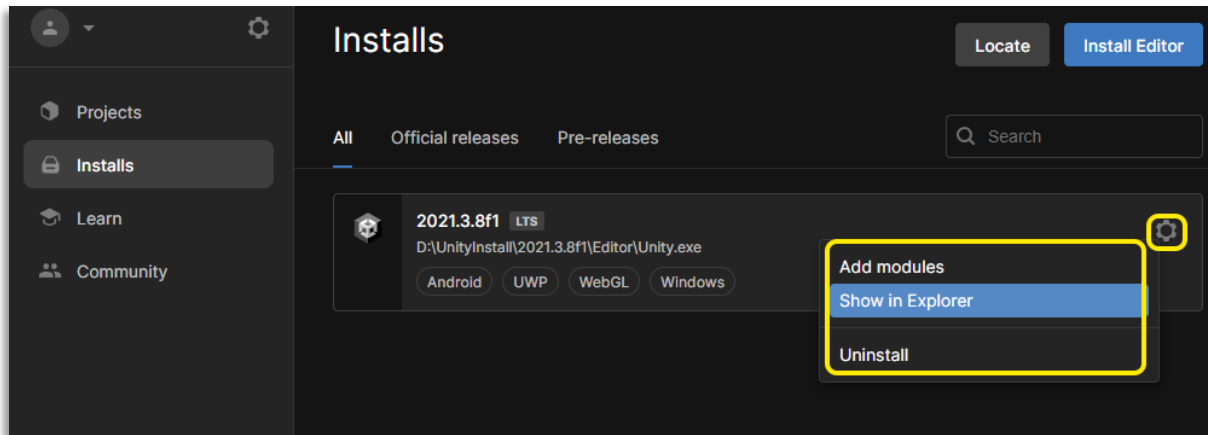


Εικόνα 14. Ρυθμίσεις του Unity Hub – Install Process

Αν για κάποιο λόγο ο χρήστης θέλει να προσθέσει και άλλες πλατφόρμες σε δεύτερο χρόνο μπορεί να το κάνει στην καρτέλα installs. Εκεί υπάρχει κουμπί που δίνει 3 επιλογές :

- 1) Add modules
- 2) Show in Explorer
- 3) Uninstall

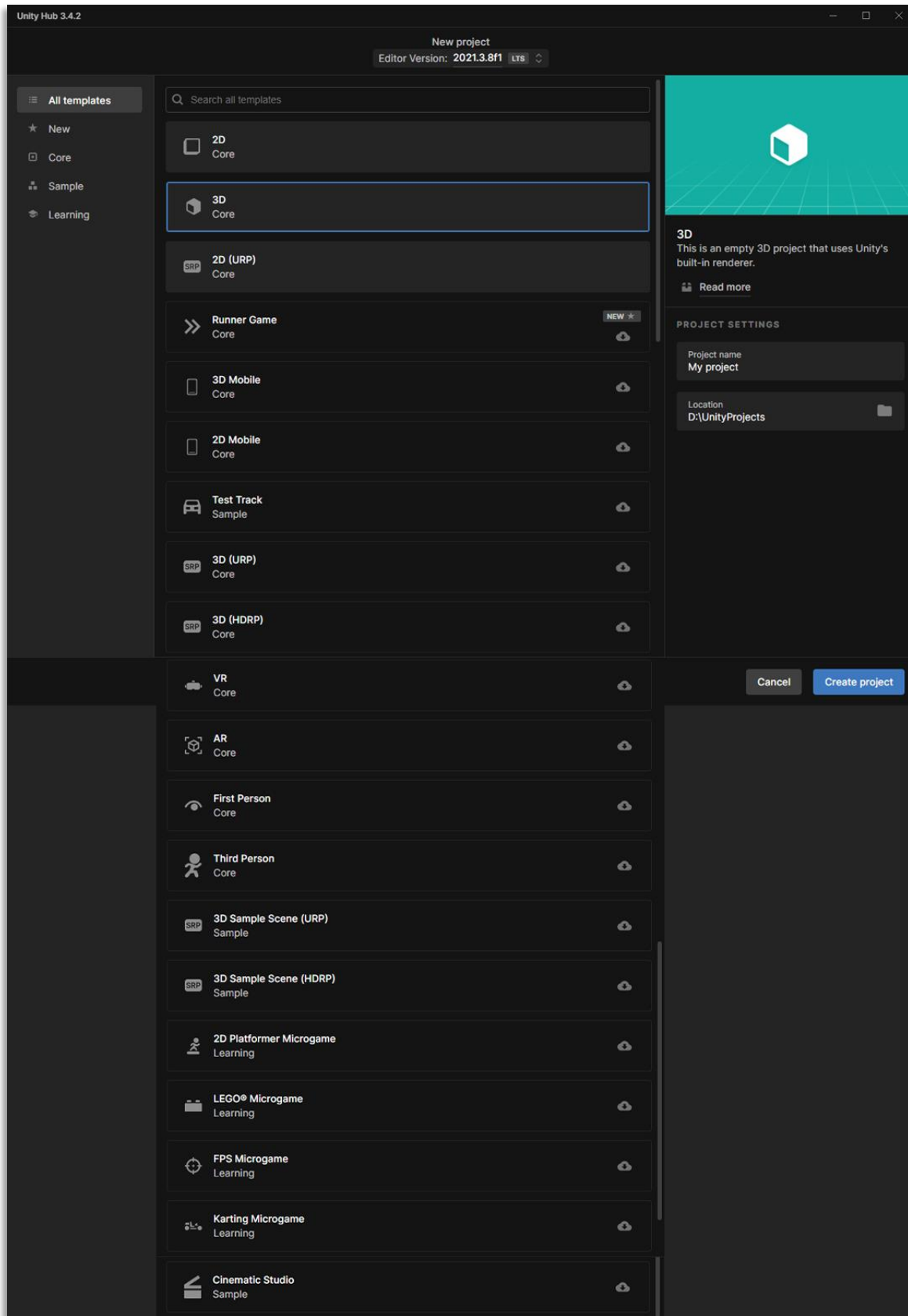
Η πρώτη επιλογή είναι για να προσθέσει ο διαχειριστής επιπλέον δυνατότητες στη συγκεκριμένη μηχανή ώστε να μπορεί να υλοποιεί project σε περισσότερες πλατφόρμες. Η δεύτερη επιλογή είναι για να ανοίξει την τοποθεσία που είναι εγκατεστημένη η συγκεκριμένη έκδοση της μηχανής μέσω του windows explorer και να κάνει μια προεπισκόπηση των αρχείων της. Η τελευταία επιλογή είναι για την απεγκατάσταση της συγκεκριμένης έκδοσης της μηχανής και μόνο.



Εικόνα 15. Ρυθμίσεις του Unity Hub – Install Settings

Ακόμα μια δυνατότητα που παρέχει το unity hub είναι τα πρότυπα έργων. Το Unity hub προσφέρει έτοιμες δομές και λειτουργίες για την ανάπτυξη παιχνιδιών. Με τη χρήση των προτύπων έργων, οι προγραμματιστές μπορούν να επιταχύνουν τη διαδικασία ανάπτυξης και να επικεντρωθούν στη δημιουργία περιεχομένου και λειτουργίας του παιχνιδιού τους. Τα πρότυπα έργων περιλαμβάνουν προκαθορισμένες ρυθμίσεις, στοιχεία UI, συστήματα φυσικής, αντικείμενα παιχνιδιού, ενδεχόμενες συμπεριφορές και άλλες λειτουργίες που μπορούν να προσαρμοστούν ανάλογα με τις ανάγκες του έργου. Η χρήση των project templates απλοποιεί την αρχική διαμόρφωση του περιβάλλοντος ανάπτυξης και παρέχει μια ευέλικτη βάση για την ανάπτυξη παιχνιδιών με γρήγορο χρόνο ανάπτυξης χωρίς αυτό να σημαίνει ότι ο χρήστης δεν μπορεί να επεκτείνει το πρότυπο που έχει επιλέξει με επιπλέον δυνατότητες κατά την διάρκεια της ανάπτυξης του έργου.


Τέλος μπορεί πλέον ο χρήστης να εκκινήσει την εφαρμογή για να ξεκινήσει να δουλεύει πάνω στο project του .



Εικόνα 16 Project Templates in Unity hub version 3.4.2

4.2 Version Control

Μια καλή πρακτική που είναι ακόμα ένα δομικό στοιχείο απαραίτητο στην υλοποίηση ενός έργου είναι το Version Control System ή αλλιώς το Repository ή Repo εν συντομία (εδώ να αναφερθεί πως το Version Control System και το Repository δεν είναι ακριβώς το ίδιο πράγμα αλλά στο πλαίσιο της ανάπτυξης λογισμικού δεν εννοείται Version control χωρίς repository ή να γίνεται αναφορά σε repository χωρίς τις βασικές λειτουργίες του Version Control). Υπάρχουν δυο τέτοιες κατηγορίες συστημάτων τα κεντροποιημένα και τα κατανεμημένα όπου το κάθε ένα έχει διαφορετική προσέγγιση για την διαχείριση και την παρακολούθηση ενός project . Ο χρήστης κατά την ανάπτυξη της εφαρμογής μπορεί να αποθηκεύει την εξέλιξή της, τις διάφορες αλλαγές/τροποποιήσεις, τις διορθώσεις και τις διάφορες εκδόσεις που μπορεί να υπάρχουν σε αυτό το αποθετήριο. Μέσα από το Version Control, αυτά μπορεί να γίνουν διαθέσιμα και για τα άλλα μέλη της ομάδας ,που μπορεί να υπάρχουν, με το μόνο μειονέκτημα να είναι, τι δυνατότητες δίνει η εταιρεία που παρέχει αυτό το αποθετήριο για δωρεάν χρήση πριν αρχίσει να χρεώνει τις υπηρεσίες αυτές. Τα τρία κύρια συστήματα διαχείρισης έκδοσης (VCS) που συνεργάζονται αρμονικά με το Unity είναι το Plastic SCM , το Git και το Perforce Helix Core .

| | |
|---|--|
|  | <p>Μια καλή πρακτική είναι να έχει γίνει μια μελέτη για το πια Version Control systems υπάρχουν στην αγορά και ποιες είναι οι απαιτήσεις της εφαρμογής. Ποσά και που θα είναι τα άτομα στην ομάδα που θα δουλεύει το project και ποιο είναι το κεφάλαιο που μπορεί να διατεθεί για αυτό το σκοπό .</p> |
|---|--|

4.2.1 Centralized Version Control Systems

Σε ένα συγκεντρωτικό VCS, υπάρχει ένα ενιαίο κεντρικό αποθετήριο που αποθηκεύει το ιστορικό και τα αρχεία ολόκληρου του έργου. Οι προγραμματιστές έχουν ένα τοπικό αντίγραφο των αρχείων στα μεμονωμένα μηχανήματά τους, αλλά όλες οι αλλαγές και οι ενημερώσεις γίνονται στο κεντρικό αποθετήριο. Για να εργαστούν σε ένα αρχείο, οι προγραμματιστές πρέπει να επικοινωνήσουν με τον κεντρικό διακομιστή και να ελέγξουν το αρχείο. Παραδείγματα κεντρικών VCS περιλαμβάνουν το Perforce και το Apache Subversion (SVN).

Τα **πλεονεκτήματα** του κεντροποιημένου συστήματος είναι :

- **Απλότητα:** Αυτά τα συστήματα είναι συχνά απλούστερα στην εκμάθηση και τη χρήση σε σύγκριση με τα κατανεμημένα συστήματα.
- **Έλεγχος πρόσβασης:** Το κεντρικό αποθετήριο επιτρέπει καλύτερο έλεγχο πρόσβασης και διαχείρισης δικαιωμάτων.
- **Ακεραιότητα του αποθετηρίου:** Ο κεντρικός διακομιστής εξασφαλίζει την ακεραιότητα και τη συνέπεια του αποθετηρίου.

Τα **μειονεκτήματα** του κεντροποιημένου συστήματος είναι :

- **Ενιαίο σημείο αποτυχίας:** Εάν ο κεντρικός διακομιστής πέσει, οι προγραμματιστές δεν μπορούν να εκτελέσουν τις περισσότερες λειτουργίες ελέγχου εκδόσεων.
- **Περιορισμένη λειτουργικότητα εκτός σύνδεσης:** Οι προγραμματιστές χρειάζονται συνεχή σύνδεση δικτύου για να έχουν πρόσβαση στο κεντρικό αποθετήριο.
- **Χαμηλότερες επιδόσεις:** Τα κεντρικά VCS μπορεί να υποφέρουν από πιο αργές επιδόσεις όταν πρόκειται για μεγάλα έργα ή συχνές λειτουργίες.

4.2.2 Distributed Version Control Systems

Σε ένα κατανεμημένο VCS, κάθε προγραμματιστής έχει ένα πλήρες αντίγραφο του αποθετηρίου του έργου, συμπεριλαμβανομένου του πλήρους ιστορικού του και όλων των αρχείων. Αυτό επιτρέπει στους προγραμματιστές να εργάζονται ανεξάρτητα στα τοπικά τους αποθετήρια και να κάνουν αλλαγές χωρίς να χρειάζονται συνεχή πρόσβαση σε έναν κεντρικό διακομιστή. Παραδείγματα κατανεμημένων VCS περιλαμβάνουν το Git, το Mercurial και το Bazaar.

Τα **πλεονεκτήματα** του κατανεμημένου συστήματος είναι :

- **Λειτουργία εκτός σύνδεσης:** Οι προγραμματιστές μπορούν να εργάζονται εκτός σύνδεσης, δεσμεύοντας αλλαγές στο τοπικό τους αποθετήριο και αργότερα να συγχρονίζονται με άλλους.

- **Ταχύτερη λειτουργία:** Οι περισσότερες λειτουργίες στο κατανεμημένο VCS εκτελούνται τοπικά, με αποτέλεσμα ταχύτερες επιδόσεις σε σύγκριση με το κεντρικό VCS.
- **Συνεργασία και διακλάδωση:** Τα κατανεμημένα συστήματα VCS προσφέρουν ισχυρές δυνατότητες διακλάδωσης και συγχώνευσης, διευκολύνοντας τη συνεργασία και την παράλληλη ανάπτυξη.

Τα **μειονεκτήματα** του κατανεμημένου συστήματος είναι :

- **Καμπύλη εκμάθησης:** Τα κατανεμημένα συστήματα VCS μπορεί να έχουν πιο απότομη καμπύλη εκμάθησης σε σύγκριση με τα συγκεντρωτικά VCS.
- **Αυξημένη χρήση χώρου στο δίσκο:** Κάθε προγραμματιστής έχει ένα πλήρες αντίγραφο του αποθετηρίου, το οποίο μπορεί να απαιτεί περισσότερο χώρο στο δίσκο.
- **Πολύπλοκη επίλυση συγκρούσεων:** Η συγχώνευση αλλαγών από πολλαπλούς κλάδους σε ένα κατανεμημένο VCS μπορεί μερικές φορές να οδηγήσει σε συγκρούσεις που πρέπει να επιλυθούν.

4.3 Best Version Control App for Unity

4.3.1 Plastic

Το Plastic SCM το παρέχει η ίδια η εταιρία της Unity ώστε ο χρήστης να έχει όλο το project του σε ένα μέρος και να το διαχειρίζεται μέσα από τον ίδιο τον editor παρέχοντας κάποιες επιπρόσθετες ευκολίες. Παλαιότερα υπήρχε και το collaborate το οποίο από την 28 Φεβρουάριου 2022 σταμάτησε να λειτουργεί και πλέον όλα τα project ενσωματώθηκαν στο Plastic SCM. Πλέον η τελευταία έκδοση του Plastic δίνει την δυνατότητα της δημιουργίας διακλαδώσεων (branching) η οποία επιτρέπει μέσα από τον ίδιο τον editor ο προγραμματιστής να δημιουργεί και να αλλάζει διακλαδώσεις. Το branching είναι ένα από τα μεγαλύτερα οφέλη του Plastic και μπορεί να βελτιώσει σημαντικά τη συνεργασία μεταξύ ομάδων διάφορων μεγεθών ακόμη και μεγάλων ομάδων που αριθμούν εκατοντάδες άτομα μέσω μιας ροής εργασίας που ταιριάζει στις ανάγκες

όλων. Το Plastic μπορεί να διαχειριστεί τις διακυμάνσεις που υπάρχουν σε ομάδες που διαχειρίζονται το project αλλά και σε μέγεθος των αρχείων και να ανταπεξέλθει στις ανάλογες απαιτήσεις που θα συναντήσει. Σε αντίθεση με το προηγούμενο Version Control system το Plastic δεν κλειδώνει των editor κάθε φορά που πραγματοποιείται ένα update και έτσι ο χρήστης μπορεί να συνεχίζει να εργάζεται πάνω στην εφαρμογή. Επίσης με το Plastic ο χρήστης μπορεί να δει όλο το ιστορικό των αλλαγών του και μπορεί να επιλέξει να επανέλθει η εφαρμογή σε μια προηγούμενη κατάσταση κάτι που στο προηγούμενο σύστημα (Collaborate) δεν ήταν δυνατό. Η δωρεάν έκδοση παρέχει μέχρι 3 χρήστες (seats) και 5 GB μηνιαίως σε χώρο. Το Unity παρέχει και σεμινάρια μέσω του Unity Learn για το πως ο χρήστης να στήσει ένα Version Control System πάνω στον editor του Unity.

Πίνακας 12. Κόστος Plastic SCM

| FREE TIER | STANDARD PRICING (once Free Tier is consumed) |
|--|--|
| Basic DevOps Seats Three (3) seats Standard Compute (Build iOS, Android, Windows, WebGL) 200 Windows minutes Storage 5 GB Machine Concurrency 1 machine | Basic DevOps Seats Seats 4-15: \$7 per seat Seats 16+: \$15 per seat Standard Compute (Build iOS, Android, Windows, WebGL) \$0.07 / build min (Mac) \$0.02 / build min (Win) Storage \$0.14 per GB Machine Concurrency Up to 3: \$0.50 per machine/day 4+: \$2 per machine/day |

Πηγή : <https://unity.com/products/unity-devops#pricing>

Το σύστημα Plastic παρέχει κάποιες επιπλέον δυνατότητες όπως :

- Ο χρήστης να κάνει μια προεπισκόπηση τα 3D μοντέλα του που είναι αποθηκευμένα στο αποθετήριο μέσω του DevOps web dashboard . Μπορεί να μετακινήσει και περιστρέψει τα μοντέλα κατευθείαν από το πρόγραμμα περιήγησης χωρίς να χρειαστεί η εκκίνηση του Unity editor ή κάποιο εργαλείο δημιουργίας ψηφιακού περιεχομένου (Maya,Blender κτλ.) . Αναπαραγωγή κινούμενων σχεδίων. Μπορεί να αναπαράγει το κινούμενο σχέδιο καρέ-καρέ σε όποια κατάσταση υπάρχει. Μπορεί να δει τις ιδιότητες του αρχείου (σύνολο αλλαγών, διακλάδωση,

διαδρομή, μέγεθος, συγγραφέας και πλήρες ιστορικό του Version Control). Υποστηριζόμενες μορφές αρχείων: gltf, .glb, .fbx, .obj

- Ο Χρήστης πλέον μπορεί με την ομάδα του να συνεργαστεί από οπουδήποτε και από οποιαδήποτε συσκευή μέσω της νέας και βελτιωμένης εμπειρίας Plastic Web στον πίνακα ελέγχου Unity Dashboard. Αυτό περιλαμβάνει πλούσιες και βελτιωμένες ροές εργασίας αναθεώρησης,
- Μια νέα εμπειρία αναθεώρησης κώδικα στο Unity Dashboard (με πολλαπλούς αναθεωρητές) εισάγει τώρα μια ροή εργασίας για την επικύρωση της εργασίας που υποβάλλεται από ένα μέλος της ομάδας με σκοπό τη συγχώνευση.
- Αυτόματες ενημερώσεις
- Εύκολη πλοήγηση μεταξύ λογαριασμών, οργανισμών, αποθετηρίων και χώρων εργασίας.
- Νέα προβολή ιστορικού (history) που προσφέρει ένα νέο βελτιωμένο σχεδιασμό με απλοποιημένη πλοήγηση που επιτρέπει στο χρήστη να περιηγηθεί εύκολα στις εκδόσεις ενός αρχείου και να τις συγκρίνει δίπλα-δίπλα.
- Η σημασιολογική συγχώνευση (Semantic Merge) υποστηρίζεται σε όλες τις πλατφόρμες. Αυτό μπορεί να καθιστά ευκολότερες πολλαπλές συγχωνεύσεις. Οι προγραμματιστές που εργάζονται παράλληλα στον ίδιο κώδικα θα επωφελούνται αυτόματα από τις νέες δυνατότητες συγχώνευσης όταν προκύψει σύγκρουση, επιτρέποντας έτσι την πολύ ταχύτερη επίλυσή της.

4.3.2 Git [3]

Το Git είναι ένα εξαιρετικά δημοφιλές Version Control System, γνωστό για την ευελιξία και την αξιοπιστία του, γεγονός που το καθιστά κορυφαία επιλογή μεταξύ των προγραμματιστών. Επιτρέπει την απρόσκοπτη συνεργασία, την αποτελεσματική διακλάδωση και συγχώνευση και την αποτελεσματική παρακολούθηση των αλλαγών εντός του Unity project. Το Git υποστηρίζει επίσης τη χρήση απομακρυσμένων αποθετηρίων, διευκολύνοντας τα μέλη της ομάδας να συνεργάζονται σε ένα Unity project. Με ένα

απομακρυσμένο αποθετήριο, τα μέλη της ομάδας μπορούν να προωθήσουν αλλαγές στο έργο και να τραβήξουν αλλαγές που έχουν γίνει από άλλα μέλη της ομάδας. Αυτό διασφαλίζει ότι όλοι εργάζονται με την πιο ενημερωμένη έκδοση του έργου και επιτρέπει την απρόσκοπτη συνεργασία ακόμη και με μέλη της ομάδας που μπορεί να εργάζονται εξ αποστάσεως. Ακόμα ένα πλεονέκτημα της χρήσης του Git με το Unity είναι η δυνατότητα εύκολης επαναφοράς αλλαγών. Εάν μια αλλαγή που γίνεται στο έργο προκαλεί προβλήματα ή δεν λειτουργεί όπως προβλέπεται, το Git σας επιτρέπει να επιστρέψετε σε μια προηγούμενη έκδοση του έργου. Αυτό μπορεί να εξοικονομήσει σημαντικό χρόνο και προσπάθεια που διαφορετικά θα ξοδευόταν στην προσπάθεια επιδιόρθωσης του προβλήματος. Επιπλέον, το Git επιτρέπει στον χρήστη να παρακολουθεί τις αλλαγές που έγιναν στο έργο, διευκολύνοντας τον εντοπισμό του πότε εισήχθησαν προβλήματα και πώς μπορούν να αντιμετωπιστούν. Το Git είναι δωρεάν και open source αλλά για να το χρησιμοποιήσει κάποιος πρέπει να αποθηκεύσει σε ένα αποθετήριο τα αρχεία του. Τα πιο γνωστά αποθετήρια είναι το GitHub[5] το οποίο έχει και δωρεάν πλάνο χρήσης, το Gitlab[8] το οποίο παρέχει και αυτό δωρεάν έκδοση και το bitBucket[12] το οποίο παρέχει και αυτό ένα δωρεάν πλάνο χρήσης.



Εικόνα 17. GIT

Πηγή : <https://git-scm.com/>

Βασικές εντολές Git για διαχείριση αποθετηρίων υπάρχουν στην ηλεκτρονική διεύθυνση <https://www.git-scm.com/doc> και είναι οι εξής:

1. **git init:** Initializes a new Git repository in the current directory.
2. **git clone <repository>:** Creates a local copy of a remote repository.
3. **git add <file>:** Adds a file or changes to the staging area.
4. **git commit -m "Commit message":** Records the changes in the staging area to the repository with a descriptive message.

5. **git status**: This shows the current status of the repository, including any untracked, modified, or staged files.
6. **git diff**: Shows the differences between the current changes and the last commit.
7. **git log**: Displays a list of commits in reverse chronological order.
8. **git branch**: Lists all the branches in the repository.
9. **git checkout <branch>**: Switches to the specified branch.
10. **git merge <branch>**: Merges changes from the specified branch into the current branch.
11. **git pull**: Fetches and merges changes from a remote repository into the current branch.
12. **git push**: Pushes local commits to a remote repository.
13. **git remote add <name> <url>**: Adds a remote repository with the given name and URL.
14. **git remote -v**: Lists all the remote repositories.
15. **git rm <file>**: Removes a file from the repository.

Το LFS είναι μια επέκταση του Git που σας επιτρέπει να διαχειρίζεστε αποτελεσματικά μεγάλα δυαδικά αρχεία αποθηκεύοντάς τα εκτός του κύριου αποθετηρίου Git. Αντί να αποθηκεύει τα πραγματικά δυαδικά αρχεία στο αποθετήριο, το LFS αποθηκεύει έναν δείκτη στο αρχείο, ενώ το περιεχόμενο του αρχείου αποθηκεύεται σε έναν ξεχωριστό διακομιστή.

Βασικές εντολές για Git LFS(Large File Storage)

1. **git lfs install**: Installs Git LFS on your system. This command needs to be run once per repository.
2. **git lfs track "<pattern>"**: Specifies the file patterns to be tracked by Git LFS. For example, `git lfs track "*.jpg"` would track all JPEG image files.
3. **git lfs untrack "<pattern>"**: Stops tracking the specified file patterns with Git LFS.
4. **git lfs ls-files**: Lists the files managed by Git LFS.

5. **git lfs pull**: Fetches files tracked by Git LFS from a remote repository.
6. **git lfs push**: Pushes local commits and associated large files to a remote repository.
7. **git lfs status**: Shows the status of Git LFS files, including any staged or modified files.
8. **git lfs migrate import --include="<pattern>"**: Converts large files in the Git history to Git LFS, based on the specified file patterns.
9. **git lfs migrate export --include="<pattern>"**: Extracts large files from Git LFS and includes them directly in the Git history, based on the specified file patterns.

Ένα ακόμα βασικό στοιχείο που είναι απαραίτητο για την διαχείριση των ψηφιακών στοιχείων καθώς αυτά προορίζονται για αποθήκευση στο αποθετήριο είναι το αρχείο που φιλτράρει ποια αρχεία ή φάκελοι δεν θα αποθηκευτούνε από το τοπικό αποθετήριο του project στο απομακρυσμένο. Αυτό το αρχείο είναι το **.gitignore** και συνήθως δημιουργείται στον αρχικό κατάλογο του έργου (project) στο απομακρυσμένο αποθετήριο. Κάθε φορά που ο χρήστης δίνει μια εντολή push για να ανεβάσει καινούργιο περιεχόμενο από το τοπικό αποθετήριο στο απομακρυσμένο τα αρχεία με επεκτάσεις που περιέχονται στο **.gitignore** θα αγνοούνται από την διαδικασία.

Το αρχείο **.gitattributes** στο Unity χρησιμοποιείται συνήθως για να καθορίσει ο χρήστης ποια αρχεία θα πρέπει να παρακολουθούνται με τη χρήση του LFS. Επιτρέπει στο χρήστη να ορίσει μοτίβα ή συγκεκριμένες επεκτάσεις αρχείων που θα πρέπει να αντιμετωπίζονται ως μεγάλα αρχεία και να διαχειρίζονται από το LFS. Από προεπιλογή, η μηχανή Unity δημιουργεί ένα αρχείο **.gitattributes** με προκαθορισμένα μοτίβα για συχνά χρησιμοποιούμενους τύπους μεγάλων αρχείων σε έργα Unity, όπως υφές(textures), αρχεία ήχου και τρισδιάστατα μοντέλα. Χρησιμοποιώντας το LFS για μεγάλα αρχεία, ο χρήστης μπορεί να αποφύγει τη διόγκωση του αποθετηρίου Git και να βελτιώσει την απόδοση κατά την διάρκεια της χρήσης του Version Control system. Αυτό επιτρέπει να επικεντρωθεί στην παρακολούθηση των αλλαγών στον πηγαίο κώδικα και σε άλλα μικρότερα αρχεία, ενώ παράλληλα μπορεί να συνεργάζεται σε έργα που περιλαμβάνουν μεγάλα ψηφιακά στοιχεία. Για παράδειγμα εάν υπάρχει ένα αρχείο **.gitattributes** με τα εξής χαρακτηριστικά :

Για παράδειγμα εάν υπάρχει ένα αρχείο **.gitignore** με τα εξής χαρακτηριστικά :

Πίνακας 13. Αρχείο .gitignore

| Δομή Αρχείου .gitignore |
|--|
| <pre> # This .gitignore file should be placed at the root of your Unity project directory /[L]ibrary/ /[Tt]emp /[Oo]bj/ /[Bb]uild/ /[Bb]uilds/ /[L]ogs/ /[Uu]ser[Ss]ettings/ # MemoryCaptures can get excessive in size. # They also could contain extremely sensitive data /[Mm]emoryCaptures/ # Recordings can get excessive in size /[Rr]ecordings/ # Uncomment this line if you wish to ignore the asset store tools plugin /[Aa]ssets/AssetStoreTools* # Autogenerated JetBrains Rider plugin /[Aa]ssets/Plugins/Editor/JetBrains* # Visual Studio cache directory .vs/ # Gradle cache directory .gradle/ # Autogenerated VS/MD/Consulo solution and project files ExportedObj/ .consulo/ *.csproj *.unityproj *.sln *.suo *.tmp *.user *.userprefs *.pidb *.booproj *.svd *.pdb *.mdb *.opendb *.VC.db # Unity3D generated meta files *.pidb.meta *.pdb.meta *.mdb.meta # Unity3D generated file on crash reports sysinfo.txt # Builds *.apk *.aab *.unitypackage *.app # Crashlytics generated file crashlytics-build.properties #Packed Addressables /[Aa]ssets/[Aa]ddressable[Aa]ssets[Dd]ata/*/*.*.bin* # Temporary auto-generated Android Assets /[Aa]ssets/[Ss]treamingAssets/aa.meta /[Aa]ssets/[Ss]treamingAssets/aa/* </pre> |

Πηγή: <https://github.com/github/gitignore/blob/main/Unity.gitignore>

Πίνακας 14. Αρχείο .gitattributes

| Δομή Αρχείου .gitattributes |
|---|
| <pre> # Macro for Unity YAML-based asset files. [attr]unityyaml -text merge=unityyamlmerge diff # Macro for all binary files that should use Git LFS. [attr]lfs -text filter=lfs diff=lfs merge=lfs # Default to auto-normalized line endings. * text=auto # Code *.cs text diff=csharp # Unity Text Assets *.meta unityyaml *.unity unityyaml *.asset unityyaml *.prefab unityyaml *.mat unityyaml *.anim unityyaml *.controller unityyaml *.overrideController unityyaml *.physicMaterial unityyaml *.physicsMaterial2D unityyaml *.playable unityyaml *.mask unityyaml *.brush unityyaml *.flare unityyaml *.fontsettings unityyaml *.guiskin unityyaml *.giparams unityyaml *.renderTexture unityyaml *.spriteatlas unityyaml *.terrainlayer unityyaml *.mixer unityyaml *.shadervariants unityyaml *.preset unityyaml *.asmdef -text diff # Unity Binary Assets *.cubemap lfs *.unitypackage lfs # Note: Unity terrain assets must have "-Terrain" suffix. *-[Tt]errain.asset -unityyaml lfs # Note: Unity navmesh assets must have "-NavMesh" suffix. *-[Nn]av[Mm]esh.asset -unityyaml lfs # Image *.jpg lfs *.jpeg lfs *.png lfs *.apng lfs *.atsc lfs *.gif lfs *.bmp lfs *.exr lfs *.tga lfs *.tiff lfs *.tif lfs *.iff lfs *.pict lfs *.dds lfs *.xcf lfs *.leo lfs *.kra lfs *.kpp lfs *.clip lfs *.webm lfs *.webp lfs </pre> |

```
*.svg lfs
*.svgz lfs
*.psd lfs
*.afphoto lfs
*.afdesign lfs

# Audio
*.mp3 lfs
*.ogg lfs
*.wav lfs
*.aiff lfs
*.aif lfs
*.mod lfs
*.it lfs
*.s3m lfs
*.xm lfs

# Video
*.mov lfs
*.avi lfs
*.asf lfs
*.mpg lfs
*.mpeg lfs
*.mp4 lfs
*.flv lfs
*.ogv lfs
*.wmv lfs

# 3D
*.fbx lfs
*.obj lfs
*.max lfs
*.blend lfs
*.blender lfs
*.dae lfs
*.mb lfs
*.ma lfs
*.3ds lfs
*.dxf lfs
*.c4d lfs
*.lwo lfs
*.lwo2 lfs
*.abc lfs
*.3dm lfs
*.bin lfs
*.glb lfs

# Executables
*.exe lfs
*.dll lfs
*.so lfs
*.pdb lfs
*.mdb lfs

# Packaging
*.zip lfs
*.7z lfs
*.gz lfs
*.rar lfs
*.tar lfs
```




Πηγή : <https://hextantstudios.com/unity-gitattributes/>

Οι πρώτες 4 γραμμές καθορίζουν τη συμπεριφορά συγχώνευσης και διαχωρισμού για συγκεκριμένους τύπους αρχείων της μηχανής Unity. Η ιδιότητα diff = unitymerge λέει στο Git να χρησιμοποιεί το εργαλείο Unity SmartMerge για την επίλυση συγκρούσεων σε

αρχεία σκηνών Unity (*.unity). Η ιδιότητα merge = unityyamlmerge χρησιμοποιείται για τη συγχώνευση συγκρούσεων σε ψηφιακά αρχεία στοιχείων Unity (*.asset) χρησιμοποιώντας το εργαλείο συγχώνευσης YAML της Unity. Η ιδιότητα eof=lf ορίζει τον χαρακτήρα τέλους γραμμής σε LF (line feed), που είναι η προτιμώμενη μορφή σε έργα Unity.

Εδώ αξίζει να αναφερθεί πως το πραγματικό περιεχόμενο του αρχείου .gitattributes σε ένα έργο Unity μπορεί να διαφέρει ανάλογα με τις ειδικές απαιτήσεις του έργου ή τις προσαρμογές που πραγματοποιούνται από την ομάδα ανάπτυξης.

Πίνακας 15. Δωρεάν Πλάνω Αποθετηρίων

| GitHub Free [5] | Gitlab Free [8] | Bitbucket Free [12] |
|---|--|--|
| <ul style="list-style-type: none"> • Unlimited public/private repositories • 500MB of Package storage • 2000 CI/CD minutes/month • Automatic security and version updates | <ul style="list-style-type: none"> • 5GB storage • 10GB transfer per month • 400 compute credits per month • 5 users per namespace | <ul style="list-style-type: none"> • Up to 5 users • Build minutes 50 / month • Git large file storage 1G • Unlimited private repositories |
|  |  |  |
| Πηγή: GitHub | Πηγή: GitLab | Πηγή: Bitbucket |

4.3.3 Perforce Helix Core [6]

Το Perforce Helix Core είναι ένα εξελιγμένο και πλούσιο σε δυνατότητες σύστημα διαχείρισης εκδόσεων που είναι πολύ γνωστό στον τομέα της ανάπτυξης λογισμικού και όχι μόνο. Σχεδιασμένο για να υποστηρίζει ομάδες που διαχειρίζονται μεγάλα project και διάφορων ψηφιακών περιεχομένων, το Helix Core προσφέρει ένα κεντρικό αποθετήριο για την αποθήκευση και την παρακολούθηση αρχείων, τη διασφάλιση της ακεραιότητας των δεδομένων και την ομαλή συνεργασία μεταξύ των μελών μιας ομάδας πάνω σε ένα project. Η εξελιγμένη διαχείριση που παρέχει το Version Control του Helix Core επιτρέπει στους προγραμματιστές να χειρίζονται αποτελεσματικά τον πηγαίο κώδικα, τα έγγραφα, τα

αρχεία πολυμέσων και άλλα πολύτιμα ψηφιακά περιεχόμενα. Η επεκτασιμότητά του και η αρχιτεκτονική υψηλών επιδόσεων το καθιστούν κατάλληλο τόσο για μικρές ομάδες όσο και για μεγάλες επιχειρήσεις, εξυπηρετώντας τις πολύπλοκες απαιτήσεις των σύγχρονων έργων ανάπτυξης λογισμικού. Επιπλέον, το Helix Core είναι συμβατό με πολλά δημοφιλή εργαλεία ανάπτυξης και ροές εργασίας, επιτρέποντας στους προγραμματιστές να αξιοποιούν τις δυνατότητές του χωρίς να διαταράσσουν τις καθιερωμένες πρακτικές τους. Η ασφάλεια έχει ύψιστη σημασία στο Helix Core, με ισχυρούς ελέγχους πρόσβασης, μηχανισμούς ελέγχου ταυτότητας χρηστών και χαρακτηριστικά κρυπτογράφησης δεδομένων που προστατεύουν τις ευαίσθητες πληροφορίες από μη εξουσιοδοτημένη πρόσβαση. Επιπλέον, τα προηγμένα εργαλεία οπτικοποίησης του Helix Core παρέχουν κρίσιμες πληροφορίες σχετικά με την εξέλιξη και τις σχέσεις των αρχείων, επιτρέποντας στους προγραμματιστές να αναλύουν και να κατανοούν τις αλλαγές με την πάροδο του χρόνου. Ο ολοκληρωμένος χαρακτήρας του Helix Core, σε συνδυασμό με την ικανότητά του να εξορθολογήσει τις ροές εργασίας ανάπτυξης και να διασφαλίζει την ακεραιότητα των δεδομένων, το καθιστά απαραίτητο εργαλείο για προγραμματιστές, δημιουργούς ψηφιακών περιεχομένων και ερευνητές που αναζητούν αποτελεσματικές και αξιόπιστες λύσεις Version Control και διαχείρισης ψηφιακών περιεχομένων.















Εικόνα 18. Perforce Helix Core

Πηγή : <https://www.perforce.com/products/helix-core>

4.4 Unity's Interface Version Control

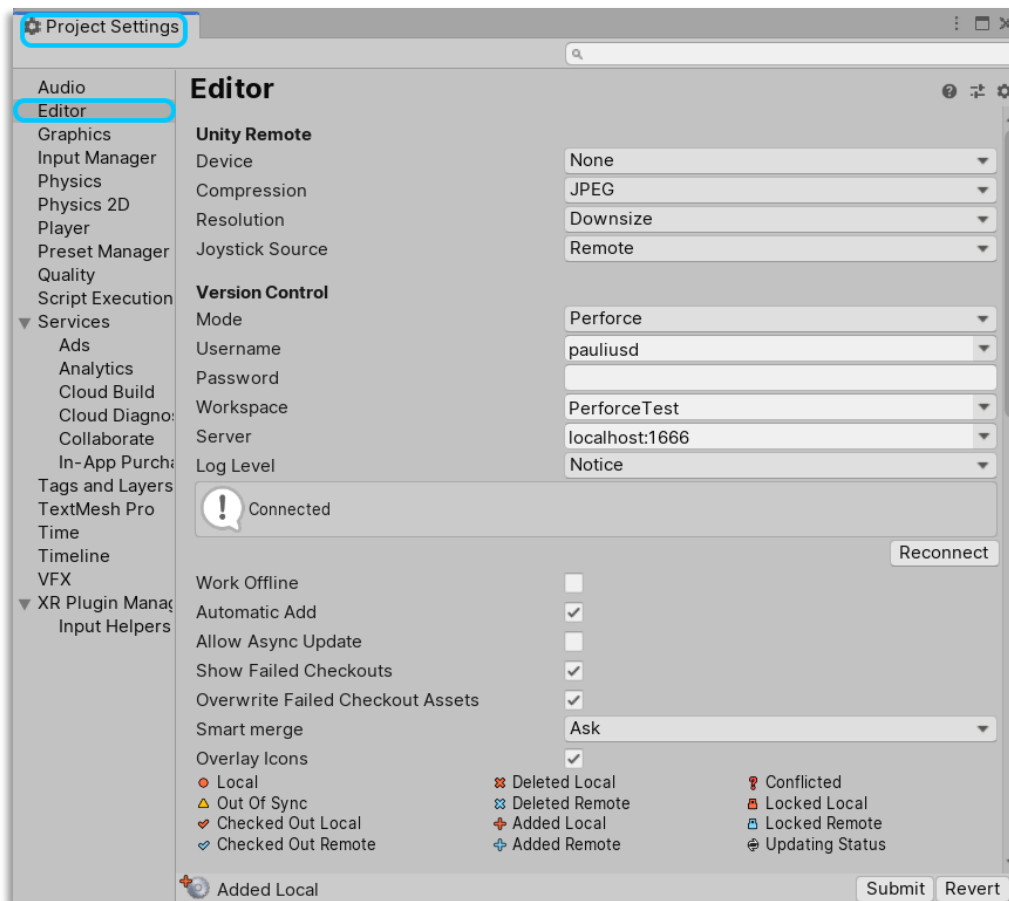
Το Unity εμφανίζει τα ακόλουθα εικονίδια για να απεικονίσει την κατάσταση του Version Control για αρχεία και ψηφιακά περιεχόμενα.

Πίνακας 16. Ενημέρωση Κατάστασης του Version Control

| Εικόνα | Σκοπός |
|---|--|
|  | File added locally, and pending addition into version control. |
|  | File added to version control by another user, and pending addition into version control. |
|  | File is checked out locally. |
|  | File is checked out by another user remotely. |
|  | There has been a conflict merging this file and it needs to be resolved. |
|  | File has been deleted locally, and is pending deletion in version control. |
|  | File has been deleted by another user and is pending deletion in version control. |
|  | File is not yet under version control. You can use the Mark Add operation to add the file manually. |
|  | File is locked by you and other users cannot modify it. |
|  | File is locked by another user and you cannot modify it. |
|  | Another user has checked in a new version of this file. Use the Apply Incoming Changes operation to get the latest version. |
|  | The server is requesting the version control status of this file, or is waiting for a response. This only appears if you use a centralized version control system like Perforce. |

Πηγή: <https://docs.unity3d.com/2023.2/Documentation/Manual/Versioncontrolintegration.html>

Για να ρυθμίσει ο χρήστης το Version Control στο Unity πρέπει να ανοίξει το project του και στην συνέχεια να κατευθυνθεί από την καρτέλα Edit > Project Settings και κατόπιν επιλέγει την κατηγορία Editor.



Εικόνα 19. Setting up version control in Unity

Πηγή : <https://docs.unity3d.com/Manual/Versioncontrolintegration.html>

Κάτω από το Version Control ο προγραμματιστής έχει τις ακόλουθες ρυθμίσεις :

| Property | Function |
|--|---|
| Mode | Select the version control mode. |
| Hidden meta files | Hides the .meta files in your operating system's file explorer. Unity does not show .meta files in the Project view, no matter which mode you choose. |
| Visible meta files | Select this option to work with a version control system that Unity doesn't support. This is the default setting. You can then manage the source Assets and metadata for those Assets with a version control system of your choice. For more information, see documentation on External Version Control Systems . |
| Perforce | Select this option if you use Perforce as your version control system. |
| Plastic SCM | Select this option if you use Plastic SCM as your version control system. |
| Username (Perforce only) | Enter the username associated with your Perforce account. |
| Password (Perforce only) | Enter the password associated with your Perforce account. |
| Workspace (Perforce only) | Enter your workspace (for example, Example_Workspace_1). |
| Server (Perforce only) | Enter the server your Unity Project is on (for example, localhost:1666). |
| Host (Perforce only) | Enter the hostname that your computer should impersonate (for example, workstation123.perforce.com) |
| Log Level | Select how much version control information to receive in Unity's console log. |
| Verbose | Unity logs every operation related to version control. This option provides very detailed logging, and is useful if you want to debug your version control setup. |
| Info | Unity logs errors, warnings, and information related to version control. |
| Notice | Unity logs errors and warnings. |
| Fatal | Unity prints only fatal errors to the console. |
| Status | Displays information on the status of the connection to the version control system. If you are not connected, select Connect to connect to the system you have configured. |
| Automatic Add | When this setting is enabled, Unity automatically adds your files to the version control system when you add them to the Project, either via the Editor or the folder on disk. When this setting is disabled, you need to add files manually to the version control system. This setting is enabled by default. |
| Work Offline (Perforce only) | Enable this setting to work offline. When this setting is enabled, you need to reconcile offline work in P4V or use the reconcile command in P4 to bring the Perforce server depot up to date with the work you did while offline. For more information, see Working offline with Perforce |
| Allow Async Update (Perforce only) | Enable this setting to use asynchronous version control status queries. When enabled, Perforce updates the version control status of files without stalling the Unity Editor. Use this setting when the connection to your version control server has high latency. Note: Only status queries are asynchronous. Unity synchronously performs operations that change the state of files, or require up-to-date knowledge of a file status. |
| Show Failed Checkouts (Perforce only) | Enable this property to show a dialog when Perforce can't perform the check out operation. This might happen if you lose connection, or if another user has exclusively checked out the Asset you want to edit. |
| Overwrite Failed Checkout Assets (Perforce only) | When you enable this setting, Unity saves any Assets that can not be checked out. This means Unity forces through a save to a file, even if Perforce cannot check out the file. This is enabled by default. If you disable it, Unity doesn't force your files to save if Perforce can't check them out. |
| Overlay Icons | Enable this setting to display version control status icons in the Editor. |
| Smart Merge | Smart Merge makes it easier for Unity to merge files that have changes on the same line. This is useful if several users are working on the same Project at the same time. |
| Off | Disable Smart Merge. |
| Ask | Enable Smart Merge, but receive a notification before you merge, if a conflict happens. This is the default setting. |
| Premerge | Automatically use Smart Merge. |

Εικόνα 20. Ρυθμίσεις του Version Control στο Unity

Πηγή : <https://docs.unity3d.com/2022.2/Documentation/Manual/Versioncontrolintegration.html>

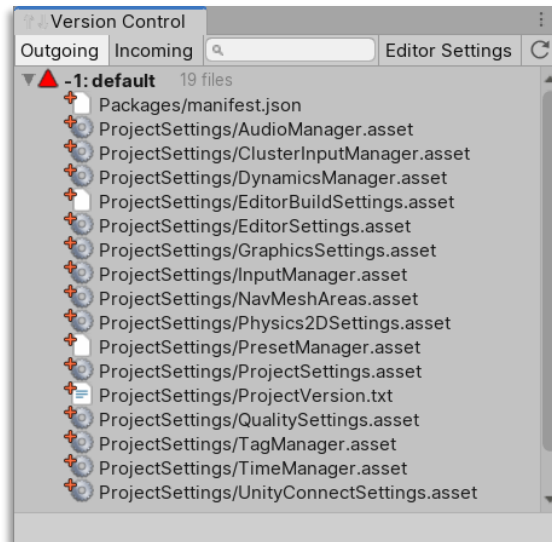
Εάν ο χρήστης ρυθμίσει σωστά τον Editor τότε θα μπορεί να εκτελεί εντολές διαχείρισης έκδοσης μέσα από αυτόν. Κάνοντας δεξί κλικ σε κάποιο ψηφιακό στοιχείο εμφανίζονται κάποιες επιλογές ανάλογα με την έκδοση του εκάστοτε Version Control που χρησιμοποιεί. Ο παρακάτω πίνακας δείχνει ποιες είναι οι επιλογές ανάλογα με την κάθε έκδοση.

Πίνακας 17. Ρυθμίσεις Version Control ανάλογα με την έκδοση

| Version control operation | Description | Perforce | Plastic SCM |
|---------------------------|--|----------|--|
| Get Latest | This updates the files on your machine to match those in the version control system. | Yes | No. To get the latest changes and update the file, you need to use the version control window (Window > Asset Management > Version Control). |
| Submit | Submits the current state of the files to the version control system. | Yes | Yes |
| Check Out | Allows changes to be made to the files. | Yes | Yes |
| Check Out (Other) | Select whether to check out both the Asset and its .meta file, or only the Asset, or only the .meta file. | Yes | No |
| Mark Add | Adds the files into version control. | Yes | Yes |
| Revert | Discards changes to open changed files. | Yes | Yes |
| Revert Unchanged | Removes the checked out status from files that have previously been checked out but you haven't modified. | Yes | Yes |
| Resolve Conflicts | Resolves conflicts on files that have been changed by multiple users. | Yes | No. Conflicts appear in the version control menu, but you need to resolve them in the Plastic SCM GUI. |
| Lock | Prevents other users from submitting changes to the files. | Yes | No. To lock or unlock files in Plastic SCM, you must edit a specific Plastic SCM lock file externally. For more information, see the page on Plastic SCM Integration . |
| Unlock | Releases the lock and allows anyone to submit changes. | Yes | No. To lock or unlock files in Plastic SCM, you must edit a specific Plastic SCM lock file externally. For more information, see the page on Plastic SCM Integration . |
| Diff | Compares the differences between the local files on your computer and the files on the server. You can choose to diff only the Asset file, or the Asset file and its .meta file. | Yes | Yes |

Πηγή : <https://docs.unity3d.com/2022.2/Documentation/Manual/Versioncontrolintegration.html>

Για να έχει μια εικόνα ο προγραμματιστής ποια αρχεία έχουν αλλάξει πρέπει να ανοίξει το παράθυρο του version control με την λίστα αλλαγής των αρχείων που βρίσκεται στο Windows > Asset Management > Version Control.



Εικόνα 21. The Version Control window docked in the Editor

Η καρτέλα εξερχομένων περιλαμβάνει όλες τις αλλαγές που εκκρεμούν και δεν έχουν δεσμευτεί από το version control System. Στην καρτέλα εισερχομένων υπάρχουν όλες οι αλλαγές που πρέπει να ανασύρουν από το version control. Με δεξί κλικ σε κάποιο ψηφιακό στοιχείο ή στις λίστες αλλαγών στο παραπάνω παράθυρο ο χρήστης θα μπορεί να εκτελέσει διάφορες λειτουργίες. Για να μετακινήσει ψηφιακά στοιχεία μεταξύ λιστών απλώς τα σέρνει από την επικεφαλίδα της μιας στην επικεφαλίδα της άλλης.



Εικόνα 22. Version Control and Project Organization

Πηγή: <https://resources.unity.com/games/version-control-project-organization-best-practices-ebook>

4.5 Unity Learn

Ένα πολύ θετικό στοιχείο που παρέχει η εταιρία είναι η διαδικτυακή πλατφόρμα Unity Learn που είναι διαθέσιμη από την Unity Technologies όπως προαναφέρθηκε και προσφέρει ένα ευρύ φάσμα εκπαιδευτικών προγραμμάτων και σεμιναρίων για άτομα που ενδιαφέρονται να μάθουν το Unity. Το Unity Learn παρέχει πρόσβαση σε διάφορα εκπαιδευτικά υλικά, συμπεριλαμβανομένων διαδραστικών σεμιναρίων, μαθημάτων, έργων, τεκμηρίωσης και ζωντανών εκπαιδευτικών συνεδριών.


Η πλατφόρμα έχει σχεδιαστεί για να ανταποκρίνεται σε διαφορετικά επίπεδα δεξιοτήτων, από αρχάριους έως προχωρημένους χρήστες, και καλύπτει διάφορα θέματα που σχετίζονται με την ανάπτυξη παιχνιδιών Unity, όπως scripting, animation, φυσική, γραφικά και άλλα. Το Unity Learn στοχεύει να εμπλουτίσει με γνώσεις και δεξιότητες τους χρήστες που ενδιαφέρονται να δημιουργήσουν τα δικά τους παιχνίδια, διαδραστικές εφαρμογές και προσομοιώσεις χρησιμοποιώντας τη μηχανή Unity.

Προσφέρει τόσο δωρεάν όσο και επί πληρωμή περιεχόμενο, με τους δωρεάν πόρους να παρέχουν μια σταθερή βάση για την εκμάθηση του Unity και το επί πληρωμή περιεχόμενο να προσφέρει πιο εμπειριστατωμένα και προχωρημένα θέματα. Η πλατφόρμα ενημερώνεται τακτικά με νέο περιεχόμενο για να συμβαδίζει με τα τελευταία χαρακτηριστικά και τις εξελίξεις στο Unity.

Συνολικά, το Unity Learn χρησιμεύει ως σημείο εκκίνησης για φιλόδοξους προγραμματιστές παιχνιδιών, φοιτητές, ερασιτέχνες και επαγγελματίες που θέλουν να αναπτύξουν τις δεξιότητες και να διευρύνουν τις γνώσεις τους στην ανάπτυξη παιχνιδιών Unity.

Learning Pathways

Pathways are guided learning experiences that help anyone interested in breaking into the gaming and real-time 3D tech industries. Gain the skills you need to obtain a job, regardless of prior experience. [Learn more about Pathways](#)




Unity Essentials

2 weeks · Foundational **+600 XP**

New to Unity? This guided learning Pathway includes everything you need to get started.

[Unity Essentials Pathway ↗](#)




Junior Programmer

12 weeks · Foundational **+3000 XP**

Ready to code? This guided learning Pathway will take you from zero to job-ready!

[Junior Programmer Pathway ↗](#)




Creative Core

10 weeks · Beginner **+3000 XP**

Ready for more? Level up your core understanding of Unity with the creative aspects of the engine.


[Creative Core Pathway ↗](#)



VR Development

6 weeks · Beginner **+600 XP**

Ready to develop for VR? This guided learning Pathway will prepare you for a job in the VR industry!



Mobile AR Development

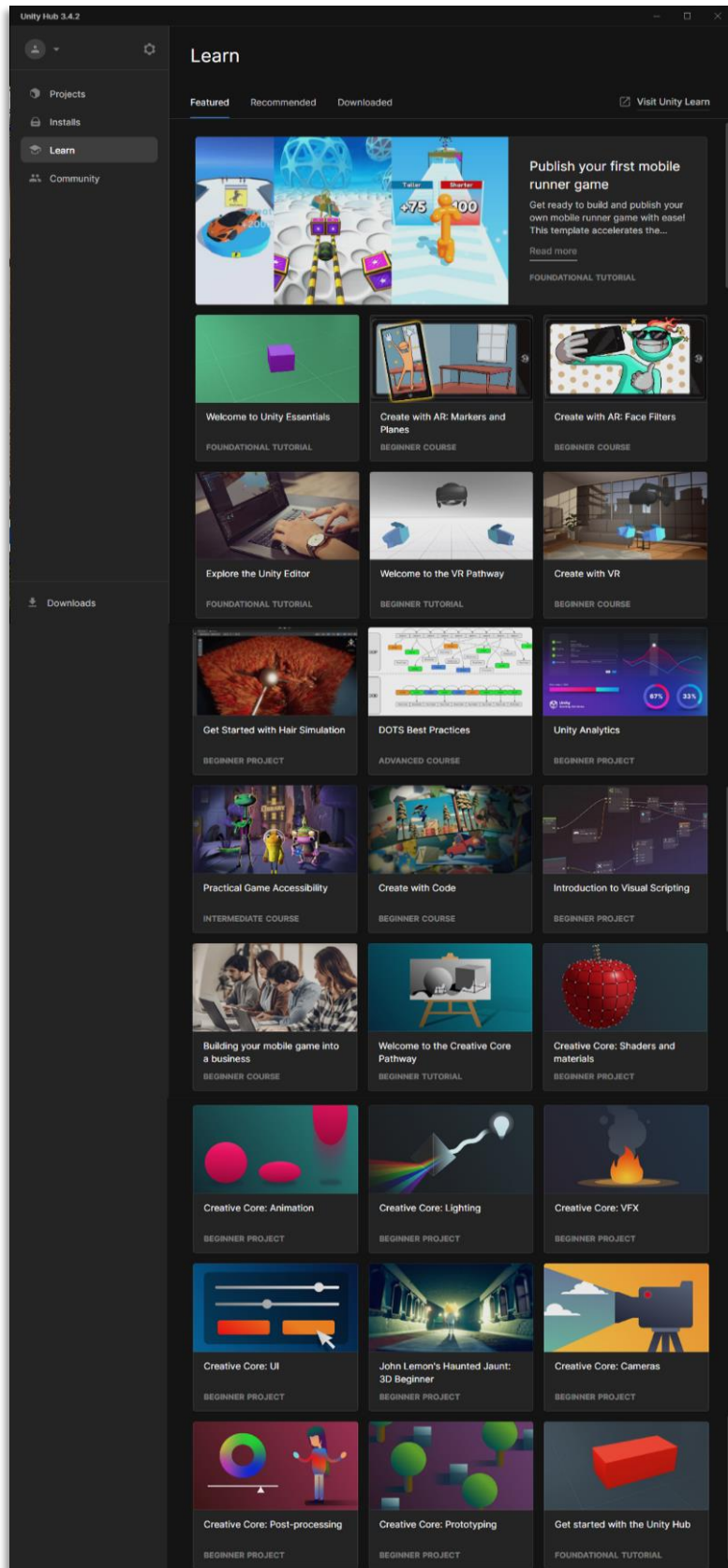
8 weeks · Foundational **+600 XP**

Ready to create AR experiences? In this learning pathway, you'll develop AR apps compatible with iOS and Android devices!

Εικόνα 23. Unity learn στο Διαδίκτυο

Πηγή : <https://learn.unity.com/>

Ένα από τα βασικά πλεονεκτήματα του Unity Learn είναι τα μονοπάτια μάθησης. Αυτά τα μονοπάτια μάθησης καθοδηγούν τους εκπαιδευόμενους μέσα από μια σειρά μαθημάτων και έργων, αναπτύσσοντας σταδιακά τις γνώσεις και τις δεξιότητές τους με λογικό και οργανωμένο τρόπο. Είτε κάποιος θέλει να μάθει τα βασικά στοιχεία του Unity, να εντρυφήσει σε συγκεκριμένους τομείς, όπως η ανάπτυξη παιχνιδιών 2D ή 3D, ή να εξερευνήσει προχωρημένα θέματα, όπως η τεχνητή νοημοσύνη το multiplayer η εικονική πραγματικότητα ή επαυξημένη πραγματικότητα, το Unity Learn προσφέρει μονοπάτια μάθησης (pathways) που ανταποκρίνονται στις ανάγκες του. Αυτή η δομημένη προσέγγιση βοηθά τους εκπαιδευόμενους να παραμείνουν συγκεντρωμένοι, να έχουν κίνητρα και διασφαλίζει ότι θα αναπτύξουν μια σταθερή βάση στην ανάπτυξη παιχνιδιών Unity.



Εικόνα 24. Unity Learn στο Unity Hub

Επιπλέον, το Unity Learn παρέχει μια ζωντανή και υποστηρικτική κοινότητα για τους εκπαιδευόμενους να συνδεθούν με συναδέλφους προγραμματιστές, να μοιραστούν τις γνώσεις τους και να αναζητήσουν καθοδήγηση σε διάφορα θέματα. Η πλατφόρμα προσφέρει φόρουμ συζητήσεων, όπου οι εκπαιδευόμενοι μπορούν να θέτουν ερωτήσεις, να συμμετέχουν σε συζητήσεις και να λαμβάνουν feedback από έμπειρους προγραμματιστές. Αυτή η κοινότητα προάγει τη συνεργασία και δίνει τη δυνατότητα στους εκπαιδευόμενους να βελτιώσουν τις δεξιότητές τους μέσω της αλληλεπίδρασης και της συνεργασίας με άλλους. Επιπλέον, το Unity Learn φιλοξενεί τακτικά ζωντανές εκδηλώσεις, διαδικτυακά σεμινάρια και προκλήσεις για την εμπλοκή της κοινότητας και την παροχή ευκαιριών στους εκπαιδευόμενους να παρουσιάσουν τη δουλειά τους και να μάθουν από ειδικούς του κλάδου. Συνολικά, το Unity Learn είναι ένα σημείο αναφοράς για όποιον θέλει να μάθει, να αναπτύξει και να διαπρέψει στην ανάπτυξη παιχνιδιών στο Unity.

4.6 Διαχείριση κώδικα στο Unity

Εδώ πρέπει να αναφερθούν κάποια βασικά εργαλεία για την διαχείριση του κώδικα που παράγει η μηχανή Unity. Τα πιο γνωστά είναι το Visual Studio Code και Visual Studio. Οι ενσωματωμένοι επεξεργαστές κώδικα, όπως το Visual Studio Code και το Visual Studio, χρησιμοποιούνται ευρέως για την ανάπτυξη παιχνιδιών στο πλαίσιο του Unity. Με τη χρήση αυτών των εργαλείων, οι προγραμματιστές μπορούν να αξιοποιήσουν τις προηγμένες δυνατότητες που προσφέρουν για τη γρήγορη και αποτελεσματική ανάπτυξη.

Τόσο το Visual Studio Code όσο και το Visual Studio έχουν τα θετικά και αρνητικά τους ως προς τη χρήση τους με το Unity. Ας εξετάσουμε μερικά από αυτά:

Θετικά του Visual Studio Code:

- **Ελαφρύ και γρήγορο:** Το Visual Studio Code είναι ένα ελαφρύ περιβάλλον ανάπτυξης με γρήγορο χρόνο φόρτωσης και απόκρισης. Αυτό το καθιστά ιδανικό για προγραμματιστές που αναζητούν ένα ελαφρύ και απλό εργαλείο.
- **Επεκτασιμότητα:** Το Visual Studio Code υποστηρίζει ένα ευρύ φάσμα πρόσθετων και επεκτάσεων που επιτρέπουν την προσαρμογή και επέκταση των δυνατοτήτων του περιβάλλοντος ανάπτυξης. Αυτό επιτρέπει στους προγραμματιστές να προσαρμόσουν το περιβάλλον στις ατομικές τους ανάγκες και προτιμήσεις.

- **Λειτουργία σε Linux και MacOS:** Το Visual Studio Code είναι ένα περιβάλλον ανάπτυξης που προσφέρει τη δυνατότητα να εκτελείται σε συστήματα Linux και Mac OS. Αυτό το χαρακτηριστικό επιτρέπει στους χρήστες αυτών των λειτουργικών συστημάτων να απολαμβάνουν τα οφέλη του Visual Studio Code και να αναπτύξουν τις εφαρμογές τους με άνεση και ευελιξία.

Θετικά του Visual Studio:

- **Αποσφαλμάτωση (Debugging):** Το Visual Studio παρέχει ένα πλούσιο σύνολο εργαλείων αποσφαλμάτωσης που επιτρέπουν την εντοπισμό και επίλυση σφαλμάτων στον κώδικα. Αυτό περιλαμβάνει το βήμα-βήμα εκτέλεση, τον έλεγχο των τιμών μεταβλητών, τον ανίχνευση των σημείων διακοπής και πολλά άλλα, που καθιστούν την αποσφαλμάτωση ευκολότερη και πιο αποτελεσματική.
- **Ενσωματωμένες λειτουργίες Unity:** Το Visual Studio έχει στενή ενσωμάτωση με το Unity και παρέχει ειδικές λειτουργίες για την ανάπτυξη σε αυτό το περιβάλλον. Αυτό περιλαμβάνει την αυτόματη συμπλήρωση κώδικα για τα στοιχεία του Unity, την οπτική ανάλυση των συνδέσεων μεταξύ στοιχείων, την αναγνώριση των συναρτήσεων Unity και άλλα, που διευκολύνουν την ανάπτυξη παιχνιδιών στο Unity.

Αρνητικά του Visual Studio Code:

- **Περιορισμένη λειτουργικότητα:** Το Visual Studio Code έχει λιγότερες λειτουργίες από το πλήρες Visual Studio, καθώς πρόκειται για ένα πιο ελαφρύ και απλό περιβάλλον ανάπτυξης. Αυτό μπορεί να περιορίζει τις δυνατότητες των προγραμματιστών και να απαιτεί τη χρήση επιπλέον εργαλείων ή προσαρμογών για να καλύψουν τις ανάγκες τους στην ανάπτυξη με το Unity.
- **Μειωμένη ενσωμάτωση Unity:** Αν και το Visual Studio Code παρέχει ορισμένες λειτουργίες για την υποστήριξη της ανάπτυξης στο Unity, η ενσωμάτωση του Unity είναι λιγότερο ολοκληρωμένη σε σχέση με το πλήρες Visual Studio. Αυτό μπορεί να οδηγήσει σε λιγότερη αυτοματοποίηση, αναγνώριση κώδικα και ενσωμάτωση λειτουργιών Unity, καθώς και πιθανώς σε μεγαλύτερη ανάγκη για χειροκίνητη διαμόρφωση και ρύθμιση.

Αρνητικά του Visual Studio μπορεί να περιλαμβάνουν:

- **Βαρύτητα:** Το Visual Studio είναι ένα πλήρες και ισχυρό περιβάλλον ανάπτυξης, και αυτό μπορεί να οδηγήσει σε μεγαλύτερη κατανάλωση πόρων στον υπολογιστή. Αυτό μπορεί να είναι ειδικά αισθητό σε παλαιότερα ή περιορισμένων πόρων συστήματα.
- **Πολυπλοκότητα:** Εξαιτίας των πολλών λειτουργιών και επιλογών που προσφέρει, το Visual Studio μπορεί να είναι πιο πολύπλοκο στην χρήση σε σχέση με άλλα απλούστερα περιβάλλοντα ανάπτυξης. Αυτό μπορεί να απαιτεί κάποιο χρόνο για εξοικείωση και εκμάθηση.
- **Απαιτήσεις χώρου αποθήκευσης:** Το Visual Studio απαιτεί συνήθως μεγαλύτερο χώρο αποθήκευσης στον υπολογιστή σε σύγκριση με άλλα ελαφρύτερα περιβάλλοντα ανάπτυξης. Αυτό μπορεί να είναι αξιοσημείωτο εάν ο χώρος αποθήκευσης είναι περιορισμένος.

Είναι σημαντικό να σημειωθεί ότι οι επιλογές ανάμεσα στο Visual Studio Code και το Visual Studio εξαρτώνται από τις προτιμήσεις και τις ανάγκες του κάθε προγραμματιστή. Καθένα από αυτά τα περιβάλλοντα ανάπτυξης έχει τα δικά του πλεονεκτήματα και περιορισμούς και πρέπει να εξεταστούν προσεκτικά πριν από την επιλογή τους. Οι προγραμματιστές που αναζητούν ένα ελαφρύ και απλό περιβάλλον ανάπτυξης με έμφαση στην επεκτασιμότητα και την προσαρμοστικότητα μπορούν να επωφεληθούν από το Visual Studio Code. Αυτό τους επιτρέπει να προσαρμόσουν το περιβάλλον ανάπτυξης σύμφωνα με τις προσωπικές τους ανάγκες. Έχουν την δυνατότητα να προσθέσουν επιπλέον χαρακτηριστικά στην εφαρμογή, από μια ποικιλία πρόσθετων για να ενισχύσουν την απόδοση και την αποτελεσματικότητά τους.

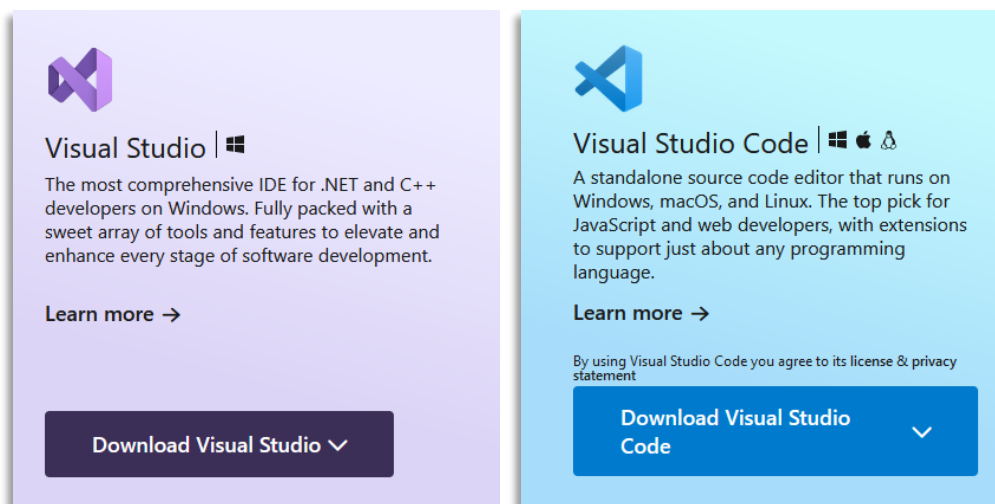
Οι προγραμματιστές που χρειάζονται μια πλήρη και ολοκληρωμένη εμπειρία ανάπτυξης με εκτεταμένες λειτουργίες και ενσωματωμένη υποστήριξη για το Unity μπορούν να επωφεληθούν περισσότερο από το Visual Studio. Αυτό τους παρέχει ένα πλούσιο σετ εργαλείων για την ανάπτυξη παιχνιδιών στο Unity και την αποσφαλμάτωσή τους, όπως η ομαδική συνεργασία, η δυνατότητα ανάλυσης κώδικα, η αυτόματη συμπλήρωση κώδικα, η αποσφαλμάτωση και πολλά άλλα. Αυτό μπορεί να καταστήσει την ανάπτυξη και τη συντήρηση του κώδικα πιο αποτελεσματική και ευκολότερη. Τέλος να σημειωθεί πως υπάρχουν και άλλες επιλογές για την σύνταξη και επεξεργασία του κώδικα από τρίτες

εταιρίες ή χρήστες μέσα και από το κατάστημα του Unity αλλά και στην αγορά όπως πχ. το Rider από την JetBrains τα οποία όμως τις περισσότερες φορές είναι επί πληρωμή.



Εικόνα 25. JetBrains Rider for Unity

Πηγή : <https://www.jetbrains.com/idea/dotnet-unity/>



Εικόνα 26. Microsoft Visual Studio and Visual Studio Code[10]

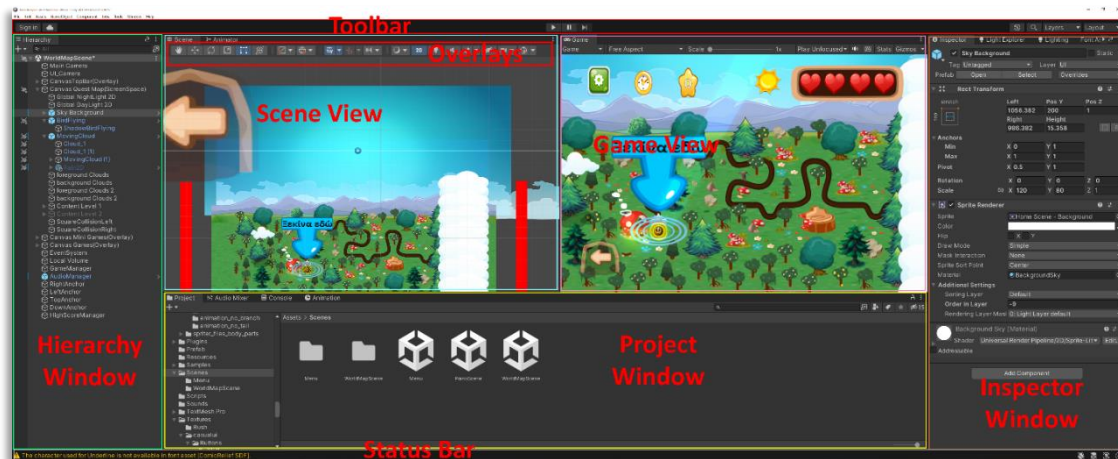
Πηγή : <https://visualstudio.microsoft.com/>

4.7 Unity editor interface [14][16][17][19][23]

Με την εκκίνηση του Unity ο χρήστης θα αντικρίσει μια εικόνα που περιλαμβάνει κάποια panel στα δεξιά, στα αριστερά, στο επάνω και στο κάτω μέρος της οθόνης του editor. Αυτά είναι τα εξής :

- Το Toolbar
- Το παράθυρο Ιεραρχίας
- Η Game View

- Η Scene View
- Overlays
- Το παράθυρο του Inspector
- Το παράθυρο Project
- Η Status bar



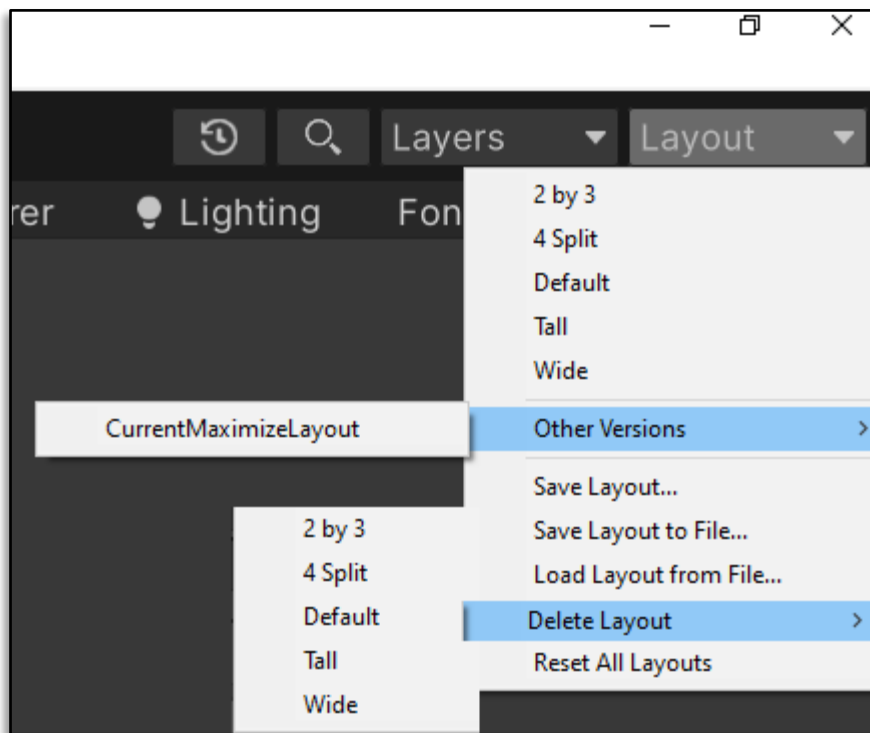
Εικόνα 27. Unity editor interface

Πηγή : <https://docs.unity3d.com/Manual/UsingTheEditor.html>

Ο Unity editor χωρίζεται σε μικρά panel όπου ο χρήστης έχει την δυνατότητα να τα πάρει και να τα προσαρμόσει στις δίκες του προτιμήσεις ανάλογα με τις ανάγκες του. Παρόλα αυτά ο editor παρέχει στον χρήστη και κάποιες προεπιλογές ώστε να μην χρειάζεται να ασχολείται με τις συγκεκριμένες ιδιότητες εάν αυτός δεν το θέλει. Οι επιλογές αυτές είναι οι εξής :

- 2 by 3
- 4 Split
- Default
- Tall
- Wide
- Other Versions -> Current Maximized Layout
- Save Layout ...
- Save Layout to File...
- Load Layout from File...

- Delete Layout -> 2 by 3 , 4 Split, Default, Tall ,Wide
- Reset All Layouts



Εικόνα 28. Unity version 2021.3.8f1 editor layout

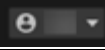

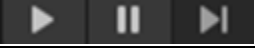


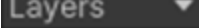

Το αρχικό layout που παρουσιάζεται στον χρήστη είναι το Default.

Η Γραμμή εργαλείων βρίσκεται πάντα στην κορυφή του περιβάλλοντος εργασίας του Unity Editor και αποτελείται από διάφορες ομάδες στοιχείων ελέγχου τα οποία είναι τα μόνα που δεν μπορούν να γίνουν αναδιάταξη:



Εικόνα 29. Main Interface Unity Toolbar

Πίνακας 18. Toolset of Unity Toolbar

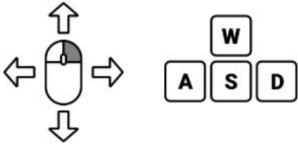






| Control | Description |
|---|--|
|  | Access your Unity Account from the Account drop-down menu. |
|  | Click the Cloud button to open the <u>Unity Services</u> window. |
|  | Use the Play, Pause, and Step buttons in the <u>Game view</u> . |
|  | Use <u>Undo History</u> to view, undo, or redo actions performed in the Editor |
|  | Open <u>Search</u> . |
|  | Control which objects appear in Scene view from the <u>Layers</u> drop-down menu. |
|  | Change the arrangement of your views and then save the new layout or load an existing from the <u>Layout</u> drop-down menu. |

Πηγή : <https://docs.unity3d.com/Manual/Toolbar.html>

4.7.1 Scene View

Η **Scene View** χρησιμοποιείται για την επιλογή και τοποθέτηση του σκηνικού, των χαρακτήρων, των καμερών, των φώτων και όλων των άλλων τύπων αντικειμένων παιχνιδιού που δημιουργείτε. Η επιλογή, ο χειρισμός και η τροποποίηση των GameObjects στην προβολή της σκηνής αποτελούν βασικές δεξιότητες που πρέπει να μάθει ο χρήστης για να ξεκινήσει την εργασία του στο Unity. Σε αυτήν την ενότητα, παρέχεται μια λεπτομερής επεξήγηση για τα ακόλουθα :

- Scene view navigation
- Pick and select GameObjects
- Overlays
- Position GameObjects
- Custom Editor tools
- Hide and show GameObjects
- Scene view control bar
- Scene view Camera
- Gizmos menu

| Scene view navigation | | Toolbar tools | | | |
|---|--|-----------------------|---|-----------------|---|
|  | | Tool | Icon | Shortcut | Purpose |
| Select a GameObject in the Scene view | Left-click the GameObject | Hand Tool |  | Q | Move your Scene around in the window |
| Move the Scene view camera | Middle-click and drag to your desired view | Move Tool |  | W | Select items and move them individually. |
| Rotate the Scene view camera | Right-click and drag to your desired view | Rotate Tool |  | E | Select items and rotate them |
| | | Scale Tool |  | R | Scale your GameObjects up and down |
| | | Rect Transform Tool |  | T | Move, rotate, or scale 2D GameObjects or UI |
| | | Rotate, Move or Scale |  | Y | Move, rotate, or scale 3D GameObjects |

Εικόνα 30. Unity Tools and Shortcuts



















Τα Overlays είναι η βασική ομάδα εργαλείων για τη διαχείριση των αντικειμένων στην Scene View. Από αριστερά προς τα δεξιά η πρώτη ομάδα εργαλείων που συναντάει ο προγραμματιστής είναι το transform tools set.



Εικόνα 31. Transform Tool Set

Η συγκεκριμένη ομάδα εργαλείων είναι υπεύθυνη για το χειρισμό του ψηφιακού στοιχείου στην σκηνή μας. Το πρώτο στοιχείο είναι το View Tool (hotkey: Q)

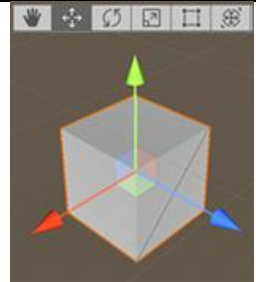
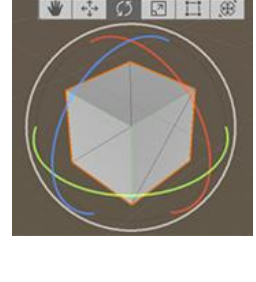
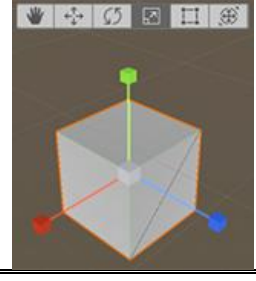
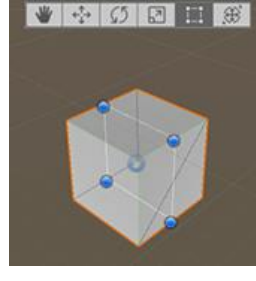
Πίνακας 19. Hand Tool Controls

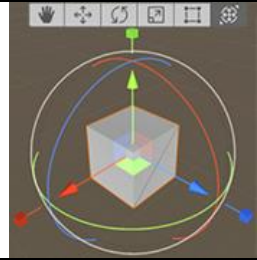
| Control | Description |
|--------------|---|
| Move |       Click and drag to move the Camera around. |
| Orbit |       Hold Alt (Windows) or Option (macOS), and left-click and drag to orbit the Camera around the current pivot point. This option isn't available in 2D mode, because the view is orthographic. |
| Zoom |       Hold Alt (Windows) or Option (macOS), and right-click and drag to zoom the Scene view. On macOS, you can also hold Control , and left-click and drag instead. |

Πηγή : <https://docs.unity3d.com/Manual/SceneViewNavigation.html>

Τα επόμενα εργαλεία στη σειρά είναι το move tool , το Rotate tool, το Scale tool, το RectTransform και το Transform tool.

Πίνακας 20. Transform tools set Move, Rotate, Scale, RectTransform and Transform

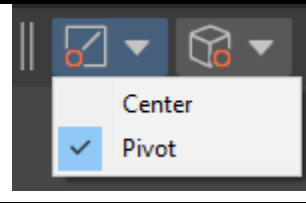
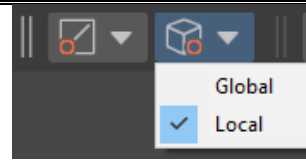
| Control | Description | Gizmo Axis Icon |
|---|---|---|
| <p>Move</p> <p>Keyboard Shortcut W</p> | <p>At the center of the Move Gizmo, there are three small squares you can use to drag the GameObject within a single plane (meaning you can move two axes at once while the third keeps still). If you hold shift while clicking and dragging in the center of the Move Gizmo, the center of the Gizmo changes to a flat square. The flat square indicates that you can move the GameObject around on a plane relative to the direction the Scene view Camera is facing.</p> |  |
| <p>Rotate</p> <p>Keyboard Shortcut E</p> | <p>With the Rotate tool selected, change the GameObject's rotation by clicking and dragging the axes of the wireframe sphere Gizmo that appears around it. As with the Move Gizmo, the last axis you changed will be colored yellow. Think of the red, green and blue circles as performing rotation around the red, green and blue axes that appear in the Move mode (red is the x-axis, green in the y-axis, and blue is the z-axis). Finally, use the outermost circle to rotate the GameObject around the Scene view z-axis. Think of this as rotating in screen space.</p> |  |
| <p>Scale</p> <p>Keyboard Shortcut R</p> | <p>The Scale tool lets you rescale the GameObject evenly on all axes at once by clicking and dragging on the cube at the center of the Gizmo. You can also scale the axes individually, but you should take care if you do this when there are child GameObjects, because the effect can look quite strange.</p> |  |
| <p>RectTransform</p> <p>Keyboard Shortcut T</p> | <p>The RectTransform is commonly used for positioning 2D elements such as Sprites or UI elements, but it can also be useful for manipulating 3D GameObjects. It combines moving, scaling and rotation into a single Gizmo:</p> <ul style="list-style-type: none"> • Click and drag within the rectangular Gizmo to move the GameObject. • Click and drag any corner or edge of the rectangular Gizmo to scale the GameObject. • Drag an edge to scale the GameObject along one axis. • Drag a corner to scale the GameObject on two axes. • To rotate the GameObject, position your cursor just beyond a corner of the rectangle. The cursor changes to display a rotation icon. Click and drag from this area to rotate the GameObject. <p>Note that in 2D mode, you can't change the z-axis in the Scene using the Gizmos. However, it is useful for certain scripting techniques to use the z-axis for other purposes, so you can still set the z-axis using the Transform component in the Inspector. For more information on transforming GameObjects, see documentation on the Transform Component.</p> |  |

| | | |
|--|--|---|
| <p>Transform</p> <p>Keyboard Shortcut Y</p> | <p>The Transform tool combines the Move, Rotate and Scale tools. Its Gizmo provides handles for movement and rotation. When the Tool Handle Rotation is set to Local (see below), the Transform tool also provides handles for scaling the selected GameObject.</p> |  |
|--|--|---|

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/PositioningGameObjects.html>

Τα επόμενα δυο εργαλεία χειρίζονται τα pivot points του position και του rotation.

Πίνακας 21. Pivot Points

| Pivot Point | Description | Tool |
|----------------------------|--|--|
| <p>For Position</p> | <p>Use the dropdown menu to switch between Pivot and Center.</p> <ul style="list-style-type: none"> • Pivot positions the Gizmo at the actual pivot point of the GameObject, as defined by the Transform component. • Center positions the Gizmo at a center position based on the selected GameObjects. |  |
| <p>For Rotation</p> | <p>Use the dropdown menu to switch between Local and Global.</p> <ul style="list-style-type: none"> • Local keeps the Gizmo's rotation relative to the GameObject's. • Global clamps the Gizmo to world space orientation. |  |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/PositioningGameObjects.html>




Τα επόμενο εργαλείο είναι το snapping . Η μηχανή παρέχει τρεις τύπους :

- World grid snapping: Σύγκλιση ενός GameObject σε ένα πλέγμα ορισμένο στους άξονες X, Y ή Z ή μετασχηματισμός ενός GameObject σε βήματα κατά μήκος των αξόνων X, Y ή Z. Αυτό είναι διαθέσιμο μόνο κατά τη χρήση του προσανατολισμού της λαβής World, ή Global.
- Surface snapping: Κλείδωμα του GameObject στην τομή οποιουδήποτε Collider
- Vertex snapping: Κλείδωμα οποιασδήποτε κορυφής από ένα δεδομένο πλέγμα (Mesh) στη θέση της κορυφής (vertex) ή της επιφάνειας (surface) ενός άλλου πλέγματος. Μπορείτε να κάνετε snapping κορυφής σε κορυφή, κορυφής σε επιφάνεια και pivot σε κορυφή (vertex).




Εικόνα 32. Grid and snap toolbar overlay

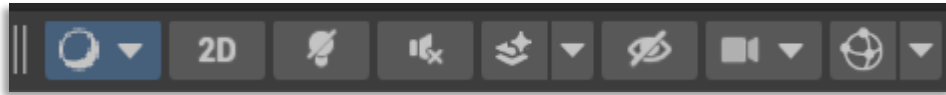
Πίνακας 22. Grid and Snap Toolbar Overlay

| Control | Icon | Actions |
|------------------------|---|---|
| Grid visibility |  | <ul style="list-style-type: none"> <u>Toggle grid on or off</u> <u>Change grid axis</u> <u>Change grid opacity</u> <u>Move grid to gizmo handle or back to origin</u> |
| Grid snapping |  | <ul style="list-style-type: none"> <u>Toggle grid snapping on or off</u> <u>Change grid size</u> <u>Align a GameObject to an axis</u> |
| Snap increment |  | <ul style="list-style-type: none"> <u>Change the default increment snap values</u> |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/GridSnapping.html>


Μετά ακολουθεί το set εργαλείων του Scene view (View Options toolbar) όπου ο χρήστης μπορεί να χρησιμοποιήσει τις επιλογές της Σκηνής προβολής View Options (Επιλογές προβολής Επικάλυψης) για να επιλέξει διάφορες επιλογές για την προβολή της Σκηνής, την λειτουργία της κάμερας, και να ενεργοποιήσει/απενεργοποιήσει το φωτισμό, τον ήχο και διάφορα άλλα εφέ όπως skybox, flares, fog, always refreshing, post processing και particle systems.

| | |
|---|---|
|  | <p>Τα προαναφερόμενα στοιχεία ελέγχου επηρεάζουν μόνο την προβολή σκηνής κατά τη διάρκεια της ανάπτυξης και δεν έχουν καμία επίδραση στο κατασκευασμένο παιχνίδι.</p> |
|---|---|



Εικόνα 33. Scene view (View Options toolbar)

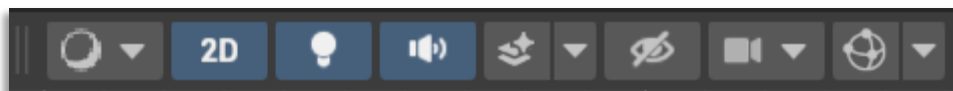
Πίνακας 23. Draw Mode Menu

| Draw Mode |  | Function |
|----------------------------|---|---|
| Shading Mode | Shaded | Show surfaces with their textures visible. |
| | Wireframe | Draw meshes with a wireframe representation. |
| | Shaded Wireframe | Show meshes textured and with wireframes overlaid. |
| Miscellaneous | Shadow Cascades | Show directional light <u>shadow cascades</u> . |
| | Render Paths | Show the <u>rendering path</u> for each GameObject using a color code: Blue indicates <u>deferred shading</u> Yellow indicates <u>forward rendering</u> Red indicates <u>vertex lit</u> |
| | Alpha Channel | Render colors with alpha. |
| | Overdraw | Render GameObjects as transparent “silhouettes”. The transparent colors accumulate, making it easy to spot places where one object is drawn over another. |
| | Mipmaps | Show ideal texture sizes using a color code: Red indicates that the texture is larger than necessary (at the current distance and resolution) Blue indicates that the texture might be larger. The ideal texture sizes depend on the resolution at which your application will run and how close the Camera can get to particular surfaces. |
| | Texture Streaming | Tint GameObjects green, red, or blue, depending on their status in the <u>Texture Streaming</u> system. For more information, see documentation on <u>Texture Streaming debugging</u> . |
| | Sprite Mask | Sprite Masks are used to either hide or reveal parts of a Sprite or group of Sprites. See <u>Sprite Masks</u> for more information. |
| Deferred | | These modes let you view each of the elements of the G-buffer (Albedo , Specular , Smoothness and Normal) in isolation. See documentation on <u>Deferred Shading</u> for more information. |
| Global Illumination | | The following modes are available to help visualize aspects of the <u>Global Illumination</u> system: Systems , Clustering , Lit Clustering , UV Charts , and Contributors/Receivers . See documentation on <u>GI Visualisations</u> for information about each of these modes. |

| | |
|-------------------------------------|---|
| Realtime Global Illumination | The following modes are available to help visualize aspects of the <u>Enlighten Realtime Global Illumination</u> system: Albedo, Emissive, Indirect, and Directionality. See documentation on <u>GI Visualisations</u> for information about each of these modes. |
| Baked Global Illumination | The following modes are available to help visualize aspects of the Baked Global Illumination system: Baked Light Map, Directionality, Shadowmask , Albedo, Emissive, UV Charts, Texel Validity, UV Overlap, Baked Lightmap Culling, Lightmap Indices, and Light Overlap. See documentation on <u>GI Visualisations</u> for information about each of these modes. |
| Material Validator | There are two Material Validator modes: Albedo and Metal Specular. These allow you to check whether your physically-based materials use values within the recommended ranges. See <u>Physically Based Material Validator</u> for more information. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/ViewModes.html>

Τα επόμενα τρία εργαλεία στο set εργαλείων Scene view (View Options toolbar) είναι το 2D switch ,ο light switch και το audio switch.



Εικόνα 34. 2D, Lighting and Audio switches

- **2D** : Αλλάζει μεταξύ 2D και 3D προβολής για τη Σκηνή. Στη λειτουργία 2D η κάμερα είναι προσανατολισμένη κοιτάζοντας προς το θετικό z, με τον άξονα x να δείχνει προς τα δεξιά και τον άξονα y προς τα πάνω.
- **Φωτισμός** : Ενεργοποιεί ή απενεργοποιεί τον φωτισμό της προβολής σκηνής (φώτα, σκίαση αντικειμένων κ.λπ.).
- **Ήχος** : Ενεργοποιεί τα εφέ ήχου της προβολής σκηνής ενεργοποιούνται ή απενεργοποιούνται.

Το επόμενο εργαλείο στο set εργαλείων Scene view (View Options toolbar) είναι το Effects button and menu.

- **Skybox** : Μια υφή (texture) skybox που αποδίδεται στο φόντο της σκηνής
- **Clouds** (Σύννεφα): Εμφανίζει τα στρώματα σύννεφων και τα ογκομετρικά σύννεφα. Διαθέσιμο μόνο όταν χρησιμοποιείτε ένα SRP που υποστηρίζει σύννεφα.
- **Fog** : Σταδιακή εξασθένιση της όψης σε ένα επίπεδο χρώμα με την απόσταση από την κάμερα.

- **Flares** : Ακτίδα φωτός που διαθλάται μέσα στο φακό της κάμερας.
- **Always Refresh** : Ορίζει εάν τα κινούμενα υλικά εμφανίζουν την κίνηση. Όταν είναι επιλεγμένο, όλα τα εφέ που βασίζονται στο χρόνο (για παράδειγμα, Shaders) θα κινούνται. Για παράδειγμα, εφέ σκηνής (όπως το γρασίδι που κυματίζει σε ένα έδαφος).
- **Post Processing** : Εμφανίζει post-processing εφέ.
- **Particle Systems** : Εμφανίζει Particle System εφέ.

Το κουμπί εργαλείου Effects button and menu ενεργοποιεί και απενεργοποιεί όλα τα παραπάνω εφέ.

Το επόμενο τρία κουμπιά στο set εργαλείων είναι το Scene Visibility Switch, το Camera settings menu και το Gizmo menu.

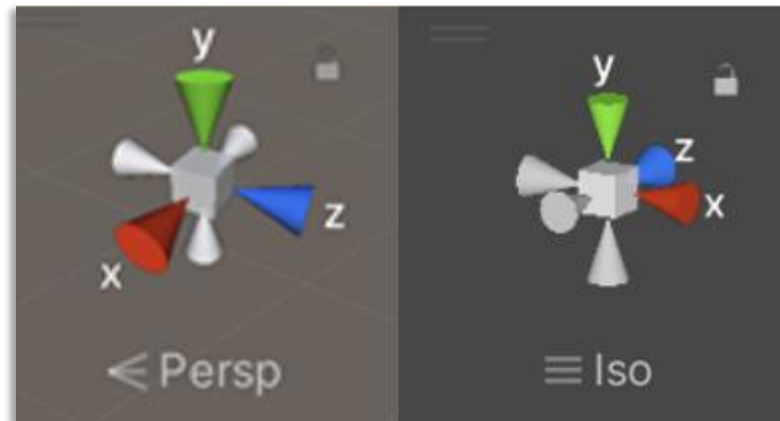


Εικόνα 35. Scene visibility switch , Camera settings menu and Gizmo menu tools

- Scene Visibility : Ο διακόπτης ορατότητας σκηνής ενεργοποιεί και απενεργοποιεί την ορατότητα σκηνής για αντικείμενα παιχνιδιού. Όταν είναι ενεργοποιημένος, η Unity εφαρμόζει τις ρυθμίσεις ορατότητας σκηνής. Όταν είναι απενεργοποιημένος, η Unity τις αγνοεί.
- Camera settings : Το μενού Ρυθμίσεις κάμερας περιέχει επιλογές για τη διαμόρφωση της κάμερας προβολής σκηνής.
- Gizmos Menu : Τα Gizmos περιέχουν επιλογές για τον τρόπο εμφάνισης των αντικειμένων, των εικονιδίων και των gizmos. Αυτό το μενού είναι διαθέσιμο τόσο στην προβολή σκηνής όσο και στην προβολή παιχνιδιού.

Τέλος όταν η Scene View είναι σε 3D απεικόνιση ο χρήστης βλέπει και το Scene Gizmo. Αυτό δείχνει τον τρέχοντα προσανατολισμό της κάμερας και επιτρέπει να αλλάξει τη γωνία προβολής και τη λειτουργία προβολής. Το Scene Gizmo διαθέτει έναν κωνικό βραχίονα σε κάθε πλευρά του κύβου. Οι βραχίονες στο μπροστινό μέρος φέρουν τις ονομασίες X, Y και Z. Ο χρήστης μπορεί να κάνει κλικ σε οποιονδήποτε από τους κωνικούς βραχίονες των αξόνων για να προσαρτήσει την Scene View Camera στον άξονα που αντιπροσωπεύει (για

παράδειγμα: προβολή από πάνω, προβολή από αριστερά και προβολή από μπροστά). Μπορεί επίσης να κάνει δεξί κλικ στον κύβο για να δει ένα μενού με μια λίστα γωνιών θέασης. Για να επιστρέψει στην προεπιλεγμένη γωνία προβολής, πρέπει να κάνει δεξί κλικ στο Gizmo σκηνής και να επιλέξει Free.




Εικόνα 36. Scene Gizmo Perspective and Isometric

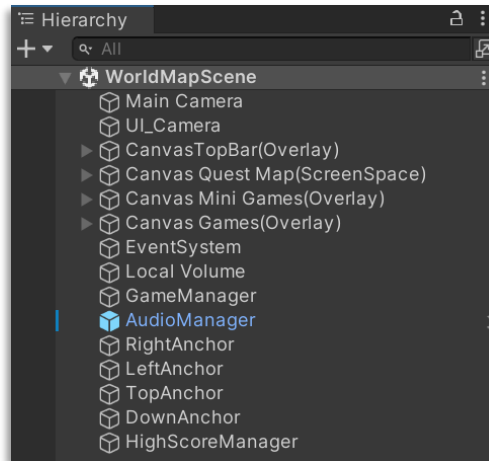
Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/SceneViewNavigation.html#gizmo>

Μπορεί επίσης να ενεργοποιήσει και να απενεργοποιήσει την προοπτική. Αυτό αλλάζει τη λειτουργία προβολής της προβολής Σκηνής μεταξύ προοπτικής και ορθογραφικής (που μερικές φορές ονομάζεται <<ισομετρική>>). Για να γίνει αυτό, αριστερό κλικ στον κύβο στο κέντρο του Scene Gizmo ή στο κείμενο κάτω από αυτό. Η ορθογραφική προβολή δεν έχει προοπτική και είναι χρήσιμη σε συνδυασμό με το κλικ σε έναν από τους βραχίονες του κωνικού άξονα για να έχει μια μπροστινή, πάνω, κάτω ή πλαϊνή όψη.

4.7.2 Hierarchy window

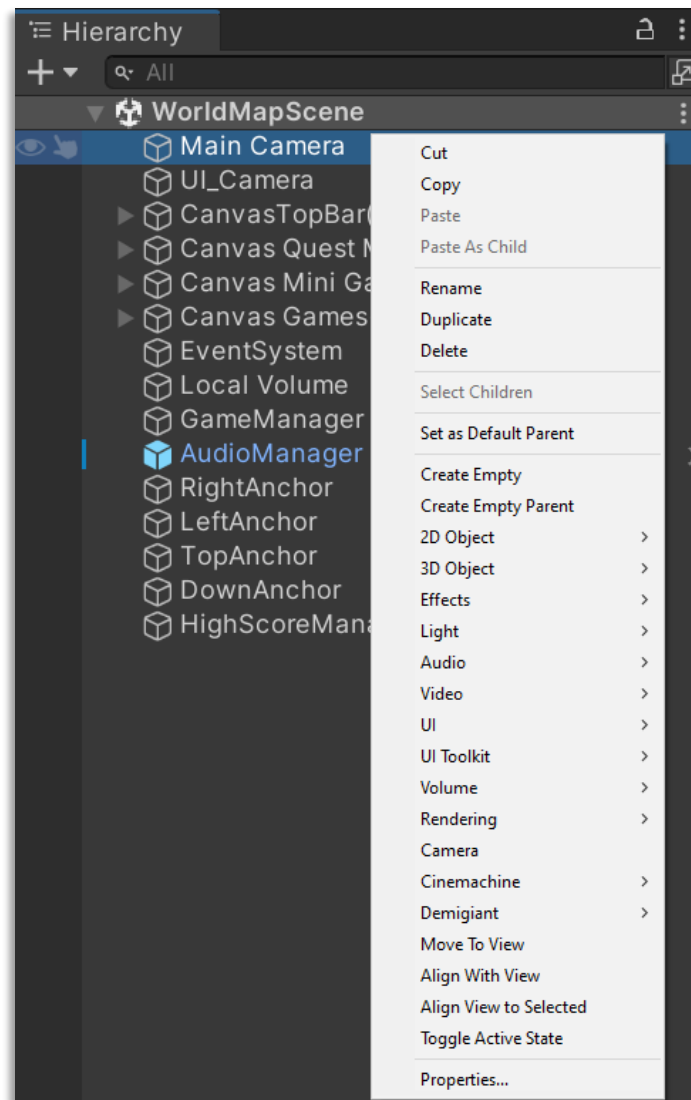
Ακολουθεί το **Hierarchy window** που είναι μια ιεραρχική αναπαράσταση κειμένου από κάθε GameObject στη σκηνή . Κάθε αντικείμενο στη Σκηνή έχει μια εγγραφή στην ιεραρχία, οπότε τα δύο παράθυρα είναι εγγενώς συνδεδεμένα. Η ιεραρχία αποκαλύπτει τη δομή του τρόπου με τον οποίο τα GameObjects συνδέονται μεταξύ τους.

| | |
|---|---|
|  | <p>Μια καλή πρακτική είναι όταν υπάρχουν πάρα πολλά GameObjects στο παράθυρο ιεραρχίας ο χρήστης να δημιουργεί κάποια κενά GameObjects, για να χρησιμοποιηθούν ως τίτλοι που θα χωρίζουν τα διάφορα τμήματα των GameObject σε ομάδες για πιο εύκολη διαχείριση.</p> |
|---|---|



Εικόνα 37. The Hierarchy window

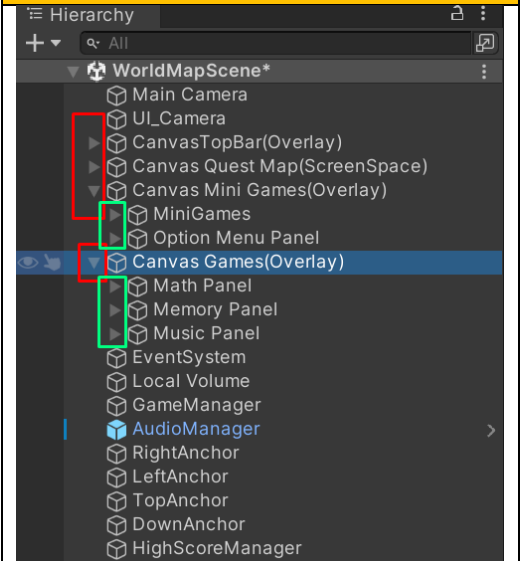
Με δεξί κλικ του ποντικιού στο παράθυρο της ιεραρχίας ή με τον σταυρό πάνω αριστερά κάτω από το όνομα της ιεραρχίας η μηχανή δίνει την δυνατότητα στο προγραμματιστή να προσθέσει καινούργια GameObject (**Ctrl+Shift+N**).



Εικόνα 38. Right Mouse Button Hierarchy GameObject Choices

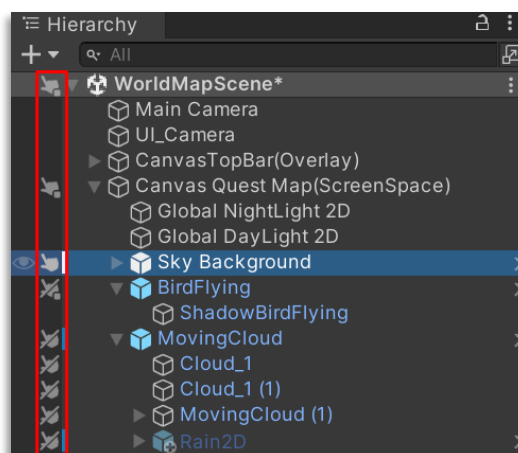
Η μηχανή Unity στο παράθυρο ιεραρχίας χρησιμοποιεί parent child δομή οπότε ο χρήστης μπορεί να δημιουργήσει ένα GameObject που είναι παιδί ενός άλλου. Αυτό μπορεί να επιτευχθεί με δυο τρόπους . Με δεξί κουμπί στην επιφάνεια του παραθύρου της ιεραρχίας πάνω στο GameObject που θέλει να αποκτήσει child GameObject και επιλογή δημιουργίας καινούργιου GameObject. Ο δεύτερος τρόπος είναι ο χρήστης να σύρει το GameObject που θέλει και να το βάλει κάτω από το parent GameObject. Αν θέλει ο χρήστης μπορεί να κρύψει τα child GameObject με τα βελόνια που υπάρχουν αριστερά του ονόματος του εκάστοτε GameObject τα οποία περιέχουν παιδιά.

Πίνακας 24. Child GameObject Visibility

| The Hierarchy window | Toggle child GameObject visibility |
|---|--|
|  | <ul style="list-style-type: none"> ❖ Select the drop-down arrow (▶) to the left of the parent GameObject. ❖ Press Alt while clicking the drop-down arrow (▶) to toggle visibility of all descendant GameObjects of the root GameObject. ❖ Select the drop-down arrow again (▼) to collapse all descendant GameObjects. |





Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/Hierarchy.html>

Άλλη μια δυνατότητα που μας παρέχει το παράθυρο της ιεραρχίας είναι να περιορίσει ο χρήστης τα αντικείμενα που μπορεί να επιλέξει στην σκηνή χωρίς να τα απενεργοποιήσει και αυτό γίνεται όταν υπάρχουν πάρα πολλά αντικείμενα στην σκηνή και είναι δύσκολη η επιλογή μέσα από αυτά. Η μηχανή Unity επιλέγει αντικείμενα με δυνατότητα επιλογής μόνο όταν ο χρήστης σχεδιάζει ένα πλαίσιο οριοθέτησης με το ποντίκι στην προβολή σκηνής δηλαδή κάνει αριστερό κλικ και τραβάει το ποντίκι για να επιλέξει μια ομάδα αντικειμένων.



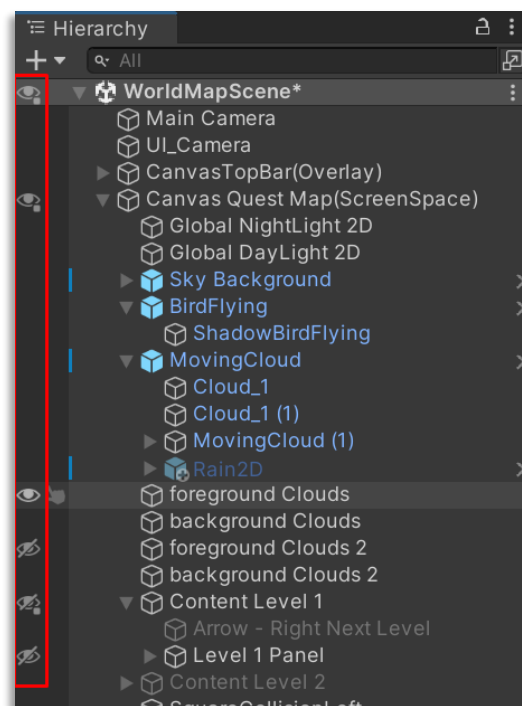
Εικόνα 39. Toggle Scene pickability

Πίνακας 25. Toggle pickability icons and their meaning

| Icon | Description |
|---|---|
|  | Ο χρήστης μπορεί να επιλέξει το GameObject, αλλά δεν μπορεί να επιλέξει κάποια από τα παιδιά του. |
|  | Ο χρήστης δεν μπορεί να επιλέξει το GameObject, αλλά μπορεί να επιλέξει κάποια από τα παιδιά του. |
|  | Ο χρήστης δεν μπορεί να επιλέξει το GameObject ούτε τα παιδιά του. |
|  | Ο χρήστης μπορεί να επιλέξει το GameObject και τα παιδιά του. Αυτό το εικονίδιο εμφανίζεται μόνο όταν περνάει με το ποντίκι πάνω από το GameObject. |





Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/ScenePicking.html>

Η επιλογή ορατότητας στα στοιχεία ελέγχου του παραθύρου ιεραρχίας επιτρέπουν στον προγραμματιστή να κρύβει και να εμφανίζει γρήγορα αντικείμενα παιχνιδιού στην προβολή σκηνής χωρίς να αλλάξει την ορατότητά τους στο παιχνίδι. Αυτό είναι χρήσιμο για την εργασία με μεγάλες ή πολύπλοκες Σκηνές όπου μπορεί να είναι δύσκολο να προβάλλει και να επιλέξει συγκεκριμένα αντικείμενα παιχνιδιού.




Εικόνα 40. Toggle Scene Visibility

Πίνακας 26. Toggle visibility icons and their meaning

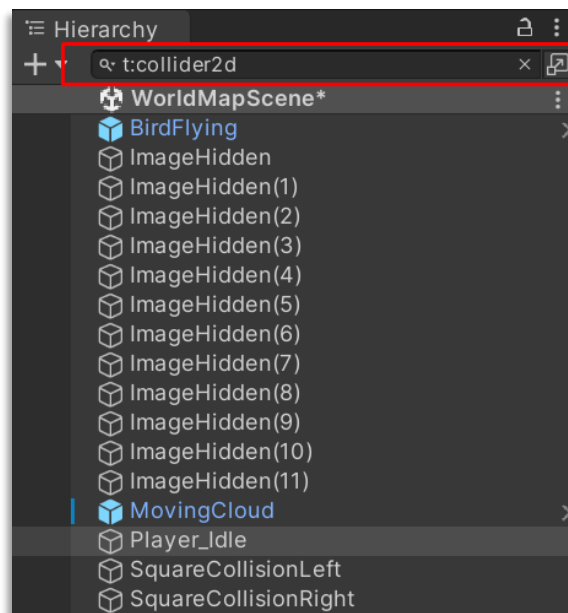
| Icon | Description |
|---|---|
|  | Το GameObject είναι ορατό, αλλά μερικά από τα παιδιά του είναι κρυμμένα. |
|  | Το GameObject είναι κρυμμένο, αλλά μερικά από τα παιδιά του είναι ορατά. |
|  | Το αντικείμενο παιχνιδιού και τα παιδιά του είναι ορατά. Αυτό το εικονίδιο εμφανίζεται μόνο όταν ο χρήστης περνάει το ποντίκι πάνω από το GameObject. |
|  | Το GameObject και τα παιδιά του είναι κρυμμένα. |

Πηγή : <https://docs.unity3d.com/2019.1/Documentation/Manual/SceneVisibility.html>



Εδώ πρέπει να αναφερθεί πως η ενεργοποίηση και απενεργοποίηση του visibility και pickability switch δεν <<λερώνει>> (dirty) τα GameObject και δεν χρειάζεται η μηχανή να τα ξανακάνει render σε αντίθεση με στοιχεία στο UI όπου με την παραμικρή αλλαγή καλείται η μηχανή για να ξανασχεδιάσει όλο τον Canvas με ότι αυτό συνεπάγεται.

Άλλη μια χρήσιμη ιδιότητα που δίνει στον προγραμματιστή το παράθυρο ιεραρχίας είναι η μηχανή αναζήτησης για GameObject. Ένα παράδειγμα εντολής για την αναζήτηση 2D Colliders στην σκηνή.



Εικόνα 41. Search Engine in Hierarchy

Πίνακας 27. Regular searches

| Provider | Function | Search token | Example |
|------------------|---|----------------------|--|
| Project | Searches Project Assets. | p: (for "project") | p:Player Searches for Assets that match the term "Player". |
| Hierarchy | Searches GameObjects in the Scene. | h: (for "hierarchy") | h:Main Camera Searches the current Scene for GameObjects that match the term "Main Camera". |
| Settings | Searches all <u>Project Settings</u> and <u>Preferences</u> . | set: | set:VFX Finds Project Settings and Preferences pages that match the term "VFX". |
| Menus | Searches the Unity main menu. | m: | m:TextMesh Pro Searches the Unity main menu for commands that contain "TextMesh Pro." |

Πηγή : <https://docs.unity.cn/2021.1/Documentation/Manual/regular-searches.html>

Πίνακας 28. Special searches

| Provider | Function | Search token | Example |
|----------------------|---|--------------|---|
| Help | Searches the Quick Search help. | ? | ?asset Searches for Search help entries containing the word "Asset". |
| Calculator | Computes mathematical expressions. | = | =2*3+29/2 Calculates the answer to the expression 2*3+29/2. |
| Files | Searches for files | find: | find:Paint Mat Searches for all assets paths containing the words paint AND the word mat (for example: PaintBrush_Mat.mat, DryWallPainted_Mat.mat) |
| Static API | Finds and executes static API methods. | # | #Mesh Searches for static API methods with "Mesh" in their names. |
| Packages | Searches the Unity package database. | pkg: | pkg:vector Searches the Unity package database for packages that match the term "vector". |
| Asset Store | Searches the <u>Unity Asset Store</u> . | store: | store:texture Searches the Unity Asset Store for Assets that match the term "texture". |
| Saved Queries | Searches for saved queries | q: | q:enemies Searches all saved searches containing the word "enemies". |
| Logs | Searches the Editor.log file | log: | log:cache Searches the Editor.log file for information that matches "cache". |

Πηγή : <https://docs.unity.cn/2021.1/Documentation/Manual/special-searches.html>

Επιπλέον υπάρχουν και οι εξής search filters αλλά θα αναφερθούν μόνο ονομαστικά :

- Prefab filters
- File filters
- Type filters
- Dependency filters

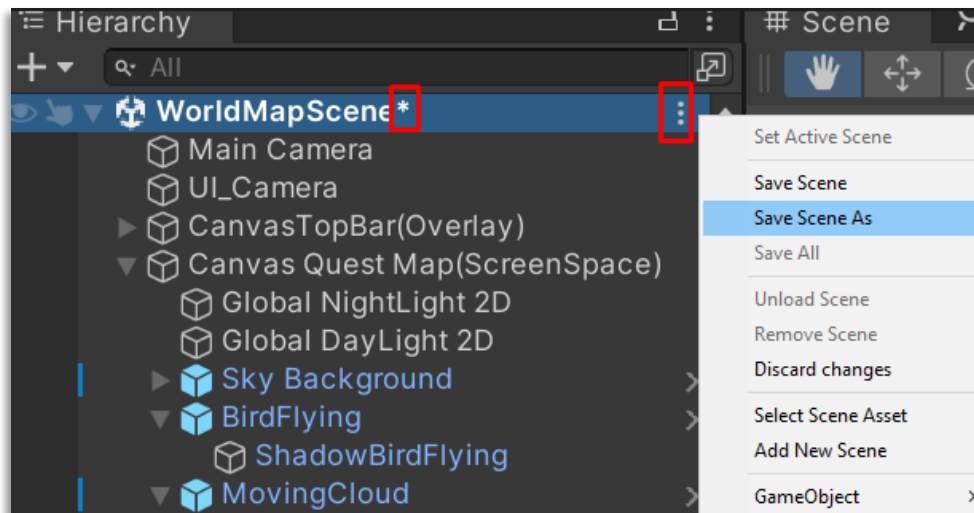
Μια πιο αναλυτική επισκόπηση υπάρχει στην ηλεκτρονική σελίδα της Unity και είναι [Additional Search filters](#). Εδώ να αναφερθεί πως υπάρχει δυνατότητα για συνδυασμό εντολών στα φίλτρα της μηχανής αναζήτησης .

Πίνακας 29. Combining search tokens

| Query | Description |
|---|--|
| h: t:meshrendererer p(castshadows)!="Off" | Searches all static meshes in a Scene that cast a shadow. |
| h: t:light p(color)=#FFFFFF p(intensity)>7.4 | Searches all lights in a Scene with a specific color with brightness higher than 7.4. |
| o: t:healthui ref:healthcanvas | Use the Object Provider to search all indexed Prefabs and Scenes for GameObjects with a HealthUI component that references the healthcanvas Prefab. |
| h: path:/Collectables t:collectable | Find all objects with a component Collectable located in the path /Collectables. |

Πηγή : <https://docs.unity.cn/2021.1/Documentation/Manual/search-filters.html#search-tokens>

Το παράθυρο ιεραρχίας παρέχει ακόμα κάποιες δυνατότητες όπως να ενημερώνει τον χρήστη πως οι αλλαγές που έχει κάνει δεν έχουν αποθηκευτεί προβάλλοντας ένα αστερίσκο δίπλα από το όνομα της ενεργής σκηνής που υπάρχει στο παράθυρο της ιεραρχίας (save scene Ctrl+S). Επίσης ο χρήστης μπορεί να φορτώσει και μια δεύτερη σκηνή, να πάρει τα assets που τον ενδιαφέρουν και να τα βάλει από την μια στην άλλη .

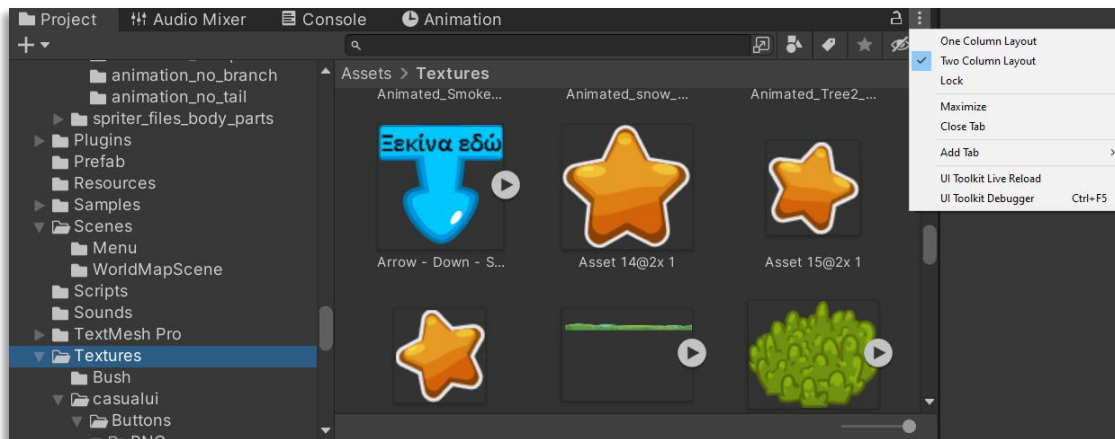


Εικόνα 42. Unsaved scene indication

4.7.3 Project window

Το **Project window** περιλαμβάνει όλα τα αρχεία που σχετίζονται με το Project και αποτελεί τον κύριο τρόπο πλοήγησης και εντοπισμού ψηφιακών στοιχείων και άλλων αρχείων στην εφαρμογή. Κατά την έναρξη ενός νέου Έργου, από προεπιλογή αυτό το παράθυρο είναι ανοιχτό. Ωστόσο, εάν δεν είναι προσβάσιμο ή είναι κλειστό, μπορεί να ανοίξει μέσω της διαδρομής Window > General > Project ή πατώντας τα πλήκτρα Ctrl+5 (macOS: Cmd+5). Το Project window μπορεί να μετακινηθεί κάνοντας κλικ με το ποντίκι στο πάνω μέρος του και κατόπιν να συρθεί στο σημείο που θέλει ο χρήστης. Μπορεί να αγκιστρωθεί είτε στη θέση του στον Editor είτε να συρθεί έξω από το παράθυρο του Editor για να χρησιμοποιηθεί ως αιωρούμενο παράθυρο.

Επιπλέον, η διάταξη του ίδιου του παραθύρου μπορεί να αλλαχθεί. Αυτό μπορεί να επιτευχθεί επιλέγοντας το μενού More menu (:) στην επάνω δεξιά γωνία του παραθύρου και επιλέγοντας είτε τη διάταξη μίας στήλης είτε τη διάταξη δύο στηλών. Η διάταξη δύο στηλών παρουσιάζει ένα επιπλέον παράθυρο που παρέχει μια οπτική προεπισκόπηση για κάθε αρχείο.



Εικόνα 43. Project Window options

Στο αριστερό τμήμα του παραθύρου περιήγησης, εμφανίζεται η δομή των φακέλων του Project ως ιεραρχική λίστα. Όταν ένας φάκελος επιλέγεται από τη λίστα, τα περιεχόμενά του εμφανίζονται στο δεξί πλαίσιο. Το τρίγωνο μπορεί να πατηθεί για να επεκταθεί ή να συμπτυχθεί ο φάκελος, αποκαλύπτοντας τυχόν εμφωλευμένους φακέλους που περιέχει. Κρατώντας πατημένο το Alt κατά το αριστερό κλικ του ποντικιού, οι εμφωλευμένοι φάκελοι μπορούν να επεκταθούν ή να συμπτυχθούν αναδρομικά. Κάθε ψηφιακό στοιχείο εμφανίζεται στο δεξί πλαίσιο ως εικονίδιο που υποδεικνύει τον τύπο του (π.χ. script, material, sub-folder). Το μέγεθος των εικονιδίων μπορεί να αλλάξει χρησιμοποιώντας τον slider στο κάτω μέρος του πλαισίου με αυτό τον τρόπο τα στοιχεία θα αντικατασταθούν από μια ιεραρχική λίστα αν ο slider μετακινηθεί προς τα αριστερά. Ο χώρος στα αριστερά του slider εμφανίζει το τρέχον επιλεγμένο στοιχείο, συμπεριλαμβανομένης της πλήρους διαδρομής προς το στοιχείο, εάν γίνεται αναζήτηση. Στο αριστερό παράθυρο πάνω από την ιεραρχική λίστα του έργου υπάρχει μια ενότητα "Αγαπημένα", όπου τα συχνά χρησιμοποιούμενα στοιχεία μπορούν να διατηρούνται για εύκολη πρόσβαση. Τα στοιχεία μπορούν να σύρονται από τη λίστα έργου στα "Αγαπημένα" και επίσης ο χρήστης μπορεί να αποθηκεύσει ερωτήματα αναζήτησης εκεί. Τέλος στην πάνω δεξιά γωνιά υπάρχουν κάποια εργαλεία που παρέχουν έξυπνη αναζήτηση των asset και είναι τα εξής :



Εικόνα 44. Project window Toolbar

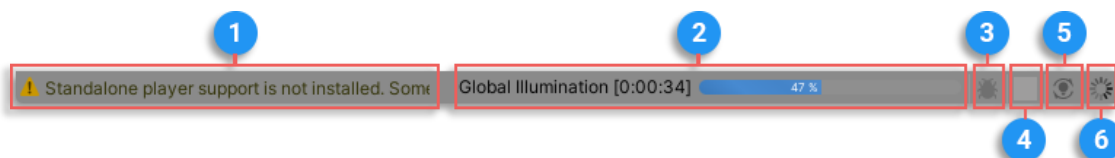
Πίνακας 30. Project window Toolbar

| Property | Description |
|-----------------------|--|
| Create menu | Ο χρήστης εμφανίζει μια λίστα με τα ψηφιακά στοιχεία και άλλους υποφακέλους που μπορεί να προσθέσει στο φάκελο που είναι επιλεγμένος αυτή τη στιγμή. |
| Search bar | Ο χρήστης χρησιμοποιεί τη γραμμή αναζήτησης για να αναζητήσει ένα αρχείο μέσα στο Έργο. Μπορεί να επιλέξει να κάνει αναζήτηση σε ολόκληρο το έργο (Όλα), στους φακέλους του κορυφαίου επιπέδου του έργου (που παρατίθενται μεμονωμένα), στο φάκελο που έχει επιλέξει αυτή τη στιγμή ή στο Asset Store. |
| Open in Search | Ο χρήστης ανοίγει το εργαλείο <u>Unity Search</u> tool για να βελτιώσει την αναζήτησή του. Ctrl+K |
| Search by Type | Ο Χρήστης επιλέγει αυτή την ιδιότητα για να περιορίσει την αναζήτησή του σε έναν συγκεκριμένο τύπο, για παράδειγμα Mesh, Prefab, Scene. |
| Search by Label | Ο χρήστης επιλέγει αυτή την ιδιότητα για να διαλέξει μια ετικέτα για αναζήτηση. |
| Save Search | Ο χρήστης αποθηκεύει την αναζήτησή του στα Αγαπημένα στον αριστερό πίνακα. |
| Hidden packages count | Ο χρήστης επιλέγει αυτή την ιδιότητα για να αλλάξει την ορατότητα των πακέτων στο παράθυρο Έργο. |

Πηγή : <https://docs.unity3d.com/Manual/ProjectView.html>

4.7.4 Status bar

Το **Status bar** παρέχει στον χρήστη κάποιες βασικές πληροφορίες για διάφορες διεργασίες της μηχανής Unity και γρήγορη πρόσβαση σε σχετικά εργαλεία και ρυθμίσεις. Η γραμμή κατάστασης εμφανίζει τα εξής:



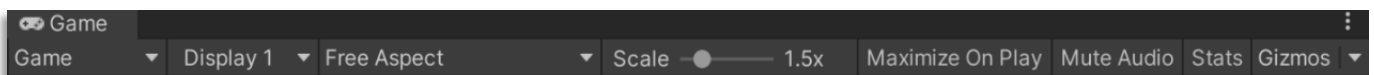
Εικόνα 45. The Status Bar

1. The most recent message logged to the Console window. Click the message to open the Console window.
2. A global progress bar for various asynchronous tasks (for example, shader compilation, lightmap baking, and occlusion culling). Click the progress bar to open the Background Tasks window, which displays progress for individual tasks and subtasks.
3. The current code optimization mode. Click the code optimization icon to switch between debug mode and release mode.

4. The cache server status. Click the icon to get additional information about the cache server, and re-establish a lost connection.
5. The automatic lighting generation status for Global illumination. Click the status text to open the Lighting window. When you generate lighting manually, or Unity generates it automatically, a progress bar appears next to the message.
6. An activity indicator (spinner) that shows when Unity compiles C# **scripts** or runs asynchronous tasks.

4.7.5 Game view

Το **Game view** απεικονίζεται από την κάμερα(ες) της εφαρμογής. Αντιπροσωπεύει την τελική, δημοσιευμένη εφαρμογή. Ο προγραμματιστής πρέπει να χρησιμοποιήσει μία ή περισσότερες κάμερες για να ελέγχει τι βλέπει ο παίκτης όταν χρησιμοποιεί την εφαρμογή.



Εικόνα 46. Game View Control Bar

Οι λειτουργίες αναπαραγωγής του επεξεργαστή και η προβολή της αναπαραγόμενης εφαρμογής μπορούν να ελεγχθούν μέσω των κουμπιών στη γραμμή εργαλείων. Κατά τη λειτουργία αναπαραγωγής, οποιεσδήποτε αλλαγές καταγράφονται προσωρινά και επανέρχονται στην αρχική κατάσταση όταν απενεργοποιηθεί η λειτουργία αναπαραγωγής. Το γραφικό περιβάλλον του Editor αλλάζει χρώμα για να υποδείξει αυτό το γεγονός.

Πίνακας 31. Game View Control Bar Function

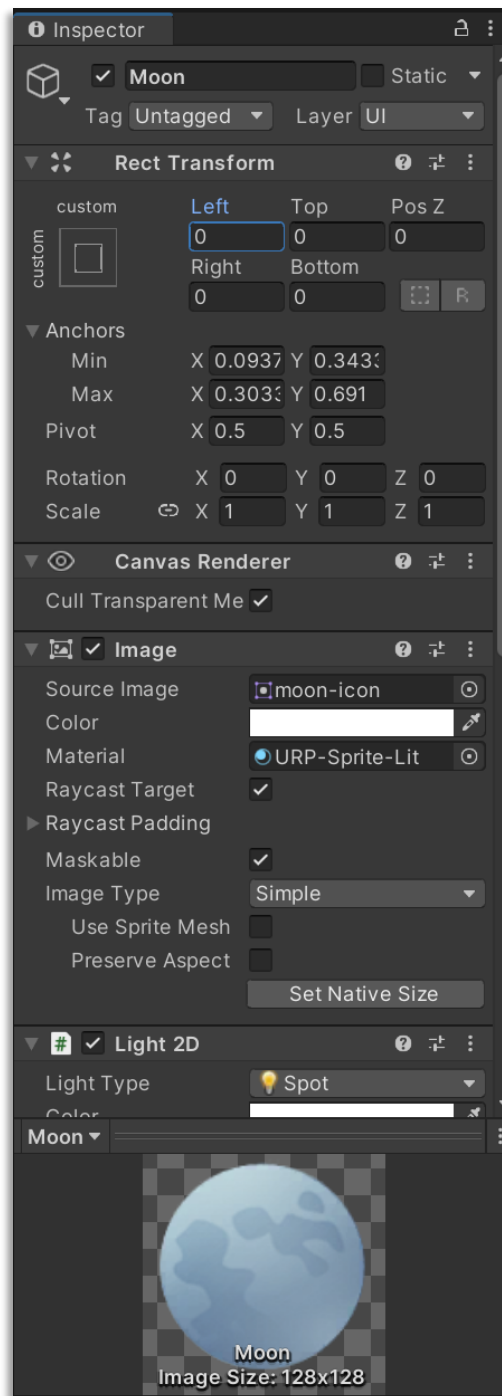
| Button | Function |
|------------------------------|--|
| Game/Simulator | Click to enable the Game or Simulator view from the drop-down menu. |
| Display | Select this to choose from a list of Cameras if you have multiple Cameras in the Scene . This is set to Display 1 by default. You can assign Displays to Cameras in the Camera module, under the Target Display drop-down menu. |
| Aspect | Select different values to test how your game looks on monitors with different aspect ratios . This is set to Free Aspect by default. |
| Low Resolution Aspect Ratios | Enable Low Resolution Aspect Ratios to emulate the pixel density of older displays: this reduces the resolution of the Game view when an aspect ratio is selected. It is always enabled when the Game view is on a non-Retina display. |

| | |
|-------------------------------|---|
| VSync (Game view only) | Enable VSync (Game view only) to give priority to the Game view. This option might add some vertical sync, which is useful when recording a video, for example. Unity attempts to render the Game view at the monitor refresh rate, though this is not guaranteed. When this option is enabled, it is still useful to maximize the Game view in Play mode to hide other views and reduce the number of views that Unity renders. |
| Scale slider | Scroll right to zoom in and examine areas of the Game screen in more detail. This slider lets you zoom out to see the entire screen where the device resolution is higher than the Game view window size. You can also use the scroll wheel and middle mouse button to do this while the application is stopped or paused. |
| Maximize on Play | Click to enable: use this to maximize the Game view (100% of your Editor window) for a full-screen preview when you enter Play mode. |
| Mute audio | Click to enable: use this to mute any in-application audio when you enter Play mode. |
| Stats | Click this to toggle the Statistics overlay, which contains <u>Rendering Statistics</u> about your application's audio and graphics. This is very useful for monitoring the performance of your application while in Play mode. |
| Gizmos | Click this to toggle the visibility of <u>Gizmos</u> . To only see certain types of Gizmo during Play mode, click the drop-down arrow next to the word Gizmos and only enable the Gizmo types you want to see. |

Πηγή : <https://docs.unity3d.com/Manual/GameView.html>

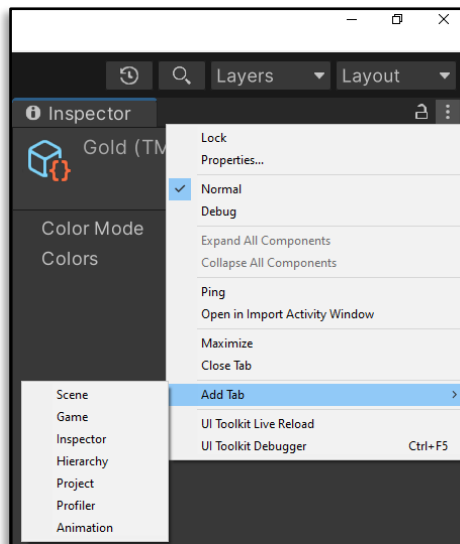
4.7.6 Inspector Window

Τέλος έχουμε το παράθυρο του **Inspector** το οποίο είναι υπεύθυνο να παρουσιάζει και να επεξεργάζεται ιδιότητες και ρυθμίσεις που υπάρχουν στον Unity Editor όπως πχ τα components μέσα σε κάθε GameObject στο παράθυρο της ιεραρχίας και στο παράθυρο Project, επίσης τα ψηφιακά στοιχεία, τα materials, τους shaders, τα scripts και ρυθμίσεις και προτιμήσεις εντός του Unity Editor.




Εικόνα 47. Inspector window layout

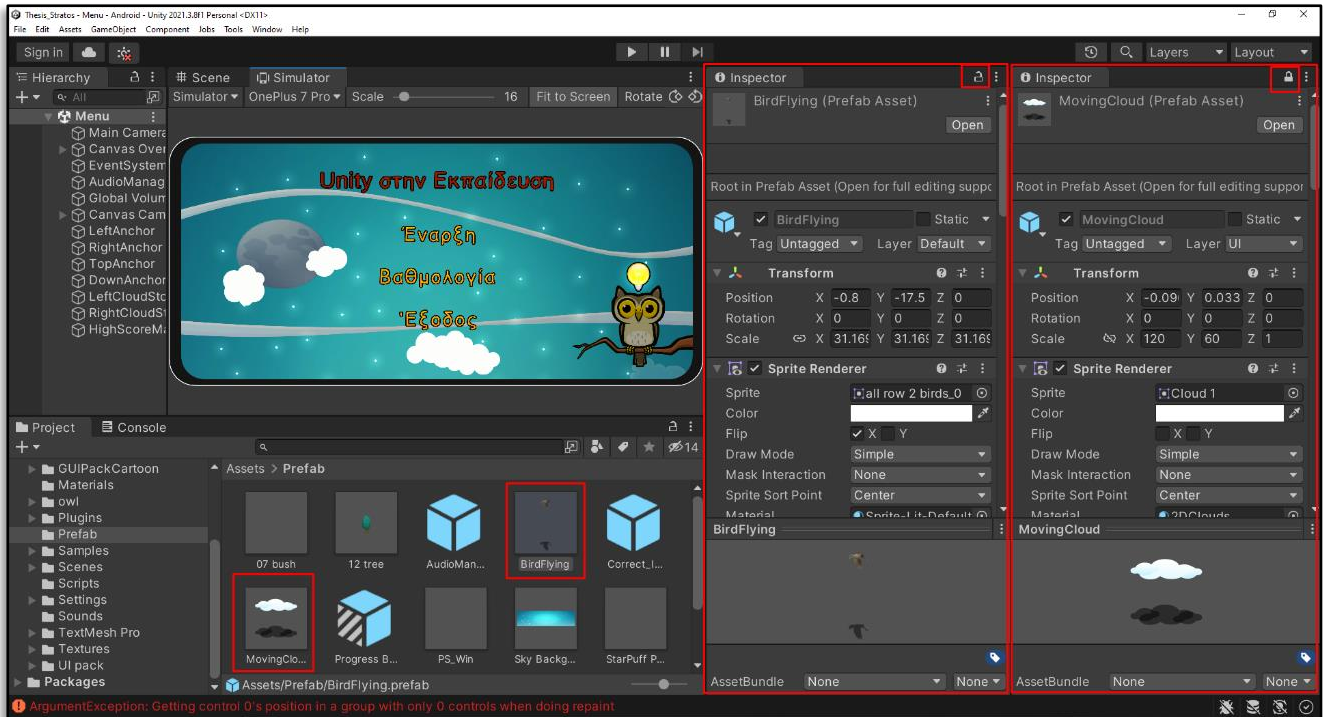
Εάν για κάποιο λόγο το εικονίδιο του παραθύρου Inspector ή και κάποιο από τα υπόλοιπα τότε ο χρήστης έχει την δυνατότητα να το ανοίξει πηγαίνοντας στο μενού Windows > General > Inspector για να ανοίξει το εικονίδιο του Inspector. Αν υπάρχει είδη ανοικτό το παράθυρο του Inspector αυτή η διαδικασία θα αγνοήσει την επιλογή του χρήστη.



Εικόνα 48. Adding Extra Inspector Window

Αν ο χρήστης θέλει και δεύτερο παράθυρο Inspector ανοιχτό τότε πρέπει να πάει από οποιοδήποτε παράθυρο του Editor στην επιλογή More Items menu (δεξιά πάνω γωνία σε οποιοδήποτε παράθυρο) και να διαλέξει τις 3 τελείες (:) κατόπιν να κάνει Add Tab > Inspector για να ανοίξει τον Inspector σε εκείνο το παράθυρο.

| | |
|---|--|
|  | <p>Υπάρχουν περιπτώσεις που θέλει ο χρήστης να συγκρίνει δυο GameObjects μεταξύ τους. Αυτό για να γίνει πρέπει να ανοίξει τον Inspector με τον δεύτερο τρόπο και στην συνέχεια να κλειδώσει τον ένα από τους δυο Inspector στο πρώτο GameObject επιλέγοντας την κλειδαριά δίπλα από τις τρεις τελείες και κατόπιν να επιλέξει το δεύτερο GameObject. Η να κάνει κλικ στο GameObject που θέλει και κατόπιν να επιλέξει <<Properties...>> αυτό είναι το λεγόμενο focused Inspector. Με επιλεγμένο GameObject και συνδυασμό πλήκτρων Alt+P.</p> |
|---|--|



Εικόνα 49. Comparing GameObjects with 2 Inspectors

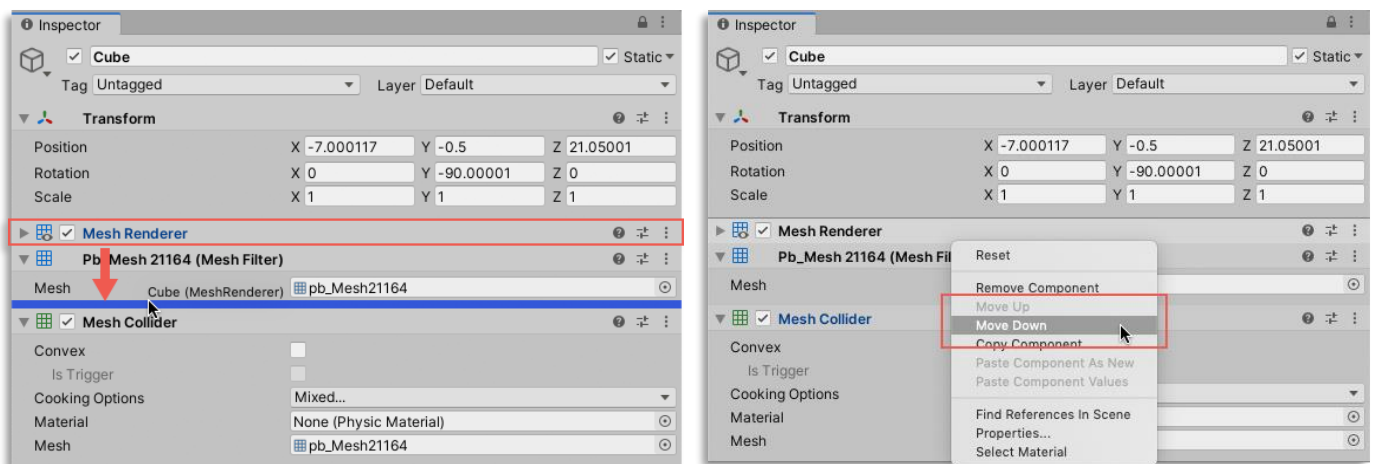
Ο **Inspector** παρουσιάζει τις πληροφορίες ανάλογα με το επιλεγμένο αντικείμενο στον Editor. Αυτές οι πληροφορίες είναι :

- ❖ Για GameObjects (παραδείγματος χάριν, στην προβολή Ιεραρχίας ή Σκηνής), οι ιδιότητες όλων των συστατικών και των Υλικών του, εμφανίζονται από τον Inspector. Οι ιδιότητες μπορούν να επεξεργαστούν και τα συστατικά να αναδιαταχθούν στο παράθυρο Inspector.
- ❖ Για Custom Script Components εμφανίζονται στον Inspector οι δημόσιες μεταβλητές. Οι μεταβλητές αυτές μπορούν να επεξεργαστούν με τον ίδιο τρόπο που επεξεργάζεται ο χρήστης άλλες ιδιότητες, πράγμα που σημαίνει ότι μπορεί να ορίσει παραμέτρους και προεπιλεγμένες τιμές στα script χωρίς να τροποποιήσει τον κώδικα.
- ❖ Όταν ένα asset επιλέγεται (για παράδειγμα, από το παράθυρο Project), ο Inspector εμφανίζει ρυθμίσεις που ελέγχουν τον τρόπο με τον οποίο το Asset εισάγεται και χρησιμοποιείται από τη Unity κατά την εκτέλεση. Για κάθε τύπο Asset, υπάρχουν δικές του ρυθμίσεις. Παραδείγματα ρυθμίσεων εισαγωγής Asset που μπορούν να επεξεργαστούν στο παράθυρο Inspector περιλαμβάνουν:

- [Model Import Settings](#) window.
- [Audio Clip Import Settings](#) window.
- [Texture Import Settings](#) window.

Ο **Inspector** παρέχει στον χρήστη και κάποιες δυνατότητες για την προσαρμογή των components που περιέχει ώστε να είναι δυνατή η παραμετροποίησή τους ανάλογα με τις προσωπικές επιλογές του κάθε χρήστη. Μερικές από αυτές είναι :

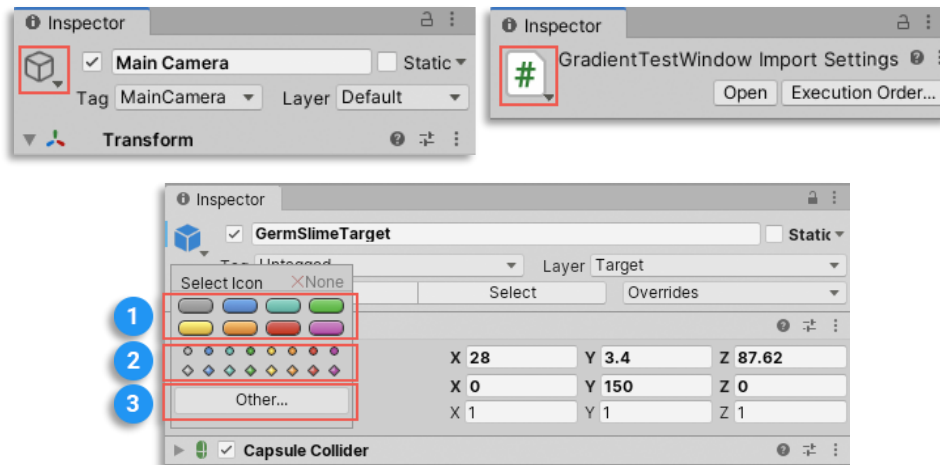
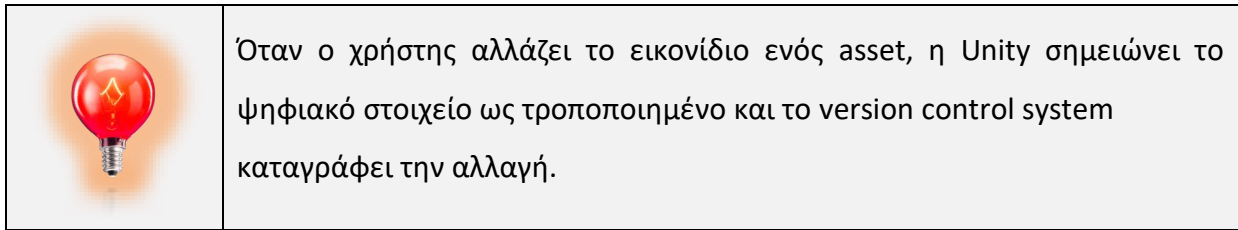
- η Αναδιάταξη των Component του GameObject. Η αναδιάταξη μπορεί να γίνει με 2 τρόπους, είτε με drag and drop του component που θέλει ο χρήστης στη σειρά που το θέλει, είτε με δεξί κλικ στον τίτλο του component και έπειτα επιλογή για μετακίνηση πάνω ή κάτω.



Εικόνα 50. Αναδιάταξη των Component του GameObject

Πηγή : <https://docs.unity3d.com/Manual/InspectorOptions.html>

- η Ανάθεση custom εικονιδίων σε GameObjects, scripts και Prefabs. Εδώ ο χρήστης μπορεί να αντιστοιχίσει custom εικονίδια σε αντικείμενα παιχνιδιού, Prefabs και scripts από το παράθυρο του Inspector. Αυτά τα εικονίδια θα εμφανίζονται στην σκηνή όπως εμφανίζονται και τα προεπιλεγμένα εικονίδια της κάμερας και των φώτων.



Εικόνα 51. Ανάθεση custom εικονιδίων σε GameObjects, scripts και Prefabs

Πηγή : <https://docs.unity3d.com/Manual/InspectorOptions.html>

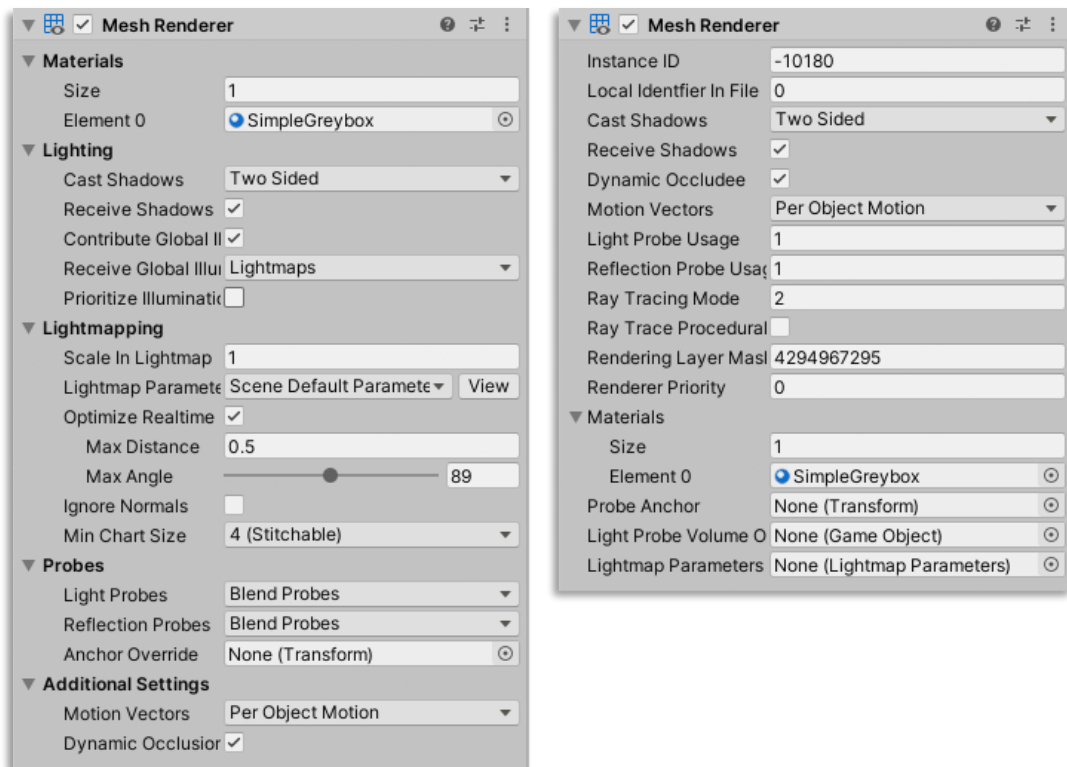
Πίνακας 32. Icon Type

| Icon | Description: |
|------|---|
| 1 | Label icons are colored capsules. In the Scene view, they display the names of the items they're assigned to. Click a label icon to assign it. |
| 2 | Image-only icons are colored divots. They're useful when you work with objects that may not have a visual representation. For example, you can assign image-only icons to navigation waypoints so you can see and select them in the Scene view. Click an image-only icon to assign it. |
| 3 | Asset icons are image Assets in your Scene that you use as custom icons. For example, you could use a skull and crossbones icon to indicate danger areas in a game level. Click the Other button to open an Object Picker window and select an image Asset to assign as an icon. |

Πηγή : <https://docs.unity3d.com/Manual/InspectorOptions.html>

- η ενεργοποίηση του Debug Mode. Είναι χρήσιμο κάποιες φορές ο χρήστης να βλέπει στον Editor μόνο τις ιδιότητες και τις τιμές στο παράθυρο του Inspector. Όταν ενεργοποιείται η επιλογή εντοπισμού σφάλματος αν το component έχει script τότε δείχνει και τις private values, παρόλα αυτά δεν έχει την δυνατότητα να τις αλλάξει.

- ❖ Για να ενεργοποιηθεί το Debug Mode ο χρήστης κάνει κλικ στο More Items (:) για να ανοίξει η επιλογή και να κάνει Debug.
- ❖ Για να γυρίσει πίσω στο Normal Mode κάνει κλικ στο More Items (:) και διαλέγει Normal.



Εικόνα 52. Toggle Debug Mode

4.7.7 Συντομεύσεις Πληκτρολογίου

Εδώ θα αναφερθούν τα πιο σημαντικά keyboard shortcuts του Unity Editor, του Visual Studio και κάποιον εντολών κατά την εγγραφή του κώδικα για να είναι πιο ευανάγνωστος στον Inspector. Οι πιο βασικές συντομεύσεις του Unity Editor είναι :

Πίνακας 33. Unity Popular Keyboard Shortcuts

| Shortcut Type | Shortcut Combination |
|--------------------|---|
| General Shortcuts: | <ul style="list-style-type: none"> • Ctrl + S: Save changes. • Ctrl + Z: Undo. • Ctrl + Y: Redo. • Ctrl + D: Duplicate selected object(s). • Ctrl + X: Cut selected object(s). • Ctrl + C: Copy selected object(s). • Ctrl + V: Paste copied/cut object(s). • Ctrl + A: Select all objects. |

| | |
|----------------------------------|--|
| Scene View Shortcuts: | <ul style="list-style-type: none"> • W: Move tool. • E: Rotate tool. • R: Scale tool. • Q: Pan tool. • F: Frame selected object(s). • Shift + F: Frame all objects in the scene. • Alt + Right-click: Orbit around pivot point. |
| Game View Shortcuts: | <ul style="list-style-type: none"> • Alt + P: Play or pause the game. • Ctrl + P: Pause the game. • Shift + P: Step the game forward one frame while paused. |
| Hierarchy View Shortcuts: | <ul style="list-style-type: none"> • Ctrl + Shift + N: Create a new empty GameObject. • Ctrl + G: Group selected objects into a new parent GameObject. • Del: Delete selected object(s). |
| Inspector View Shortcuts: | <ul style="list-style-type: none"> • Tab: Switch between Inspector and Scene/Game view. • Spacebar: Toggle component foldout in the Inspector. • Shift + Spacebar: Toggle all component foldouts in the Inspector. |
| Project View Shortcuts: | <ul style="list-style-type: none"> • Ctrl + Shift + N: Create a new folder. • F2: Rename Selected Object. • Ctrl + Shift + D: Duplicate selected object(s) in the Project view. • Del: Delete selected object(s) in the Project view. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/ShortcutsManager.html>

Πηγή : <https://docs.unity3d.com/2018.1/Documentation/Manual/UnityHotkeys.html>

Οι πιο βασικές συντομεύσεις για το IDE του Visual Studio είναι :

| Microsoft Visual Studio 2019 | | DEFAULT KEYBOARD SHORTCUTS |
|---|---|----------------------------|
| SEARCH AND NAVIGATION | | |
| Visual Studio search | Ctrl + Q | |
| Go to All | Ctrl + T or Ctrl + _ | |
| Go to Type / File / Member / Symbol | Ctrl + 1 + T / F / M / S | |
| Navigate Backward / Forward | Ctrl + - / Ctrl + ^ + - | |
| Go to Definition / Peek to Definition | F12 / Alt + F12 | |
| Go to Implementation | Ctrl + F12 | |
| Go to Next Error | Ctrl + ^ + F12 | |
| Go to Next / Previous Result in List | F8 / ^ + F8 | |
| EDITING AND REFACTORING | | |
| Quick Actions / Refactoring Suggestions | Alt + <←> or Ctrl + . | |
| Method Info | Ctrl + K, Ctrl + I | |
| Comment / Uncomment | Ctrl + K, Ctrl + C / Ctrl + K, Ctrl + U | |
| Delete Line (without copying it) | Ctrl + ^ + L | |
| Paste from keyboard buffer ring | Ctrl + ^ + V | |
| Move Code Up / Down | Alt + ↑ / ↓ | |
| Format Document / Selection | Ctrl + K, Ctrl + D / Ctrl + K, Ctrl + F | |
| Surround with (... if/try/foreach) | Ctrl + K, Ctrl + S | |
| Rename | Ctrl + R, Ctrl + R | |
| Encapsulate Field | Ctrl + R, Ctrl + E | |
| Remove and Sort Usings | Ctrl + R, Ctrl + G | |
| Extract Method | Ctrl + R, Ctrl + M | |
| DEBUGGING AND TESTING | | |
| Debug / Run / Stop | F5 / Ctrl + S / ^ + S | |
| Toggle Breakpoint | F9 | |
| Step Over | F10 | |
| Step Into | F11 | |
| Step Out | ^ + F11 | |
| Debug All Tests / Run All Tests | Ctrl + R, Ctrl + A / Ctrl + R, A | |
| WINDOW MANAGEMENT | | |
| Select Active File in Solution Explorer | Ctrl + [, S | |
| Open Tool Windows | | |
| Solution Explorer | Ctrl + Alt + L | |
| Output Window | Ctrl + Alt + O | |
| Error List | Ctrl + \ , E | |
| Team Explorer | Ctrl + \ , Ctrl + M | |
| Breakpoints | Ctrl + Alt + B | |
| Next / Previous Tool Window | Alt + F6 / ^ + Alt + F6 | |
| Close Current Tool Window | ^ + esc | |
| Go to Document to the Left / Right | Ctrl + Alt + Pg Up / Pg Dn | |
| Most Recent / Least Recent Open Document | Ctrl + tab / Ctrl + ^ + tab | |
| Keep Preview Window Open | Ctrl + Alt + ⌘ | |
| Full Screen(max window size/reduced menus) | ^ + Alt + <↵> | |
| Configure keyboard shortcuts: Tools → Options; Environment → Keyboard | | |
| Revised August 2021 | | |

Εικόνα 53. Popular keyboard shortcuts for visual studio

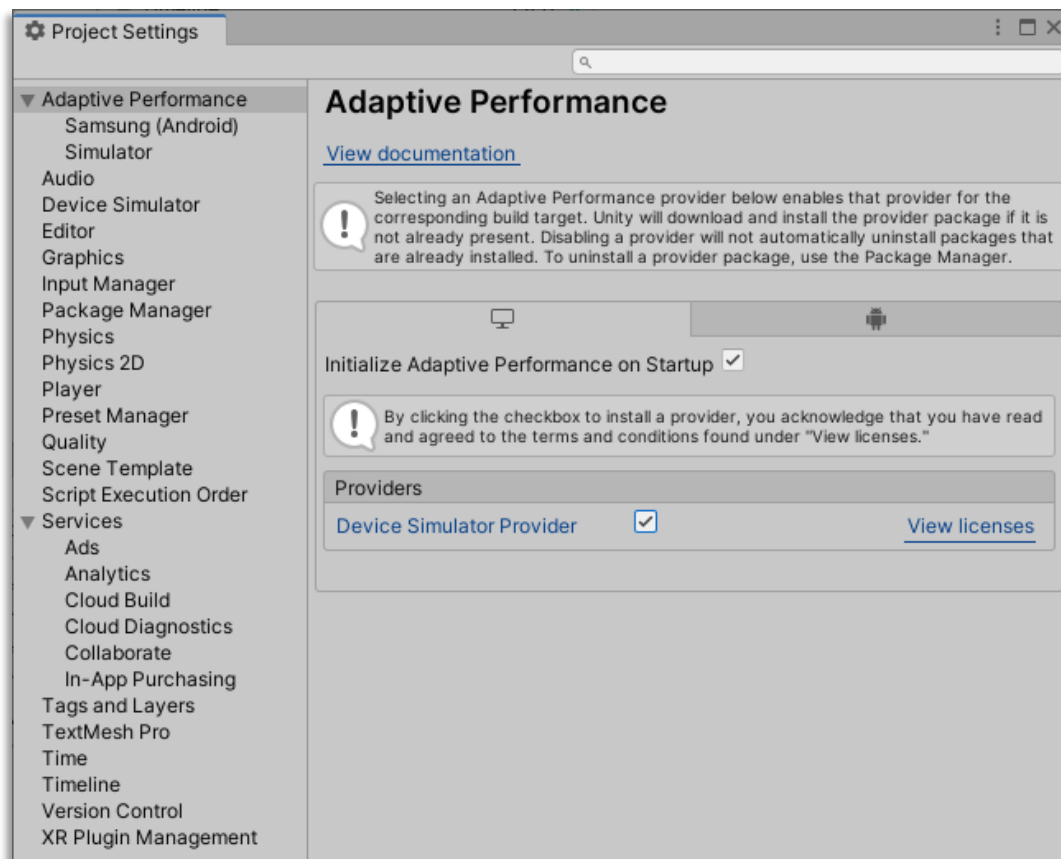
Πηγή : <https://learn.microsoft.com/en-us/visualstudio/ide/default-keyboard-shortcuts-in-visual-studio?view=vs-2022>

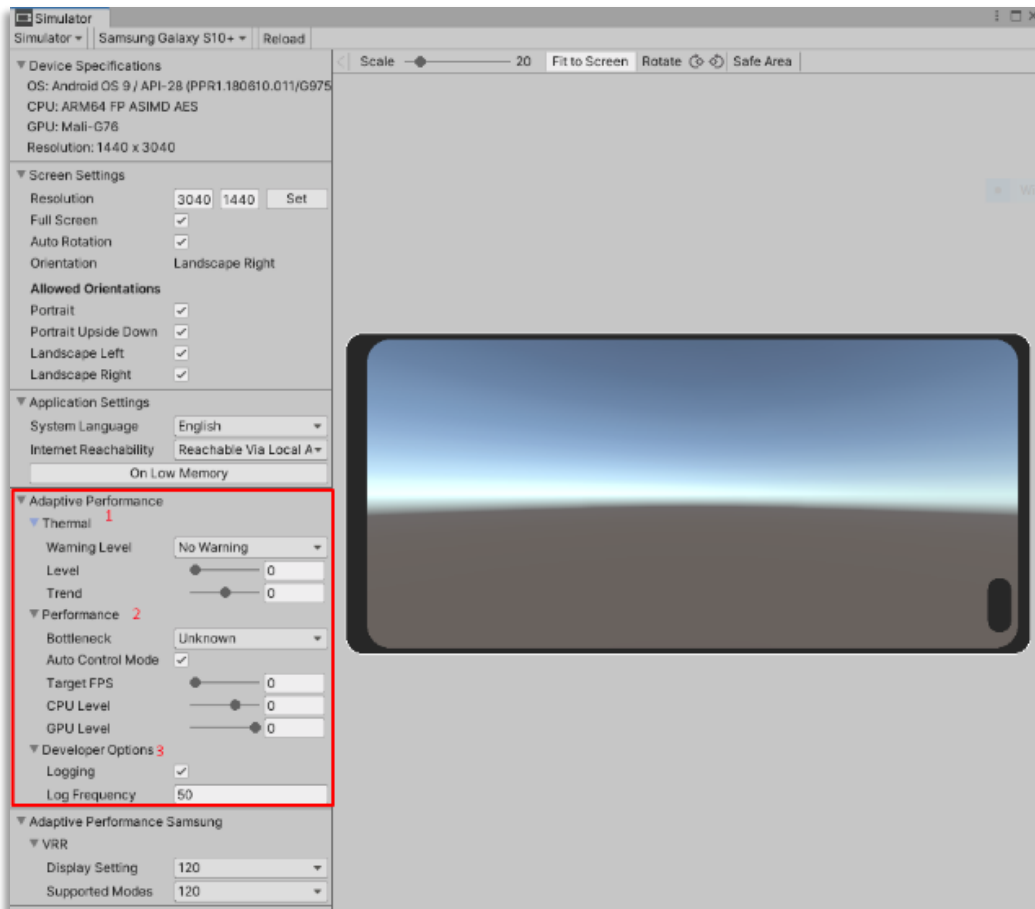
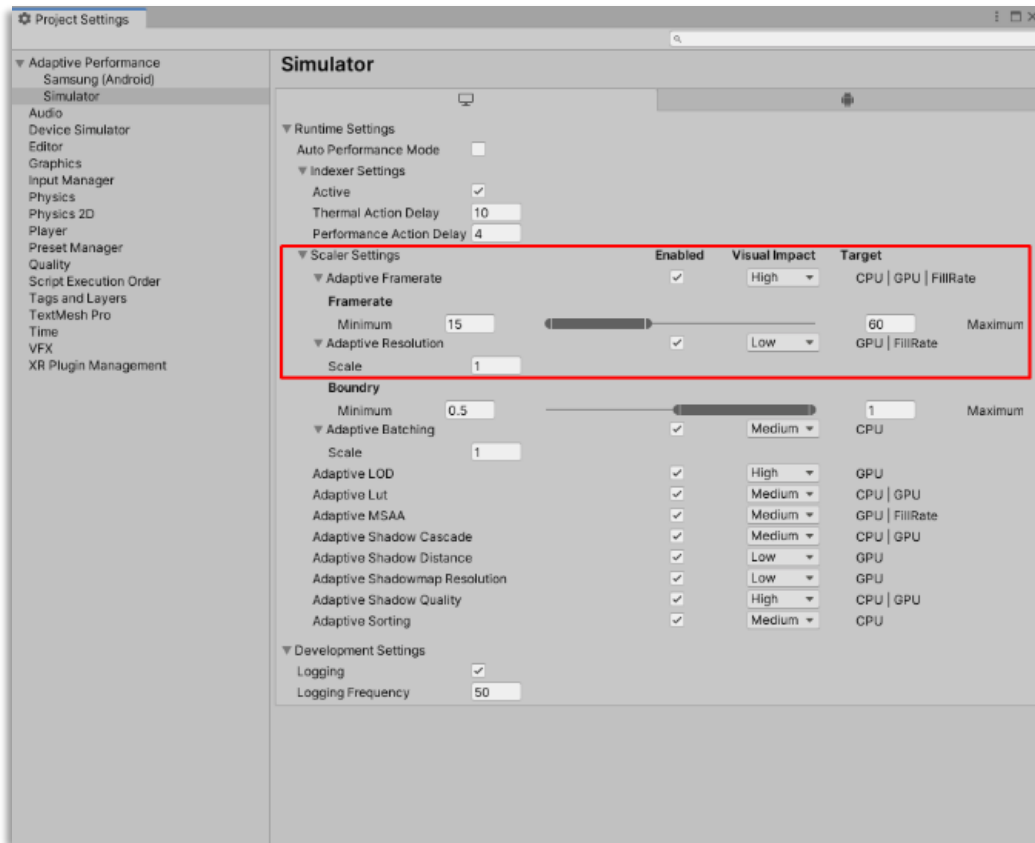
4.8 Unity Project Settings

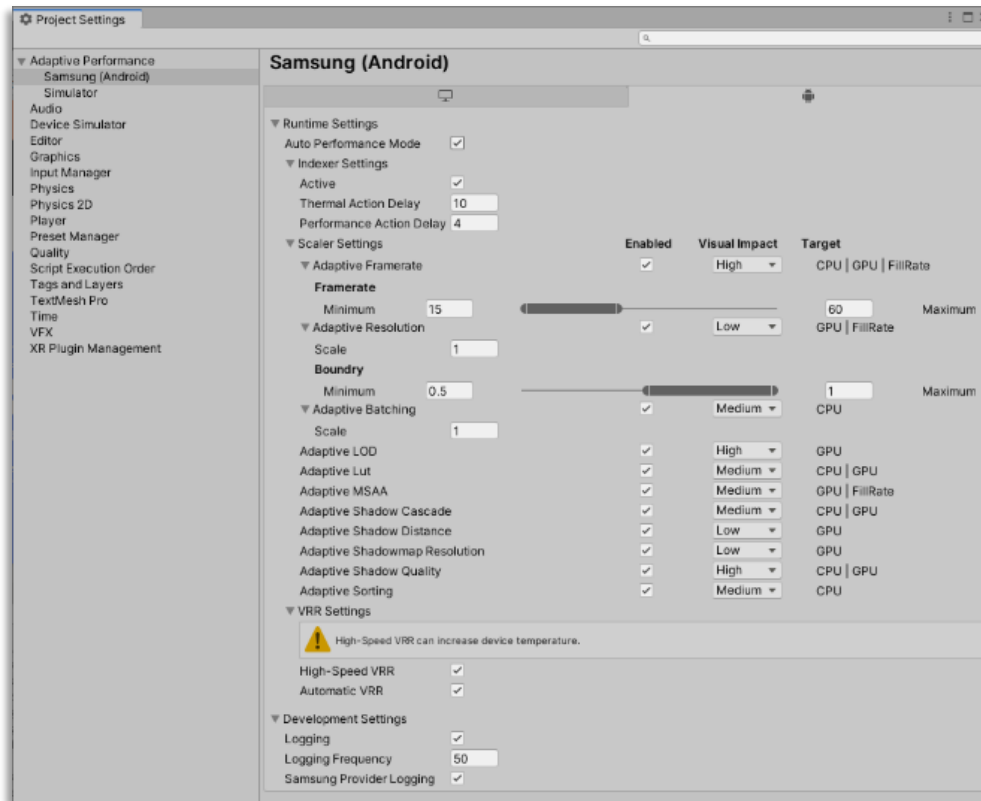
Εδώ θα γίνει μια παρουσίαση των βασικών στοιχείων που περιέχονται στο μενού του project settings και όλων των ρυθμίσεων που είναι απαραίτητα για την ανάπτυξη και της εκπαιδευτικής εφαρμογής για σταθερούς υπολογιστές και έξυπνες κινητές πλατφόρμες.

4.8.1 Ρύθμιση Adaptive Performance

Adaptive Performance : Η adaptive performance επιτρέπει στο χρήστη να λαμβάνει feedback σχετικά με τη θερμική και ενεργειακή κατάσταση της κινητής συσκευής και να αντιδρά αντιστοίχως. Ο χρήστης μπορεί να δημιουργήσει εφαρμογές που αντιδρούν στις αυξομειώσεις των τάσεων της θερμοκρασίας και στα συμβάντα της συσκευής, προκειμένου να εξασφαλίζονται σταθεροί ρυθμοί καρέ για μεγαλύτερο χρονικό διάστημα και να αποτρέπεται το θερμικό throttling.







Εικόνα 54. Adaptive Performance

Πίνακας 34. Various thermal and bottleneck conditions

| Condition | Message |
|---------------------|--|
| 1)Thermal | <ul style="list-style-type: none"> • Warning Level - Simulates a thermal warning level. The options are No Warning, Throttling Imminent, and Throttling. • Level - The temperature level simulates the skin temperature level of a device. Options range between nominal temperature (0) and critical throttling temperature (maximum level of 1.0). • Trend - Temperature trends with a value below 0 simulate a temperature drop. Values higher than 0 simulate a temperature increase. |
| 2)Performance | <ul style="list-style-type: none"> • Bottleneck - Simulate the device bottleneck, which can be CPU, GPU, or target frame rate bound. Target framerate is either VSync or the value set in Application.TargetFramerate. • Auto Control Mode - Sets CPU and GPU level automatically. If the system is throttling performance, the Control Mode is set to System and Auto Control Mode has no influence. • Target FPS - Simulates the application target framerate in frames per second. • CPU Level - Simulates the frequency cap of the CPU in discrete levels. • GPU Level - Simulates the frequency cap of the GPU in discrete levels. |
| 3)Developer Options | <ul style="list-style-type: none"> • Logging - Enable or disable logging from the Adaptive Performance subsystem. • Logging Frequency - Control Adaptive Performance logging frequency in frames |

Πηγή : <https://docs.unity3d.com/Packages/com.unity.adaptiveperformance@2.1/manual/simulator.html>

Πίνακας 35. Scaler

| Setting | Description |
|----------------------|--|
| Enabled | Defines if a scaler is enabled or disable. You can change this setting during runtime. |
| Visual Impact | The visual impact can be high, medium or low and depends on how much visual change a quality settings change causes. The higher the visual impact the higher the cost. this means a Scaler level is increased more often than a level of a Scaler with a lower cost. You can change this setting during runtime. |
| Target | This setting can not be changed. It defines which bottleneck a Scaler targets. Those targets can be CPU, GPU or Fillrate. If a bottleneck aligns with the target of a scaler it triggers the highest cost for this Scaler and the level is often increase. |

Πηγή : <https://docs.unity3d.com/Packages/com.unity.adaptiveperformance@2.1/manual/user-guide.html>

Για κάθε Scaler, διαμοιράζεται ένα σύνολο ρυθμίσεων που περιγράφουν το κόστος του Scaler. Το επίπεδο αυξάνεται πιο συχνά για το Scaler όσο υψηλότερο είναι το κόστος. Η αλλαγή του επιπέδου ενεργοποιεί μια αλλαγή ποιότητας στο Scaler. Οι ακόλουθες ρυθμίσεις είναι διαθέσιμες για κάθε Scaler.

Πίνακας 36. The target of a Scaler

| Target | Bottleneck |
|-----------------|-------------------------|
| CPU | CPU bound |
| GPU | GPU bound |
| FillRate | Target Frame Rate bound |

Το Adaptive Framerate απαιτεί ελάχιστη και μέγιστη ταχύτητα καρέ. Το Scaler θέτει το Application.targetFrameRate σε μια κατάλληλη τιμή μεταξύ αυτών των τιμών.

Πίνακας 37. Adaptive Framerate

| Setting | Description |
|----------------|---|
| Minimum | Defines the lowest framerate acceptable. |
| Maximum | Defines the highest framerate the Scaler should target. |

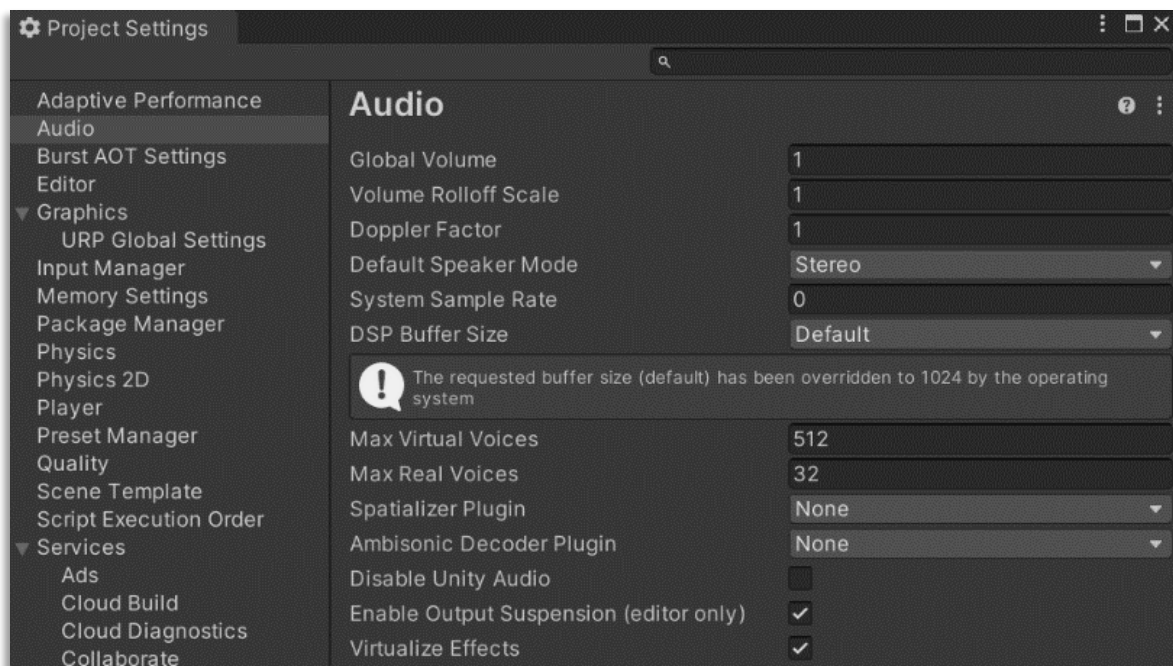
Η Adaptive Resolution απαιτεί ένα ελάχιστο και ένα μέγιστο όριο. Το Scaler αλλάζει την ανάλυση μεταξύ αυτών των ορίων με βήματα που ορίζονται από το μέγιστο επίπεδο.

Πίνακας 38. Adaptive Resolution

| Setting | Description |
|------------------|---|
| Minimum Boundary | Defines the lowest scale acceptable. |
| Maximum Boundary | Defines the highest scale desirable. |
| Maximum Level | Defines the scale steps which are taken between minimum and maximum boundary. |

4.8.2 Ρυθμίσεις Ήχου

[Audio](#) : Οι ρυθμίσεις ήχου στο Unity επιτρέπουν στο χρήστη να ρυθμίζει διάφορες πτυχές της αναπαραγωγής ήχου στο παιχνίδι. Ο προεπιλεγμένος μίκτης ήχου μπορεί να οριστεί, οι ρυθμίσεις ποιότητας ήχου μπορούν να προσαρμοστούν, παράμετροι χωρικού ήχου μπορούν να οριστούν και η μορφή εξόδου ήχου μπορεί να επιλεγεί. Αυτές οι ρυθμίσεις επιτρέπεται να ελέγχονται ο τρόπος επεξεργασίας, ανάμιξης και αναπαραγωγής του ήχου, εξασφαλίζοντας βέλτιστη απόδοση και πιστότητα ήχου.



Εικόνα 55. Ρυθμίσεις Ήχου

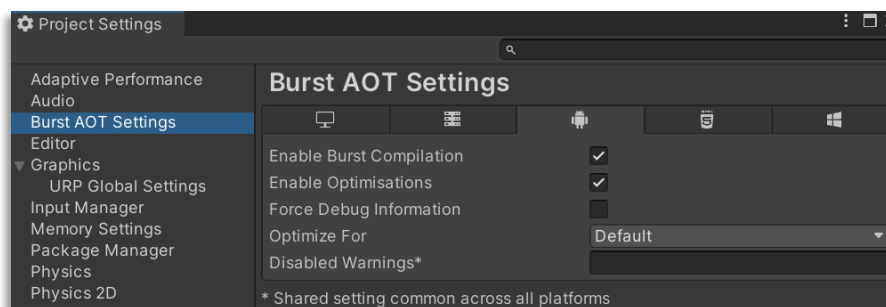
Πίνακας 39. Ιδιότητες ρυθμίσεων ήχου

| Property | Function |
|---|---|
| Global Volume | Set the volume for all sounds during playback. |
| Volume Rolloff Scale | Set the global attenuation rolloff factor for Logarithmic rolloff-based sources . The higher the value, the faster the volume attenuates. Conversely, the lower the value, the slower it attenuates. Tip: A value of 1 simulates the “real world”. |
| Doppler Factor | Set how audible the Doppler effect is. Use 0 to disable it. Use 1 make it audible for fast moving objects. Tip: After setting the Doppler Factor to 1, you can tweak both Speed of Sound and Doppler Factor until you are satisfied. |
| Default Speaker Mode | Set which speaker mode should be the default for your project. The default is 2, which corresponds to stereo speakers. For the full list of modes, see the AudioSpeakerMode API reference. Note: You can also change the speaker mode at runtime through scripting. See Audio Settings for details. |
| System Sample Rate | Set the output sample rate. If set to 0, Unity uses the sample rate of the system. Note: This only serves as a reference only, since certain platforms allow you to change the sample rate, such as iOS or Android. |
| DSP Buffer Size | Set the size of the DSP buffer to optimize for latency or performance. |
| Default | Default buffer size. |
| Best Latency | Trade off performance in favour of latency. |
| Good Latency | Balance between latency and performance. |
| Best Performance | Trade off latency in favour of performance. |
| Max Virtual Voices | Set the number of virtual voices that the audio system manages. This value should always be larger than the number of voices played by the game. If not, Unity displays warnings in the console. |
| Max Real Voices | Set the number of real voices that can play at the same time. At every frame, the loudest voice is picked. |
| Spatializer Plugin | Choose which native audio plugin to use in order to perform spatialized filtering of 3D sources. |
| Ambisonic Decoder Plugin | Choose which native audio plugin to perform ambisonic-to-binaural filtering of sources. |
| Disable Unity Audio | Enable to deactivate the audio system in standalone builds. In the Editor the audio system is still on and supports previewing audio clips, but Unity does not handle calls to AudioSource.Play and AudioSource.playOnAwake in order to simulate behavior of the standalone build. |
| Enable Output Suspension (Editor Only) | When enabled automatically suspends audio output after detecting that the output has been silent for a long duration (editor only) Suspending the audio system disables a mechanism in the operating system that prevents the computer from going into sleep mode |
| Virtualize Effect | Enable to dynamically turn off effects and spatializers on AudioSources that are culled in order to save CPU. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-AudioManager.html>

4.8.3 Ρυθμίσεις Burst Compiler

Burst AOT Settings: Οι ρυθμίσεις του Burst AOT (Ahead-Of-Time) παρέχουν επιλογές στο χρήστη για τη βελτιστοποίηση της εκτέλεσης του κώδικα μέσω του μεταγλωττιστή Burst. Η μεταγλώττιση του Burst μπορεί να ενεργοποιηθεί για συγκεκριμένες πλατφόρμες και η κάθε μια από αυτές έχει διάφορες ρυθμίσεις για το μεταγλωττιστή. Η τεχνολογία Burst μεταφράζει τον κώδικά σας από IL/.NET bytecode σε βελτιστοποιημένο εγγενή κώδικα CPU που χρησιμοποιεί τον μεταγλωττιστή LLVM.



Εικόνα 56. Ρυθμίσεις Burst Compiler για Android

Πηγή : <https://docs.unity3d.com/Packages/com.unity.burst@1.8/manual/index.html>

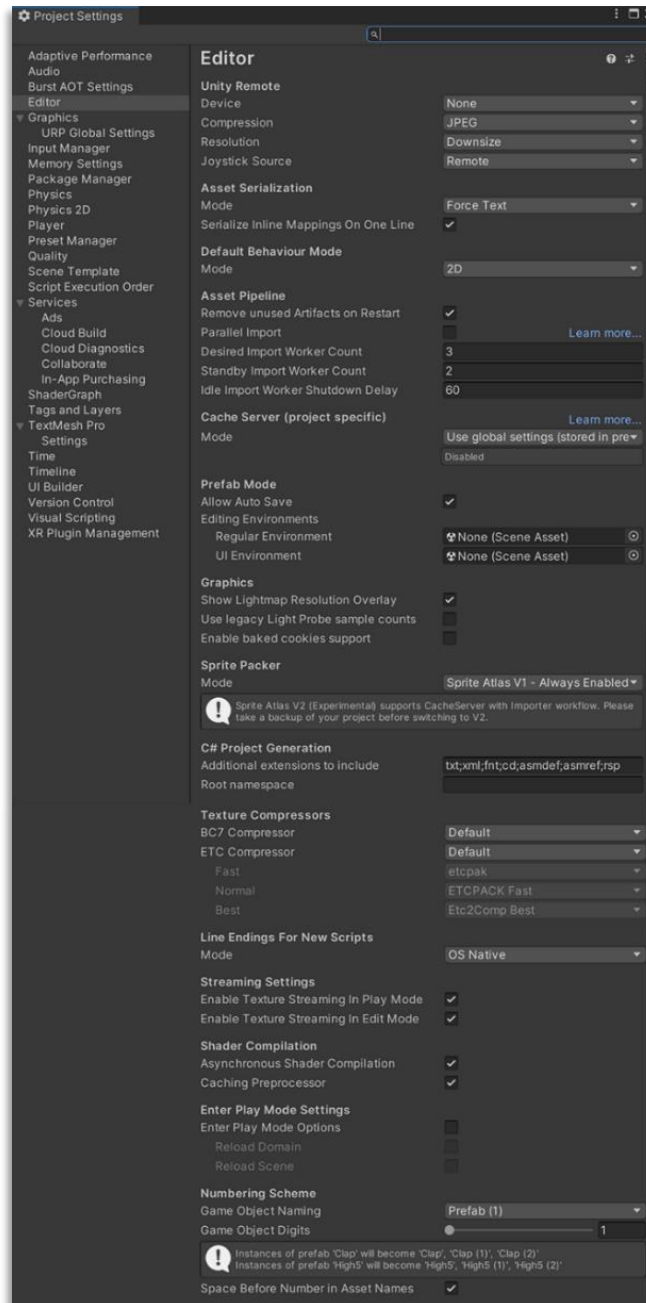
4.8.4 Ρυθμίσεις Editor

Editor Settings: Το Unity Remote είναι μια εφαρμογή που μπορείτε να κατεβάσετε και η οποία έχει σχεδιαστεί για να βοηθήσει στην ανάπτυξη Android, iOS και tvOS. Επιτρέπει τη δοκιμή της εφαρμογής σας σε κινητές συσκευές. Μπορεί να επιλεγεί ο τύπος της συσκευής για τη δοκιμή Unity Remote. Κατά τη μετάδοση της οθόνης του παιχνιδιού στη συσκευή μέσω του Unity Remote, μπορεί να επιλεγεί ο τύπος συμπίεσης εικόνας. Οι επιλογές περιλαμβάνουν JPEG ή PNG, με το JPEG να είναι η προεπιλεγμένη επιλογή. Η ανάλυση στην οποία εκτελείται το παιχνίδι στο Unity Remote μπορεί επίσης να επιλεγεί, με επιλογές για Downsize και Normal. Μπορεί να επιλεγεί η πηγή των joysticks που χρησιμοποιούνται, είτε Remote (συνδεδεμένο σε συσκευή που εκτελεί Unity Remote) είτε Local (συνδεδεμένο στον υπολογιστή που εκτελεί τον Editor). Το Unity Remote παρέχει επίσης επιλογές για ρυθμίσεις ελέγχου εκδόσεων, τρόπους serialization ψηφιακών στοιχείων, προεπιλεγμένους τρόπους συμπεριφοράς, asset pipeline versions και πολλά άλλα.



Εικόνα 57. Εφαρμογή Unity Remote στο Google Play Store

Πηγή : <https://play.google.com/store/apps/details?id=com.unity3d.mobileremote>



Εικόνα 58. Ρυθμίσεις του Editor

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-EditorManager.html>



Η εφαρμογή Unity Remote 5 είναι μια πολύ καλή πρακτική για να βλέπει ο χρήστης πως θα φαίνεται η εφαρμογή όταν ολοκληρωθεί χωρίς να πρέπει να εγκαταστήσει το εκτελέσιμο αρχείο στην συσκευή.

4.8.5 Ρυθμίσεις Γραφικών

[Graphics Settings](#): Αυτές οι ρυθμίσεις επιτρέπουν στους προγραμματιστές να εφαρμόζουν σφαιρικές ρυθμίσεις για τα γραφικά στα έργα τους. Οι ρυθμίσεις γραφικών περιλαμβάνουν διάφορες ομάδες ιδιοτήτων, συμπεριλαμβανομένων των ρυθμίσεων Scriptable Render Pipeline Settings, Camera Settings, Tier Settings, Built-in shader settings, Always-included Shaders, Shader stripping, Shader loading και Culling settings.

- ❖ Η επιλογή **Scriptable Render Pipeline Settings** επιτρέπει στους προγραμματιστές να ορίσουν τον προεπιλεγμένο αγωγό απόδοσης γραφικών για το έργο τους. Αυτό περιλαμβάνει την ανάθεση ενός Render Pipeline Asset που καθορίζει τον αγωγό απόδοσης που θα χρησιμοποιηθεί.
- ❖ Η ομάδα **Camera Settings** ελέγχει ρυθμίσεις απόδοσης, όπως η λειτουργία ταξινόμησης διαφάνειας, ο άξονας ταξινόμησης διαφάνειας και ο προσαρμοσμένος άξονας για την ταξινόμηση αντικειμένων με βάση την απόσταση ή την προοπτική.
- ❖ Η κατηγορία **Tier Settings** επιτρέπει στους προγραμματιστές να τροποποιούν τις ρυθμίσεις απόδοσης και σύνταξης σκίασης με βάση διαφορετικούς τύπους υλικού. Περιλαμβάνει επιλογές για την προσαρμογή της τυπικής ποιότητας shader, των ανιχνευτών αντανάκλασης, των λεπτομερών κανονικών χαρτών, των σκιών, της απόδοσης HDR και της επιλογής διαδρομής απόδοσης.
- ❖ Η ενότητα **Built-in shader settings** παρέχει επιλογές για τον καθορισμό των shader που χρησιμοποιούνται για το Deferred rendering, τους deferred reflection probes (παλαιός και καινούργιος τρόπος), τις σκιές χώρου οθόνης, τα διανύσματα κίνησης τα Depth Normals, τα lens flares και τα light halos, με επιλογές για ενσωματωμένους shaders ή προσαρμοσμένους shaders.
- ❖ Η επιλογή **Always-included Shaders** επιτρέπει στους προγραμματιστές να καθορίζουν shaders ή παραλλαγές shaders που περιλαμβάνονται σε κάθε build. Οι ρυθμίσεις

απογύμνωσης shaders καθορίζουν τη συμπεριφορά της απογύμνωσης των αχρησιμοποίητων παραλλαγών shaders στη δημιουργία και οι ρυθμίσεις φόρτωσης shaders διαμορφώνουν τον τρόπο με τον οποίο οι συλλογές παραλλαγών shaders προθερμαίνονται στην εφαρμογή.

- ❖ Τέλος, οι ρυθμίσεις **Culling** προσφέρουν τον έλεγχο του camera-relative culling για τα φώτα και τις σκιές, επιτρέποντας στους προγραμματιστές να βελτιστοποιήσουν το culling με βάση τη θέση της κάμερας.

Πίνακας 40. Camera Settings

| Property | Function |
|-------------------------------|--|
| Transparency Sort Mode | Define the order for rendering objects by their distance along a specific axis. Renderers in Unity are sorted by several criteria, such as their layer number or their distance from the camera . This is generally only useful in 2D development: for example, sorting Sprites by height or along the Y-axis. |
| <i>Default</i> | Sort objects based on the Camera mode. |
| <i>Perspective</i> | Sort objects based on perspective view. |
| <i>Orthographic</i> | Sort objects based on orthographic view. |
| <i>Custom Axis</i> | Sort objects based on the sort mode defined with the Transparency Sort Axis . |
| Transparency Sort Axis | Define a custom Transparency Sort Mode . |

Πηγή : <https://docs.unity3d.com/Manual/class-GraphicsSettings.html>

Πίνακας 41. Ενσωματωμένα Shader Settings

| Rendering path | Shader to use |
|-----------------------------|---|
| Deferred | Use with Deferred shading . |
| Deferred Reflection | Use with Reflection Probes along deferred lighting. |
| Screen Space shadows | Use with cascaded shadow maps for directional lights on PC/console platforms. |
| Legacy deferred | Use with Legacy Deferred lighting. |
| Motion vectors | Use for object-based motion vector calculations. |
| Lens Flare | Use with Lens Flares . |
| Light Halo | Use with Light Halos . |

Πίνακας 42. Shader Stripping properties and functions

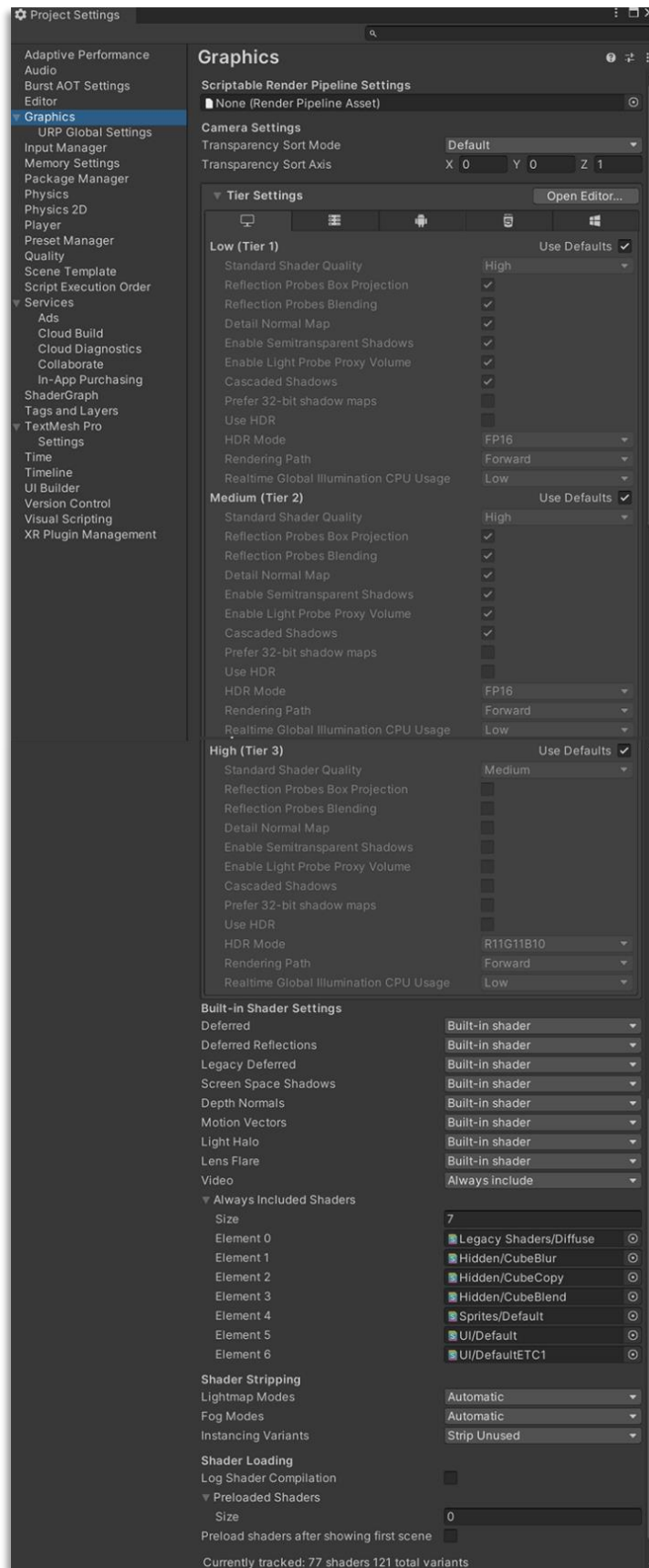
| Property | Function |
|----------------------------|---|
| Lightmap Modes | Determine the shader variant stripping behavior for lightmap-related shaders. |
| <i>Automatic</i> | Unity examines the scenes in the build and automatically strips variants that are not used in those scenes. This is the default setting. |
| <i>Custom</i> | Choose to manually include or exclude variants for the following lightmap modes: <ul style="list-style-type: none"> * Baked Non-Directional * Baked Directional * Realtime Non-Directional * Realtime Directional * Baked Shadowmask * Baked Subtractive |
| Fog Modes | Determine the shader variant stripping behavior for shaders that relate to Unity's built-in fog effect. |
| <i>Automatic</i> | Unity examines the scenes in the build and automatically strips variants that are not used in those scenes. This is the default setting. |
| <i>Custom</i> | Choose to manually include or exclude variants for the following fog modes: <ul style="list-style-type: none"> * Linear * Exponential * Exponential Squared |
| Instancing Variants | Determine the shader variant stripping behavior for shaders that relate to GPU instancing. |
| <i>Strip Unused</i> | Unity only includes variants for GPU instancing for a given shader if at least one material that uses that shader has Enable instancing enabled. This is the default setting. |
| <i>Strip All</i> | Strip all variants for GPU instancing, even if they are being used. |
| <i>Keep All</i> | Include all variants for GPU instancing, even if they are being used. |

Πίνακας 43. Shader Loading properties and functions

| Property | Function |
|--|---|
| Preloaded shaders | The shader variant collections to prewarm on application start. |
| Preload shaders after showing first scene | When Unity prewarms the shader variants specified in Preloaded shaders . If enabled, Unity loads and prewarms them after the first scene has fully loaded. Otherwise, Unity loads and prewarms them before showing the first scene. |
| Preloaded shaders | How Unity prewarms the shader variants specified in Preloaded shaders , if Preload shaders after showing first scene is enabled. If the value is 0, Unity preloads all shader variants on the next frame after it shows the first scene. Otherwise, Unity preloads new shader variants over multiple frames after it shows the first scene. In a given frame, it prewarms shader variants until it reaches that time limit set here. After that, it does not begin prewarming new shader variants until the next frame. |

Πίνακας 44. Tier Settings properties and functions

| Property | Function |
|--|---|
| Standard Shader Quality | Set the quality of the Standard Shader to <i>High</i> , <i>Medium</i> , or <i>Low</i> . |
| Reflection Probes Box Projection | Enable projection for reflection UV mappings on Reflection Probes . |
| Reflection Probes Blending | Enable blending on Reflection Probes . |
| Detail Normal Map | Enable Detail Normal Map sampling, if assigned. |
| Enable Semitransparent Shadows | Enable Semitransparent Shadows. This adds or removes the UNITY_USE_DITHER_MASK_FOR_ALPHABLENDED_SHADOWS shader compiler define. |
| Enable Light Probe Proxy Volume | Enable rendering a 3D grid of interpolated Light Probes . |
| Cascaded Shadows | Enable using cascaded shadow maps. This adds or removes the UNITY_NO_SCREENSPACE_SHADOWS shader compiler define. |
| Prefer 32 bit shadow maps | Enable 32-bit float shadow map when you are targeting PS4 or platforms using DX11 or DX12. Most platforms have a fixed shadow map format that you can't adjust. These vary in format, and can be 16-bit, 24-bit, or 32-bit, and can also be either float- or integer-based. 32-bit shadow maps give higher quality shadows than 16-bit, but use increased memory and bandwidth on the GPU. Note: To use 32-bit shadow maps, make sure the depth buffer is also set to 32-bit. |
| Use HDR | Enable High Dynamic Range rendering for this tier. |
| HDR Mode | Select the format to use for the HDR buffer when HDR is enabled for the current Graphics Tier. By default, this is set to <i>FP16</i> . |
| FP16 | Color render texture format, 16-bit floating point per channel. |
| R11G11B10 | Color render texture format . R and G channels are 11-bit floating point, B channel is 10-bit floating point. |
| Rendering Path | Choose how Unity should render graphics. Different rendering paths affect the performance of your game, and how lighting and shading are calculated. Some paths are more suited to different platforms and hardware than others. <i>Deferred</i> rendering is not supported when using Orthographic projection. If the camera's projection mode is set to Orthographic, these values are overridden, and the camera always uses <i>Forward</i> rendering. For more information, see Rendering Paths . |
| Forward | The traditional rendering path . This supports all the typical Unity graphics features (normal maps, per-pixel lights, shadows etc.). However under default settings, only a small number of the brightest lights are rendered in per-pixel lighting mode. The rest of the lights are calculated at object vertices or per-object. |
| Deferred | Deferred shading has the most lighting and shadow fidelity, and is best suited if you have many real-time lights. It requires a certain level of hardware support. |
| Legacy Vertex Lit | Legacy Vertex Lit is the rendering path with the lowest lighting fidelity and no support for real-time shadows. It is a subset of <i>Forward</i> rendering path. |
| Legacy Deferred (light prepass) | Legacy Deferred is similar to Deferred Shading, just using a different technique with different trade-offs. It does not support the Unity 5 physically-based standard shader. |
| Realtime Global Illumination CPU Usage | The CPU budget you allow Enlighten Realtime Global Illumination to use for lighting calculations at runtime. Increasing this makes the system react faster to changes in lighting at a cost of using more CPU time. Note: Some platforms allow all CPUs to be occupied by worker threads whereas some enforce maximums. For example, some gaming consoles allow a maximum of 4 CPU cores. For Android devices, if it is a bigLittle architecture, only the little CPUs are used; otherwise, the maximum is one less than the total number of CPUs. |
| Low | 25% of the allowed CPU threads are used as worker threads. |
| Medium | 50% of the allowed CPU threads are used as worker threads. |
| High | 75% of the allowed CPU threads are used as worker threads. |
| Unlimited | 100% of the allowed CPU threads are used as worker threads. |



Εικόνα 59. Graphics Settings

4.8.6 Ρυθμίσεις Χειρισμού

[Input Manager](#): Εδώ ο χρήστης μπορεί να ορίσει τον τρόπο που θα γίνεται η διαδραση του με την εφαρμογή. Η μηχανή υποστηρίζει εισοδο από πολλούς τυπους συσκευων εισοδου, όπως:

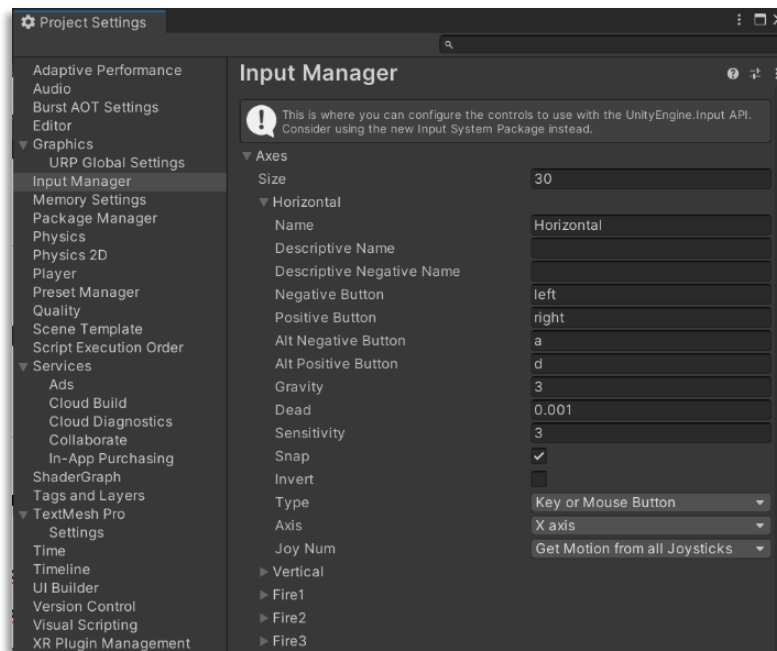
- Πληκτρολόγια και ποντίκια
- Joysticks
- Οθόνες αφής
- Δυνατότητες ανίχνευσης κίνησης κινητών συσκευών, όπως επιταχυνσιόμετρα ή γυροσκόπια
- VR & AR controllers

Η Unity υποστηρίζει την είσοδο μέσω δύο ξεχωριστών συστημάτων:

- Input Manager ο οποίος αποτελεί την default επιλογή της μηχανής
- Το Input System που είναι ένα πακέτο που πρέπει να εγκατασταθεί μέσω του Διαχειριστή πακέτων πριν ο χρήστης μπορέσει να το χρησιμοποιήσει. Απαιτεί το περιβάλλον εκτέλεσης .NET 4 και δεν λειτουργεί σε έργα που χρησιμοποιούν το παλιό περιβάλλον εκτέλεσης .NET 3.5.

Ο Διαχειριστής εισόδου χρησιμοποιεί τους ακόλουθους τύπους για έλεγχο εισόδου :

- Key αναφέρεται σε οποιοδήποτε πλήκτρο πάνω σε ένα πληκτρολόγιο, όπως το W, το Shift ή το spacebar.
- Το κουμπί αναφέρεται σε οποιοδήποτε κουμπί σε ένα φυσικό χειριστήριο (πχ. gamepads), όπως το κουμπί X σε ένα τηλεχειριστήριο.
- Ένας εικονικός άξονας αντιστοιχίζεται σε ένα χειριστήριο, όπως ένα κουμπί ή ένα πλήκτρο. Όταν ο χρήστης ενεργοποιεί το χειριστήριο, ο άξονας λαμβάνει μια τιμή στο εύρος [-1..1]. Μπορεί να χρησιμοποιήσει αυτή την τιμή στα σενάρια του.

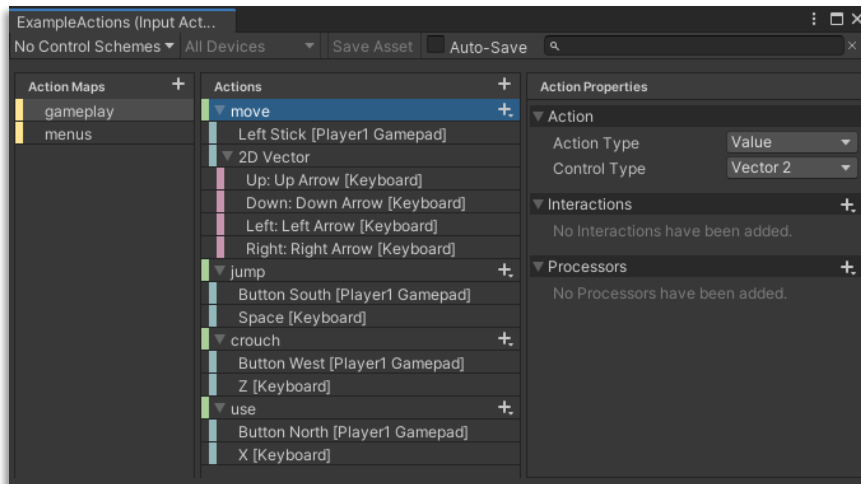


Εικόνα 60. Input Manager

Πίνακας 45. Input Axis Properties and Functions

| Property | Function |
|--|--|
| Name | Axis name. You can use this to access the axis from scripts. |
| Descriptive Name, Descriptive Negative Name | These values are deprecated and do not work. Previously, they displayed for the user on the Rebind Controls screen at startup, but this screen has also been deprecated. |
| Negative Button, Positive Button | The controls to push the axis in the negative and positive direction respectively. These can be keys on a keyboard, or buttons on a joystick or mouse. |
| Alt Negative Button, Alt Positive Button | Alternative controls to push the axis in the negative and positive direction respectively. |
| Gravity | Speed in units per second that the axis falls toward neutral when no input is present. |
| Dead | How far the user needs to move an analog stick before your application registers the movement. At runtime, input from all analog devices that falls within this range will be considered null. |
| Sensitivity | Speed in units per second that the axis will move toward the target value. This is for digital devices only. |
| Snap | If enabled, the axis value will reset to zero when pressing a button that corresponds to the opposite direction. |
| Type | The type of input that controls the axis. Select from these values: - Key or Mouse button - Mouse Movement - Joystick Axis |
| Axis | The axis of a connected device that controls this axis. |
| JoyNum | The connected Joystick that controls this axis. You can select a specific joystick, or query input from all joysticks. |

Πηγή : <https://docs.unity3d.com/Manual/class-InputManager.html>

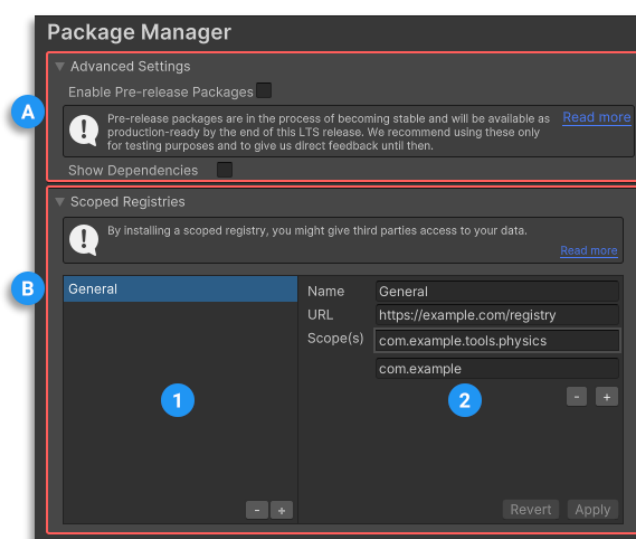


Εικόνα 61. Input System

Πηγή : <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.6/manual/index.html>





4.8.7 Package Manager

Package Manager: Ο διαχειριστής πακέτων στο Unity χρησιμοποιείται για τη διαχείριση των πακέτων λογισμικού που χρησιμοποιούνται στο περιβάλλον ανάπτυξης της μηχανής Unity. Μέσω του διαχειριστή πακέτων, τα πακέτα μπορούν να εισαχθούν, να ενημερωθούν ή να αφαιρεθούν από ένα project. Επιπλέον, η μηχανή Unity υποστηρίζει τη δυνατότητα εγκατάστασης πακέτων από εξωτερικές πηγές, όπως το Unity Asset Store ή το GitHub, για να επεκτείνει τις δυνατότητες και τις λειτουργίες του περιβάλλοντος ανάπτυξης ενός project.





Εικόνα 62. Διαχειριστής Πακέτων

Πίνακας 46. Settings for the Package Manager

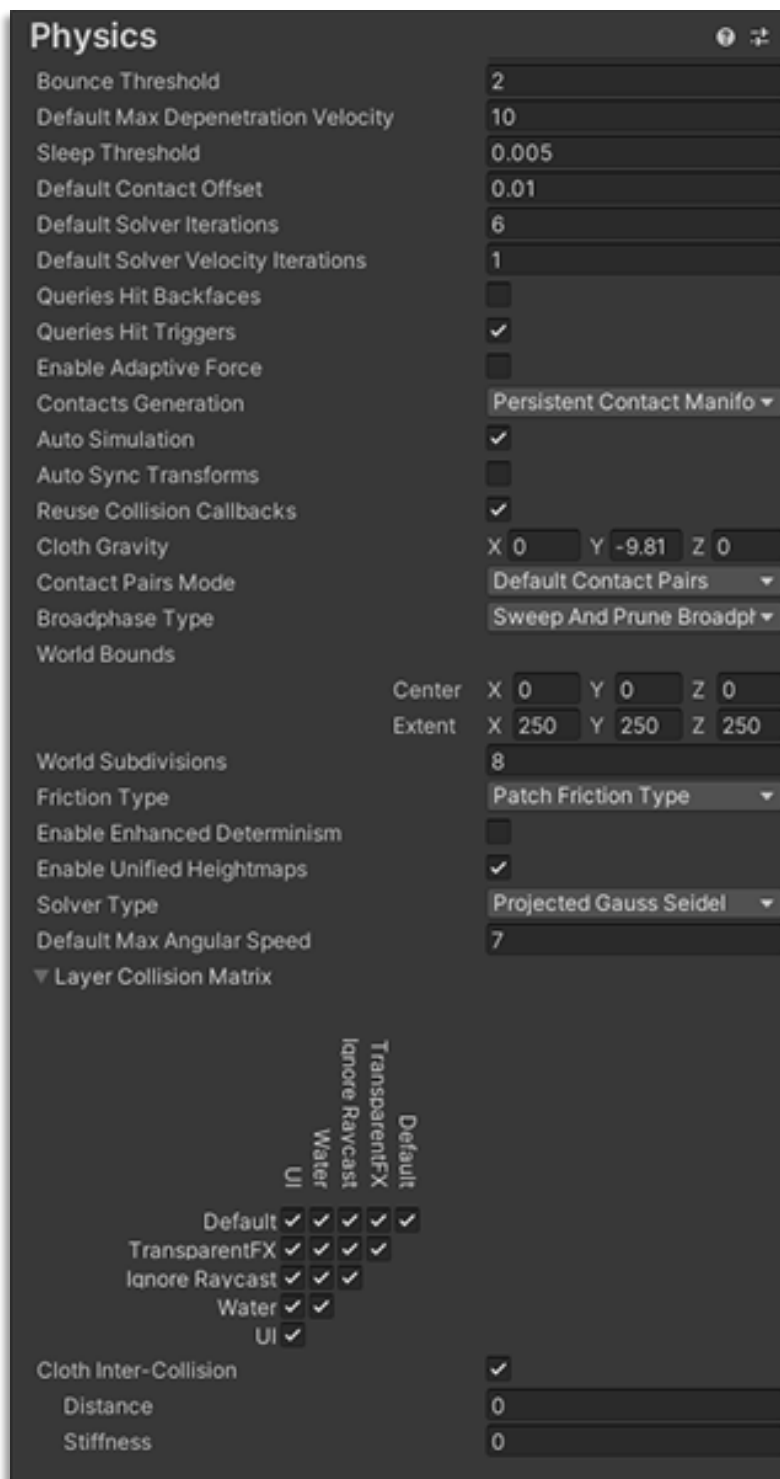
| Icon | Description |
|---|--|
|  | Under the Advanced Settings section, toggle whether or not pre-release packages and package dependencies are visible in the Package Manager window. |
|  | Add , modify , and remove scoped registries for your project under the Scoped Registries group, which contains the following areas: |
|  | The list of scoped registries currently defined for this project |
|  | The details for the selected scoped registry |

Πηγή : <https://docs.unity3d.com/Manual/class-PackageManager.html>

| | |
|---|--|
|  | Μια καλή πρακτική είναι η ενεργοποίηση της εμφάνισης πακέτων pre release για να έχει την δυνατότητα ο χρήστης να επιλέξει τα πιο καινούργια εργαλεία για το project του. |
|  | Εδώ θέλει προσοχή γιατί αυτά τα εργαλεία επειδή είναι ακόμα σε πειραματικό στάδιο ενδεχόμενος να μην δουλεύουν σωστά. |

4.8.8 Ρυθμίσεις Μηχανής Φυσικής 3D

[Physics](#): Ρυθμίσεις της Unity μηχανής Φυσικής για όλο το project (global settings) για 3D φυσική. Οι ρυθμίσεις αυτές προσδιορίζουν τα όρια της ακρίβειας της φυσικής προσομοίωσης, με τη γενική αρχή ότι μια πιο ακριβής προσομοίωση επιφέρει αυξημένη απαίτηση της επεξεργαστικής ισχύος. Έτσι, αυτές οι ρυθμίσεις παρέχουν έναν τρόπο για την απόκτηση μιας ισορροπίας μεταξύ ακρίβειας και απόδοσης.



Εικόνα 63. Ρυθμίσεις Φυσικής 3D

Πίνακας 47. Physics 3D properties and functions

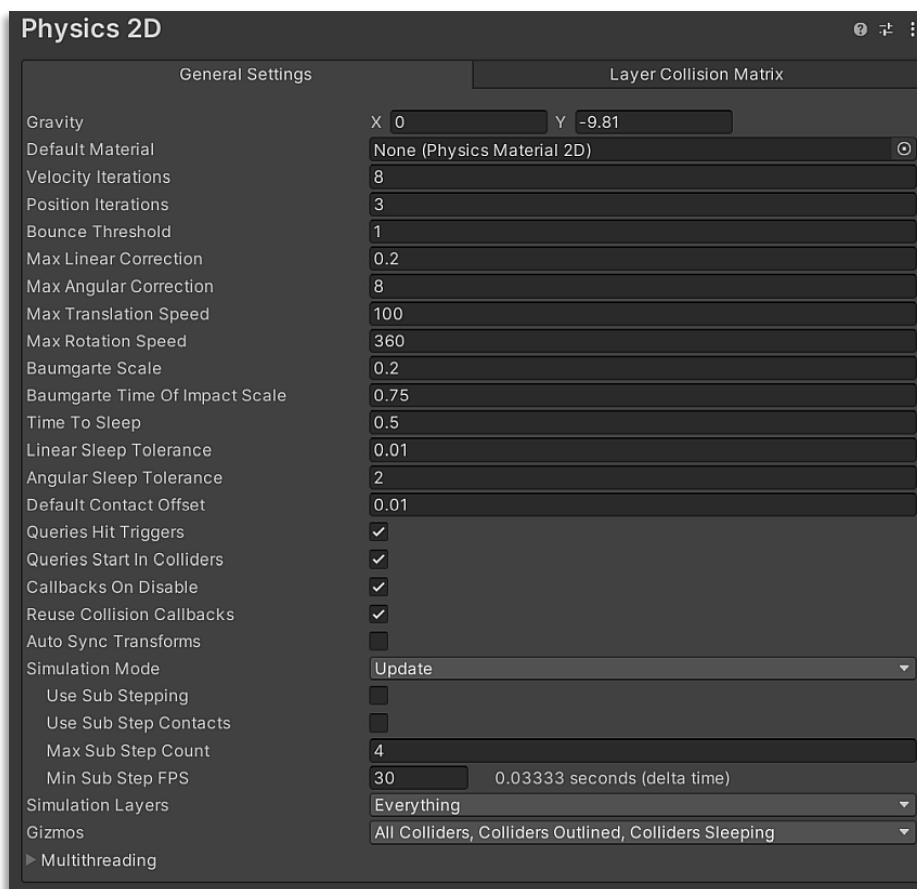
| Property | Function |
|---|---|
| Gravity | Use the x, y and z axes to set the amount of gravity applied to all Rigidbody components. For realistic gravity settings, apply a negative number to the y axis. Gravity is defined in world units per seconds squared. Note: If you increase the gravity, you might need to also increase the Default Solver Iterations value to maintain stable contacts. |
| Default Material | Set a reference to the default Physics Material to use if none has been assigned to an individual Collider . |
| Bounce Threshold | Set a velocity value. If two colliding objects have a relative velocity below this value, they do not bounce off each other. This value also reduces jitter, so it is not recommended to set it to a very low value. |
| Default Max Depenetration Velocity | Define the default value for the maximum depenetration velocity (velocity that the solver can set to a body while trying to pull it out of overlap with the other bodies). |
| Sleep Threshold | Set a global energy threshold, below which a non-kinematic Rigidbody (that is, one that is not controlled by the physics system) may go to sleep. When a Rigidbody is sleeping, it is not updated every frame, making it less resource-intensive. If a Rigidbody's kinetic energy divided by its mass is below this threshold, it is a candidate for sleeping. |
| Default Contact Offset | Set the distance the collision detection system uses to generate collision contacts. The value must be positive, and if set too close to zero, it can cause jitter. This is set to 0.01 by default. Colliders only generate collision contacts if their distance is less than the sum of their contact offset values. |
| Default Solver Iterations | Define how many solver processes Unity runs on every physics frame. Solvers are small physics engine tasks which determine a number of physics interactions, such as the movements of joints or managing contact between overlapping Rigidbody components. This affects the quality of the solver output and it's advisable to change the property in case non-default Time.fixedDeltaTime is used, or the configuration is extra demanding. Typically, it's used to reduce the jitter resulting from joints or contacts. |
| Default Solver Velocity Iterations | Set how many velocity processes a solver performs in each physics frame. The more processes the solver performs, the higher the accuracy of the resulting exit velocity after a Rigidbody bounce. If you experience problems with jointed Rigidbody components or Ragdolls moving too much after collisions, try increasing this value. |
| Queries Hit Backfaces | Enable this option if you want physics queries (such as Physics.Raycast) to detect hits with the backface triangles of MeshColliders. By default, this setting is disabled. |
| Queries Hit Triggers | Enable this option if you want physics hit tests (such as Raycasts, SphereCasts and SphereTests) to return a hit when they intersect with a Collider marked as a Trigger. Individual raycasts can override this behavior. By default, this setting is enabled. |
| Enable Adaptive Force | Enable this option to enable the adaptive force. The adaptive force affects the way forces are transmitted through a pile or stack of objects, to give more realistic behaviour. By default, this setting is disabled. |
| Auto Simulation | Enable this option to run the physics simulation automatically or allow explicit control over it. |
| Auto Sync Transforms | Enable this option to automatically sync transform changes with the physics system whenever a Transform component changes. By default, this setting is disabled. |
| Contact Pairs Mode | Choose the type of contact pair generation to use. |
| Default Contact Pairs | Receive collision and trigger events from all contact pairs except kinematic-kinematic and kinematic-static pairs. |
| Enable Kinematic Kinematic Pairs | Receive collision and trigger events from kinematic-kinematic contact pairs. |
| Enable Kinematic Static Pairs | Receive collision and trigger events from kinematic-static contact pairs. |

| Property | Function |
|-------------------------------|--|
| Enable All Contact Pairs | Receive collision and trigger events from all contact pairs. |
| Broadphase Type | Choose which broad-phase algorithm to use in the physics simulation. See NVIDIA's documentation on PhysX SDK and Rigid Body Collision . |
| Sweep and Prune Broadphase | Use the sweep-and-prune broad phase collision method (that is, sorting objects along a single axis to rule out having to check pairs that are far apart). |
| Automatic Box Pruning | This algorithm is similar to the Multibox Pruning one, except that it can also automatically compute the world boundaries and number of subdivisions. It maintains the set of grid cells and uses the regular sweep-and-prune approach to work out potentially overlapping pairs of colliders. It usually helps with big scenes where a single Sweep and Prune would produce lots of extra false positives. |
| Friction Type | Choose the friction algorithm used for simulation. |
| Patch Friction Type | A basic strong friction algorithm which typically leads to the most stable results at low solver iteration counts. This method uses only up to four scalar solver constraints per pair of touching objects. |
| One Directional Friction Type | A simplification of the Coulomb friction model, in which the friction for a given point of contact is applied in the alternating tangent directions of the contact's normal. This requires more solver iterations than patch friction but is not as accurate as the two-directional model. For Articulation bodies to work with this friction type, set the Solver Type to Temporal Gauss Seidel . |
| Two Directional Friction Type | Like the one-directional model, but applies friction in both tangent directions simultaneously. This requires more solver iterations but is more accurate. More expensive than patch friction for scenarios with many contact points because it is applied at every contact point. For Articulation bodies to work with this friction type, set the Solver Type to Temporal Gauss Seidel . |
| Enable Enhanced Determinism | Simulation in the scene is consistent regardless the actors present, provided that the game inserts the actors in a deterministic order. This mode sacrifices some performance to ensure this additional determinism. |
| Enable Unified Heightmaps | Enable this option to process Terrain collisions in the same way as Mesh collisions. |
| Solver Type | Choose the PhysX solver type to use for the simulation. |
| Projected Gauss Seidel | The default PhysX solver. |
| Temporal Gauss Seidel | This solver offers a better convergence and a better handling of high-mass ratios, minimizes energy introduced when correcting penetrations and improves the resistance of joints to overstretch. It usually helps when you experience some erratic behavior during simulation with the default solver. |
| Layer Collision Matrix | Define how the layer-based collision detection system behaves. Select which layers on the Collision Matrix interact with the other layers by checking them. |
| Cloth Inter-Collision | Specify one set of particles for cloth inter-collision to make fabrics in your game move more realistically. |
| Distance | Define the diameter of a sphere around each intercolliding Cloth particle. Unity ensures that these spheres do not overlap during simulations. Distance should be smaller than the smallest distance between two particles in the configuration. If the distance is larger, cloth collision may violate some distance constraints and result in jittering. |
| Stiffness | How strong the separating impulse between intercolliding Cloth particles should be. The cloth solver calculates this and it should be enough to keep the particles separated. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-PhysicsManager.html>

4.8.9 Ρυθμίσεις Μηχανής Φυσικής 2D

Physics 2D: Οι παρακάτω ρυθμίσεις διαχειρίζονται τις παγκόσμιες ρυθμίσεις του Project για τα Physics 2D στο σύστημα φυσικής της Unity, οι οποίες καθορίζουν τα όρια της ακρίβειας της προσομοίωσης φυσικής για τα 2D GameObjects. Η πιο ακριβής προσομοίωση απαιτεί μεγαλύτερη επεξεργαστική ισχύ, και αυτές οι ρυθμίσεις επιτρέπουν τον χρήστη να τις παραμετροποιήσει μεταξύ ακρίβειας και απόδοσης όπου ταιριάζει καλύτερα στο έργο του.



Εικόνα 64. Ρυθμίσεις Φυσικής 2D

Πίνακας 48. Physics 2D properties and functions

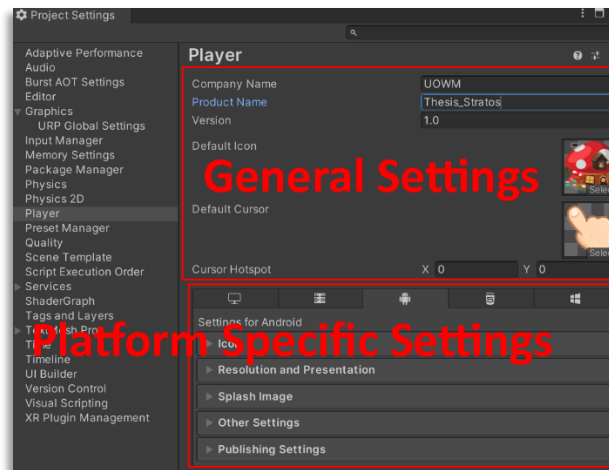
| Property | Function |
|---------------------------------------|---|
| Gravity | Set the amount of gravity applied to all Rigidbody 2D GameObjects. Usually you only set gravity for the negative direction of the y-axis. |
| Default Material | Set a reference to the Physics Material 2D to use if none has been assigned to an individual Collider 2D . |
| Velocity Iterations | Set the number of iterations made by the physics system to resolve velocity effects. Higher numbers result in more accurate physics calculations but the Editor requires more CPU time. |
| Position Iterations | Set the number of iterations made by the physics system to resolve position changes. Higher numbers result in more accurate physics calculations but also requires more CPU time. |
| Bounce Threshold | Set the threshold for elastic collisions . Unity treats any collisions with a relative linear velocity below this threshold as inelastic, so no bounce will occur. |
| Max Linear Correction | Set the maximum linear position correction used when solving constraints (from a range between 0.0001 to 1000000). This helps to prevent overshooting. |
| Max Angular Correction | Set the maximum angular correction used when solving constraints (from a range between 0.0001 to 1000000). This helps to prevent overshooting. |
| Max Translation Speed | Set the maximum linear speed of a Rigidbody 2D GameObject during any physics update. |
| Max Rotation Speed | Set the maximum rotation speed of a Rigidbody 2D GameObject during any physics update. |
| Baumgarte Scale | Set the scale factor that determines how fast Unity resolves collision overlaps. |
| Baumgarte Time of Impact Scale | Set the scale factor that determines how fast Unity resolves time-of-impact overlaps. |
| Time to Sleep | Set the time (in seconds) that must pass after a Rigidbody 2D stops moving before it goes to sleep. |
| Linear Sleep Tolerance | Set the linear speed below which a Rigidbody 2D goes to sleep after the Time to Sleep elapses. |
| Angular Sleep Tolerance | Set the rotational speed below which a Rigidbody 2D goes to sleep after Time to Sleep elapses. |
| Default Contact Offset | Set a proximity distance value for Colliders to be considered in contact, even when they aren't actually in contact. Colliders whose distance is less than the sum of their <code>contactOffset</code> values generate contacts, which causes the collision detection system to predictively enforce the contact constraint even when the objects are slightly separated. Caution: Reducing this value too far can slow down Unity's ability to calculate continuous polygon collisions. Conversely, increasing the value too much might create artifacts for vertex collisions. |
| Simulation Mode | Select when Unity executes the 2D physics simulation. |
| | Fixed Update Select this to have Unity execute the physics simulation immediately after MonoBehaviour.FixedUpdate is called. |
| | Update Select this to have Unity execute the physics simulation immediately after MonoBehaviour.Update is called. When you select this mode, additional properties are visible. |
| | Script Select this to manually execute the physics simulation via Physics2D.Simulate . |
| Simulation Layers | Select which layer(s) Unity simulates when Simulation Mode is set to FixedUpdate or Update . The Everything option is selected by default, which automatically selects all layers and includes all of them in the physics simulation. You can select which specific layer(s) to include in the simulation, and Unity will only simulate the Rigidbody2D , joints , effectors , and contacts between Collider2D on those selected layers. |

| Property | Function | |
|-----------------------------------|---|--|
| Queries Hit Triggers | Enable this option if you want Collider 2Ds marked as Triggers to return a hit when any physics query (such as Linecasts or Raycasts) intersects with them. Defaults to enabled. | |
| Queries Start In Colliders | Enable this option if you want physics queries that start inside a Collider 2D to detect the Collider they start in. | |
| Callbacks On Disable | Enable this option to produce collision callbacks when a Collider with contacts is disabled. | |
| Reuse Collision Callbacks | Enable this setting to have the physics system reuse a single Collision2D instance for all collision callbacks. Disable to have the physics system create a new Collision2D instance for each collision callback instead. | |
| Auto Sync Transforms | Enable this option to automatically sync transform changes with the physics system. | |
| Gizmos | Select the types of physics 2D gizmos to be drawn within the Editor. You may select multiple options. | |
| | Nothing | Select this to deselect every option. No physics 2D gizmo will be drawn. |
| | Everything | Select this to select every option. |
| | All Colliders | Select this to have all Colliders drawn without having to select them in the Hierarchy window. |
| | Colliders Outlined | Select this to have Colliders drawn with an outline (you can customize the outline's color in 2D Physics Preferences). |
| | Colliders Filled | Select this to have all Colliders drawn using the Fill color specified in the 2D Physics Preferences . |
| | Colliders Sleeping | Select this to have Colliders drawn to show when the Rigidbody 2D they are attached to is sleeping using the Awake or Asleep colors specified in the 2D Physics Preferences . |
| | Collider Contacts | Select this to have Collider contacts shown as a directional arrow that starts at the contact point in the direction of the contact normal. You can specify the Contact color of the arrow in the 2D Physics Preferences . |
| | Collider Bounds | Select this to have Collider bounds drawn for all PhysicsShape2D that a Collider creates. The bounds are an Axis-Aligned Bounding Box (AABB). |
| Multithreading | Expand this to adjust the multithreading settings. Refer to Multithreading for information about each property. | |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-Physics2DManager.html>

4.8.10 Ρυθμίσεις παίκτη

Player: Οι ρυθμίσεις παίκτη είναι αν όχι η πιο σημαντική ρύθμιση στο project settings, μια από τις πιο σημαντικές για την ανάπτυξη της εφαρμογής γιατί περιέχουν ρυθμίσεις που καθορίζουν τον τρόπο με τον οποίο η Unity κατασκευάζει και εμφανίζει την τελική εφαρμογή. Οι ρυθμίσεις του Player διαφέρουν ανάλογα με τις πλατφόρμες που έχουν εγκατασταθεί. Κάθε πλατφόρμα έχει τις δικές της ρυθμίσεις Player τις οποίες θα πρέπει ο χρήστης να ορίσει για κάθε έκδοση της εφαρμογής που θέλει να υλοποιήσει.



Εικόνα 65. Player Settings

Πίνακας 49. Γενικές Ρυθμίσεις Παίκτη που μοιράζονται όλες οι πλατφόρμες

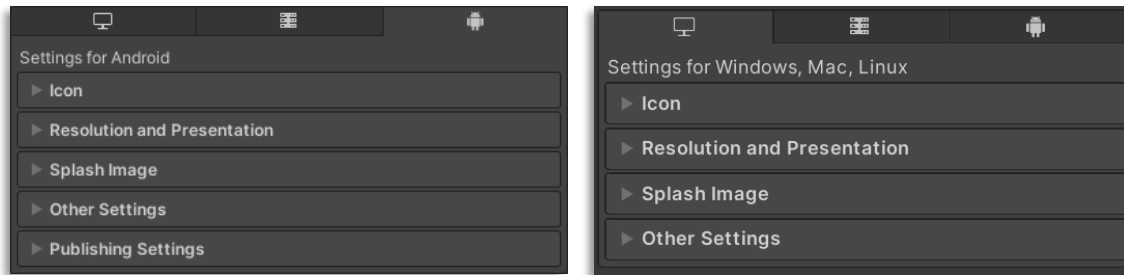
| Property | Function |
|-----------------------|--|
| Company Name | Enter the name of your company. Unity uses this to locate the preferences file. |
| Product Name | Enter the name that appears on the menu bar when your application is running. Unity also uses this to locate the preferences file. |
| Version | Enter the version number of your application. |
| Default Icon | Pick the Texture 2D file that you want to use as a default icon for the application on every platform. You can override this for specific platforms. |
| Default Cursor | Pick the Texture 2D file that you want to use as a default cursor for the application on every supported platform. |
| Cursor Hotspot | Set the pixel offset value from the top left of the default cursor to the location of the cursor hotspot. The cursor hotspot is the point in the cursor image that Unity uses to trigger events based on cursor position. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-PlayerSettings.html>

Πίνακας 50. Συγκεκριμένες Ρυθμίσεις Παίκτη ανάλογα την πλατφόρμα

| Property | Function |
|------------------------------------|--|
| Icon | The game icon(s) as shown on the desktop. You can choose icons from 2D image assets in the Project, such as sprites or imported images. |
| Resolution and Presentation | The settings for screen resolution and other presentation details such as whether the game should default to fullscreen mode. |
| Splash Image | The image shown while the game is launching. This section also includes common settings for creating a Splash Screen. For more information, refer to the Splash Image documentation. |
| Other Settings | Any remaining settings specific to the platform. |
| Publishing Settings | Details of how the built application is prepared for delivery from the app store or host webpage. |

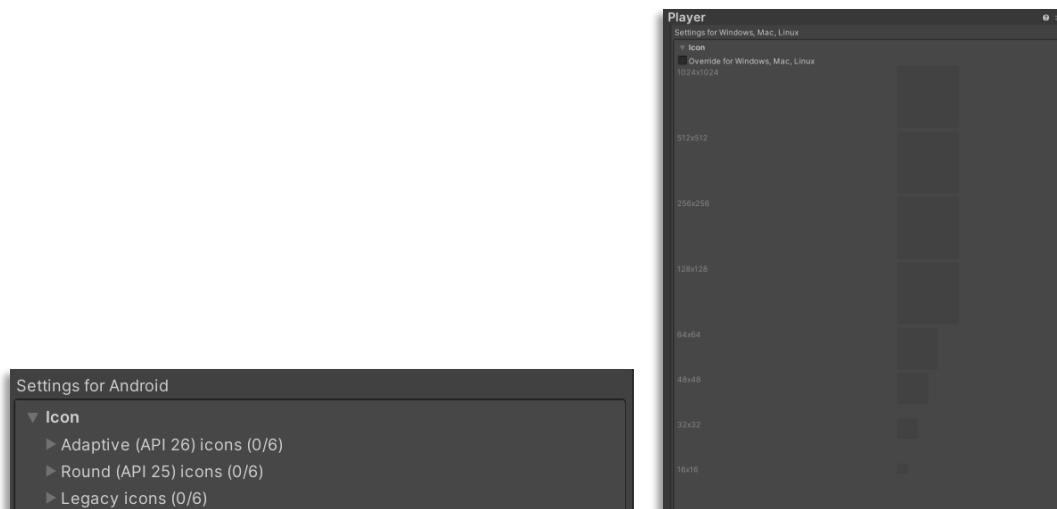
Εδώ θα γίνει μια ανάλυση μόνο των ρυθμίσεων για την πλατφόρμα Android και την πλατφόρμα Windows.



Εικόνα 66. Player Settings for Android and Windows

4.8.10.1 Icon

Στην ρύθμιση **Icon** υπάρχουν οι επιλογές για να καθοριστούν τα εικονίδια που θα αναπαριστούν την εφαρμογή στη συσκευή και είναι οι εξής:



Εικόνα 67. Ρυθμίσεις Icon Android και Windows

Ακολουθεί αναλυτική περιγραφή της κάθε μια ιδιότητας του Icon.

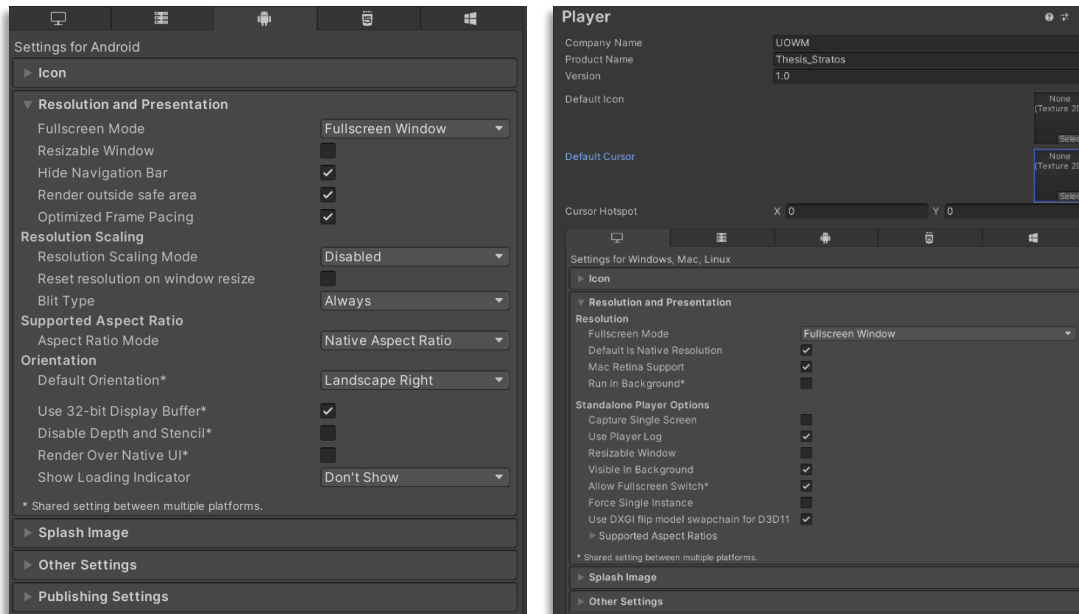
Πίνακας 51. Icon (Android)

| Setting | Description |
|-----------------|--|
| Adaptive | Specify textures of various sizes to represent your application on devices running Android 8.0 (API level 26) or higher. |
| Round | Specify textures of various sizes to represent your application on devices running Android 7.1 (API level 25) or higher. |
| Legacy | Specify textures of various sizes to represent your application on devices running versions earlier than Android 7.1 (API level 25). |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-PlayerSettingsAndroid.html>

4.8.10.2 Resolution And Presentation

Στην ρύθμιση **Resolution And Presentation** υπάρχουν οι επιλογές για παραμετροποίηση των πτυχών της εμφάνισης της οθόνης και είναι οι εξής:



Εικόνα 68. Ρυθμίσεις Resolution and Presentation Android και Windows

Ακολουθεί αναλυτική περιγραφή της κάθε μια ιδιότητας του Resolutions and Presentation.

Πίνακας 52. Resolution And Presentation (Android)

| Setting | Description |
|------------------------------|--|
| Fullscreen Mode | Specifies how the window appears on the device. This sets the default window mode at startup. |
| <i>Fullscreen Window</i> | The application window fills the full-screen native resolution of the device. To fill the full-screen window, Unity scales the application contents. Resolution Scaling Mode controls how Unity scales the contents. In this mode, the navigation bar is always hidden. |
| <i>Windowed</i> | The application uses a standard, non-full-screen, movable window. The size of the window depends on the application's resolution. In this mode, the window is resizable by default. To disable this, disable Resizable Window . |
| Resizable Window | Indicates whether the user can resize the application's window. This setting enables multi-window capabilities in your application on Android phones and tablets. For more information, see Google's developer documentation . |
| Default Window Width | The default width of the application window in pixels . This option is only available if the Fullscreen Mode is set to Windowed . |
| Default Window Height | The default height of the application window in pixels. This option is only available if the Fullscreen Mode is set to Windowed . |
| Minimum Window Width | The minimum width of the application window in pixels. This option is only available if the Fullscreen Mode is set to Windowed . |
| Minimum Window Height | The minimum height of the application window in pixels. This option is only available if the Fullscreen Mode is set to Windowed . |
| Hide Navigation Bar | Indicates whether to hide the navigation bar that appears at the top of the window. |

| | |
|---------------------------------|---|
| Render outside safe area | Indicates whether the application should use all available screen space to render, including areas of the display that are cut out (notched). For more information, see Android's display cutout support documentation. |
| Optimized Frame Pacing | Indicates whether Unity should evenly distribute frames for less variance in framerate. Enable this option to create a smoother experience. |
| Run In background | Enable this option to not pause the application when it loses focus. When disabled, the application pauses when it loses focus. |

Επίσης υπάρχουν στο παρών μενού και οι εξής ομαδοποιημένες ρυθμίσεις:

- ❖ **To resolution scaling** είναι χρήσιμο για την ρύθμιση της ανάλυσης της οθόνης του κινητού. Σε χαμηλότερη ανάλυση από την native του κινητού βελτιώνει πολύ την απόδοση στο gameplay και της μπαταρίας της συσκευής.

Πίνακας 53. Resolution Scaling (Android)

| Setting | Description |
|--|---|
| Resolution Scaling Mode | Specifies whether and how the application scales its resolution. You can set the scaling to be equal to or lower than the native screen resolution. Using a lower resolution can improve performance and battery life. |
| <i>Disabled</i> | Doesn't apply resolution scaling and the application renders to the device's native screen resolution. |
| <i>FixedDPI</i> | Applies resolution scaling using a target DPI. Use this to optimize performance and battery life or target a specific DPI setting. |
| <i>Letterboxed</i> | Adds black bars to the rendered output so the content doesn't stretch. This process is called letterboxing . |
| Target DPI | The resolution of the application. If the device's native screen DPI is higher than this value, Unity downscales the application's resolution to match this setting. To calculate the scale, Unity uses the following: $\min(\text{Target DPI} * \text{Factor} / \text{Screen DPI}, 1)$ Where Factor is the Resolution Scaling Fixed DPI Factor from Quality settings . Note: This option only appears when you set Resolution Scaling Mode to Fixed DPI . |
| Reset resolution on window resize | Indicates whether to set the screen resolution to the new native window size when the native window size changes. If you set Resolution Scaling Mode to Fixed DPI , Unity recalculates the resolution based on Fixed DPI property. |
| Blit Type | Controls whether to use a blit to render the final image to the screen. Using a blit is compatible with most devices but is usually slower than not using a blit. |
| <i>Always</i> | Unity renders to an offscreen buffer and then uses a blit to copy the contents of the buffer to the device's framebuffer. This is compatible with most devices but is usually slower than not using blit. |
| <i>Never</i> | Unity renders to the framebuffer provided by the device's operating system. If this fails, the application prints a one-time warning to the device log. This is usually faster than using blit, but it isn't compatible with all devices. |
| <i>Auto</i> | Unity renders to the framebuffer provided by the device's operating system if possible. If this fails, Unity prints a warning to the device console and uses a blit to render the final image to the screen. |

- ❖ Το **Supported Aspect Ratio** είναι υπεύθυνο για τις αναλογίες διαστάσεων της οθόνης που θα υποστηρίζονται από το κινητό.

Πίνακας 54. Supported Aspect Ratio (Android)

| Setting | Description |
|----------------------------------|--|
| Aspect Ratio Mode | Specifies the largest aspect ratio the application supports. If the device's aspect ratio is greater than this aspect ratio, Unity uses this aspect ratio for the application and adds black bars so the application doesn't stretch.. |
| <i>Legacy Wide Screen (1.86)</i> | The application supports aspect ratios up to Android's legacy wide-screen aspect ratio. |
| <i>Native Aspect Ratio</i> | The application supports aspect ratios up to Android's native aspect ratio. |
| <i>Custom</i> | The application supports aspect ratios up to the aspect ratio you set in Up To . |
| Minimum | The custom minimum aspect ratio. This value must be less than Maximum . This setting only appears when you set Aspect Ratio Mode to Custom . |
| Maximum | The custom maximum aspect ratio. This value must be greater than Minimum . This setting only appears when you set Aspect Ratio Mode to Custom . |

- ❖ Το **Orientation** αφορά ρυθμίσεις που έχουν να κάνουν με τον προσανατολισμό της εφαρμογής

Πίνακας 55. Orientation (Android)

| Setting | Description |
|-----------------------------|---|
| Default Orientation | Specifies the screen orientation the application uses. Note: Unity shares the value you set for this setting between Android and iOS. |
| <i>Portrait</i> | The application uses portrait screen orientation where the bottom of the application's window aligns with the bottom of the device's screen. |
| <i>Portrait Upside Down</i> | The application uses portrait screen orientation where the bottom of the application's window aligns with the top of the device's screen. |
| <i>Landscape Right</i> | The application uses landscape screen orientation where the right side of the application's window aligns with the bottom of the device's screen. |
| <i>Landscape Left</i> | The application uses landscape screen orientation where the right side of the application's window aligns with the top of the device's screen. |
| <i>Auto Rotation</i> | The screen can rotate to any of the orientations you specify in the Allowed Orientations for Auto Rotation section. |

- ❖ Η ενότητα <<**Allowed Orientations for Auto Rotation**>> για ποιους προσανατολισμούς υποστηρίζει η εφαρμογή όταν ο χρήστης ορίζει τον <<Προεπιλεγμένο προσανατολισμό>> σε <<Αυτόματη περιστροφή>>. Αυτό είναι χρήσιμο, για παράδειγμα, για να κλειδώσει την εφαρμογή σε οριζόντιο προσανατολισμό, αλλά να επιτρέψει στο χρήστη να εναλλάσσεται μεταξύ

αριστερού και δεξιού οριζόντιου προσανατολισμού. Αυτή η ενότητα εμφανίζεται μόνο όταν ορίζετε τον <<Προεπιλεγμένο προσανατολισμό>> σε <<Αυτόματη περιστροφή>>.

Πίνακας 56. Allowed Orientations for Auto Rotation (Android)

| Setting | Description |
|-----------------------------|---|
| Portrait | Indicates whether the application supports portrait screen orientation where the bottom of the application's window aligns with the bottom of the device's screen. |
| Portrait Upside Down | Indicates whether the application supports portrait screen orientation where the bottom of the application's window aligns with the top of the device's screen. |
| Landscape Right | Indicates whether the application supports landscape screen orientation where the right side of the application's window aligns with the bottom of the device's screen. |
| Landscape Left | Indicates whether the application supports landscape screen orientation where the right side of the application's window aligns with the top of the device's screen. |

❖ Και τέλος κάποιες γενικές ρυθμίσεις στη συγκεκριμένη καρτέλα όπως :

Πίνακας 57. General Settings (Android)

| Setting | Description | |
|----------------------------------|---|--|
| Use 32-bit Display Buffer | Indicates whether the display buffer holds 32-bit color values instead of 16-bit color values. Enable this setting if you see banding, or need alpha values in post-processing effects . Some post-processing effects require this because they create Render Textures in the same format as the display buffer. | |
| Disable Depth and Stencil | Indicates whether to disable depth and stencil buffers . | |
| Render Over Native UI | Indicates whether to render on top of native UI on Android or iOS. For this setting to take effect, set your Camera's Clear Flags to use a solid color with an alpha value lower than 1. | |
| Show Loading Indicator | Specifies if and how the loading indicator appears. | |
| <input type="checkbox"/> | <i>Don't Show</i> | The loading indicator doesn't appear. |
| <input type="checkbox"/> | <i>Large</i> | A large loading indicator appears. |
| <input type="checkbox"/> | <i>Inversed Large</i> | A large loading indicator appears with inversed color. |
| <input type="checkbox"/> | <i>Small</i> | A small loading indicator appears. |
| <input type="checkbox"/> | <i>Inversed Small</i> | A small loading indicator appears with inversed color. |

Πίνακας 58. Resolution And Presentation (Windows)

| Property | Description |
|--------------------------|---|
| Fullscreen Mode | Choose the full-screen mode. This defines the default window mode at startup. |
| Fullscreen Window | Set your app window to the full-screen native display resolution, covering the whole screen. This mode is also known as borderless full-screen. Unity renders the app content at the resolution set by a script, or the native display resolution if none is set and scales it to fill the window. When scaling, Unity adds black bars to the rendered output to match the display aspect ratio to prevent content stretching. This process is called letterboxing . The OS overlay UI displays on top of the full-screen window (such as IME input windows). All platforms support this mode. |

| Property | Description |
|--|--|
| Exclusive Fullscreen (Windows only) | Set your app to maintain sole full-screen use of a display. Unlike <i>Fullscreen Window</i> , this mode changes the OS resolution of the display to match the app's chosen resolution. This option is only supported on Windows. |
| Maximized Window (Mac only) | Set the app window to the operating system's definition of maximized , which is typically a full-screen window with a hidden menu bar and dock on macOS. This option is only supported on macOS. Fullscreen Window is the default setting for other platforms. |
| Windowed | Set your app to a standard, non-full-screen movable window, the size of which is dependent on the app resolution. In this mode, the window is resizable by default. Use the Resizable Window setting to disable this. All desktop platforms support this full-screen mode. |
| Default Is Native Resolution | Enable this option to make the game use the default resolution used on the target machine. This option isn't available if the Fullscreen Mode is set to <i>Windowed</i> . |
| Default Screen Width | Set the default width of the game screen in pixels . This option is only available if the Fullscreen Mode is set to <i>Windowed</i> . |
| Default Screen Height | Set the default height of the game screen in pixels. This option is only available if the Fullscreen Mode is set to <i>Windowed</i> . |
| Mac Retina Support | Enable this option to enable support for high DPI (Retina) screens on a Mac. Unity enables this by default. This enhances Projects on a Retina display, but it's somewhat resource-intensive when active. |
| Run In background | Enable this option to have the game running in the background instead of pausing if the app loses focus. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/playersettings-windows.html>

Πίνακας 59. Standalone Player Options (Windows)


| Property | Function |
|---|--|
| Capture Single Screen | Enable this option to ensure standalone games in Fullscreen Mode do not darken the secondary monitor in multi-monitor setups. |
| Use Player Log | Enable this option to write a log file with debugging information. Defaults to enabled. |
| Resizable Window | Enable this option to allow resizing of the desktop player window. Note: If you disable this option, your application can't use the <i>Windowed Fullscreen Mode</i> . |
| Visible in Background | Enable this option to display the application in the background if <i>Windowed Fullscreen Mode</i> is used. Note: This will prevent keyboard shortcuts that minimize the app window, for example, Alt+Tab and Windows+M, from functioning. See PlayerSettings.visibleInBackground for more details. |
| Allow Fullscreen Switch | Enable this option to allow default OS full-screen key presses to toggle between full-screen and windowed modes. |
| Force Single Instance | Enable this option to restrict desktop players to a single concurrent running instance. |
| Use DXGI flip model swap chain for D3D11 | Using the flip model ensures the best performance. This setting affects the D3D11 graphics API. Disable this option to fall back to the Windows 7-style BitBlt model. For more information, see PlayerSettings.useFlipModelSwapchain . |

4.8.10.3 Splash Image

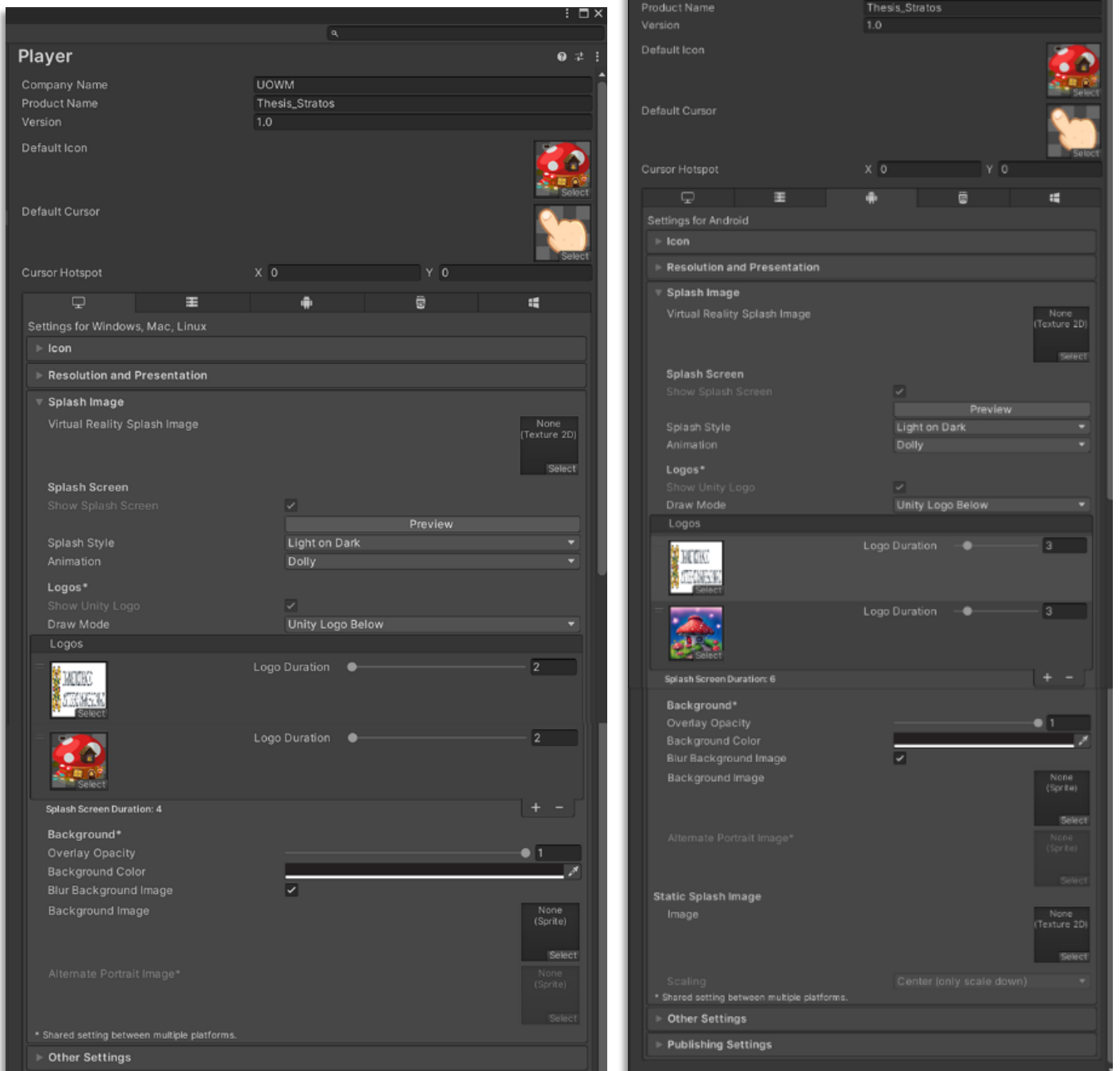
Ακολουθεί η καρτέλα **Splash Image** όπου ο χρήστης έχει την δυνατότητα να επιλέξει μια προσαρμοσμένη εικόνα προβολής (κάτι σαν λογότυπο) για τις οθόνες Virtual

Reality και όχι μόνο. Η οθόνη εκκίνησης εμφανίζεται κατά την εκκίνηση της εφαρμογής. Μέσω των ρυθμίσεων στο παράθυρο Splash Image, ο χρήστης μπορεί να προσαρμόσει τον τρόπο που η οθόνη εκκίνησης εμφανίζεται στην εφαρμογή του, όπως η απενεργοποίηση του λογότυπου της Unity, η προσθήκη δικών του λογότυπων ή πλήρη απενεργοποίηση της οθόνης εκκίνησης. Ωστόσο, η πλήρης προσαρμογή της οθόνης εκκίνησης είναι διαθέσιμη μόνο για τους κατόχους προγραμμάτων Unity Plus, Pro ή Enterprise. Εάν χρησιμοποιεί την προσωπική έκδοση του Unity, υπάρχουν περιορισμοί σχετικά με τις ρυθμίσεις της εικόνας εκκίνησης:

- Δεν είναι δυνατή η απενεργοποίηση της ρύθμισης **Show Splash Screen** για να απενεργοποιηθεί στην οθόνη εκκίνησης.
- Δεν είναι δυνατή η απενεργοποίηση της ρύθμισης **Show Unity Logo** για να απενεργοποιηθεί το λογότυπο Unity.
- Η ρύθμιση **Overlay Opacity** έχει ελάχιστη τιμή 0,5.

| | |
|---|--|
|  | <p>Εδώ να αναφερθεί πως αν ο χρήστης στοχεύει να δημιουργήσει μια εφαρμογή για κινητές συσκευές Android τότε θα πρέπει υποχρεωτικά να βάλει Splash Screen για το build της εφαρμογής ή να κρατήσει αυτό που είδη έχει η ίδια η εταιρία. Χωρίς την splash screen η εφαρμογή δεν θα δουλέψει. Εάν ο χρήστης έχει την Unity Plus, Pro, or Enterprise έκδοση και δεν έχει την επιλογή για απενεργοποίηση της splash screen θα πρέπει ίσως να ξανά ενεργοποιήσει την άδειά του.</p> |
|---|--|

Η οθόνη εκκίνησης της Unity είναι πανομοιότυπη σε όλες τις πλατφόρμες και εμφανίζεται αμέσως καθώς η πρώτη σκηνή της εφαρμογής φορτώνεται ασύγχρονα στο παρασκήνιο. Ωστόσο, όταν ο χρήστης χρησιμοποιεί δικές του εισαγωγικές οθόνες ή κινούμενα σχέδια, μπορεί να χρειαστεί περισσότερος χρόνος για να εμφανιστεί, καθώς η Unity πρέπει να φορτώσει ολόκληρο την μηχανή καθώς και την πρώτη σκηνή πριν εμφανίσει την προσαρμοσμένη οθόνη εκκίνησης. Υπάρχει επίσης δυνατότητα του ελέγχου των ρυθμίσεων splash screen μέσω του [SplashScreen API](#).



Εικόνα 69. Ρυθμίσεις Splash Image Android και Windows

Αναλυτικά οι ιδιότητες της κάθε επιλογής.

Πίνακας 60. Splash Image

| Setting | | Description |
|------------------------------------|----------------------------------|--|
| Virtual Reality Splash Image | | Set an image to use for applications that use virtual reality. The image must be a 2D Texture . |
| Splash Screen | | Settings for displaying the splash screen |
| Show Splash Screen | | Enabled by default. Disable this setting to not display a splash screen at the start of your application. Important: If you have a Unity Personal plan you can't disable this setting. |
| Preview | | Select the Preview button to see a preview of the Splash Screen in the Game view. The preview reflects the resolution and aspect ratio of the Game view. |
| Splash Style | | Settings for the color style of the Unity logo. |
| | Light on Dark | A light version of the Unity logo best suited to display on dark colored backgrounds. This is the default setting. |
| | Dark on Light | A dark version of the Unity logo best suited to display on light colored backgrounds. |
| Animation | | Settings for how the splash screen appears and disappears on screen. |
| | Static | Doesn't apply any animation. |
| | Dolly | The logo and background zoom to create a visual dolly effect. This is the default setting. |
| | Custom | Choose this setting to configure the background and logo zoom amounts to modify the dolly effect. |
| | Logo Zoom | The target zoom (from 0 to 1) for the logo when it reaches the end of the logo animation's total duration. |
| | Background Zoom | The target zoom (from 0 to 1) for the background when it reaches the end of the splash screen animation's total duration. |
| Logos | | Settings to customize the logos in your splash screen |
| Show Unity Logo | | Enabled by default. Disable this option to not display the Unity logo on the splash screen. Important: If you have a Unity Personal plan you can't disable this setting. |
| Draw Mode | | Select the mode for the order that the logos appear on the splash screen |
| | Unity Logo Below | Displays the Unity logo underneath all logos in the Logos list. |
| | All Sequential | Displays the Unity logo sequentially with all logos in the Logos list. |
| Logos | | The logos to include on the splash screen. Each logo must be a Sprite Asset . Set the Sprite Mode to Single to use the entire sprite as the logo. If you want to change the logo's aspect ratio, set the Sprite Mode to Multiple and edit the dimensions with the Sprite Editor . To add and remove logos, use the plus (+) and minus (-) buttons. To reorder logos, drag them in the list. To create a delay between logos, add an entry to the Logos list with no Sprite Asset assigned. |
| | Logo Duration | Set the length of time each Sprite Asset appears on the screen. Use a value between 2 and 10 seconds. |
| Splash Screen Duration | | Displays the total duration of the splash screen sequence, in seconds. This is the total of all logos plus 0.5 seconds for fading out. Note: Your application's splash screen might display for longer if the first Scene isn't ready to play. In this case, the splash screen shows only the background image or color and then fades out when the first Scene is ready to play. |

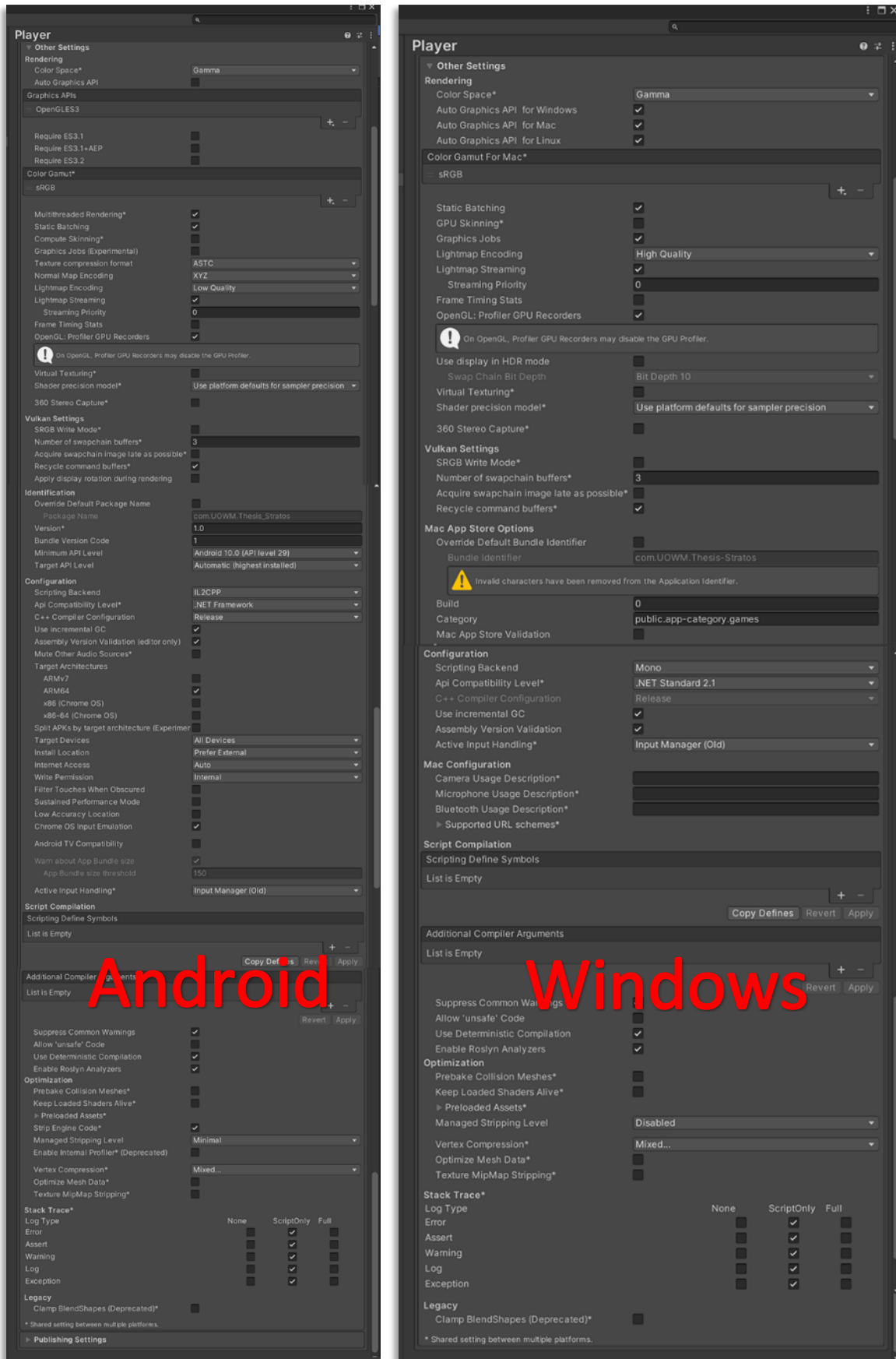
| Setting | Description |
|--|--|
| Background | Settings to customize the background of your application's splash screen. |
| Overlay Opacity | Adjust the value of the Overlay Opacity setting to make the logos stand out. This affects the background color and image color of the logo, based on how you set the Splash Style (Light on Dark, or Dark on Light). Set the opacity to a lower value to reduce this effect. You can also disable the effect by setting it to 0. For example, if the Splash Style is Light on Dark , with a white background, the background becomes gray if you set the Overlay Opacity to 1, and white if set to 0. Important: If you have a Unity Personal plan, this setting has a minimum value of 0.5. |
| Background Color | Sets a color for the splash screen. Unity uses this color if you don't set a Background Image . Note: The Overlay Opacity setting has an effect on the display of the Background Color, and might not match the assigned color. |
| Blur Background Image | Enable this setting to blur the Background Image you set. If you disable this setting, it displays the Background Image with no blur effect. |
| Background Image | Set a reference to a Sprite image to use as a background. Unity adjusts the background image so that it fills the screen. It uniformly scales the image until it fits both the width and height of the screen. This means that parts of the image might extend beyond the screen edges in some aspect ratios. To adjust the background image's response to aspect ratio, change the Sprite's Position values in the Sprite Editor . |
| Alternate Portrait Image | Set an alternative image to use with portrait aspect ratios (for example, a mobile device in portrait mode). If you don't assign an Alternate Portrait Image Sprite, the Unity Editor uses the Sprite assigned as the Background Image for both portrait and landscape mode. You can adjust the Position and dimensions of the Sprite in the Sprite Editor to control the aspect ratio and position of the background image on the Splash Screen. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-PlayerSettingsSplashScreen.html>

Τέλος στο splash Image settings υπάρχουν δυο ακόμα επιλογές. Αυτές είναι :

Πίνακας 61. Static Splash Image And Scaling

| Setting | Description |
|------------------------------------|--|
| Image | Specifies the texture that the application uses for the Android splash screen. The standard size for the splash screen image is 320x480. |
| Scaling | Specifies how to scale the splash image to fit the device's screen. |
| <i>Center (only scale down)</i> | Draws the image at its native size unless it's too large, in which case Unity scales the image down to fit. |
| <i>Scale to Fit (letter-boxed)</i> | Scales the image so that the longer dimension fits the screen size exactly. Unity fills in the empty space around the sides in the shorter dimension in black. |
| <i>Scale to Fill (cropped)</i> | Scales the image so that the shorter dimension fits the screen size exactly. Unity crops the image in the longer dimension. |



Εικόνα 70. Other Settings

4.8.10.4 Other Settings

Η άμεσος επόμενη καρτέλα είναι η **Other Settings**. Αυτή η καρτέλα επιτρέπει τον χρήστη να παραμετροποιήσει διάφορες ρυθμίσεις και είναι ομαδοποιημένη με τους εξής τρόπους :

Πίνακας 62. Rendering (Android)

| Property | Description |
|-------------------------------------|---|
| Color Space | Choose which color space to use for rendering. For more information, refer to Linear rendering overview . |
| Gamma | Gamma color space is typically used for calculating lighting on older hardware restricted to 8 bits per channel for the framebuffer format. Even though monitors today are digital, they might still take a gamma-encoded signal as input. |
| Linear | Linear color space rendering gives more precise results. When you select to work in linear color space, the Editor defaults to using sRGB sampling. If your Textures are in linear color space, you need to work in linear color space and deactivate sRGB sampling for each Texture. |
| Auto Graphics API | Disable this option to manually pick and reorder the graphics APIs. By default, this option is enabled, and Unity tries to use Vulkan. If the device doesn't support Vulkan, Unity falls back to GLES3.2, GLES3.1 or GLES3.0. Important: Unity adds the GLES3/GLES3.1/AEP/3.2 requirement to your Android App Manifest only if GLES2 is not in the list of APIs when Auto Graphics API is disabled. In this case only, your application does not appear on unsupported devices in the Google Play Store. |
| Graphics APIs | Specifies which graphics APIs the application supports. Unity tries to use the graphics API at the top of the list first and, if the device doesn't support it, Unity falls back to the next graphics APIs in the list. This property is visible only when Auto Graphics API is set to false. |
| Vulkan | Adds Vulkan as an API to target. |
| OpenGLES3 | Adds OpenGLES 3 as an API to target. |
| Require ES3.1 | Indicates whether to require that the minimum OpenGL ES 3 minor version is 3.1. This property is visible only if you enable Auto Graphics API or if Graphics APIs includes OpenGLES3 . |
| Require ES3.1+AEP | Indicates whether to require that the minimum OpenGL ES 3 minor version is 3.1+AEP. This property is visible only if you enable Auto Graphics API or if Graphics APIs includes OpenGLES3 . |
| Require ES3.2 | Indicates whether to require that the minimum OpenGL ES 3 minor version is 3.2. This property is visible only if you enable Auto Graphics API or if Graphics APIs includes OpenGLES3 . |
| Color Gamut | You can add or remove color gamuts to use for rendering. Click the plus (+) icon to see a list of available gamuts. A color gamut defines a possible range of colors available for a given device (such as a monitor or screen). The sRGB gamut is the default (and required) gamut. |
| Multithreaded Rendering | Enable this option to use multithreaded rendering. This is only supported on Metal. |
| Static Batching | Enable this option to use static batching . |
| Dynamic Batching | Check this box to use Dynamic Batching on your build (enabled by default). |
| Compute Skinning | Enable this option to use DX11/DX12/ES3 GPU compute skinning, which frees up CPU resources. |
| Graphics Jobs (Experimental) | Enable this option to instruct Unity to offload graphics tasks (render loops) to worker threads running on other CPU cores. This is intended to reduce the time spent in Camera.Render on the main thread, which is often a bottleneck. Note: This feature is experimental. It may not deliver a performance improvement for your project, and may introduce new crashes. |
| Texture | Choose between ASTC, ETC2 and ETC (ETC1 for RGB, ETC2 for RGBA), DXTC and DXT + RGTC(BC4, BC5). |

| Property | Description |
|-------------------------------------|--|
| compression formats | Refer to texture compression format overview for more information on how to pick the right format. It's possible to add multiple texture compression formats to this list for texture compression targeting support, however you can't have both DXT and DXT + RGTC(BC4, BC5) in this list at the same time. If you enable texture compression targeting, Unity also enables, Split Application Binary . See Texture compression settings for more details on how this interacts with the texture compression setting in the Build Settings . Note: If you export or build your application as an APK, Unity only uses the first texture compression format in this list. |
| Normal Map Encoding | Choose <i>XYZ</i> or <i>DXT5nm-style</i> to set the normal map encoding. This setting affects the encoding scheme and compression format used for normal maps. <i>DXT5nm-style</i> normal maps are of higher quality, but more expensive to decode in shaders. |
| Lightmap Encoding | Defines the encoding scheme and compression format of the lightmaps. You can choose from Low Quality , Normal Quality , or High Quality |
| HDR Cubemap Encoding | Defines the encoding scheme and compression format of the HDR Cubemaps. You can choose from Low Quality , Normal Quality , or High Quality . For more information, refer to Lightmaps: Technical information . |
| Lightmap Streaming | Enable this option to use Mipmap Streaming for lightmaps. Unity applies this setting to all lightmaps when it generates them. Note: To use this setting, you must enable the Texture Streaming Quality setting. |
| Streaming Priority | Sets the priority for all lightmaps in the Mipmap Streaming system . Unity applies this setting to all lightmaps when it generates them. Positive numbers give higher priority. Valid values range from -128 to 127. |
| Frame Timing Stats | Enable this option to gather CPU/GPU frame timing statistics. |
| Virtual Texturing | Indicates whether to enable Virtual Texturing . Note: Virtual Texturing isn't compatible with Android. |
| Shader precision model | Controls the default precision of samplers used in shaders. For more information, refer to Shader data types and precision . |
| 360 Stereo Capture | Indicates whether Unity can capture stereoscopic 360 images and videos. For more information, see Stereo 360 Image and Video Capture . Note: 360 stereoscopic capturing isn't compatible with Android. |
| Load/Store Action Debug Mode | Highlights undefined pixels that may cause rendering problems on mobile platforms. This affects the Unity Editor Game view, and your built application if you select Development Build in Build Settings. See LoadStoreActionDebugModeSettings . |
| Editor Only | Indicates whether the Load/Store Action Debug Mode only runs in the Unity Editor. This property is visible only when Load/Store Action Debug Mode is set to true. |

Ρυθμίσεις για την απόδοση γραφικών με το Vulkan API.

Πίνακας 63. Vulkan Settings (Android)

| Property | Description |
|------------------------------------|---|
| sRGB Write Mode | Enable this option to allow Graphics.SetSRGBWrite() renderer to toggle the sRGB write mode during runtime. That is, if you want to temporarily turn off Linear-to-sRGB write color conversion, you can use this property to achieve that. Enabling this has a negative impact on performance on mobile tile-based GPUs; therefore, do NOT enable this for mobile. |
| Number of swapchain buffers | Set this option to 2 for double-buffering, or 3 for triple-buffering to use with Vulkan renderer. This setting may help with latency on some platforms, but in most cases you should not change this from the default value of 3. Double-buffering might have a negative impact on performance. Do not use this setting on Android. |
| Acquire swapchain | If enabled, Vulkan delays acquiring the backbuffer until after it renders the frame to an offscreen |

| Property | Description |
|---|---|
| image late as possible | image. Vulkan uses a staging image to achieve this. Enabling this setting causes an extra blit when presenting the backbuffer. This setting, in combination with double-buffering, can improve performance. However, it also can cause performance issues because the additional blit takes up bandwidth. |
| Recycle command buffers | Indicates whether to recycle or free CommandBuffers after Unity executes them. |
| Apply display rotation during rendering | Enable this to perform all rendering in the native orientation of the display. This has a performance benefit on many devices. For more information, see documentation on Framebuffer orientation . |

Πίνακας 64. Identification (Android)

| Property | Function |
|-------------------------------|--|
| Override Default Package Name | Indicates whether to override the default package name for your application. Note: This setting affects macOS, iOS, tvOS, and Android. |
| Package Name | Set the application ID, which uniquely identifies your app on the device and in Google Play Store. The application ID must follow the convention com.YourCompanyName.YourProductName and must contain only alphanumeric and underscore characters. Each segment must start with an alphabetical character. For more information, see Set the application ID . Important: Unity automatically removes any invalid characters you type. To set this property, enable Override Default Package Name . |
| Version | Enter the build version number of the bundle, which identifies an iteration (released or unreleased) of the bundle. The version is specified in the common format of a string containing numbers separated by dots (eg, 4.3.2). (Shared between iOS and Android.) |
| Bundle Version Code | An internal version number. This number is used only to determine whether one version is more recent than another, with higher numbers indicating more recent versions. This is not the version number shown to users; that number is set by the versionName attribute. The value must be set as an integer, such as "100". You can define it however you want, as long as each successive version has a higher number. For example, it could be a build number. Or you could translate a version number in "x.y" format to an integer by encoding the "x" and "y" separately in the lower and upper 16 bits. Or you could simply increase the number by one each time a new version is released. Keep this number under 100000 if Split APKs by target architecture is enabled. Each APK must have a unique version code so Unity adds 100000 to the number for ARMv7, and 200000 for ARM64. |
| Minimum API Level | Minimum Android version (API level) required to run the application. |
| Target API Level | Target Android version (API level) against which to compile the application. |

Πίνακας 65. Configuration (Android)

| Property | Description |
|-------------------------|---|
| Scripting Backend | Choose the scripting backend you want to use. The scripting backend determines how Unity compiles and executes C# code in your Project. |
| Mono | Compiles C# code into .NET Common Intermediate Language (CIL) and executes that CIL using a Common Language Runtime. For more information, refer to Mono . |
| IL2CPP | Compiles C# code into CIL, converts the CIL to C++ and then compiles that C++ into native machine code, which executes directly at runtime. For more information, refer to IL2CPP . |
| API Compatibility Level | Choose which .NET APIs you can use in your project. This setting can affect compatibility with third-party libraries. However, it has no effect on Editor-specific code (code in an Editor directory, or within an Editor-specific Assembly Definition). Tip: If you are having problems with a third-party assembly, you can try the suggestion in the API Compatibility Level section below. |

| Property | Description |
|--|---|
| <input type="checkbox"/> .Net Framework | Compatible with the .NET Framework 4 (which includes everything in the .NET Standard 2.0 profile plus additional APIs). Choose this option when using libraries that access APIs not included in .NET Standard 2.0. Produces larger builds and any additional APIs available aren't necessarily supported on all platforms. Refer to Referencing additional class library assemblies for more information. |
| <input type="checkbox"/> .Net Standard 2.1 | Produces smaller builds and has full cross-platform support. |
| Editor Assemblies Compatibility Level | Select which .NET APIs to use in your Editor assemblies. |
| <input type="checkbox"/> .NET Framework | Compatible with the .NET Framework 4 (which includes everything in the .NET Standard 2.1 profile plus additional APIs). Choose this option when using libraries that access APIs not included in .NET Standard 2.1. Produces larger builds and any additional APIs available aren't necessarily supported on all platforms. Refer to Referencing additional class library assemblies for more information. |
| <input type="checkbox"/> .NET Standard | Compatible with .NET Standard 2.1 . Produces smaller builds and has full cross-platform support. |
| IL2CPP Code Generation | Defines how Unity manages IL2CPP code generation. Note: To use this, set Scripting Backend to IL2CPP . |
| C++ Compiler Configuration | Choose the C++ compiler configuration used when compiling IL2CPP generated code. |
| <input type="checkbox"/> Debug | Debug configuration turns off all optimizations, which makes the code quicker to build but slower to run. |
| <input type="checkbox"/> Release | Release configuration enables optimizations, so the compiled code runs faster and the binary size is smaller but it takes longer to compile. |
| <input type="checkbox"/> Master | Master configuration enables all possible optimizations, squeezing every bit of performance possible. For instance, on platforms that use the MSVC++ compiler, this option enables link-time code generation. Compiling code using this configuration can take significantly longer than it does using the Release configuration. Unity recommends building the shipping version of your game using the Master configuration if the increase in build time is acceptable. |
| Use incremental GC | Uses the incremental garbage collector, which spreads garbage collection over several frames to reduce garbage collection-related spikes in frame duration. For more information, refer to Automatic Memory Management . |
| Allow downloads over HTTP | Indicates whether to allow downloading content over HTTP. The default option is Not allowed due to the recommended protocol being HTTPS, which is more secure. |
| <input type="checkbox"/> Not Allowed | Never allow downloads over HTTP. |
| <input type="checkbox"/> Allowed in Development Builds | Only allow downloads over HTTP in development builds. |
| <input type="checkbox"/> Always Allowed | Allow downloads over HTTP in development and release builds. |
| Mute Other Audio Sources | Indicates whether your Unity application should stop audio from applications running in the background. Otherwise, Audio from background applications continues to play alongside your Unity application. |
| Target Architectures | Specifies which architecture to target. |
| ARMv7 | Enable support for ARMv7 architecture. |
| ARM64 | Enable support for ARM64 architecture. |
| x86 (ChromeOS) | Enable support for x86 architecture. |
| x86-64 (ChromeOS and Magic Leap 2) | Enable support for x86-64 architecture. |
| Enable Armv9 Security Features for Arm64 | Enable Pointer Authentication (PAuth, PAC) and Branch Target Identification (BTI) for Arm64 builds. This property is visible only when Arm64 is set to true. |
| Split APKs by target architecture | Enable this option to create a separate APK for each CPU architecture selected in Target Architectures . This makes download size smaller for Google Play Store users. This is primarily a Google Play store feature and may not work in other stores. For more details, refer to Multiple APK Support . |

| Property | Description |
|--|--|
| Target Devices | Specifies the target devices on which the APK is allowed to run. |
| <input type="checkbox"/> All Devices | The APK is allowed to run on all Android and ChromeOS devices. |
| <input type="checkbox"/> Phones, Tablets, and TV Devices Only | The APK is allowed to run on Android phones, tablets, and TVs, but not on ChromeOS devices. |
| <input type="checkbox"/> ChromeOS Devices Only | The APK is allowed to run on ChromeOS devices, but not on Android phones or tablets. |
| Install Location | Specifies application install location on the device (for detailed information, refer to Android Developer documentation on install locations). |
| <input type="checkbox"/> Automatic | Let the operating system decide. User will be able to move the app back and forth. |
| <input type="checkbox"/> Prefer External | Install the application to external storage (SD card) if possible. The operating system doesn't guarantee it; if not possible, the app will be installed to internal memory. |
| <input type="checkbox"/> Force Internal | Force the application to be installed to internal memory. The user will be unable to move the app to external storage. |
| Internet Access | Choose whether to always add the networking (INTERNET) permission to the Android App Manifest , even if you aren't using any networking APIs. Set to <i>Require</i> by default for development builds. |
| <input type="checkbox"/> Auto | Only add the internet access permission if you are using a networking API. |
| <input type="checkbox"/> Require | Always add the internet access permission. |
| Write Permission | Choose whether to enable write access to the external storage (such as the SD card) and add a corresponding permission to the Android App Manifest. Set to External(SDCard) by default for development builds. |
| <input type="checkbox"/> Internal | Only grant write permission to internal storage. |
| <input type="checkbox"/> External(SDCard) | Enable write permission to external storage. |
| Filter Touches When Obscured | Enable this option to discard touches received when another visible window is covering the Unity application. This is to prevent tapjacking. |
| Sustained Performance Mode | Enable this option to set a predictable and consistent level of device performance over longer periods of time, without thermal throttling. Overall performance might be lower when this setting is enabled. Based on the Android Sustained Performance API . |
| Maximum Java Heap Size | Set the maximum Java heap size to use for building (in megabytes). Defaults to 4096. |
| Low Accuracy Location | Enable this option to use low accuracy values with Android location APIs instead. |
| ChromeOS Input Emulation | ChromeOS's default behaviour is to convert mouse and touchpad input events into touchscreen input events. Un-check this setting to disable the default behavior. |
| Android TV Compatibility | Enable this option to mark the application as Android TV compatible. |
| Android Game | Enable this option to mark the output package (APK) as a game rather than a regular application. This property is visible only when Android TV Compatibility is set to true. |
| Android Gamepad Support Level | Choose the level of support your application offers for a gamepad. This property is visible only when Android TV Compatibility is set to true. |
| <input type="checkbox"/> Works with D-Pad | The application is fully operational with a D-pad. No gamepad is needed. |
| <input type="checkbox"/> Supports Gamepad | The application works with a gamepad, but does not require it. |
| <input type="checkbox"/> Requires Gamepad | The application requires a gamepad to use. |
| Warn about App Bundle size | Enable this option to receive a warning when the size of the Android App Bundle exceeds a certain threshold. This option is selected by default and you can only configure it if you enable the Build App Bundle (Google Play) option in the Build settings . |
| App Bundle size threshold | Enter a size in Mb. When your App Bundle exceeds this size, Unity will display a warning. |
| Application Entry Point | The application entry points to generate for your application. For more information, refer to Android application entry points . |
| Activity | Generate an Activity application entry point . |
| GameActivity | Generate a GameActivity application entry point . |
| Active Input Handling | Choose how to handle input from users. |

| Property | Description |
|-----------------------------------|--|
| Input Manager (Old) | Uses the traditional Input settings. |
| Input System Package (New) | Uses the Input system. This option requires you to install the InputSystem package . |
| Both | Use both systems. |

Οι επόμενες ρυθμίσεις ελέγχουν ποσό μνήμη θα πιάσουν οι shader κατά το χρόνο εκτέλεσης.

Πίνακας 66. Shader Variant Loading (Android)

| Setting | Description |
|--------------------------------|---|
| Default chunk size (MB) | Sets the maximum size of compressed shader variant data chunks Unity stores in your built application for all platforms. The default is 16. Refer to Shader loading for more information. |
| Default chunk count | Sets the default limit on how many decompressed chunks Unity keeps in memory on all platforms. The default is 0, which means there's no limit. |
| Override | Enables overriding Default chunk size and Default chunk count for this build target. |
| Chunk size (MB) | Overrides the value of Default chunk size (MB) on this build target. |
| Chunk count | Overrides the value of Default chunk count on this build target. |

Ρυθμίσεις οι οποίες επιτρέπουν το χρήστη να συμπεριλάβει ή να αποκλείσει επιλεκτικά κώδικα από τη μεταγλώττιση, ανάλογα με το αν ορισμένα σύμβολα σεναρίων έχουν οριστεί ή όχι.

Πίνακας 67. Script Compilation (Android)

| Property | Description |
|--------------------------------------|--|
| Scripting Define Symbols | Sets custom compilation flags. For more details, see Platform dependent compilation . |
| Additional Compiler Arguments | Adds entries to this list to pass additional arguments to the Roslyn compiler. Use one new entry for each additional argument. To create a new entry, click Add (+) . To remove an entry, click Remove (-) . When you have added all desired arguments, click Apply to include your additional arguments in future compilations. Click Revert to reset this list to the most recent applied state. |
| Suppress Common Warnings | Indicates whether to display the C# warnings CS0169 and CS0649 . |
| Allow 'unsafe' Code | Enables support for compiling 'unsafe' C# code in a pre-defined assembly (for example, Assembly-CSharp.dll). For Assembly Definition Files (.asmdef), click on one of your .asmdef files and enable the option in the Inspector window that appears. |
| Use Deterministic Compilation | Indicates whether to prevent compilation with the <code>-deterministic</code> C# flag. With this setting enabled, compiled assemblies are byte-for-byte identical each time they are compiled. For more information, see Microsoft's deterministic compiler option . |
| Enable Roslyn Analyzers | Indicates whether to compile user-written scripts without Roslyn analyzer DLLs that might be present in your project. |

Ρυθμίσεις για συμπίεση τελικού μεγέθους της εφαρμογής.

Πίνακας 68. Optimization (Android)

| Property | Description |
|---------------------------------------|---|
| Prebake Collision Meshes | Adds collision data to Meshes at build time. |
| Preloaded Assets | Sets an array of Assets for the player to load on startup. To add new Assets, increase the value of the Size property and then set a reference to the Asset to load in the new Element box that appears. |
| Strip Engine Code | Enable this option if you want the Unity Linker tool to remove code for Unity Engine features that your Project doesn't use. This setting is only available with the IL2CPP scripting backend . Most apps do not use every available DLL. This option strips out DLLs that your app doesn't use to reduce the size of the built Player. If your app is using one or more classes that would normally be stripped out under your current settings, Unity displays a debug message when you try to build the app. |
| Managed Stripping Level | Chooses how aggressively Unity strips unused managed (C#) code. When Unity builds your app, the Unity Linker process can strip unused code from the managed DLLs your Project uses. Stripping code can make the resulting executable smaller, but can sometimes remove code that's in use. For more information about these options and bytecode stripping with IL2CPP, refer to ManagedStrippingLevel . |
| <input type="checkbox"/> Minimal | Use this to strip class libraries, UnityEngine, Windows Runtime assemblies, and copy all other assemblies. |
| <input type="checkbox"/> Low | Remove unreachable managed code to reduce build size and Mono/IL2CPP build times. |
| <input type="checkbox"/> Medium | Run UnityLinker to reduce code size beyond what Low can achieve. You might need to support a custom link.xml file, and some reflection code paths might not behave the same. |
| <input type="checkbox"/> High | UnityLinker will strip as much code as possible. This will further reduce code size beyond what Medium can achieve but managed code debugging of some methods might no longer work. You might need to support a custom link.xml file, and some reflection code paths might not behave the same. |
| Enable Internal Profiler (Deprecated) | This feature is deprecated and will be retired in a future version of Unity. Use the Profiler window instead (menu: Window > Analytics > Profiler). The Profiler collects application performance data and prints a report to the console. The report contains the number of milliseconds each Unity subsystem takes to execute on each frame, averaged across 30 frames. |
| Vertex Compression | Sets vertex compression per channel. This affects all the meshes in your project. Typically, Vertex Compression is used to reduce the size of mesh data in memory, reduce file size, and improve GPU performance. For more information on how to configure vertex compression and limitations of this setting, refer to Compressing mesh data . |
| Optimize Mesh Data | Enable this option to strip unused vertex attributes from the mesh used in a build. This option reduces the amount of data in the mesh, which can help reduce build size, loading times, and runtime memory usage. Warning: If you have this setting enabled, you should remember to not change material or shader settings at runtime. Refer to PlayerSettings.stripUnusedMeshComponents for more information. |
| Texture MipMap Stripping | Enables mipmap stripping for all platforms. It strips unused mipmaps from Textures at build time. Unity determines unused mipmaps by comparing the value of the mipmap against the quality settings for the current platform. If a mipmap value is excluded from every quality setting for the current platform, then Unity strips those mipmaps from the build at build time. If QualitySettings.masterTextureLimit is set to a mipmap value that has been stripped, Unity will set the value to the closest mipmap value that hasn't been stripped. |

Ο χρήστης μπορεί να επιλέξει τον τύπο [Logs](#) που θα επιτρέπεται σε συγκεκριμένα πλαίσια. Επιλέγει τη μέθοδο ανίχνευσης στοιβάς που προτιμάει, ενεργοποιώντας την επιλογή που αντιστοιχεί σε κάθε τύπο καταγραφής (Σφάλμα, Βεβαίωση, Προειδοποίηση, Καταγραφή και Εξαίρεση) ανάλογα με τον τύπο καταγραφής που χρειάζεται. Παρακάτω παρουσιάζονται παραδείγματα:

- ScriptOnly: Καταγράφει μόνο κατά την εκτέλεση σεναρίων.
- Πλήρης: Καταγραφεί συνεχώς.

- Κανένα: Δεν καταγράφονται ποτέ αρχεία καταγραφής.

Και τέλος ένα στοιχείο [Legacy](#) που ενεργοποιεί την επιλογή Clamp BlendShapes (Deprecated) για να περιορίσει το εύρος των βαρών των σχημάτων ανάμειξης σε SkinnedMeshRenderers.

Ακολουθούν οι ρυθμίσεις σε **Windows** πλατφόρμα.

Οι ρυθμίσεις Rendering είναι για να προσαρμόσει ο χρήστης τον τρόπο με τον οποίο η Unity αποδίδει το παιχνίδι για πλατφόρμες σταθερών υπολογιστών.

Πίνακας 69. Rendering (Windows)

| Property | Function |
|--------------------------------------|---|
| Color Space | Choose which color space should be used for rendering: Gamma or Linear . See the Linear rendering overview for an explanation of the difference between the two. |
| Auto Graphics API for Windows | Enable this option to use the best Graphics API on the Windows machine the game is running on. Disable it to add and remove supported Graphics APIs. |
| Auto Graphics API for Mac | Enable this option to use the best Graphics API on the Mac the game is running on. Disable it to add and remove supported Graphics APIs. Windows does not support this property. |
| Auto Graphics API for Linux | Enable this option to use the best Graphics API on the Linux machine it runs on. Disable it to add and remove supported Graphics APIs. |
| Color Gamut for Mac | You can add or remove color gamuts for the Mac platform to use for rendering. Click the plus (+) icon to see a list of available gamuts. A color gamut defines a possible range of colors available for a given device (such as a monitor or screen). The sRGB gamut is the default (and required) gamut. Windows does not support this property. |
| Metal API Validation | Enable this option when you need to debug Shader issues. Note: Validation increases CPU usage, so use it only for debugging. |
| Metal Write-Only Backbuffer | Allow improved performance in non-default device orientation. This sets the framebufferOnly flag on the back buffer, which prevents readback from the back buffer but enables some driver optimization. |
| Memoryless Depth | Choose when to use memoryless render textures . Memoryless render textures are temporarily stored in the on-tile memory when rendered, not in CPU or GPU memory. This reduces memory usage of your app but you cannot read or write to these render textures. Note: Memoryless render textures are only supported on iOS, tvOS 10.0+ Metal and Vulkan. Render textures are read/write protected and stored in CPU or GPU memory on other platforms. |
| <i>Unused</i> | Never use memoryless framebuffer depth. |
| <i>Forced</i> | Always use memoryless framebuffer depth. |
| <i>Automatic</i> | Let Unity decide when to use memoryless framebuffer depth. |
| Static Batching | Enable this option to use Static batching. |
| Dynamic Batching | Enable this option to use Dynamic Batching on your build (enabled by default). Note: Dynamic batching has no effect when a Scriptable Render Pipeline is active, so this setting is only visible when nothing is set in the Scriptable Render Pipeline Asset Graphics setting. |
| Compute Skinning | Enable this option to enable DX11/DX12/ES3 GPU compute skinning , freeing up CPU resources. |

| Property | Function |
|---|--|
| Graphics Jobs | Enable this option to instruct Unity to offload graphics tasks (render loops) to worker threads running on other CPU cores. This is intended to reduce the time spent in Camera.Render on the main thread, which is often a bottleneck. |
| Lightmap Encoding | Choose <i>Normal Quality</i> or <i>High Quality</i> to set the lightmap encoding. This setting affects the encoding scheme and compression format of the lightmaps. |
| HDR Cubemap Encoding | Choose <i>Low Quality</i> , <i>Normal Quality</i> , or <i>High Quality</i> to set the HDR Cubemap encoding. This setting affects the encoding scheme and compression format of the HDR Cubemaps. |
| Lightmap Streaming | Whether to use Mipmap Streaming for lightmaps. Unity applies this setting to all lightmaps when it generates them. Note: To use this setting, you must enable the Texture Streaming Quality setting. |
| Streaming Priority | Set the priority for all lightmaps in the Mipmap Streaming system . Unity applies this setting to all lightmaps when it generates them. Positive numbers give higher priority. Valid values range from -128 to 127. |
| Frame Timing Stats | Enable this property to gather CPU and GPU frame time statistics. Use this together with the Dynamic Resolution camera setting to determine if your application is CPU or GPU bound. |
| Use Display In HDR Mode (Windows Only) | Enable this checkbox to make the game automatically switch to HDR mode output when it runs. This only works on displays that support this feature. If the display does not support HDR mode, the game runs in standard mode. |
| Swap Chain Bit Depth | Select the number of bits in each color channel for swap chain buffers. Only available if HDR Mode is enabled. |
| <i>Bit Depth 10</i> | Unity will use the R10G10B10A2 buffer format and Rec2020 primaries with ST2084 PQ encoding. |
| <i>Bit Depth 16</i> | Unity will use the R16G16B16A16 buffer format and Rec709 primaries with linear color (no encoding). |
| Load/Store Action Debug Mode | Highlights undefined pixels that might cause rendering problems in your built application. The highlighting appears only in the Game view, and your built application if you select Development Build in Build Settings. See LoadStoreActionDebugModeSettings . |
| <i>Editor Only</i> | Highlights undefined pixels in the Game view, but not in your built application. |

Πίνακας 70. Vulkan Settings (Windows)

| Property | Description |
|---|---|
| SRGB Write Mode | Enable this option to allow Graphics.SetSRGBWrite() renderer to toggle the sRGB write mode during runtime. That is, if you want to temporarily turn off Linear-to-sRGB write color conversion, you can use this property to achieve that. Enabling this has a negative impact on performance on mobile tile-based GPUs; therefore, do NOT enable this for mobile. |
| Number of swapchain buffers | Set this option to 2 for double-buffering, or 3 for triple-buffering to use with Vulkan renderer. This setting may help with latency on some platforms, but in most cases you should not change this from the default value of 3. Double-buffering might have a negative impact on performance. Do not use this setting on Android. |
| Acquire swapchain image late as possible | If enabled, Vulkan delays acquiring the backbuffer until after it renders the frame to an offscreen image. Vulkan uses a staging image to achieve this. Enabling this setting causes an extra blit when presenting the backbuffer. This setting, in combination with double-buffering, can improve performance. However, it also can cause performance issues because |

| Property | Description |
|--------------------------------|--|
| | the additional blit takes up bandwidth. |
| Recycle command buffers | Indicates whether to recycle or free CommandBuffers after Unity executes them. |

Πίνακας 71. Configuration (Windows)

| Property | Description |
|-----------------------------------|--|
| Scripting Backend | Choose the scripting backend you want to use. The scripting backend determines how Unity compiles and executes C# code in your Project. |
| Mono | Compiles C# code into .NET Common Intermediate Language (CIL) and executes that CIL using a Common Language Runtime. For more information, refer to the Mono page. |
| IL2CPP | Compiles C# code into CIL, converts the CIL to C++ and then compiles that C++ into native machine code, which executes directly at runtime. Refer to IL2CPP for more information. |
| API Compatibility Level | Choose which .NET APIs you can use in your project. This setting can affect compatibility with third-party libraries. However, it has no effect on Editor-specific code (code in an Editor directory, or within an Editor-specific Assembly Definition). Tip: If you are having problems with a third-party assembly, you can try the suggestion in the API Compatibility Level section below. |
| .Net Standard 2.1 | Compatible with .NET Standard 2.1 . Produces smaller builds and has full cross-platform support. |
| .Net Framework | Compatible with the .NET Framework 4 (which includes everything in the .NET Standard 2.0 profile plus additional APIs). Choose this option when using libraries that access APIs not included in .NET Standard 2.0. Produces larger builds and any additional APIs available aren't necessarily supported on all platforms. Refer to Referencing additional class library assemblies for more information. |
| IL2CPP Code Generation | Defines how Unity manages IL2CPP code generation. This option is only available if you use the IL2CPP scripting backend. |
| Faster runtime | Generates code optimized for runtime performance. This setting is activated by default. |
| Faster (smaller) builds | Generates code optimized for build size and iteration. This setting generates less code and produces a smaller build, but can reduce runtime performance for generic code. Use this option when faster build times are important, such as when iterating on changes. |
| C++ Compiler Configuration | Choose the C++ compiler configuration used when compiling IL2CPP generated code. Note: This property is disabled unless Scripting Backend is set to <i>IL2CPP</i> . |
| Use incremental GC | Uses the incremental garbage collector, which spreads garbage collection over several frames to reduce garbage collection-related spikes in frame duration. For more information, refer to Automatic Memory Management . |
| Allow downloads over HTTP | Indicates whether to allow downloading content over HTTP. The options are Not allowed, Allowed in Development builds only, and Always allowed. The default option is Not allowed due to the recommended protocol being HTTPS, which is more secure. |
| Active Input Handling | Choose how to handle input from users. |
| Input Manager (old) | Use the default Input window. |

| Property | Description |
|-----------------------------------|--|
| Input System Package (New) | Use the newer Input system. To try the new Input System, install the InputSystem package . |
| Both | Use both systems side by side. |

Χρησιμοποιήστε αυτές τις ρυθμίσεις για να ελέγξετε πόση μνήμη shaders χρησιμοποιείτε κατά το χρόνο εκτέλεσης.

Πίνακας 72. Shader Variant Loading (Windows)

| Setting | Description |
|--------------------------------|---|
| Default chunk size (MB) | Sets the maximum size of compressed shader variant data chunks Unity stores in your built application for all platforms. The default is 16. Refer to Shader loading for more information. |
| Default chunk count | Sets the default limit on how many decompressed chunks Unity keeps in memory on all platforms. The default is 0, which means there's no limit. |
| Override | Enables overriding Default chunk size and Default chunk count for this build target. |
| Chunk size (MB) | Overrides the value of Default chunk size (MB) on this build target. |
| Chunk count | Overrides the value of Default chunk count on this build target. |

Πίνακας 73. Optimization (Windows)


| Property | Description |
|----------------------------------|---|
| Prebake Collision Meshes | Adds collision data to Meshes at build time. |
| Keep Loaded Shaders Alive | Indicates whether to prevent shaders from being unloaded. For more information, see Shader Loading . |
| Preloaded Assets | Sets an array of Assets for the player to load on startup. To add new Assets, increase the value of the Size property and then set a reference to the Asset to load in the new Element box that appears. |
| AOT compilation options | Additional options for Ahead of Time (AOT) compilation. This helps optimize the size of the built iOS player . |
| Strip Engine Code | Enable this option if you want the Unity Linker tool to remove code for Unity Engine features that your Project doesn't use. This setting is only available with the IL2CPP scripting backend. Most apps don't use every available DLL. This option strips out DLLs that your app doesn't use to reduce the size of the built Player. If your app is using one or more classes that would normally be stripped out under your current settings, Unity displays a debug message when you try to build the app. |
| Managed Stripping Level | Chooses how aggressively Unity strips unused managed (C#) code. The options are Minimal , Low , Medium , and High . When Unity builds your app, the Unity Linker process can strip unused code from the managed DLLs your Project uses. Stripping code can make the resulting executable significantly smaller, but can sometimes accidentally remove code that's in use. For more information about these options and bytecode stripping with IL2CPP, refer to ManagedStrippingLevel . |
| Vertex Compression | Sets vertex compression per channel. This affects all the meshes in your project. Typically, Vertex Compression is used to reduce the size of mesh data in memory, |

| Property | Description |
|---------------------------------|--|
| | reduce file size, and improve GPU performance. For more information on how to configure vertex compression and limitations of this setting, refer to Compressing mesh data . |
| Optimize Mesh Data | Enable this option to strip unused vertex attributes from the mesh used in a build. This option reduces the amount of data in the mesh, which can help reduce build size, loading times, and runtime memory usage. Warning: If you have this setting enabled, you should remember to not change material or shader settings at runtime. Refer to PlayerSettings.stripUnusedMeshComponents for more information. |
| Texture MipMap Stripping | Enables mipmap stripping for all platforms. This strips unused mipmaps from Textures at build time. Unity determines unused mipmaps by comparing the value of the mipmap against the Quality Settings for the current platform. If a mipmap value is excluded from every Quality Setting for the current platform, then Unity strips those mipmaps from the build at build time. If QualitySettings.masterTextureLimit is set to a mipmap value that has been stripped, Unity will set the value to the closest mipmap value that has not been stripped. |

Οι υπόλοιπες ρυθμίσεις είναι οι ίδιες με το Android.

4.8.10.5 Publishing Settings

Η τελευταία καρτέλα είναι το [Publishing Settings](#) και είναι μόνο στην Android πλατφόρμα. Εδώ ο χρήστης μπορεί να ρυθμίσει πως η μηχανή Unity δημιουργεί την εφαρμογή. Το παράθυρο **Keystore Manager** χρησιμοποιείται για τη δημιουργία, διαμόρφωση και φόρτωση των κλειδιών και των κλειδοθηκών του χρήστη. Οι υπάρχουσες κλειδοθήκες και κλειδιά μπορούν να φορτωθούν είτε από το παράθυρο Διαχείρισης κλειδοθηκών είτε από τον κύριο πίνακα δημοσίευσης Android. Εάν επιλεγεί η φόρτωσή τους από το εσωτερικό του Keystore Manager, τα πεδία Project Keystore και Project Key συμπληρώνονται αυτόματα από την Unity.

| | |
|---|---|
|  | Για λόγους ασφαλείας η Unity δεν αποθηκεύει το Keystore ή το Project key passwords. |
|---|---|

Πίνακας 74. Project Keystore properties and description

| Property | Description |
|------------------------|--|
| Custom Keystore | Enable Custom Keystore to load and use an existing Keystore. |
| Select | When Custom Keystore is enabled, use this to select the keystore you want to use. The keystores below the partition in the Select dropdown are stored in a predefined dedicated location. For more |

| Property | Description |
|----------|---|
| | details, refer to Choose the keystore location . |
| Path | You don't need to enter your keystore path. Unity provides this based on the keystore you choose. |
| Password | Enter your keystore password to load your chosen keystore. |

Όταν ο χρήστης φορτώνει ένα keystore η μηχανή Unity φορτώνει όλα τα κλειδιά σε αυτό. Χρησιμοποιεί ρυθμίσεις Project Key για να επιλέξει ένα κλειδί από αυτό το keystore για να χρησιμοποιηθεί ως ενεργό κλειδί για το ανοιχτό έργο.

Πίνακας 75. Project key properties and descriptions

| Property | Description |
|----------|--|
| Alias | Select the key you want to use for the open project. |
| Password | Enter your key Password. |

Από προεπιλογή, η εφαρμογή κατασκευάζεται από τη Unity χρησιμοποιώντας τα παρεχόμενα αρχεία manifest, πρότυπα Gradle και αρχεία Proguard που είναι εγκατεστημένα. Ο χρήστης μπορεί να τροποποιήσει αυτά τα αρχεία χρησιμοποιώντας την ενότητα Build στις ρυθμίσεις δημοσίευσης Android.

Για να χρησιμοποιήσετε ένα προσαρμοσμένο αρχείο manifest, ένα πρότυπο Gradle ή ένα αρχείο Proguard:

Πίνακας 76. The manifest files, Gradle templates and Proguard files

| Property | Description |
|-----------------------------------|---|
| Upgrade templates to C# | Opens the Android Templates Upgrader window which you can use to upgrade your custom Gradle template files to use the Android Project Configuration Manager . |
| Custom Main Manifest | Customizable version of the Android LibraryManifest.xml file. This file contains important metadata about your Android application. For more information about the responsibilities of the Main/Unity Library Manifest refer to Unity Library Manifest . |
| Custom Unity Launcher Manifest | Customizable version of the Android LauncherManifest.xml file. This file contains important metadata about your Android application's launcher. For more information about the responsibilities of the Unity Launcher Manifest refer to Unity Launcher Manifest . |
| Custom Main Gradle Template | Customizable version of the mainTemplate.gradle file. This file contains information on how to build your Android application as a library. For more information refer to the documentation on Gradle project files . |
| Custom Launcher Gradle Template | Customizable version of the launcherTemplate.gradle_ file. This file contains instructions on how to build your Android application. For more information refer to the documentation on Gradle project files . |
| Custom Base Gradle Template | Customizable version of the baseProjectTemplate.gradle file. This file contains configuration that's shared between all other templates and Gradle projects. For more information see documentation on Gradle project files . |
| Custom Gradle Properties Template | Customizable version of the gradle.properties file. This file contains configuration settings for the Gradle build environment. This includes: <ul style="list-style-type: none"> The JVM (Java Virtual Machine) memory configuration. |

| Property | Description |
|----------------------|---|
| | <ul style="list-style-type: none"> • A property to allow Gradle to build using multiple JVMs. • A property for choosing the tool to do the minification. • A property to not compress native libs when building an app bundle. |
| Custom Proguard File | Customizable version of the proguard.txt file. This file contains configuration settings for the minification process. If minification removes some Java code which should be kept, you should add a rule to keep that code in this file. For more information refer to the documentation on Minification . |

Η **Minification** είναι μια διαδικασία που εφαρμόζεται στον κώδικα της εφαρμογής με σκοπό τη συρρίκνωση, την κρυπτογράφηση και τη βελτιστοποίησή του. Μπορεί να προκαλέσει μείωση του μεγέθους του κώδικα και να δυσκολέψει την αποσυναρμολόγησή του. Οι ρυθμίσεις Minify χρησιμοποιούνται για να καθορίσετε πότε και πώς θα εφαρμοστεί η ελαχιστοποίηση από τη Unity κατά τη δημιουργία της εφαρμογής. Οι ρυθμίσεις στην ενότητα Minify ισχύουν μόνο για τη διαδικασία κατασκευής του τρέχοντος έργου.


Πίνακας 77. Rendering Player settings for desktop platforms (Windows)

| Property | Function |
|-------------------------------|---|
| Color Space | Choose which color space should be used for rendering: Gamma or Linear . See the Linear rendering overview for an explanation of the difference between the two. |
| Auto Graphics API for Windows | Enable this option to use the best Graphics API on the Windows machine the game is running on. Disable it to add and remove supported Graphics APIs. |
| Auto Graphics API for Mac | Enable this option to use the best Graphics API on the Mac the game is running on. Disable it to add and remove supported Graphics APIs. Windows does not support this property. |
| Auto Graphics API for Linux | Enable this option to use the best Graphics API on the Linux machine it runs on. Disable it to add and remove supported Graphics APIs. |
| Color Gamut for Mac | You can add or remove color gamuts for the Mac platform to use for rendering. Click the plus (+) icon to see a list of available gamuts. A color gamut defines a possible range of colors available for a given device (such as a monitor or screen). The sRGB gamut is the default (and required) gamut. Windows does not support this property. |
| Metal API Validation | Enable this option when you need to debug Shader issues. Note: Validation increases CPU usage, so use it only for debugging. |
| Metal Write-Only Backbuffer | Allow improved performance in non-default device orientation. This sets the frame-BufferOnly flag on the back buffer, which prevents readback from the back buffer but enables some driver optimization. |
| Memoryless Depth | Choose when to use memoryless render textures . Memoryless render textures are temporarily stored in the on-tile memory when rendered, not in CPU or GPU memory. This reduces memory usage of your app but you cannot read or write to these render textures. Note: Memoryless render textures are only supported on iOS, tvOS 10.0+ Metal and Vulkan. Render textures are read/write protected and stored in CPU or GPU memory on other platforms. |
| Unused | Never use memoryless framebuffer depth. |

| Property | Function |
|---|---|
| <i>Forced</i> | Always use memoryless framebuffer depth. |
| <i>Automatic</i> | Let Unity decide when to use memoryless framebuffer depth. |
| Static Batching | Enable this option to use Static batching. |
| Dynamic Batching | Enable this option to use Dynamic Batching on your build (enabled by default). Note: Dynamic batching has no effect when a Scriptable Render Pipeline is active, so this setting is only visible when nothing is set in the Scriptable Render Pipeline Asset Graphics setting. |
| Compute Skinning | Enable this option to enable DX11/DX12/ES3 GPU compute skinning , freeing up CPU resources. |
| Graphics Jobs | Enable this option to instruct Unity to offload graphics tasks (render loops) to worker threads running on other CPU cores. This is intended to reduce the time spent in Camera.Render on the main thread, which is often a bottleneck. |
| Lightmap Encoding | Choose <i>Normal Quality</i> or <i>High Quality</i> to set the lightmap encoding. This setting affects the encoding scheme and compression format of the lightmaps. |
| HDR Cubemap Encoding | Choose <i>Low Quality</i> , <i>Normal Quality</i> , or <i>High Quality</i> to set the HDR Cubemap encoding. This setting affects the encoding scheme and compression format of the HDR Cubemaps. |
| Lightmap Streaming | Whether to use Mipmap Streaming for lightmaps. Unity applies this setting to all lightmaps when it generates them. Note: To use this setting, you must enable the Texture Streaming Quality setting. |
| Streaming Priority | Set the priority for all lightmaps in the Mipmap Streaming system . Unity applies this setting to all lightmaps when it generates them. Positive numbers give higher priority. Valid values range from -128 to 127. |
| Frame Timing Stats | Enable this property to gather CPU and GPU frame time statistics. Use this together with the Dynamic Resolution camera setting to determine if your application is CPU or GPU bound. |
| Use Display In HDR Mode (Windows Only) | Enable this checkbox to make the game automatically switch to HDR mode output when it runs. This only works on displays that support this feature. If the display does not support HDR mode, the game runs in standard mode. |
| Swap Chain Bit Depth | Select the number of bits in each color channel for swap chain buffers. Only available if HDR Mode is enabled. |
| <i>Bit Depth 10</i> | Unity will use the R10G10B10A2 buffer format and Rec2020 primaries with ST2084 PQ encoding. |
| <i>Bit Depth 16</i> | Unity will use the R16G16B16A16 buffer format and Rec709 primaries with linear color (no encoding). |
| Load/Store Action Debug Mode | Highlights undefined pixels that might cause rendering problems in your built application. The highlighting appears only in the Game view, and your built application if you select Development Build in Build Settings. See LoadStoreActionDebugModeSettings . |
| <i>Editor Only</i> | Highlights undefined pixels in the Game view, but not in your built application. |



Εδώ θέλει προσοχή γιατί η διαδικασία Minify μπορεί να αφαιρέσει κώδικα που χρειάζεται η εφαρμογή με αποτέλεσμα να καταστραφεί το εκτελέσιμο αρχείο της εφαρμογής.

| | |
|---|--|
|  | <p>Συνήθως, είναι καλή πρακτική ο χρήστης να εφαρμόζει την ελαχιστοποίηση μόνο στις εκδόσεις παραγωγής της εφαρμογής και όχι στις εκδόσεις αποσφαλμάτωσης. Αυτό συμβαίνει διότι η διαδικασία ελαχιστοποίησης απαιτεί χρόνο και μπορεί να επηρεάσει την ταχύτητα κατασκευής. Επίσης, μπορεί να δυσκολέψει την αποσφαλμάτωση λόγω της βελτιστοποίησης που εφαρμόζεται στον κώδικα.</p> |
|---|--|

Πίνακας 78. Minify properties and descriptions

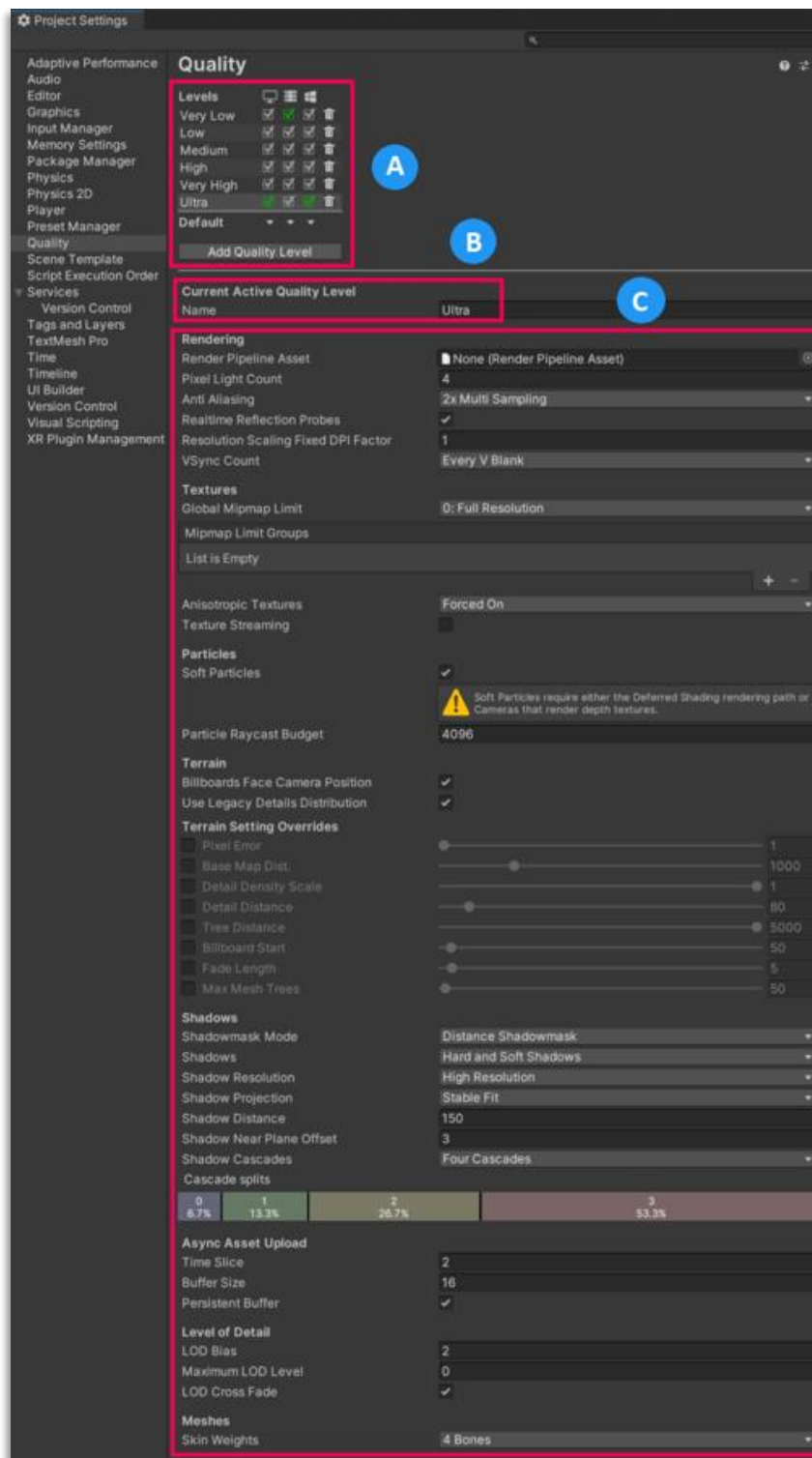
| Property | Description |
|----------|---|
| Release | Enable this checkbox if you want Unity to minify your application's code in release builds. |
| Debug | Enable this checkbox if you want Unity to minify your application's code in debug builds. |

Η τελευταία επιλογή στην συγκεκριμένη καρτέλα ενεργοποιεί την επιλογή **Split Application Binary** για να χωρίσει ο χρήστης την εφαρμογή του σε κύριο πακέτο (APK) εξόδου σε και πακέτο επέκτασης (OBB). Το Google Play Store το απαιτεί αυτό αν θέλει να δημοσιεύσει εφαρμογές μεγαλύτερες από 100 MB.

4.8.11 Quality




Η επόμενη σημαντική επιλογή στις ρυθμίσεις Project Settings που θα πρέπει να παραμετροποιήσει ο προγραμματιστής είναι το **Quality**. Εδώ ο χρήστης δύναται να ορίσει την ποιότητα γραφικών που θέλει να συμπεριλάβει στο build της εφαρμογής. Στο επάνω μέρος υπάρχει το Quality matrix το οποίο παρέχει μια οπτική απεικόνιση των διάφορων επιπέδων γραφικών τα οποία παίζουν το ρόλο των template δηλαδή των ομαδοποιημένων ρυθμίσεων κάτω από ένα όνομα για την εύκολη εναλλαγή τους ανά πάσα στιγμή. Εδώ ανάλογα την πλατφόρμα που στοχεύει να πλασάρει την εφαρμογή του ο χρήστης ρυθμίζει πόσο καλά γραφικά θα χρησιμοποιήσει γιατί αυτό έχει άμεσο αντίκτυπο στο framerate και κατά επέκταση στο gameplay. Μπορεί να φτιάξει το δικό του template και να ορίσει ένα δικό του όνομα. Το πράσινο πλαίσιο επιλογής σε μια στήλη υποδηλώνει το επίπεδο που έχει επιλεγεί επί του παρόντος για τη συγκεκριμένη πλατφόρμα. Η Unity παρέχει έξι προκαθορισμένα επίπεδα ποιότητας, αλλά προγραμματιστής μπορεί να προσθέσει τα δικά

του επίπεδα. Στο κάτω μέρος εμφανίζονται οι ρυθμίσεις για το επιλεγμένο επίπεδο ποιότητας με τις εξής ιδιότητες:



Εικόνα 71. Quality settings

Πίνακας 79. Περιγραφή τμημάτων Quality settings

| Icon | Description |
|---|--|
|  | Το Επίπεδο Ποιότητας που έχετε ορίσει σε αυτό το έργο. |
|  | Το τρέχον ενεργό Επίπεδο Ποιότητας. |
|  | Η διαμόρφωση του τρέχοντος επιπέδου ποιότητας. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/class-QualitySettings.html>

Πίνακας 80. Rendering Properties and Description

| Property | Description |
|--|--|
| Render Pipeline | The Render Pipeline Asset to use for this quality level. |
| Pixel Light Count | Set the maximum number of pixel lights when Unity uses Forward Rendering . |
| AntiAliasing | Choose the level of Multi-Sample Anti-aliasing (MSAA) that the GPU performs. The options are <i>Disabled</i> , <i>2x Multi Sampling</i> , <i>4x Multi Sampling</i> and <i>8x Multi Sampling</i> . Anti aliasing smooths the appearance of polygon edges. As the level of anti-aliasing increases, so does the smoothness and the performance cost on the GPU. MSAA is compatible only with Forward rendering. For more information on other types of anti-aliasing and compatibility, see Post processing . |
| Realtime Reflection Probes | Enable this option to update reflection probes during gameplay. |
| Resolution Scaling Fixed DPI Factor | Downscales the device's screen resolution below its native resolution. For more details, see Android Player settings and iOS Player settings. |
| V Sync Count | Choose to synchronize rendering with vertical blanks or not to synchronize at all. Unity can synchronize rendering with the refresh rate of the display device to avoid tearing artifacts . The available options are <i>Every V Blank</i> , <i>Every Second V Blank</i> , or <i>Don't Sync</i> . |

Πίνακας 81. Textures Properties and descriptions

| Property | Description |
|----------------------------|---|
| Global Mipmap Limit | Choose the highest resolution mipmap level that Unity uses when it renders textures. Higher mipmap levels have lower resolutions, which means that they require less GPU memory and GPU processing time. This property only affects textures with a texture shape of 2D or 2D Array. The options are: <ul style="list-style-type: none"> • 0: Full Resolution • 1: Half Resolution • 2: Quarter Resolution • 3: Eighth Resolution For more detailed information about what these resolutions mean, see Mipmaps introduction . Textures that do not have mipmaps render at their full resolution, regardless of the option you choose. |
| Mipmap Limit Groups | Use these groups to designate specific textures that should disregard the globalTextureMipmapLimit or add a bias. This makes it possible to allocate more of your memory budget for important textures and less of it for less important textures. This property only affects textures with a texture shape of 2D or 2D Array. The options are: <ul style="list-style-type: none"> • Offset Global Mipmap Limit: -3 • Offset Global Mipmap Limit: -2 • Offset Global Mipmap Limit: -1 • Use Global Mipmap Limit • Offset Global Mipmap Limit: +1 |

| Property | Description |
|-----------------------------|---|
| | <ul style="list-style-type: none"> • Offset Global Mipmap Limit: +2 • Offset Global Mipmap Limit: +3 • Override Global Mipmap Limit: Full Resolution • Override Global Mipmap Limit: Half Resolution • Override Global Mipmap Limit: Quarter Resolution • Override Global Mipmap Limit: Eighth Resolution <p>The Offsets bias the Mipmap Limit value for the applicable group of textures. For example, if the Global Texture Mipmap Limit is Half Resolution and you select Offset Global Mipmap Limit: -1, then the new Mipmap Limit for the applicable group of textures is Full Resolution. The Overrides replace the Global Mipmap Limit for the applicable group of textures. For example, if the Global Texture Mipmap Limit is Half Resolution and you select Override Global Mipmap Limit: Full Resolution, then the new Mipmap Limit for the applicable group of textures is Full Resolution.</p> <p>Note: If you delete or rename a Mipmap Limit Group, this triggers a dialog box that provides you with the option to re-import the textures in that group. Undo does not revert these import changes.</p> |
| Anisotropic Textures | Choose if and how Unity uses anisotropic Textures. The options are <i>Disabled</i> , <i>Per Texture</i> and <i>Forced On</i> (that is, always enabled). |
| Texture Streaming | Enable this checkbox to use Mipmap Streaming . If you are not planning to use the Mipmap Streaming system, disable this feature to avoid any overhead. |
| Add All Cameras | Enable this checkbox to make Unity calculate Mipmap Streaming for all active Cameras in the Project. This is enabled by default. For more information, see Mipmap Streaming system: Configuring Cameras . |
| Memory Budget | Set the total amount of memory you want to assign to all loaded textures (in MB) when using the Mipmap Streaming system . This is set to 512 MB by default. For more information, see Mipmap Streaming system: Setting the memory budget . |
| Renderers Per Frame | This controls the CPU processing overhead for the Mipmap Streaming system for the main thread and associated jobs. This is 512 by default (that is, processing 512 Mesh renderers per frame). Lower values reduce processing time, but increase delays to Unity loading the mipmaps. |
| Max Level Reduction | Set the maximum number of mipmaps that the Mipmap Streaming system can discard if the Mipmap Streaming system reaches the Memory Budget . This is set to 2 by default (which means that the system discards no more than two mipmaps). This value is also the mipmap level that the Mipmap Streaming system initially loads at startup. For example, when this is set to 2, Unity skips the two highest mipmaps on first load. For more information, see Mipmap Streaming system: Setting the memory budget . |
| Max IO Requests | Set the maximum number of texture file in/out (IO) requests from the Mipmap Streaming system that are active at any one time. This is set to 1024 by default. This default is set high enough to prevent any IO cap beyond what is already active due to the Async Upload pipeline or file system itself. If the Scene Texture content changes significantly and rapidly, the system might attempt to load more Texture mipmaps than the file IO can keep up with. Lowering this value reduces the IO bandwidth that the Mipmap Streaming system generates. The result is a more rapid response to changing mipmap requirements. |

Πίνακας 82. Particles Properties and Descriptions

| Property | Description |
|--------------------------------|--|
| Soft Particles | Indicates whether to fade particles as they approach the edges of opaque GameObject . For more information, see Soft particles . |
| Particle Raycast Budget | Set the maximum number of raycasts to use for approximate particle system collisions (those with <i>Medium</i> or <i>Low</i> quality). See Particle System Collision Module . |

Πίνακας 83. Terrain Properties and Descriptions

| Property | Description |
|--|--|
| Billboards Face Camera Position | Enable this option to force billboards to face the camera while rendering instead of the camera plane. This produces a better, more realistic image, but is more expensive to render. |
| Use Legacy Details Distribution | Enable this option to use the previously supported scattering algorithm that often resulted in overlapping details. Included for backward compatibility with Terrains authored in Unity 2022.1 and earlier. |
| Terrain Setting Overrides | Various override settings that, when enabled, override the value of all active terrains (except those with the “Ignore Quality Settings” setting enabled). For more information about these settings, see Terrain Settings . |
| Pixel Error | Value set to Terrain Pixel Error. See Terrain Settings . |
| Base Map Dist. | Value set to Terrain Basemap Distance. See Terrain Settings . |
| Detail Density Scale | Value set to Terrain Density Scale. See Terrain Settings . |
| Detail Distance | Value set to Terrain Detail Distance. See Terrain Settings . |
| Tree Distance | Value set to Terrain Tree Distance. See Terrain Settings . |
| Billboard Start | Value set to Terrain Billboard Start. See Terrain Settings . |
| Fade Length | Value set to Terrain Fade Length. See Terrain Settings . |
| Max Mesh Trees | Value set to Terrain Max Mesh Trees. See Terrain Settings . |

Πίνακας 84. Shadows Properties and Descriptions

| Property | Description |
|---------------------------------|---|
| Shadowmask Mode | Choose the shadowmask behavior when using the Shadowmask Mixed lighting mode. Use the Lighting window (menu: Window > Rendering > Lighting) to set this up in your Scene. |
| Distance Shadowmask | Unity uses real-time shadows up to the Shadow Distance , and baked shadows beyond it. |
| Shadowmask | Static GameObjects that cast shadows always cast baked shadows. |
| Shadows | Choose which type of shadows to use. The available options are <i>Hard and Soft Shadows</i> , <i>Hard Shadows Only</i> and <i>Disable Shadows</i> . |
| Shadow Resolution | Choose which resolution to render shadows at. The available options are <i>Low Resolution</i> , <i>Medium Resolution</i> , <i>High Resolution</i> and <i>Very High Resolution</i> . The higher the resolution, the greater the processing overhead. |
| Shadow Projection | Choose which method to use for projecting shadows from a directional light. |
| Close Fit | Renders higher resolution shadows but they can sometimes wobble slightly if the camera moves. |
| Stable Fit | Renders lower resolution shadows but they don't wobble with camera movements. |
| Shadow Distance | Enter the maximum distance from the Camera at which shadows are visible. Unity does not render shadows that fall beyond this distance. |
| Shadow Near Plane Offset | Enter the offset shadow near plane to account for large triangles being distorted by shadow pancaking. |
| Shadow Cascades | Choose the number of shadow cascades to use. The available options are <i>No Cascades</i> , <i>Two Cascades</i> , or <i>Four Cascades</i> . A higher number of cascades gives better quality but at the expense of processing overhead (see Shadow Cascades for further details). |
| Cascade splits | Adjust the cascade shadow split(s) by moving the vertical line between each cascade left or right. Depending on what value you chose for the Shadow Cascades setting, you can see two or four different colors. If Shadow Cascades is set to <i>No Cascades</i> , then this entire control is hidden. |

Πίνακας 85. Async Asset Upload Properties and Descriptions

| Property | Description |
|--------------------------|--|
| Time Slice | Set the amount of CPU time in milliseconds per frame Unity spends uploading buffered Texture and Mesh data to the GPU. See LoadingTextureandMeshData . |
| Buffer Size | Set the size in megabytes of the Async Upload Buffer Unity uses to stream Texture and Mesh data for the to the GPU. See LoadingTextureandMeshData . |
| Persistent Buffer | Indicates whether the upload buffer should persist even when there is nothing left to upload. |

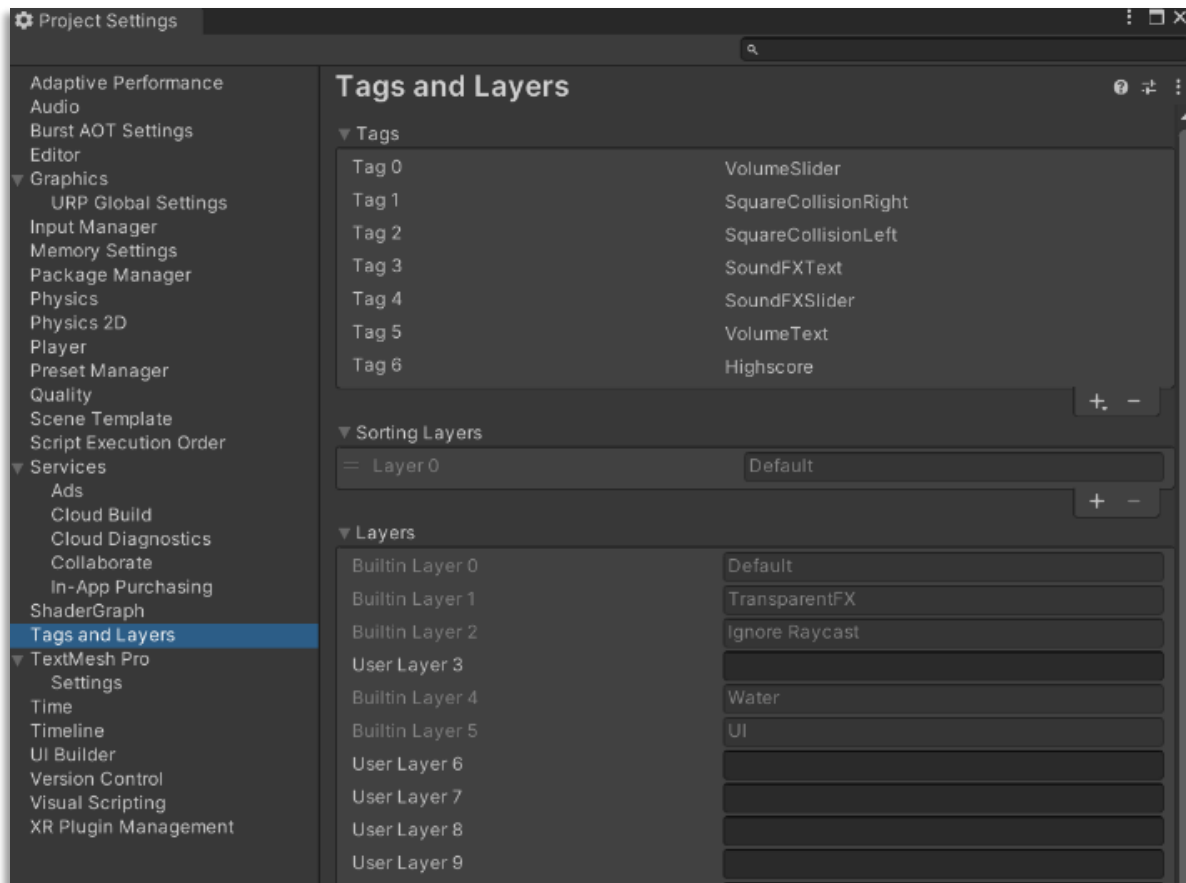
Πίνακας 86. Level of Detail Properties and Descriptions

| Property | Description |
|--------------------------|---|
| Lod Bias | Set the level-of-detail (LOD) bias. LOD levels are chosen based on the onscreen size of an object. When the size is between two LOD levels, the choice can be biased toward the less detailed or more detailed of the two Models available. This is set as a fraction from 0 to +infinity. When it is set between 0 and 1 it favors less detail. A setting of more than 1 favors greater detail. For example, setting LOD Bias to 2 and having it change at 50% distance, LOD actually only changes on 25%. |
| Maximum LOD Level | Set the highest LOD that the game uses. See Maximum LOD level for more information. |



Πίνακας 87. Meshes Properties and Descriptions

| Property | Description |
|---------------------|---|
| Skin Weights | Choose the number of bones that can affect a given vertex during an animation. The available options are <i>1 Bone</i> , <i>2 Bones</i> , <i>4 Bones</i> , and <i>Unlimited</i> . |

Επόμενη σημαντική ρύθμιση στο project settings είναι τα Tags and Layers τα οποία παίζουν πολύ σημαντικό ρόλο στην ανάπτυξη εφαρμογών μέσω της μηχανής Unity.



Εικόνα 72. Ρυθμίσεις Tags and Layers

| | |
|---|--|
|  | <p>Εδώ να αναφερθεί ότι αφού ο χρήστης ονομάσει μια ετικέτα, δεν μπορεί να τη μετονομάσει πάρα μόνο να την καταργήσει και να την δημιουργήσει από την αρχή. Ένα αντικείμενο παιχνιδιού μπορεί να έχει μόνο μία ετικέτα που του έχει εκχωρηθεί.</p> |
|  | <p>Μια καλή πρακτική είναι η χρήση των tags γιατί αποφεύγεται η ανάγκη για χειροκίνητη προσθήκη GameObjects σε public ιδιότητες μέσω drag and drop, εξοικονομώντας έτσι χρόνο όταν χρησιμοποιείτε ο ίδιος κώδικας για πολλά GameObjects.</p> |

Οι ετικέτες χρησιμοποιούνται ως λέξεις αναφοράς που μπορούν να αντιστοιχιστούν σε ένα ή περισσότερα αντικείμενα του παιχνιδιού. Για παράδειγμα, η ετικέτα <<Player>> μπορεί να αντιστοιχιστεί σε χαρακτήρες που ελέγχονται από τον παίκτη, ενώ η ετικέτα <<Enemy>> μπορεί να αντιστοιχιστεί σε χαρακτήρες που δεν ελέγχονται από τον παίκτη. Σε μια σκηνή, μπορεί να οριστούν αντικείμενα που ο παίκτης μπορεί να συλλέξει με την ετικέτα

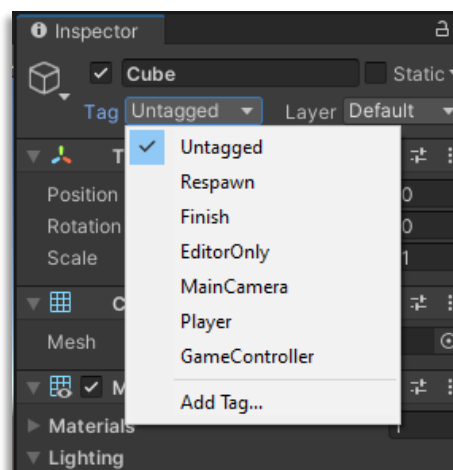
<<Collectable>>. Η χρήση ετικετών βοηθά στον προσδιορισμό των GameObjects μέσα στα c# scripts. Οι ετικέτες είναι χρήσιμες για τον έλεγχο στους collision controllers, όπου καθορίζεται εάν ο παίκτης αλληλοεπιδρά με έναν εχθρό, ένα prop ή ένα collectible αντικείμενο. Ο χρήστης μπορεί να χρησιμοποιήσει τη μέθοδο `GameObject.FindWithTag()` για να αναζητήσει οποιοδήποτε GameObject που περιέχει μια συγκεκριμένη ετικέτα που έχει καθορίσει. Το παρακάτω παράδειγμα χρησιμοποιεί τη μέθοδο `GameObject.FindWithTag()`. Ενεργοποιεί το `respawnPrefab` στις θέσεις των GameObjects με την ετικέτα "Respawn".

```
using UnityEngine;
using System.Collections;

public class Example : MonoBehaviour {
    public GameObject respawnPrefab;
    public GameObject respawn;
    void Start() {
        if (respawn == null)
            respawn = GameObject.FindWithTag("Respawn");

        Instantiate(respawnPrefab, respawn.transform.position, respawn.transform.rotation) as GameObject;
    }
}
```

Η Unity περιλαμβάνει ορισμένες ενσωματωμένες ετικέτες που δεν εμφανίζονται στη Διαχείριση ετικετών και παρουσιάζονται στην παρακάτω εικόνα.



Εικόνα 73. Inspector Tags Attribute

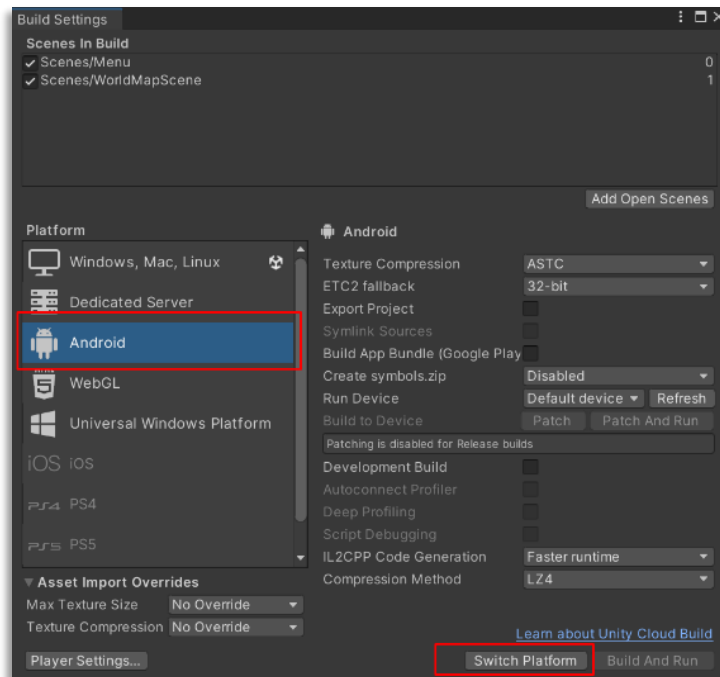
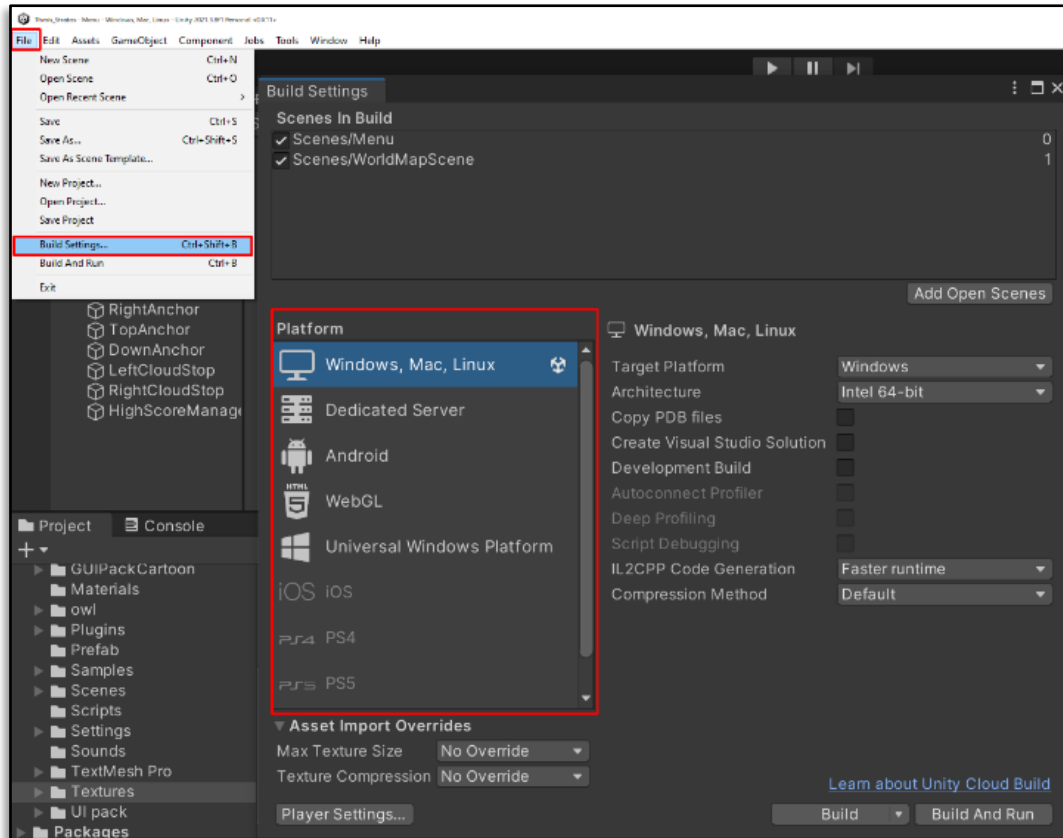
Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/Tags.html>

Τα Layers στον Unity Editor είναι ένας τρόπος δημιουργίας ομάδων αντικειμένων με κοινά χαρακτηριστικά. Τα Layers χρησιμοποιούνται κυρίως για περιορισμό λειτουργιών, όπως το raycasting ή το rendering, έτσι ώστε να εφαρμόζονται μόνο στις σχετικές ομάδες αντικειμένων. Τα Layers που έχουν επισημανθεί ως Builtin Layer είναι προεπιλεγμένα layers που χρησιμοποιεί το Unity και δεν μπορούν να τροποποιηθούν. Ο χρήστης μπορεί να προσαρμόσει τα layers που έχουν επισημανθεί ως User Layer .


4.9 Android and Windows Build Settings

Οι ρυθμίσεις κατασκευής Android και Windows χρησιμοποιούνται για τη διαμόρφωση και δημιουργία της εφαρμογής για αντίστοιχες συσκευές. Ακολουθήστε τα παρακάτω βήματα για να προβάλετε τις ρυθμίσεις κατασκευής Android και Windows:

- ❖ Μενού Αρχείο > Ρυθμίσεις κατασκευής.(File > Build Settings)
- ❖ Από τη λίστα των διαθέσιμων συσκευών στο παράθυρο Πλατφόρμα, ο χρήστης επιλέγει την αντίστοιχη επιλογή.
- ❖ Εάν το κουμπί Build δεν είναι ορατό και το Build And Run είναι απενεργοποιημένο (γκριζαρισμένο), ο χρήστης πρέπει να επιλέξει την επιλογή Switch Platform (Εναλλαγή πλατφόρμας). Αυτό θα ενεργοποιήσει τις αντίστοιχες ρυθμίσεις κατασκευής και θα έχουν επίδραση στη διαδικασία δημιουργίας.



Εικόνα 74. Build Settings

| | |
|---|---|
|  | <p>Εάν το Android ή το Windows είναι απενεργοποιημένο (γκριζαρισμένο), ο χρήστης θα πρέπει να κάνει τα απαραίτητα βήματα για τη ρύθμιση του αντίστοιχου περιβάλλοντος πχ. το Android environment setup.</p> |
|---|---|

Ακολουθώντας αυτά τα βήματα, μπορεί να ανοίξει τις ρυθμίσεις κατασκευής και να προχωρήσει στη διαμόρφωση των απαιτούμενων ρυθμίσεων για την εφαρμογή στην πλατφόρμα που στοχεύει.

Πίνακας 88. Build Settings (Android)

| Setting | Description |
|--------------------------------|---|
| Texture Compression | <p>The texture compression format to use for the build. The options are:</p> <ul style="list-style-type: none"> • Use Player Settings: Uses the texture compression format you set in Player Settings • ETC (GL ES 2.0): Uses ETC format. • ETC2 (GL ES 3.0): Uses ETC2 format. • ASTC: Uses ASTC format. • DXT (Tegra): Uses DXT format. • PVRTC (PowerVR): Uses PVRTC format (deprecated). You can also change this setting from a script or using the <code>-setDefaultPlatformTextureFormat</code> command-line switch. Note: If you enable texture compression targeting, Unity disables this setting. For more information about texture compression formats, refer to Recommended, default, and supported texture compression formats, by platform. |
| Export Project | <p>Indicates whether to export the Unity project as a Gradle project that you can import into Android Studio. For more information, refer to Exporting an Android project.</p> |
| Symlink Sources | <p>Indicates whether to share Java and Kotlin source files between the Unity and exported Gradle project.</p> <p>Enable this setting to create symbolic links so the Gradle project references Java and Kotlin source files in the Unity project. This helps you test and iterate Java and Kotlin code because it means any changes you make to Java and Kotlin source files in the exported Gradle project persist if you re-export the Unity project. Disable this setting to make Unity copy Java and Kotlin source files from the Unity project into the exported Gradle project. You can only interact with this setting if you enable Export Project.</p> |
| Build App Bundle (Google Play) | <p>Indicates whether to build the application as an Android App Bundle (AAB) to distribute on Google Play. If you enable this setting, Unity builds the application as an AAB. If you disable this setting, Unity builds the application as an APK. The setting only appears if you disable Export Project.</p> |
| Export for App Bundle | <p>Indicates whether to configure the exported Gradle project to build as an Android App Bundle. The setting only appears if you enable Export Project.</p> |
| Create symbols.zip | <p>Specifies how Unity generates a symbols package when it builds your application. The options are:</p> <ul style="list-style-type: none"> • Disabled: Unity doesn't generate a symbols package. • Public: Unity generates a public symbols package for your application. • Debugging: Unity generates a debugging symbols package for your application. |
| Run Device | <p>Specifies which attached device to test the build on. Devices connected via USB should appear in the list automatically. If you connect a new device or don't find an attached device in the list,</p> |

| Setting | Description |
|----------------------------------|--|
| | click Refresh . To connect a new device wirelessly over Android Debug Bridge, select the <Enter IP> option. For more information, refer to Debugging on an Android device or Debugging on a ChromeOS device , depending on the platform you want to build for. |
| Build to Device | A build pipeline that doesn't create a full build and instead deploys single files that changed since the last patch directly to the device. Patch deploys changed files to the devices and Patch And Run deploys changed files and then runs the application on the device. For more information, refer to Application patching . You can only interact with this setting if you enable Development Build . |
| Development Build | Indicates whether to include scripting debug symbols and the Profiler in your build. Use this setting when you want to test your application. When you select this option, Unity sets the DEVELOPMENT_BUILD scripting define. Your build then includes pre-processor directives that set DEVELOPMENT_BUILD as a condition. For more information, refer to Platform-dependent compilations . |
| Autoconnect Profiler | Indicates whether to automatically connect the Profiler to the built application when it runs. You can only interact with this setting if you enable Development Build . |
| Deep Profiling Support | Indicates whether to enable Deep Profiling in the Profiler. This makes the Profiler instrument every function call in your application so it returns more detailed profiling data. This option might slow down script execution. You can only interact with this setting if you enable Development Build . |
| Script Debugging | Indicates whether to allow debugging for the application's script code. You can only interact with this setting if you enable Development Build . |
| Wait For Managed Debugger | Indicates whether the application waits for a debugger to attach before it runs any script code. This setting only appears if you enable Script Debugging . Also, you can only interact with this setting if you enable Development Build . |
| Compression Method | Specifies the method Unity uses to compress the data in your Project when it builds the Player. This includes Assets , Scenes , Player settings , and GI data . The options are: <ul style="list-style-type: none"> • Default: Uses ZIP which gives slightly better compressions results than LZ4 and LZ4HC but data is slower to decompress. • LZ4: Uses LZ4, a fast compression format that's useful for development builds. Using LZ4 compression instead of ZIP can significantly improve loading time for Unity applications. For more information, refer to BuildOptions.CompressWithLz4. • LZ4HC: A high compression variant of LZ4 that's slower to build but produces better results for release builds. Using LZ4HC compression instead of ZIP can significantly improve loading time for Unity applications. For more information, refer to BuildOptions.CompressWithLz4. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/android-build-settings.html>

Πίνακας 89. Build Settings (Windows)

| Setting | Function |
|--|--|
| Target platform | Select Windows to build your app for the Windows platform. |
| Architecture | Select the CPU to build for (only applies to Build And Run). |
| <input type="checkbox"/> Intel 64-bit | Intel/AMD 64-bit CPU architecture. |
| <input type="checkbox"/> Intel 32-bit | Intel 32-bit CPU architecture. |
| <input type="checkbox"/> ARM 64-bit | ARM 64-bit CPU architecture. |
| Copy PDB files | Enable this setting to include Microsoft program database (PDB) files in the built Windows Player. PDB files contain debugging information for your application, but might increase the size of your Player. For more information, see Windows debugging . |

| Setting | Function |
|---|--|
| Create Visual Studio Solution | Enable this setting to generate Visual Studio solution files for the project such that you can build the project from Microsoft Visual Studio directly. |
| Development Build | Enable this setting to include scripting debug symbols and the Profiler in your build. When you enable this, Unity sets the DEVELOPMENT_BUILD#define for testing purposes. When you select Development Build , Autoconnect Profiler, Deep Profiling Support, Script Debugging, and Wait for Managed Debugger settings also become available. |
| Autoconnect Profiler | Enable this setting to automatically connect the Unity Profiler to your build. |
| Deep Profiling Support | Enable this setting to select Deep Profiling in the Profiler. This makes the Profiler instrument every function call in your application and returns more detailed profiling data. When you enable Deep Profiling Support, it might slow down script execution. |
| Script debugging | Enable this setting to allow your script code to be debugged. Not available on WebGL . |
| Wait for Managed Debugger | Enable this setting to be prompted to attach a debugger before Unity executes any script code. |
| Compression Method (Not available on Lumin or WebGL platforms) | Compress the data in your application when you build your application. This includes Assets , Scenes , Player settings, and GI data . Choose from the following methods: |
| Default | On Windows, Mac, Linux, and iOS, there is no compression by default. On Android, the default compression is ZIP, which provides better compression results than LZ4HC; however, data is slower to decompress. |
| LZ4 | A fast compression format that's useful for development builds. For more information, see BuildOptions.CompressWithLz4 . |
| LZ4HC | A high compression variant of LZ4 that's slower to build but produces better results for release builds. For more information, see BuildOptions.CompressWithLz4HC . |

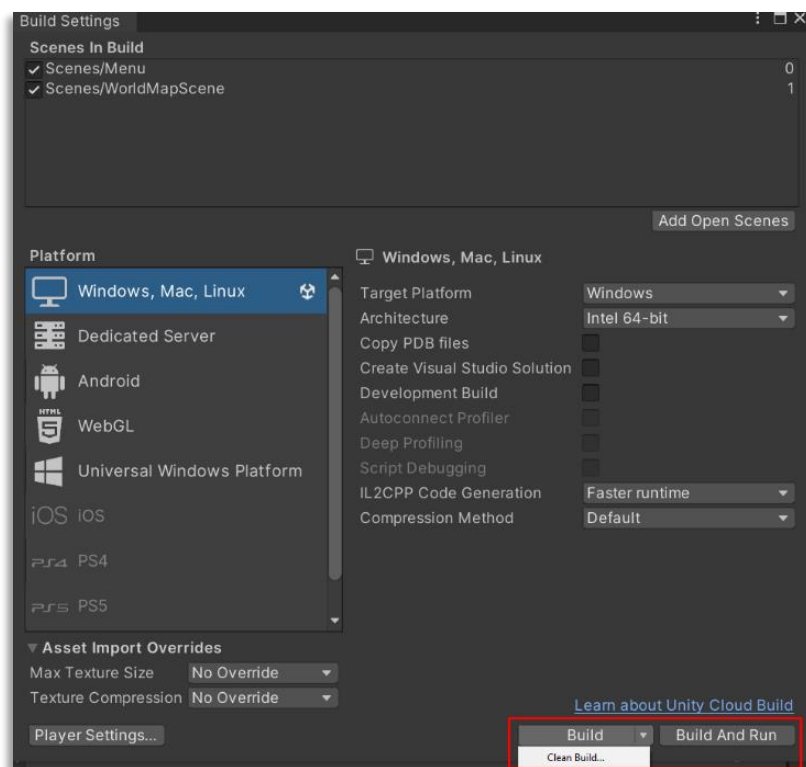
Πηγή : <https://docs.unity3d.com/2023.2/Documentation/Manual/WindowsStandaloneBinaries.html>

Η επιλογή **Asset Import Overrides** είναι υπεύθυνη για πιο γρήγορους χρόνους εναλλαγής πλατφόρμας και εισαγωγής ψηφιακών στοιχείων. Αυτός ο τρόπος τοπικά παρακάμπτει όλες τις ρυθμίσεις εισαγωγής textures πράγμα που πρέπει να αποφεύγετε όταν η εφαρμογή του χρήστη είναι σε κατάσταση final build και θα πρέπει να χρησιμοποιείτε μόνο για το στάδιο του development.

Πίνακας 90. Asset Import Overrides Options

| Setting | Function |
|-------------------------|---|
| Max Texture Size | Override the maximum imported texture size. Unity imports textures in the lower of two values: this value, or the Max Size value specified in Texture import settings . <ul style="list-style-type: none"> • Max 64 • Max 128 • Max 256 • Max 512 • Max 1024 • Max 2048 |

| Setting | Function |
|------------------------------|---|
| | <ul style="list-style-type: none"> The time it takes to import a texture is roughly proportional to the amount of pixels it contains; therefore, reducing maximum allowed texture size can speed up the import times. As this setting might result in lower resolution textures, use it only for development purposes. |
| Texture Compression | Override the texture compression options set in Texture import settings . This only affects textures imported into one of the compressed texture formats . |
| <i>Force Fast Compressor</i> | Use a faster yet lower quality texture compression mode for formats where this is applicable (BC7, BC6H, ASTC, ETC, ETC2). Typically, this results in more compression artifacts, but for many formats the compression itself is many times faster (2 to 20 times faster). This setting also disables the Crunch texture compression format on any textures that have it. The effect is the same as if all textures had their Compressor Quality set to Fast setting in their platform's Texture import settings . |
| <i>Force Uncompressed</i> | Don't compress the textures; use uncompressed formats instead. Note that while this is faster to import because it skips the whole texture compression process, the resulting textures take up more memory, game data size, and can impact rendering performance. Texture import settings . This option has the same effect as all textures that have their Compression set to None in their platforms' Texture Import settings . |



Εικόνα 75. Windows Clean Build and Incremental.

Ο χρήστης αφού έχει καταλήξει με τις υπόλοιπες ρυθμίσεις της μηχανής το τελευταίο βήμα για την δημιουργία της αφορά τον τρόπο της κατασκευής της. Εδώ υπάρχουν δυο τρόποι υλοποίησης :

1. **Build** :Ο προεπιλεγμένος τρόπος είναι μια αυξητική μέθοδος εκτός της πρώτης φοράς η οποία είναι πάντα πλήρης μη αυξητική η λεγομένη καθαρή κατασκευή. Αν για κάποιο λόγο ο χρήστης θέλει να κάνει μια καθαρή κατασκευή για δεύτερη φορά μπορεί να την επιβάλει επιλέγοντας το βελάκι στο κουμπί build και πατώντας την επιλογή clean build από το αναπτυσσόμενο μενού που εμφανίζεται
2. **Build and Run** : Εδώ ο χρήστης χρησιμοποιώντας τον αυξητικό τρόπο κατασκευής της εφαρμογής επιλέγει κατόπιν της δημιουργίας της να εκτελεστεί και η εγκατάστασή της στην πλατφόρμα της επιλογής του.

Τα εκτελέσιμα αρχεία που δημιουργούνται σε περιβάλλον **Windows** όπου το ProjectName είναι το όνομα της εφαρμογής είναι τα εξής :

- ProjectName.exe : Αυτό περιλαμβάνει το σημείο εισόδου του προγράμματος που καλεί τη μηχανή Unity κατά την εκκίνησή της.
- UnityPlayer.dll : Το αρχείο DLL περιέχει τον εγγενή κώδικα της μηχανής Unity. Αυτό το αρχείο έχει υπογραφεί με το πιστοποιητικό της Unity Technologies, το οποίο επιβεβαιώνει την ακεραιότητα της μηχανής και εξασφαλίζει ότι δεν έχει αλλοιωθεί από κακόβουλους χρήστες ή εφαρμογές.
- *.pdb files : Αρχεία συμβόλων για αποσφαλμάτωση. Η Unity τα αντιγράφει στον κατάλογο κατασκευής, εάν ο χρήστης ενεργοποιήσει την επιλογή Αντιγραφή αρχείων PDB στο παράθυρο Ρυθμίσεις κατασκευής..
- WinPixEventRuntime.dll : Αυτό το DLL ενεργοποιεί την υποστήριξη της εισαγωγής του PIX στα Windows (beta). Η Unity δημιουργεί αυτό το αρχείο μόνο αν ο χρήστης ενεργοποιήσει την επιλογή Development Build στο παράθυρο Build Settings.
- ProjectName_Data folder : Αυτός ο φάκελος περιέχει όλα τα δεδομένα που απαιτούνται για την εκτέλεση της εφαρμογής.

Τα εκτελέσιμα αρχεία που δημιουργούνται σε περιβάλλον **Android** όπου το ProjectName είναι το όνομα της εφαρμογής είναι τα εξής :

- ProjectName.apk : Το εκτελέσιμο αρχείο του project είναι η εφαρμογή.

4.10 C# και Unity [23][4][16]

Εδώ θα αναλυθούν κάποιες βασικές κλάσεις με τις εντολές, έννοιες και μεθόδους τους για να υπάρχει μια πιο σφαιρική εικόνα των δυνατοτήτων της μηχανής Unity. Μια πλήρη αναφορά υπάρχει στην ηλεκτρονική διεύθυνση στο [Unity Documentation](#).

4.10.1 Object

Η Unity χρησιμοποιεί την κλάση UnityEngine.Object ως βασική κλάση για όλα τα αντικείμενα στα οποία μπορεί να γίνει αναφορά. Με τη δημιουργία μιας νέας κλάσης από το MonoBehaviour, το ScriptableObject ή το ScriptableObject, ο χρήστης μπορεί να ορίσει προσαρμοσμένους τύπους αντικειμένων Unity στα σενάρια του. Οι δημόσιες μεταβλητές που προέρχονται από την κλάση Object εμφανίζονται στον editor ως θέσης έτοιμες να δεχτούν τιμές μέσα από το γραφικό περιβάλλον. Συνήθως, τα scripts χρησιμοποιούν τύπους που προέρχονται από αυτήν την κλάση, όπως GameObject, Material και Mesh, προκειμένου να έχουν πρόσβαση σε συγκεκριμένες ιδιότητες και μεθόδους για αυτούς τους τύπους. Ωστόσο, ορισμένα API έχουν σχεδιαστεί έτσι ώστε να λειτουργούν με οποιοδήποτε αντικείμενο της Unity, επομένως η κλάση Object εμφανίζεται ως τύπος στις υπογραφές αυτών των μεθόδων. Παραδείγματα αυτών των μεθόδων είναι το Resources.LoadAll, το EditorJsonUtility.ToJson και το SerializedObject.

Πίνακας 91. Ιδιότητες Object

| Name | Description |
|---------------------------|--|
| hideFlags | Should the object be hidden, saved with the Scene or modifiable by the user? |
| name | The name of the object. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/ScriptReference/Object.html>

Πίνακας 92. Δημόσιες Μέθοδοι

| Name | Description |
|-------------------------------|-------------------------------------|
| GetInstanceID | Gets the instance ID of the object. |
| ToString | Returns the name of the object. |

Πίνακας 93. Στατικές Μέθοδοι

| Name | Description |
|---------------------------------------|---|
| Destroy | Removes a GameObject, component or asset. |
| DestroyImmediate | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| DontDestroyOnLoad | Do not destroy the target Object when loading a new Scene. |
| FindAnyObjectByType | Retrieves any active loaded object of Type type. |
| FindFirstObjectByType | Retrieves the first active loaded object of Type type. |
| FindObjectsByType | Retrieves a list of all loaded objects of Type type. |
| Instantiate | Clones the object original and returns the clone. |

Πίνακας 94. Τελεστές

| Name | Description |
|-----------------------------|---|
| bool | Does the object exist? |
| operator != | Compares if two objects refer to a different object. |
| operator == | Compares two object references to see if they refer to the same object. |

4.10.2 GameObject

Το βασικό δομικό στοιχείο όλων των παιχνιδιών της Unity είναι το **GameObject**. Είτε πρόκειται για έναν χαρακτήρα, ένα όπλο, ένα κτίριο, ένα στοιχείο διεπαφής, είτε ακόμη και για την κάμερα ή το `directional` φως, κάθε μία από αυτές τις οντότητες κατασκευάζεται από το `GameObject`. Ενώ τα περισσότερα αντικείμενα στο παιχνίδι εισάγονται από εξωτερικά εργαλεία δημιουργίας ψηφιακών στοιχείων, όπως το Blender, το Maya, το 3ds Max, το Photoshop και άλλα, το Unity παρέχει μια συλλογή βασικών αντικειμένων τα λεγόμενα `primals` με τα οποία μπορεί ο χρήστης να εργαστεί σε επίπεδο `prototyping`. Οι κύβοι, οι σφαίρες, τα `planes` και οι κύλινδροι είναι μερικά από τα βασικά αντικείμενα που είναι διαθέσιμα.

Πίνακας 95. Ιδιότητες GameObject

| Name | Description |
|-----------------------------------|---|
| activeInHierarchy | Defines whether the GameObject is active in the Scene. |
| activeSelf | The local active state of this GameObject. (Read Only) |
| isStatic | Gets and sets the GameObject's StaticEditorFlags. |
| layer | The layer the GameObject is in. |
| scene | Scene that the GameObject is part of. |
| sceneCullingMask | Scene culling mask Unity uses to determine which scene to render the GameObject in. |
| tag | The tag of this GameObject. |
| transform | The Transform attached to this GameObject. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/ScriptReference/GameObject.html>

Πίνακας 96. Constructors

| Name | Description |
|----------------------------|---------------------------------------|
| GameObject | Creates a new GameObject, named name. |

Πίνακας 97. Δημόσιες Μέθοδοι

| Name | Description |
|---|--|
| AddComponent | Adds a component class of type componentType to the GameObject. C# Users can use a generic version. |
| BroadcastMessage | Calls the method named methodName on every MonoBehaviour in this GameObject or any of its children. |
| CompareTag | Is this GameObject tagged with tag ? |
| GetComponent | Gets a reference to a component of type T on the specified GameObject. |
| GetComponentInChildren | Gets a reference to a component of type T on the specified GameObject, or any child of the GameObject. |
| GetComponentInParent | Gets a reference to a component of type T on the specified GameObject, or any parent of the GameObject. |
| GetComponents | Gets references to all components of type T on the specified GameObject. |
| GetComponentsInChildren | Gets references to all components of type T on the specified GameObject, and any child of the GameObject. |
| GetComponentsInParent | Gets references to all components of type T on the specified GameObject, and any parent of the GameObject. |
| SendMessage | Calls the method named methodName on every MonoBehaviour in this GameObject. |
| SendMessageUpwards | Calls the method named methodName on every MonoBehaviour in this |

| | |
|---------------------------------|---|
| | GameObject and on every ancestor of the behaviour. |
| SetActive | Activates/Deactivates the GameObject, depending on the given true or false value. |
| TryGetComponent | Gets the component of the specified type, if it exists. |

Πίνακας 98. Στατικές Μέθοδοι

| Name | Description |
|--|--|
| CreatePrimitive | Creates a GameObject with a primitive mesh renderer and appropriate collider. |
| Find | Finds a GameObject by name and returns it. |
| FindGameObjectsWithTag | Returns an array of active GameObjects tagged tag. Returns empty array if no GameObject was found. |
| FindWithTag | Returns one active GameObject tagged tag. Returns null if no GameObject was found. |

Τα μέλη που κληρονομούνται από την GameObject είναι :

Πίνακας 99. Κληρονομημένες Ιδιότητες

| Name | Description |
|---------------------------|--|
| hideFlags | Should the object be hidden, saved with the Scene or modifiable by the user? |
| name | The name of the object. |

Πίνακας 100. Κληρονομημένες Δημόσιες Μέθοδοι

| Name | Description |
|-------------------------------|-------------------------------------|
| GetInstanceID | Gets the instance ID of the object. |
| ToString | Returns the name of the object. |

Πίνακας 101. Κληρονομημένες Στατικές Μέθοδοι

| Name | Description |
|-------------------------------------|---|
| Destroy | Removes a GameObject, component or asset. |
| DestroyImmediate | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| DontDestroyOnLoad | Do not destroy the target Object when loading a new Scene. |
| FindAnyObjectByType | Retrieves any active loaded object of Type type. |

| | |
|---------------------------------------|--|
| FindFirstObjectByType | Retrieves the first active loaded object of Type type. |
| FindObjectsByType | Retrieves a list of all loaded objects of Type type. |
| Instantiate | Clones the object original and returns the clone. |


Πίνακας 102. Τελεστές

| Name | Description |
|-----------------------------|---|
| bool | Does the object exist? |
| operator != | Compares if two objects refer to a different object. |
| operator == | Compares two object references to see if they refer to the same object. |

4.10.3 MonoBehaviour



Η κλάση **MonoBehaviour** κληρονομείται από προεπιλογή κατά τη δημιουργία και τη σύνδεση ενός component σε ένα αντικείμενο παιχνιδιού. Κληρονομώντας από αυτή τη βασική κλάση, ο κώδικας που ορίζει τα συστήματα παιχνιδιών επωφελείται από τις χρήσιμες συναρτήσεις που παρέχει η μηχανή Unity. Τα MonoBehaviours υπάρχουν πάντα ως component ενός GameObject και μπορούν να δημιουργηθούν με την εντολή `GameObject.AddComponent`. Αντικείμενα που πρέπει να υπάρχουν ανεξάρτητα από ένα GameObject θα πρέπει να προέρχονται από το `ScriptableObject`. Η έννοια της κληρονομικότητας παίζει σημαντικό ρόλο στον OOP με τη γλώσσα C#. Επιτρέπει τη δημιουργία νέων κλάσεων, scripts ή components based σε προ υπάρχοντα component, κληρονομώντας έτσι τη λειτουργικότητά τους χωρίς την ανάγκη περιττής επανεγγραφής κώδικα. Η σωστή χρήση της κληρονομικότητας επιτρέπει την ανάπτυξη αποδοτικού και αξιόπιστου κώδικα. Κατά την εκκίνηση της δέσμης ενεργειών, αρκετές προκαθορισμένες λειτουργίες είναι διαθέσιμες για χρήση.

Η μέθοδος **Awake()** καλείται όταν δημιουργείται ένα αντικείμενο στη σκηνή και χρησιμεύει ως κατάλληλο μέρος για την αρχικοποίηση τιμών, αν και ορισμένα στοιχεία και αντικείμενα ενδέχεται να μην είναι προσβάσιμα σε αυτό το σημείο. Η `Awake()` θα τρέξει ανεξάρτητα αν είναι Active/enabled το GameObject στην σκηνή.

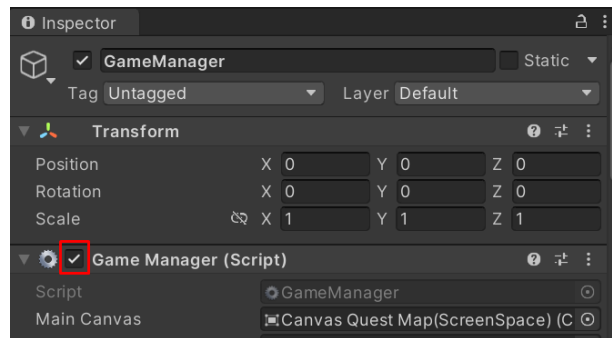
| | |
|---|--|
|  | <p>Ο κώδικας αρχικοποίησης που εξαρτάται από άλλα στοιχεία θα πρέπει να τοποθετείται στη συνάρτηση Start().</p> |
|---|--|

Η μέθοδος **Start()**, από την άλλη πλευρά, καλείται στην πρώτη εμφάνιση του αντικειμένου στη σκηνή και μπορεί να χρησιμοποιηθεί για εργασίες αρχικοποίησης που βασίζονται στη φόρτωση άλλων objects και components. Η Start() θα εκτελεστεί κατά την ενεργοποίηση της μεθόδου και πάντα μετά την Awake() δηλαδή όταν το GameObject είναι Active /enabled στην σκηνή και αφού έχει τελειώσει να φορτώνεται η Awake().

Η μέθοδος **Update()** καλείται σε κάθε καρτέ ή όσο πιο συχνά γίνεται με βάση το ρυθμό καρτέ και συνήθως περιέχει το μεγαλύτερο μέρος της λογικής ενός component. Είναι ζωτικής σημασίας να διατηρηθεί η οργάνωση στη συνάρτηση Update() για να αποφευχθεί η υπερβολική πολυπλοκότητά της, οπότε θα πρέπει να χρησιμοποιούνται υπό-συναρτήσεις για τον αποτελεσματικό διαχωρισμό και την οργάνωση των update calls.

| | |
|---|---|
|  | <p>Μια καλή πρακτική είναι να αποφεύγει ο χρήστης να ζητάει από την μηχανή να τρέξει μια μέθοδο για να ελέγξει κάποια κατάσταση στην Update() και να χρησιμοποιήσει είτε events, είτε delegates, είτε Coroutines, είτε Physics callbacks είτε Input Events.</p> |
|  | <p>Η μέθοδος Update() θέλει προσοχή γιατί αν γίνει αλόγιστη χρήση θα έχει δραματική επίπτωση στην απόδοση του παιχνιδιού και κατά επέκταση στο gameplay.</p> |

Όταν μέσα σε ένα component ο χρήστης χρησιμοποιήσει μια από τις μεθόδους [Start\(\)](#), [Update\(\)](#), [FixedUpdate\(\)](#), [LateUpdate\(\)](#), [OnGUI\(\)](#), [OnDisable\(\)](#), [OnEnable\(\)](#) τότε εμφανίζεται ένα checkbox για την χειροκίνητη ενεργοποίηση και απενεργοποίησή του μέσα από τον editor. Εάν δεν υπάρχει κάτι από όλα αυτά τότε το checkbox δεν εμφανίζεται.



Εικόνα 76. Checkbox for enabling or disabling in editor

Οι προαναφερόμενες μέθοδοι εκτελούν τις εξής λειτουργίες :

```

void Start()
{
    // Κώδικας αρχικοποίησης
    // Αυτός ο κώδικας εκτελείται μία φορά, όταν το αντικείμενο γίνεται ενεργό στη σκηνή
}
void Update()
{
    // Κώδικας ενημέρωσης
    // Αυτός ο κώδικας εκτελείται κάθε καρέ και χρησιμοποιείται για την τακτική ενημέρωση και τη λογική του παιχνιδιού
}
void FixedUpdate()
{
    // Κώδικας σταθερής ενημέρωσης
    // Αυτός ο κώδικας εκτελείται σε σταθερά διαστήματα, συγχρονισμένος με τη μηχανή φυσικής
    // Χρησιμοποιείται για φυσικά υπολογισμένα και ενημερώσεις σχετικές με τη φυσική
}
void LateUpdate()
{
    // Κώδικας καθυστερημένης ενημέρωσης
    // Αυτός ο κώδικας εκτελείται μετά από όλες τις κλήσεις των Update() συναρτήσεων
    // Χρησιμοποιείται για κίνηση κάμερας και άλλες ενέργειες που εξαρτώνται από την ενημέρωση των θέσεων των
    αντικειμένων
}
void OnGUI()
{
    // Κώδικας GUI
    // Χρησιμοποιείται για την απεικόνιση στοιχείων γραφικού περιβάλλοντος στην οθόνη
    // Καλείται πολλές φορές ανά καρέ, μετά τις κλήσεις των Update() συναρτήσεων
}
void OnDisable()
{
    // Κώδικας απενεργοποίησης
    // Αυτός ο κώδικας εκτελείται όταν το αντικείμενο γίνεται ανενεργό ή απενεργοποιημένο στη σκηνή
    // Χρησιμοποιείται για την απελευθέρωση πόρων και την εκτέλεση ενεργειών πριν το αντικείμενο απενεργοποιηθεί
}
void OnEnable()
{
    // Κώδικας ενεργοποίησης
    // Αυτός ο κώδικας εκτελείται όταν το αντικείμενο γίνεται ενεργό ή ενεργοποιημένο στη σκηνή
    // Χρησιμοποιείται για αρχικοποίηση και ρύθμιση της κατάστασης του αντικειμένου
}

```

Ιδιότητες της MonoBehaviour είναι :

Πίνακας 103. Ιδιότητες MonoBehaviour

| Name | Description |
|--|--|
| destroyCancellationToken | Cancellation token raised when the MonoBehaviour is destroyed (Read Only). |
| didAwake | Returns a boolean value which represents if Awake was called. |
| didStart | Returns a boolean value which represents if Start was called. |
| runInEditMode | Allow a specific instance of a MonoBehaviour to run in edit mode (only available in the editor). |
| useGUILayout | Disabling this lets you skip the GUI layout phase. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/ScriptReference/MonoBehaviour.html>

Δημόσιες Μέθοδοι της MonoBehaviour είναι :

Πίνακας 104. Δημόσιες Μέθοδοι

| Name | Description |
|-----------------------------------|---|
| CancelInvoke | Cancels all Invoke calls on this MonoBehaviour. |
| Invoke | Invokes the method methodName in time seconds. |
| InvokeRepeating | Invokes the method methodName in time seconds, then repeatedly every repeatRate seconds. |
| IsInvoking | Is any invoke on methodName pending? |
| StartCoroutine | Starts a Coroutine. |
| StopAllCoroutines | Stops all coroutines running on this behaviour. |
| StopCoroutine | Stops the first coroutine named methodName, or the coroutine stored in routine running on this behaviour. |

Στατικές Μέθοδοι της MonoBehaviour είναι :

Πίνακας 105. Στατικές Μέθοδοι

| Name | Description |
|-----------------------|---|
| print | Logs message to the Unity Console (identical to Debug.Log). |

Μηνύματα της MonoBehaviour είναι :

Πίνακας 106. Messages

| Name | Description |
|------------------------------|--|
| Awake | Awake is called when an enabled script instance is being loaded. |
| FixedUpdate | Frame-rate independent MonoBehaviour.FixedUpdate message for physics calculations. |
| LateUpdate | LateUpdate is called every frame, if the Behaviour is enabled. |
| OnAnimatorIK | Callback for setting up animation IK (inverse kinematics). |

| | |
|---|--|
| OnAnimatorMove | Callback for processing animation movements for modifying root motion. |
| OnApplicationFocus | Sent to all GameObjects when the player gets or loses focus. |
| OnApplicationPause | Sent to all GameObjects when the application pauses. |
| OnApplicationQuit | Sent to all GameObjects before the application quits. |
| OnAudioFilterRead | If OnAudioFilterRead is implemented, Unity will insert a custom filter into the audio DSP chain. |
| OnBecameInvisible | OnBecameInvisible is called when the renderer is no longer visible by any camera. |
| OnBecameVisible | OnBecameVisible is called when the renderer became visible by any camera. |
| OnCollisionEnter | OnCollisionEnter is called when this collider/rigidbody has begun touching another rigidbody/collider. |
| OnCollisionEnter2D | Sent when an incoming collider makes contact with this object's collider (2D physics only). |
| OnCollisionExit | OnCollisionExit is called when this collider/rigidbody has stopped touching another rigidbody/collider. |
| OnCollisionExit2D | Sent when a collider on another object stops touching this object's collider (2D physics only). |
| OnCollisionStay | OnCollisionStay is called once per frame for every Collider or Rigidbody that touches another Collider or Rigidbody. |
| OnCollisionStay2D | Sent each frame where a collider on another object is touching this object's collider (2D physics only). |
| OnConnectedToServer | Called on the client when you have successfully connected to a server. |
| OnControllerColliderHit | OnControllerColliderHit is called when the controller hits a collider while performing a Move. |
| OnDestroy | Destroying the attached Behaviour will result in the game or Scene receiving OnDestroy. |
| OnDisable | This function is called when the behaviour becomes disabled. |
| OnDisconnectedFromServer | Called on the client when the connection was lost or you disconnected from the server. |
| OnDrawGizmos | Implement OnDrawGizmos if you want to draw gizmos that are also pickable and always drawn. |
| OnDrawGizmosSelected | Implement OnDrawGizmosSelected to draw a gizmo if the object is selected. |
| OnEnable | This function is called when the object becomes enabled and active. |
| OnFailedToConnect | Called on the client when a connection attempt fails for some reason. |
| OnFailedToConnectToMasterServer | Called on clients or servers when there is a problem connecting to the MasterServer. |
| OnGUI | OnGUI is called for rendering and handling GUI events. |
| OnJointBreak | Called when a joint attached to the same game object broke. |
| OnJointBreak2D | Called when a Joint2D attached to the same game object breaks. |
| OnMasterServerEvent | Called on clients or servers when reporting events from the MasterServer. |
| OnMouseDown | OnMouseDown is called when the user has pressed the mouse button while over the Collider. |
| OnMouseDrag | OnMouseDrag is called when the user has clicked on a Collider and is still holding down the mouse. |
| OnMouseEnter | Called when the mouse enters the Collider. |
| OnMouseExit | Called when the mouse is not any longer over the Collider. |
| OnMouseOver | Called every frame while the mouse is over the Collider. |

| | |
|--|--|
| OnMouseUp | OnMouseUp is called when the user has released the mouse button. |
| OnMouseUpAsButton | OnMouseUpAsButton is only called when the mouse is released over the same Collider as it was pressed. |
| OnNetworkInstantiate | Called on objects which have been network instantiated with Network.Instantiate. |
| OnParticleCollision | OnParticleCollision is called when a particle hits a Collider. |
| OnParticleSystemStopped | OnParticleSystemStopped is called when all particles in the system have died, and no new particles will be born. New particles cease to be created either after Stop is called, or when the duration property of a non-looping system has been exceeded. |
| OnParticleTrigger | OnParticleTrigger is called when any particles in a Particle System meet the conditions in the trigger module. |
| OnParticleUpdateJobScheduled | OnParticleUpdateJobScheduled is called when a Particle System's built-in update job has been scheduled. |
| OnPlayerConnected | Called on the server whenever a new player has successfully connected. |
| OnPlayerDisconnected | Called on the server whenever a player disconnected from the server. |
| OnPostRender | Event function that Unity calls after a Camera renders the scene. |
| OnPreCull | Event function that Unity calls before a Camera culls the scene. |
| OnPreRender | Event function that Unity calls before a Camera renders the scene. |
| OnRenderImage | Event function that Unity calls after a Camera has finished rendering, that allows you to modify the Camera's final image. |
| OnRenderObject | OnRenderObject is called after camera has rendered the Scene. |
| OnSerializeNetworkView | Used to customize synchronization of variables in a script watched by a network view. |
| OnServerInitialized | Called on the server whenever a Network.InitializeServer was invoked and has completed. |
| OnTransformChildrenChanged | This function is called when the list of children of the transform of the GameObject has changed. |
| OnTransformParentChanged | This function is called when a direct or indirect parent of the transform of the GameObject has changed. |
| OnTriggerEnter | When a GameObject collides with another GameObject, Unity calls OnTriggerEnter. |
| OnTriggerEnter2D | Sent when another object enters a trigger collider attached to this object (2D physics only). |
| OnTriggerExit | OnTriggerExit is called when the Collider other has stopped touching the trigger. |
| OnTriggerExit2D | Sent when another object leaves a trigger collider attached to this object (2D physics only). |
| OnTriggerStay | OnTriggerStay is called once per physics update for every Collider other that is touching the trigger. |
| OnTriggerStay2D | Sent each frame where another object is within a trigger collider attached to this object (2D physics only). |
| OnValidate | Editor-only function that Unity calls when the script is loaded or a value changes in the Inspector. |
| OnWillRenderObject | OnWillRenderObject is called for each camera if the object is visible and not a UI element. |
| Reset | Reset to default values. |
| Start | Start is called on the frame when a script is enabled just before any of the Update methods are called the first time. |
| Update | Update is called every frame, if the MonoBehaviour is enabled. |

Τα μέλη που κληρονομούνται από την MonoBehaviour είναι :

Πίνακας 107. Κληρονομημένες Ιδιότητες

| Name | Description |
|------------------------------------|---|
| enabled | Enabled Behaviours are Updated, disabled Behaviours are not. |
| isActiveAndEnabled | Reports whether a GameObject and its associated Behaviour is active and enabled. |
| gameObject | The game object this component is attached to. A component is always attached to a game object. |
| tag | The tag of this game object. |
| transform | The Transform attached to this GameObject. |
| hideFlags | Should the object be hidden, saved with the Scene or modifiable by the user? |
| name | The name of the object. |

Πίνακας 108. Κληρονομημένες Δημόσιες Μέθοδοι

| Name | Description |
|---|--|
| BroadcastMessage | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| CompareTag | Checks the GameObject's tag against the defined tag. |
| GetComponent | Gets a reference to a component of type T on the same GameObject as the component specified. |
| GetComponentInChildren | Gets a reference to a component of type T on the same GameObject as the component specified, or any child of the GameObject. |
| GetComponentInParent | Gets a reference to a component of type T on the same GameObject as the component specified, or any parent of the GameObject. |
| GetComponents | Gets references to all components of type T on the same GameObject as the component specified. |
| GetComponentsInChildren | Gets references to all components of type T on the same GameObject as the component specified, and any child of the GameObject. |
| GetComponentsInParent | Gets references to all components of type T on the same GameObject as the component specified, and any parent of the GameObject. |
| SendMessage | Calls the method named methodName on every MonoBehaviour in this game object. |
| SendMessageUpwards | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| TryGetComponent | Gets the component of the specified type, if it exists. |
| GetInstanceID | Gets the instance ID of the object. |
| ToString | Returns the name of the object. |

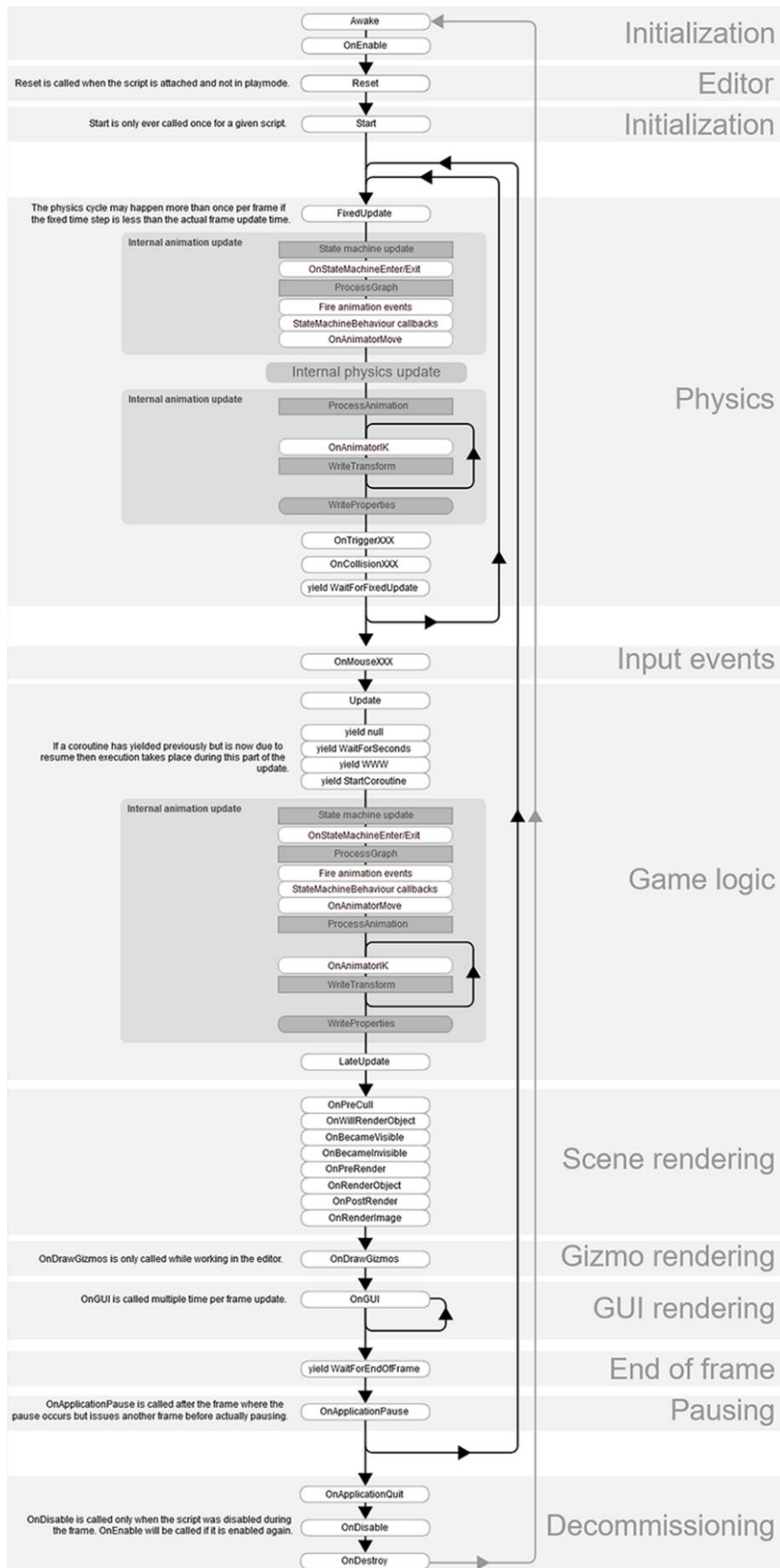
Πίνακας 109. Κληρονομημένες Στατικές Μέθοδοι

| Name | Description |
|---------------------------------------|---|
| Destroy | Removes a GameObject, component or asset. |
| DestroyImmediate | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| DontDestroyOnLoad | Do not destroy the target Object when loading a new Scene. |
| FindAnyObjectByType | Retrieves any active loaded object of Type type. |
| FindFirstObjectByType | Retrieves the first active loaded object of Type type. |
| FindObjectsByType | Retrieves a list of all loaded objects of Type type. |
| Instantiate | Clones the object original and returns the clone. |

Πίνακας 110. Κληρονομημένοι Τελεστές

| Name | Description |
|-----------------------------|---|
| bool | Does the object exist? |
| operator != | Compares if two objects refer to a different object. |
| operator == | Compares two object references to see if they refer to the same object. |

Ακολουθεί ένα διάγραμμα ροής που δείχνει την σειρά προτεραιότητας του κάθε event σε όλη την διάρκεια του κύκλου ζωής μιας εφαρμογής από την εκκίνησή της μέχρι και το κλείσμό της.



Εικόνα 77. Order of Execution for event functions

4.10.4 Component

Βασική κλάση για όλα όσα συνδέονται με ένα GameObject παιχνιδιού. Είναι σημαντικό να σημειωθεί ότι ο κώδικας δεν δημιουργεί ποτέ άμεσα ένα Component (αν εξαιρεθεί το Transform που έχουν όλα τα GameObject καθώς δημιουργούνται μέσα στον ευκλείδειο χώρο του editor). Ο χρήστης γράφει ένα script και στη συνέχεια το προσαρτίζει σε ένα GameObject. Αυτός ο τρόπος προσθέτει επιπλέον λειτουργικότητα στο GameObject παιχνιδιού με τη χρήση συγκεκριμένων script. Υπάρχουν έτοιμα script φτιαγμένα από την μηχανή για να γίνεται πιο γρήγορη η ανάπτυξη της εφαρμογής.

Επιπλέον, ο ScriptableObject είναι ένας τρόπος δημιουργίας script που δεν προσαρτώνται σε κανένα GameObject. Ο προγραμματιστής μπορείτε να χρησιμοποιήσει το ScriptableObject για να δημιουργήσει ανεξάρτητα script που μπορούν να χρησιμοποιηθούν από πολλά αντικείμενα παιχνιδιού.

Πίνακας 111. Ιδιότητες

| Name | Description |
|----------------------------|---|
| gameObject | The game object this component is attached to. A component is always attached to a game object. |
| tag | The tag of this game object. |
| transform | The Transform attached to this GameObject. |

Πηγή : <https://docs.unity3d.com/2023.2/Documentation/ScriptReference/Component.html>

Πίνακας 112. Δημόσιες Μέθοδοι

| Name | Description |
|---|---|
| BroadcastMessage | Calls the method named methodName on every MonoBehaviour in this game object or any of its children. |
| CompareTag | Checks the GameObject's tag against the defined tag. |
| GetComponent | Gets a reference to a component of type T on the same GameObject as the component specified. |
| GetComponentInChildren | Gets a reference to a component of type T on the same GameObject as the component specified, or any child of the GameObject. |
| GetComponentInParent | Gets a reference to a component of type T on the same GameObject as the component specified, or any parent of the GameObject. |
| GetComponents | Gets references to all components of type T on the same GameObject as the component specified. |
| GetComponentsInChildren | Gets references to all components of type T on the same GameObject as the component specified, and any child of the GameObject. |

| | |
|--------------------------------------|--|
| GetComponentInParent | Gets references to all components of type T on the same GameObject as the component specified, and any parent of the GameObject. |
| SendMessage | Calls the method named methodName on every MonoBehaviour in this game object. |
| SendMessageUpwards | Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the behaviour. |
| TryGetComponent | Gets the component of the specified type, if it exists. |

Τα μέλη που κληρονομούνται από την Component είναι :

Πίνακας 113. Κληρονομημένες Ιδιότητες

| Name | Description |
|---------------------------|--|
| hideFlags | Should the object be hidden, saved with the Scene or modifiable by the user? |
| name | The name of the object. |

Πίνακας 114. Κληρονομημένες Δημόσιες Μέθοδοι

| Name | Description |
|-------------------------------|-------------------------------------|
| GetInstanceID | Gets the instance ID of the object. |
| ToString | Returns the name of the object. |

Πίνακας 115. Κληρονομημένες Στατικές Μέθοδοι



| Name | Description |
|---------------------------------------|---|
| Destroy | Removes a GameObject, component or asset. |
| DestroyImmediate | Destroys the object obj immediately. You are strongly recommended to use Destroy instead. |
| DontDestroyOnLoad | Do not destroy the target Object when loading a new Scene. |
| FindAnyObjectByType | Retrieves any active loaded object of Type type. |
| FindFirstObjectByType | Retrieves the first active loaded object of Type type. |
| FindObjectsByType | Retrieves a list of all loaded objects of Type type. |
| Instantiate | Clones the object original and returns the clone. |

Πίνακας 116. Κληρονομημένοι Τελεστές

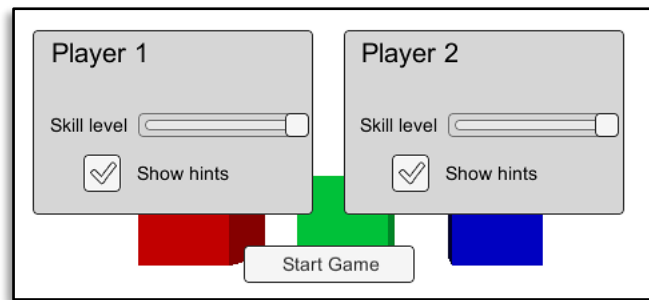
| Name | Description |
|-----------------------------|---|
| bool | Does the object exist? |
| operator != | Compares if two objects refer to a different object. |
| operator == | Compares two object references to see if they refer to the same object. |

4.10.5 Canvas

Ο καμβάς στην Unity αποτελείται από ένα σύστημα που χρησιμοποιείται για τη δημιουργία και την απεικόνιση γραφικών στο παιχνίδι. Μέσω του καμβά, μπορούν να προστεθούν διάφορα στοιχεία, όπως εικόνες, κείμενο, κουμπιά και πλαίσια, σε μια σκηνή του παιχνιδιού. Ο καμβάς χρησιμοποιείται για τη δημιουργία του γραφικού περιβάλλοντος του παιχνιδιού και την αλληλεπίδραση του παίκτη με τα διάφορα στοιχεία. Τα γραφικά που απεικονίζονται στον καμβά μπορούν να αλλάξουν δυναμικά κατά τη διάρκεια του παιχνιδιού, προσφέροντας έτσι δυνατότητες προσαρμογής και αλληλεπίδρασης. Ο καμβάς αποτελεί ένα σημαντικό στοιχείο στη δημιουργία παιχνιδιών με την Unity και η κατανόηση της λειτουργίας του είναι ζωτικής σημασίας για τους προγραμματιστές.

| | |
|---|---|
|  | <p>Μια καλή πρακτική είναι να αποφεύγει ο χρήστης να κάνει animations με την χρήση του animator και αντί αυτού να χρησιμοποιήσει κώδικα ή πακέτα από τρίτους που παρέχουν παρόμοιες δυνατότητες όπως π.χ. το DoTween Pro.</p> |
|  | <p>Κάθε φορά που αλλάζει ένα στοιχείο του UI με κάποιο animation που χρησιμοποίησε τον animator η μηχανή πρέπει να κάνει Render ξανά όλο τον καμβά με όλα τα αντικείμενα μέσα είτε αυτά επηρεάστηκαν είτε όχι. Αν παρόλα αυτά ο χρήστης θέλει να χρησιμοποιήσει τον animator τότε η μόνη λύση είναι να χρησιμοποιήσει πολλούς καμβάδες και να χωρίσει τα UI elements σε ομάδες.</p> |

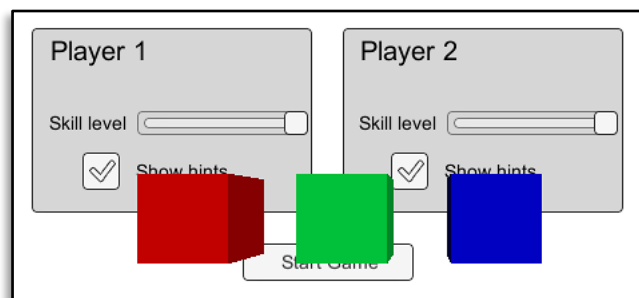
Σε λειτουργία **Screen Space - Overlay**, ο καμβάς προσαρμόζεται κλιμακωτά για να ταιριάζει στην οθόνη και στη συνέχεια απεικονίζεται άμεσα χωρίς να απαιτείται αναφορά στη σκηνή ή σε μια κάμερα (το UI θα εμφανιστεί ακόμα και αν δεν υπάρχει κάμερα στη σκηνή). Όταν αλλάζει το μέγεθος ή η ανάλυση της οθόνης, το UI προσαρμόζει αυτόματα την κλίμακά του για να προσαρμοστεί. Το UI σχεδιάζεται πάνω από οποιοδήποτε άλλα γραφικά, όπως η προβολή της κάμερας.



Εικόνα 78. Παράδειγμα Screen Space Overlay

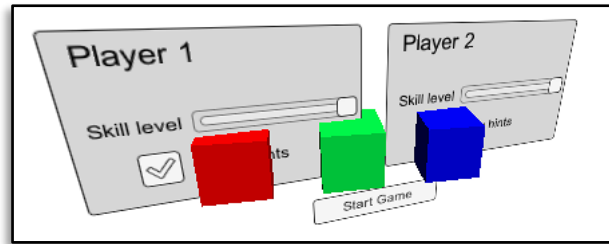
Πηγή : <https://docs.unity3d.com/Packages/com.unity.ugui@2.0/manual/class-Canvas.html>

Σε αυτή τη λειτουργία, ο καμβάς απεικονίζεται ως να έχει σχεδιαστεί σε ένα επίπεδο αντικείμενο που βρίσκεται σε μια συγκεκριμένη απόσταση μπροστά από μια κάμερα. Το μέγεθος του UI στην οθόνη δεν επηρεάζεται από την απόσταση, καθώς πάντα προσαρμόζεται κλιμακωτά ώστε να χωράει ακριβώς εντός του πεδίου θέασης (frustum) της κάμερας. Εάν αλλάξει το μέγεθος ή η ανάλυση της οθόνης ή το πεδίο θέασης της κάμερας, το UI θα προσαρμόζει αυτόματα την κλίμακά του για να προσαρμοστεί. Οποιαδήποτε τρισδιάστατα αντικείμενα στη σκηνή που βρίσκονται πιο κοντά στην κάμερα από το επίπεδο του UI θα εμφανίζονται μπροστά από το UI, ενώ τα αντικείμενα που βρίσκονται πίσω από το επίπεδο θα αποκρύπτονται.



Εικόνα 79. Screen Space Camera

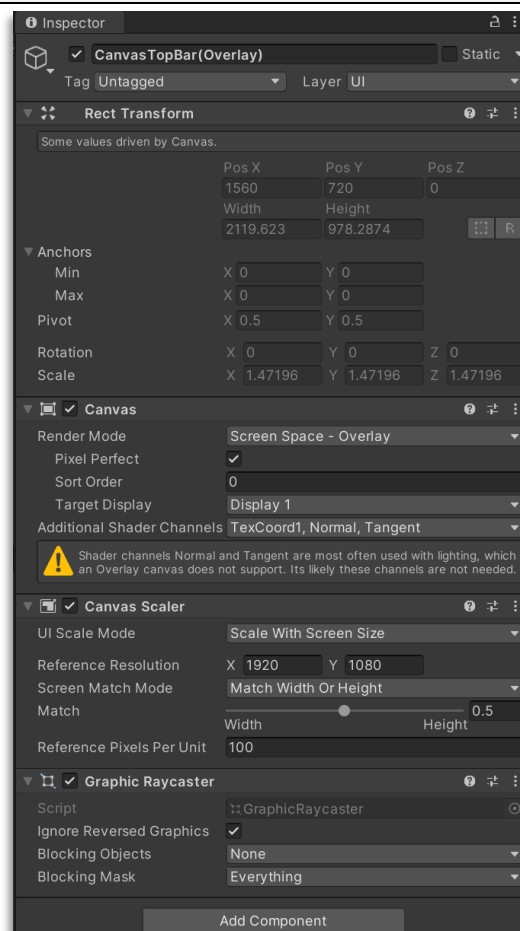
Εδώ το UI αντιμετωπίζεται ως ένα επίπεδο αντικείμενο στη σκηνή. Αντίθετα από τη λειτουργία Screen Space - Camera, το επίπεδο δεν απαιτεί να είναι στραμμένο προς την κάμερα και μπορεί να προσανατολιστεί όπως επιθυμεί ο χρήστης. Το μέγεθος του καμβά μπορεί να οριστεί χρησιμοποιώντας το Rect Transform, αλλά το μέγεθος του στην οθόνη εξαρτάται από τη γωνία θέασης και την απόσταση της κάμερας. Άλλα αντικείμενα στη σκηνή μπορούν να περάσουν πίσω, μέσα ή μπροστά από τον καμβά.



Εικόνα 80. World Space



Μια καλή πρακτική είναι να χρησιμοποιούνται τα anchor points που παρέχει το Rect Transform για να διατηρεί τα UI elements ο χρήστης στο ίδιο σημείο ανεξάρτητα του Screen Size και Resolution.



Εικόνα 81. Canvas

Το στοιχείο **Canvas Scaler** χρησιμοποιείται για τον έλεγχο της συνολικής κλίμακας και της πυκνότητας εικονοστοιχείων των στοιχείων του UI στον καμβά. Αυτή η κλιμάκωση επηρεάζεται από όλα τα στοιχεία κάτω από τον καμβά, συμπεριλαμβανομένων των

μεγεθών γραμματοσειράς και των περιθωρίων εικόνων. Για έναν καμβά που έχει οριστεί σε Screen Space Overlay ή Screen Space Camera, η ιδιότητα UI Scale Mode του καμβά μπορεί να οριστεί σε Constant Pixel Size, Scale With Screen Size ή Constant Physical Size.

Χρησιμοποιώντας τη λειτουργία **Constant Pixel Size**, οι θέσεις και τα μεγέθη των στοιχείων UI καθορίζονται σε εικονοστοιχεία στην οθόνη. Αυτή είναι επίσης η προεπιλεγμένη λειτουργία του καμβά, όταν δεν είναι προσαρτημένο το Canvas Scaler. Ωστόσο, με τη ρύθμιση στο Canvas Scaler, μπορεί να εφαρμοστεί μια σταθερή κλιμάκωση σε όλα τα στοιχεία UI του Canvas.

Χρησιμοποιώντας τη λειτουργία **Scale With Screen Size**, οι θέσεις και τα μεγέθη μπορούν να καθοριστούν σύμφωνα με τα εικονοστοιχεία μιας προκαθορισμένης ανάλυσης. Εάν η τρέχουσα ανάλυση της οθόνης είναι μεγαλύτερη από την ανάλυση αυτή, ο καμβάς θα διατηρήσει την ανάλυση αυτή, αλλά θα κάνει scale up προκειμένου να γεμίσει την οθόνη. Εάν η τρέχουσα ανάλυση της οθόνης είναι μικρότερη από την προκαθορισμένη, ο καμβάς θα μειωθεί ανάλογα για να χωρέσει σε αυτήν. Εάν η τρέχουσα ανάλυση της οθόνης έχει διαφορετικό λόγο διαστάσεων από την ανάλυση αναφοράς, η κλιμάκωση κάθε άξονα ξεχωριστά για να προσαρμοστεί στην οθόνη θα οδηγήσει σε παραμόρφωση, η οποία είναι γενικά ανεπιθύμητη οπότε με την ιδιότητα Reference Resolution θα καθορίσει την ανάλυση του καμβά ώστε να διατηρηθεί ο λόγος διαστάσεων της οθόνης. Ο χρήστης μπορεί να ελέγξει τον τρόπο με τον οποίο θα πραγματοποιηθεί αυτή η απόκλιση χρησιμοποιώντας τη ρύθμιση Screen Match Mode.

Χρησιμοποιώντας τη λειτουργία **Constant Physical Size**, οι θέσεις και τα μεγέθη των στοιχείων του UI καθορίζονται σε φυσικές μονάδες, όπως εκατοστά, χιλιοστά, ίντσες, σημεία και picas. Αυτή η λειτουργία βασίζεται στο DPI που αναφέρει σωστά η συσκευή για την οθόνη της. Ο χρήστης μπορεί να ορίσει ένα εναλλακτικό DPI που θα χρησιμοποιείται για συσκευές που δεν αναφέρουν το DPI τους. Για έναν καμβά που έχει οριστεί σε World Space, το component Canvas Scaler μπορεί να χρησιμοποιηθεί για τον έλεγχο της πυκνότητας εικονοστοιχείων των component του UI στον καμβά.

Πίνακας 117. Canvas Scaler Properties

| Property | Function |
|-------------------------------|---|
| UI Scale Mode | Determines how UI elements in the Canvas are scaled. |
| Constant Pixel Size | Makes UI elements retain the same size in pixels regardless of screen size. |
| Scale With Screen Size | Makes UI elements bigger the bigger the screen is. |
| Constant Physical Size | Makes UI elements retain the same physical size regardless of screen size and resolution. |

Πηγή : <https://docs.unity3d.com/Packages/com.unity.ugui@2.0/manual/script-CanvasScaler.html>

Ρυθμίσεις για Constant Pixel Size:

Πίνακας 118. Constant Pixel Size

| Property | Function |
|----------------------------------|---|
| Scale Factor | Scales all UI elements in the Canvas by this factor. |
| Reference Pixels Per Unit | If a sprite has this 'Pixels Per Unit' setting, then one pixel in the sprite will cover one unit in the UI. |

Ρυθμίσεις για Scale With Screen Size:

Πίνακας 119. Scale With Screen Size

| Property | Function |
|----------------------------------|--|
| Reference Resolution | The resolution the UI layout is designed for. If the screen resolution is larger, the UI will be scaled up, and if it's smaller, the UI will be scaled down. |
| Screen Match Mode | A mode used to scale the canvas area if the aspect ratio of the current resolution doesn't fit the reference resolution. |
| Match Width or Height | Scale the canvas area with the width as reference, the height as reference, or something in between. |
| Expand | Expand the canvas area either horizontally or vertically, so the size of the canvas will never be smaller than the reference. |
| Shrink | Crop the canvas area either horizontally or vertically, so the size of the canvas will never be larger than the reference. |
| Match | Determines if the scaling is using the width or height as reference, or a mix in between. |
| Reference Pixels Per Unit | If a sprite has this 'Pixels Per Unit' setting, then one pixel in the sprite will cover one unit in the UI. |

Ρυθμίσεις για Constant Physical Size:

Πίνακας 120. Constant Physical Size

| Property | Function |
|----------------------------------|--|
| Physical Unit | The physical unit to specify positions and sizes in. |
| Fallback Screen DPI | The DPI to assume if the screen DPI is not known. |
| Default Sprite DPI | The pixels per inch to use for sprites that have a 'Pixels Per Unit' setting that matches the 'Reference Pixels Per Unit' setting. |
| Reference Pixels Per Unit | If a sprite has this 'Pixels Per Unit' setting, then its DPI will match the 'Default Sprite DPI' setting. |

Ρυθμίσεις για το World Space Canvas (εμφανίζεται όταν το στοιχείο Canvas έχει οριστεί σε World Space):

Πίνακας 121. World Space Canvas

| Property | Function |
|----------------------------------|---|
| Dynamic Pixels Per Unit | The amount of pixels per unit to use for dynamically created bitmaps in the UI, such as Text. |
| Reference Pixels Per Unit | If a sprite has this 'Pixels Per Unit' setting, then one pixel in the sprite will cover one unit in the world. If the 'Reference Pixels Per Unit' is set to 1, then the 'Pixels Per Unit' setting in the sprite will be used as-is. |

Το component **Graphics Raycaster** είναι υπεύθυνο για την εντολές του χρήστη στο interface της εφαρμογής δηλαδή όταν ο παίκτης κάνει αριστερό κλικ με το ποντίκι ή με πατώντας με το δάκτυλο κάποιο σημείο αναφοράς στην οθόνη τότε αυτό εξετάζει όλα τα γραφικά στον καμβά και ανιχνεύει εάν ένα από αυτά έχει προσκρούσει με την ακτίνα που δημιούργησε το κλικ του χρήστη. Το Graphic Raycaster μπορεί να διαμορφωθεί έτσι ώστε να αγνοεί τα γραφικά που βρίσκονται πίσω του και να αποκλείεται από τα 2D ή 3D αντικείμενα που υπάρχουν μπροστά του. Μπορεί επίσης να εφαρμοστεί μέσω των layers μια χειροκίνητη προτεραιότητα, εάν επιθυμεί ο προγραμματιστής όπου η επεξεργασία αυτού του στοιχείου να επιβάλλεται στο μπροστινό ή στο πίσω μέρος του Raycasting.

Πίνακας 122. Graphics Raycaster

| Property | Function |
|---------------------------------|---|
| Ignore Reversed Graphics | Should graphics facing away from the raycaster be considered? |
| Blocked Objects | Type of objects that will block graphic raycasts. |
| Blocking Mask | Type of objects that will block graphic raycasts. |

Πηγή : <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-GraphicRaycaster.html>

5. Παρουσίαση Εφαρμογής

Στο παρόν κεφάλαιο θα γίνει παρουσίαση της τελικής μορφής της εφαρμογής και θα αναλυθούν οι καταστάσεις και οι δυνατότητες που δίνει η εφαρμογή σε κάθε παίκτη κατά την διάδραση με αυτήν. Επίσης μέσω διαγραμμάτων UML θα γίνει πιο κατανοητή η διασύνδεση των επιμέρους στοιχείων της εφαρμογής και πως μέσω αυτής της συνεργασίας παράγεται το τελικό αποτέλεσμα.

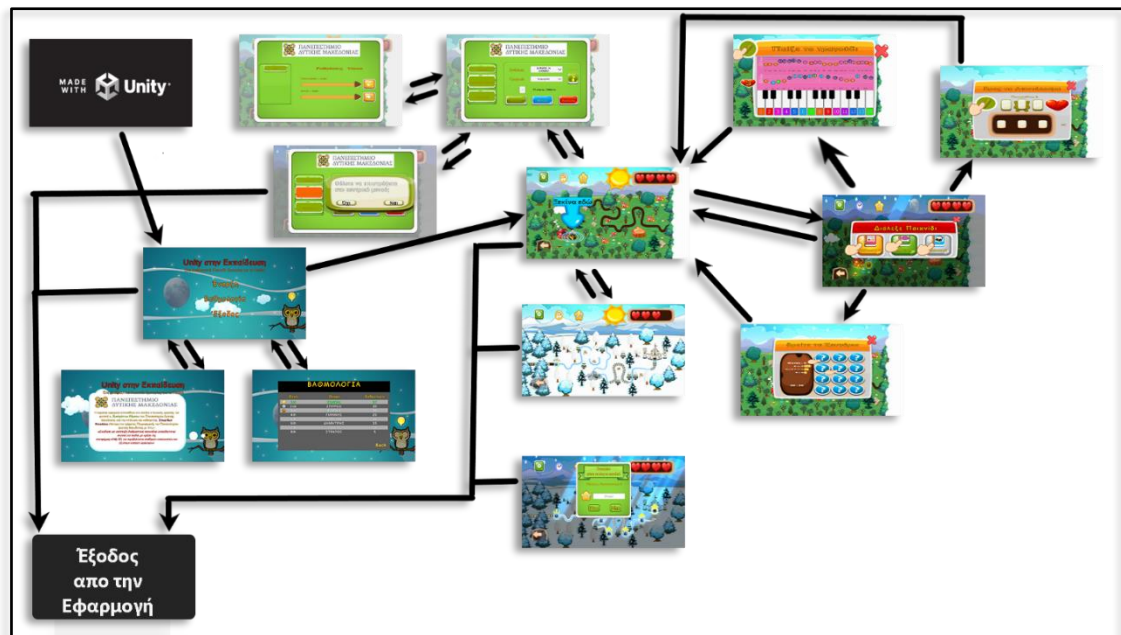


Εικόνα 82. User Interface Implementation in Unity

Πηγή : <https://resources.unity.com/games/user-interface-design-and-implementation-in-unity>

5.1 Καταστάσεις μετάβασης της εφαρμογής

Διάφορες καταστάσεις που μπορεί να βρεθεί η εφαρμογή κατά την διάρκεια διάδρασης ενός παίκτη με την εφαρμογή και περιγραφή των δυνατοτήτων που έχει σε κάθε επιμέρους κατάσταση.



Εικόνα 83. Καταστάσεις που μπορεί να βρεθεί η εφαρμογή

Με την εκκίνηση της εφαρμογής η Unity εμφανίζει το λογότυπο της εταιρίας για κάποια δευτερόλεπτα. Το λογότυπο είναι υποχρεωτικό για την δωρεάν έκδοση από την εταιρία. Η επιπλέον δυνατότητα που έχει ο χρήστης στην pro έκδοση της Unity είναι πως μπορεί να απενεργοποιήσει τη splash screen πηγαίνοντας Project Settings > Player > Splash Image. Μπορεί να βάλει δικά του splash screen και δικό του λογότυπο.



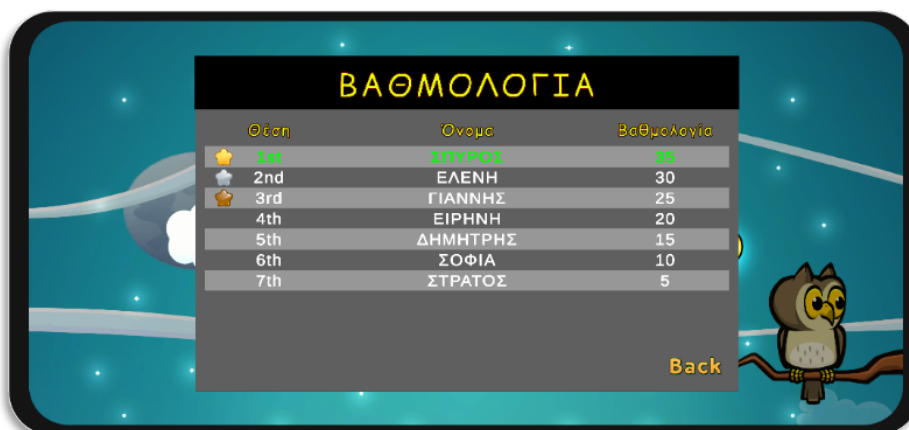
Εικόνα 84. Logo Unity during App Start

Αμέσως μετά ο παίκτης μεταφέρετε στην πρώτη σκηνή του παιχνιδιού που είναι το κεντρικό μενού με τον τίτλο της εφαρμογής και κάποιες επιλογές που έχουν να κάνουν με την παρουσίαση των high score, πληροφορίες σχετικά με την εφαρμογή, έναρξη και έξοδος εφαρμογής. Από εδώ ξεκινάει η διάδραση του χρήστη με την εφαρμογή .



Εικόνα 85. Κεντρικό μενού εφαρμογής

Εδώ υπάρχουν τρεις πιθανές επιλογές για τον χρήστη οι οποίες είναι είτε επιλέγει να κάνει έξοδο από την εφαρμογή, είτε να ζητήσει να δει την βαθμολογία του, είτε να ξεκινήσει το παιχνίδι.



Εικόνα 86. Πίνακας Βαθμολογίας

Επίσης με παρατεταμένο αριστερό κλικ πάνω στο ψηφιακό στοιχείο στην δεξιά κάτω γωνία που αντιπροσωπεύεται από την κουκουβάγια ο χρήστης ενημερώνεται για κάποιες πληροφορίες σχετικά με την εφαρμογή



Εικόνα 87. Πληροφορίες σχετικά με την εφαρμογή

Στην επόμενη οθόνη υπάρχει η κεντρική σκηνή με όλο το interface που είναι υπεύθυνο για την διάδραση του παίκτη με την μηχανή.



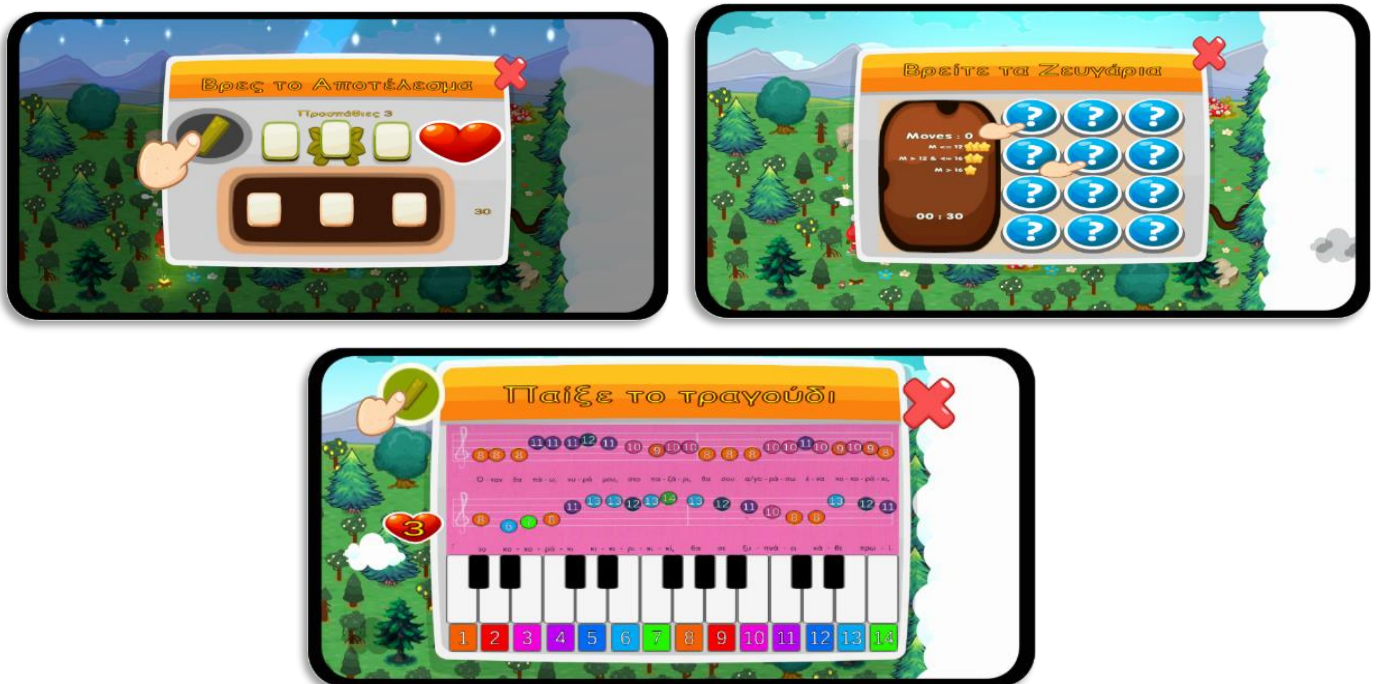
Εικόνα 88. Κεντρική Σκηνή παιχνιδιού

Έντονα χρώματα και εναλλαγή νύχτας μέρας προσφέρουν ένα ευχάριστο περιβάλλον στον παίκτη. Τα particles effects και το φως μαζί με την κίνηση στην σκηνή είτε από τα σύννεφα , είτε από τα animations με την χλωρίδα και την πανίδα προσδίδουν μια ζωντάνια στο όλο παιχνίδι. Ο παίκτης επιλέγει το πρώτο επίπεδο με την ένδειξη να τον βοηθάει που να κάνει αριστερό κλικ με το ποντίκι ή να πατήσει με το δάχτυλό του. Με την επιλογή αυτή ανοίγει ένα παράθυρο το οποίο παρουσιάζει στον παίκτη τα διαθέσιμα παιχνίδια που υπάρχουν και από τα οποία μπορεί να διαλέξει ένα από αυτά κάθε φορά.



Εικόνα 89. Διαθέσιμα Μίνι Παιχνίδια

Κατόπιν εμφανίζεται το αντίστοιχο μίνι παιχνίδι που έχει επιλέξει ο παίκτης και με χρήση φωνητικής ενημέρωσης τις προσπάθειες, τις ζωές ή το χρόνο που έχει για να ολοκληρώσει την παρτίδα. Οι προσπάθειες, οι ζωές και ο χρόνος έχουν να κάνουν με το τι επίπεδο δυσκολίας έχει βάλει ο παίκτης στις ρυθμίσεις μενού.



Εικόνα 90. Εκκίνηση Μίνι Παιχνιδιού

Όταν τελειώσει το μίνι παιχνίδι ο χρήστης ενημερώνεται για το αποτέλεσμα δηλαδή αν κέρδισε ή αν έχασε. Αν κέρδισε ενημερώνεται με ένα animation για την βαθμολογία του δηλαδή πόσα αστέρια κέρδισε. Αν έχασε εμφανίζεται ένα animation που πληροφορεί τον παίκτη για την απώλεια μιας ζωής.



Εικόνα 91. Αποτέλεσμα Μίνι Παιχνιδιού

Όλες αυτές οι πληροφορίες εμφανίζονται μετά στην πάνω μπάρα της οθόνης στην κεντρική σκηνή.



Εικόνα 92. Επάνω Μπάρα Εικόνων Κεντρικής Σκηνής

Η επάνω μπάρα παρέχει στον χρήστη από αριστερά προς τα δεξιά τις γενικές ρυθμίσεις όπου μπορεί να αλλάξει την ανάλυση γραφικών, τις διαστάσεις της οθόνης, τον τρόπο που θα παιχτεί η εφαρμογή δηλαδή αν θα είναι σε περιβάλλον window ή full screen και το επίπεδο δυσκολίας που θα διαλέξει ο παίκτης (εύκολο, μέτριο, δύσκολο). Το κουμπί ήχου έχει τις ανάλογες ρυθμίσεις για τον background ήχο της εφαρμογής και των sound effects. Από αυτό το μενού τέλος ο παίκτης μπορεί να κλείσει το παράθυρο αυτό, να τερματίσει την εφαρμογή ή να γυρίσει στην εισαγωγική οθόνη.



Εικόνα 93. Κεντρικό Μενού Παιχνιδιού

Μετά το κουμπί μενού υπάρχει το ρολόι για να ενημερώνει τον χρήστη ότι είναι μέρα ή νύχτα στο παιχνίδι και το αστεράκι που κρατάει το σύνολο των κερδισμένων αστερίων που έχει μαζέψει ο χρήστης κατά την διάρκεια της συνεδρίας. Ακολουθεί ένδειξη για την κατάσταση που βρίσκεται εκείνη την στιγμή η εφαρμογή δηλαδή ημέρα ή νύχτα όπως και με το ρολόι με το αντίστοιχο εικονίδιο το οποίο για την μέρα είναι ο ήλιος και για την νύχτα το φεγγάρι. Σε όλο αυτό το σκηνικό προστίθεται το post process το οποίο είναι υπεύθυνο για το σκοτεινιάσμα του περιβάλλοντος στην κεντρική σκηνή κατά την διάρκεια της νύχτας ή την επαναφορά του φωτός κατά την διάρκεια της μέρας.



Εικόνα 94. Κεντρική Σκηνή παιχνιδιού Νύχτα

Τέλος τέρμα δεξιά υπάρχουν οι ζωές που έχει ο παίκτης διαθέσιμες. Όταν ο παίκτης φτάσει στο τέλος της πρώτης πίστας εμφανίζεται ένα βέλος για να συνεχίσει στη επόμενη πίστα αφού το πατήσει.



Εικόνα 95. Ξεκλείδωμα Επόμενης Πίστας



Εικόνα 96. Κεντρική Σκηνή Δεύτερη Πίστα

Εάν δεν φτάσει μέχρι το τέλος της πίστας και χάσει όλες τις ζωές του, τότε το παιχνίδι περνάει σε κατάσταση game over. Εάν καταφέρει να φτάσει στο τέλος της διαδρομής τότε έχει την δυνατότητα να βάλει το όνομά του στον πίνακα με τα high score.



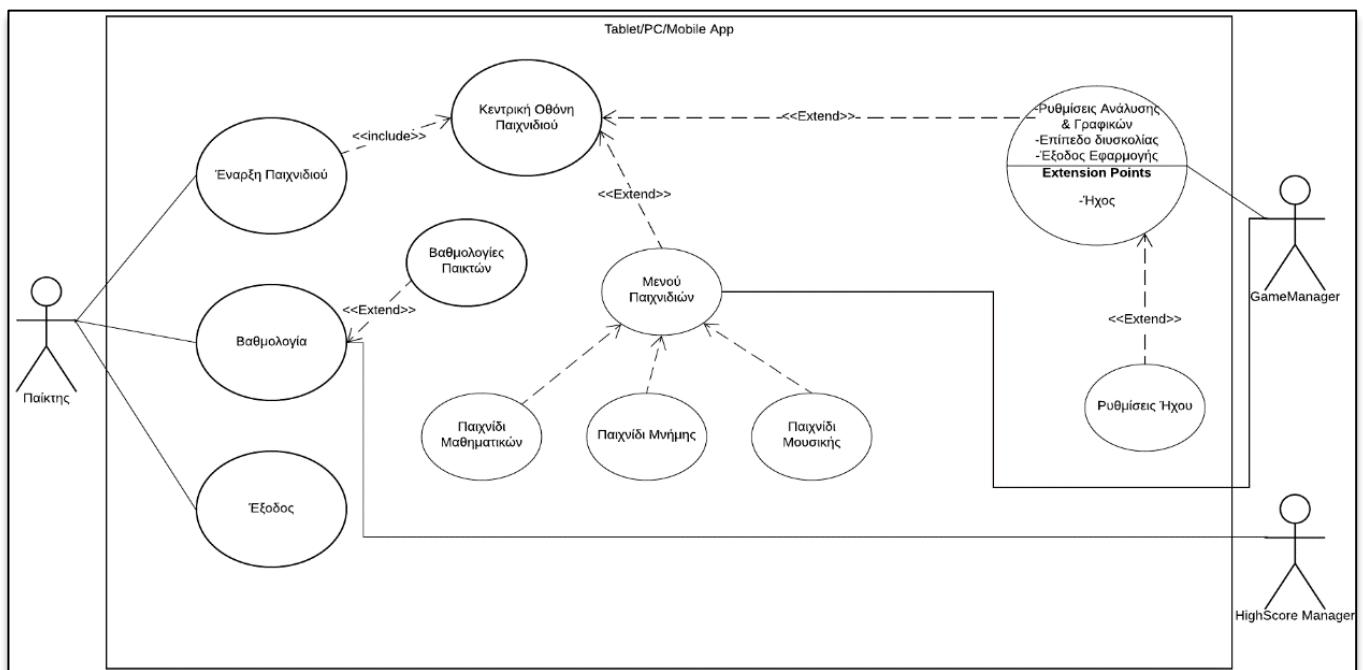
Εικόνα 97. Game Over



Εικόνα 98. The End

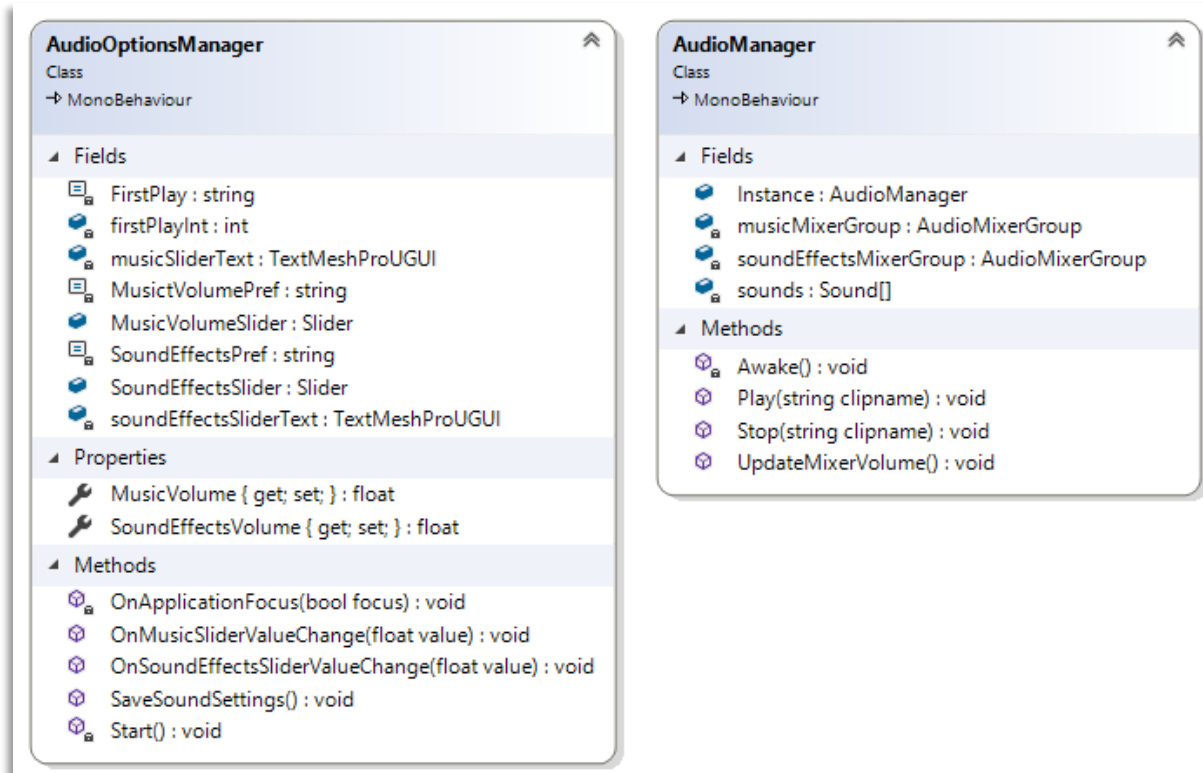
5.2 Διαγράμματα UML

Με το **διάγραμμα case UML** θα περιγράψει γενικά η δομή της εφαρμογής και πως ο παίκτης αλληλοεπιδρά μαζί της κατά την διάρκεια μιας συνεδρίας.



Εικόνα 99. UML case Diagram

Με το **διάγραμμα class UML** θα γίνει μια πιο αναλυτική επίδειξη για το πως είναι δομημένη η εφαρμογή και το πως συνδέονται τα επιμέρους στοιχεία για να δημιουργήσουν το συγκεκριμένο αποτέλεσμα στην τελική υλοποίησή της.



Εικόνα 100. AudioManager GameObject

Το **AudioManager** GameObject όπως λέει και το όνομά του είναι υπεύθυνο για την μουσική στο παιχνίδι. Μέσα στον πίνακα sounds αποθηκεύονται όλες οι μουσικές και τα ηχητικά εφέ.

Το **GameManager** είναι η καρδιά του παιχνιδιού γιατί είναι υπεύθυνο για την σωστή λειτουργία της εφαρμογής. Είναι το GameObject που αναζητούν όλες οι λειτουργίες ώστε να ενημερώνει μέσω του UI για την κατάσταση που βρίσκεται η εφαρμογή τον εκάστοτε παίκτη. Μέσα στο αντικείμενο αυτό υπάρχει η λογική για τα τρία μίνι παιχνίδια αλλά και για το κεντρικό μενού. Επειδή είναι όλα μέσα στο ίδιο αντικείμενο η αναφορά από το ένα στο άλλο είναι πιο εύκολη.



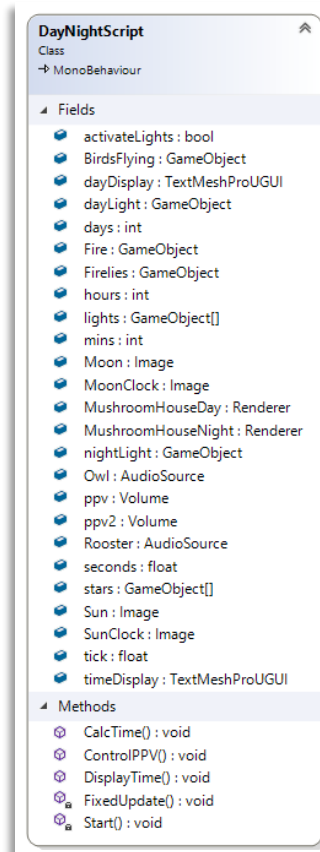
Μια καλή πρακτική είναι ο προγραμματιστής να χρησιμοποιεί private μεταβλητές γιατί αν έχει ένα αντικείμενο που η τιμή του αλλάζει από 2 η περισσότερες μεθόδους τότε αν δημιουργηθεί πρόβλημα δεν θα ξέρει από που προκλήθηκε.

The image displays four class windows from Visual Studio, showing the structure of the GameManager, MathProblems, MemoryGameLogic, and MusicGameLogic classes. Each window lists fields, properties, and methods.

- GameManager** (Class, MonoBehaviour):
 - Fields:** audioManager: AudioManager, btnNextLevel: GameObject, checkpoint: GameObject[], clouds: Transform, contentLevel1: GameObject, contentLevel1Reached: GameObject, contentLevel2: GameObject, contentLevel2Reached: GameObject, CountWinStars: int, currentPlace: int, debugcheck: bool, difficulty: string, fogOfWar: RectTransform, GameOverMessage: GameObject, HandButtonDownLeft: GameObject, HandButtonDownMiddle: GameObject, HandButtonDownRight: GameObject, HeartFirst: GameObject, HeartFourth: GameObject, hearts: int, HeartSecond: GameObject, HeartThird: GameObject, Highscore: GameObject, infoPanel: GameObject, instance: GameManager, leftEnd: RectTransform, Level13checkpointStar1: SpriteRenderer, Level13checkpointStar2: SpriteRenderer, Level13checkpointStar3: SpriteRenderer, level1Panel: RectTransform, level1Path: GameObject, level1Props: GameObject, level2Panel: RectTransform, level2Path: GameObject, level2Props: GameObject, level2SignAnimation: GameObject, Level6checkpointStar1: SpriteRenderer, Level6checkpointStar2: SpriteRenderer, Level6checkpointStar3: SpriteRenderer, levelSignAnimation: GameObject, mainCanvas: Canvas, MathPanel: RectTransform, MemoryPanel: RectTransform, miniGamePanel: RectTransform, MusicPanel: RectTransform, OptionMenu: RectTransform, OptionMenuPanel: RectTransform, Player: GameObject, rightEnd: RectTransform, SoundMenu: RectTransform, Stars: GameObject[], startMiniGameBtn: Button, startMiniGameBtn2: Button, startMiniGameBtn3: Button, TotalStars: GameObject, totalStars: TextMeshProUGUI.
 - Methods:** Awake(): void, CheckDifficulty(): void, CheckMath(): void, CheckMemory(): void, CheckMusic(): void, CountTotalStars(): void, GetCheckpointNumber(): int, HideMathPanel(): void, HideMemoryPanel(): void, HideMiniGamesPanel(): void, HideMusicPanel(): void, HideOptionPanel(): void, Lose(): void, MainMenuBtn(): void, MovePlayerToNextCheckpoint(): void, NextLevel(): void, Restart(): void, ReturnToMainMenuAfterATime(): void, SetDifficultyToEasy(): void, SetDifficultyToHard(): void, SetDifficultyToNormal(): void, ShowMathPanel(): void, ShowMemoryPanel(): void, ShowMiniGamesPanel(): void, ShowMusicPanel(): void, ShowOptionPanel(): void, Start(): void, Update(): void, Win(): void.
- MathProblems** (Class, MonoBehaviour):
 - Fields:** allowChoice: bool, audioManager: AudioManager, calculations: string[], CanvasTopBar: Canvas, choice: int, correct: bool, correctAnswer1: Image, correctAnswer2: Image, correctAnswer3: Image, finalNumber: int, firstBtn: Button, firstNumber: int, firstNumberText: TextMeshProUGUI, firstResult: TextMeshProUGUI, imageListOperation: List<Image>, life: int, lifeLeft: TextMeshProUGUI, loselImage: GameObject, nextQuestion: GameObject, number: int, pickedOperation: int, secondBtn: Button, secondNumber: int, secondNumberText: TextMeshProUGUI, secondResult: TextMeshProUGUI, starImage1: GameObject, starImage2: GameObject, starImage3: GameObject, startMiniGameBtn: Button, temporaryNumber: int, thirdBtn: Button, thirdResult: TextMeshProUGUI, TimeLeft: float, TimerOn: bool, TimerText: TextMeshProUGUI, tries: int, TriesLeft: TextMeshProUGUI, winImage: GameObject, wintext: TextMeshProUGUI, wrongAnswer1: Image, wrongAnswer2: Image, wrongAnswer3: Image.
 - Methods:** BtnOneChoice(): void, BtnThreeChoice(): void, BtnTwoChoice(): void, CalcRandomNumbers(string pickedOperation): void, CheckDifficulty(): void, CheckLifeAndTries(int life): string, CheckOperations(): void, OnBtnStart(): void, PlayersChoice(int choice): int, RandomNumberCalc(): int, ResetBtn(): void, SetStars(string result): int, StarAnimations(string result): IEnumerator, StartResetMiniGame(): IEnumerator, Update(): void, UpdateTimer(float currentTime): void.
- MemoryGameLogic** (Class, MonoBehaviour):
 - Fields:** audioManager: AudioManager, CanvasTopBar: Canvas, count: int, CoverImageArray: GameObject[], firstOpen: MainImageScript, GameFinished: bool, GridPanel: GameObject, HiddenImageArray: Image[], IntRandomArray: int[], IntRandomArray2: int[], LoselImage: GameObject, Moves: int, MovesText: TextMeshProUGUI, PairImages: Sprite[], PauseButton: GameObject, PauseImage: Image, secondOpen: MainImageScript, stars: int, StartButton: GameObject, startObject: MainImageScript, TimeLeft: float, TimerOn: bool, TimerText: TextMeshProUGUI, WinImage: GameObject, WinningStars: GameObject[].
 - Properties:** CanOpen { get; } : bool.
 - Methods:** CheckGameDifficulty(): void, CheckGuessed(): IEnumerator, CheckVictory(): void, HideImages(int ImageToCover): void, ImageOpened(MainImageScript startObject): void, PauseMemoryGame(): void, ResetMemoryBtn(): void, StarAnimations(int Moves): IEnumerator, StartMemoryGame(): void, StartResetMiniGame(): IEnumerator, Update(): void, UpdateTimer(float currentTime): void.
- MusicGameLogic** (Class, MonoBehaviour):
 - Fields:** audioManager: AudioManager, CanvasTopBar: Canvas, choice: int, correctAnswersAxKounelaki: List<GameObject>, correctAnswersPazari: List<GameObject>, count: int, CurrentSong: List<int>, CurrentSongSprites: List<Sprite>, gamefinished: bool, GameStarted: bool, HorizontalProgressBarTimer: GameObject, k: int, lifes_tries: TextMeshProUGUI, ListOfSongs: List<GameObject>, LoselImage: GameObject, PauseButton: GameObject, pianoButtonCurrentChoice: int, pianoKeyNumbers: GameObject[], pianoKeys: GameObject[], SongListSprites: List<Sprite>, songNoteCurrentChoice: int, SongNotesKounelaki1: List<int>, SongNotesPazari2: List<int>, starImage1: GameObject, starImage2: GameObject, starImage3: GameObject, startMiniGameBtn: GameObject, TimeLeftMusic: float, TimerOnMusic: bool, TimerTextMusicMM: TextMeshProUGUI, userinputs: List<int>, WinImage: GameObject, wrongAnswersAxKounelaki: List<GameObject>, wrongAnswersPazari: List<GameObject>.
 - Methods:** AddToList(int choice): void, CheckForErrors(): void, CheckGameDifficulty(): void, CheckProgressOfSong(): bool, Lose(): void, PauseMusicGame(): void, PianoBtnChoice(int number): void, RandomList(): List<int>, ResetMusicBtn(): void, ShowProgress(): void, StarAnimations(int count): IEnumerator, StartMusicGame(): void, StartResetMiniGame(): IEnumerator, Update(): void, UpdateTimerMusic(float currentTime): void, Win(): void.
- MainMenu** (Class, MonoBehaviour):
 - Fields:** resolutionDropdown: TMP_Dropdown, resolutions: Resolution[].
 - Methods:** DelayApplicationQuit(): void, PlayGame(): void, QuitPlay(): void, SetFullscreen(bool isFullscreen): void, SetQuality(int qualityIndex): void, SetResolution(int resolutionIndex): void, Start(): void.

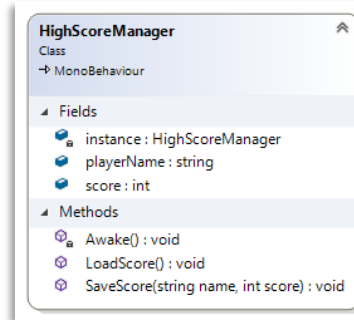
Εικόνα 101. GameManager GameObject

Το **Local Volume** αντικείμενο είναι υπεύθυνο για να δημιουργεί την αίσθηση της μέρας και της νύχτας στον παίκτη, αυξάνοντας ή ελαττώνοντας τις ρυθμίσεις post process με ένα συγκεκριμένο τρόπο. Το DayNightScript ανάλογα με το βάρος (weight) που θα πάρει από τις ρυθμίσεις του component Volume θα ελαττώσει ή θα αυξήσει τον αντίστοιχο φωτισμό στο παιχνίδι.



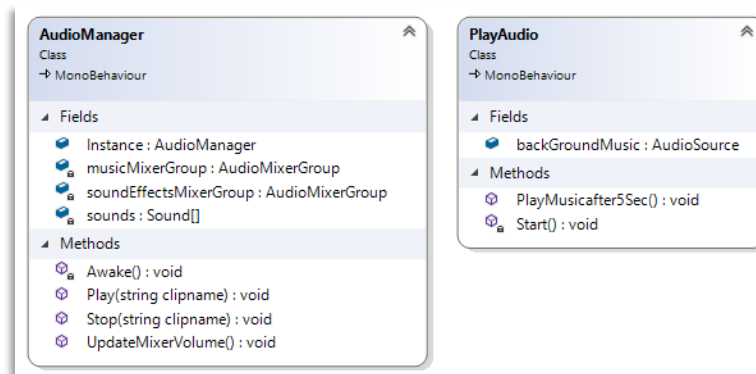
Εικόνα 102. Local Volume GameObject

Το αντικείμενο **HighScoreManager** είναι υπεύθυνο για να διατηρεί το score του παίκτη από σκηνή σε σκηνή γιατί δεν καταστρέφεται όταν αλλάζει σκηνή ο παίκτης αλλά παραμένει με αποθηκευμένες τις καινούργιες τιμές της βαθμολογίας. Είναι αυτό που θα ενημερώσει τον πίνακα βαθμολογίας στην αρχική σκηνή με τις καινούργιες τιμές αν υπάρχουν και οι οποίες θα πρέπει να είναι τουλάχιστον μεγαλύτερες από την τελευταία βαθμολογία του πίνακα των High Score.



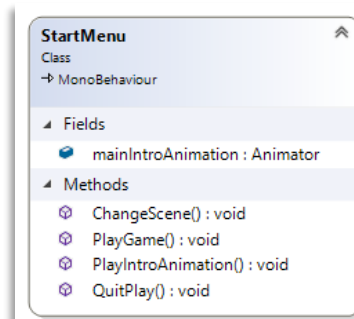
Εικόνα 103. HighScoreManager GameObject

Το **AudioManager** αντικείμενο έχει παρόμοιες ρυθμίσεις με αυτό της κεντρικής σκηνής αλλά είναι πιο απλό στην λειτουργία του και αφορά μόνο αυτή την σκηνή και καμία άλλη.



Εικόνα 104. AudioManager GameObject Εισαγωγική Σκηνή

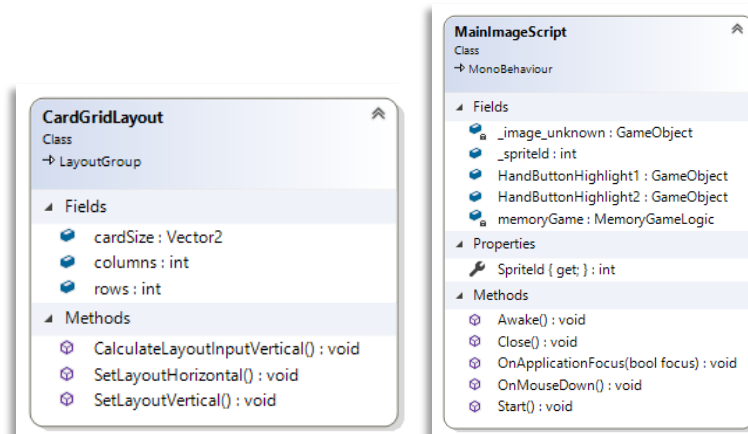
Το αντικείμενο **MainMenu** στην αρχική σκηνή έχει κάποιες βασικές μεθόδους για την διαχείριση μόνο της συγκεκριμένης σκηνής όπως το κλείσιμο της εφαρμογής, την εκκίνηση του παιχνιδιού αλλάζοντας σκηνή και το εισαγωγικό Animation.



Εικόνα 105. MainMenu GameObject Εισαγωγική Σκηνή

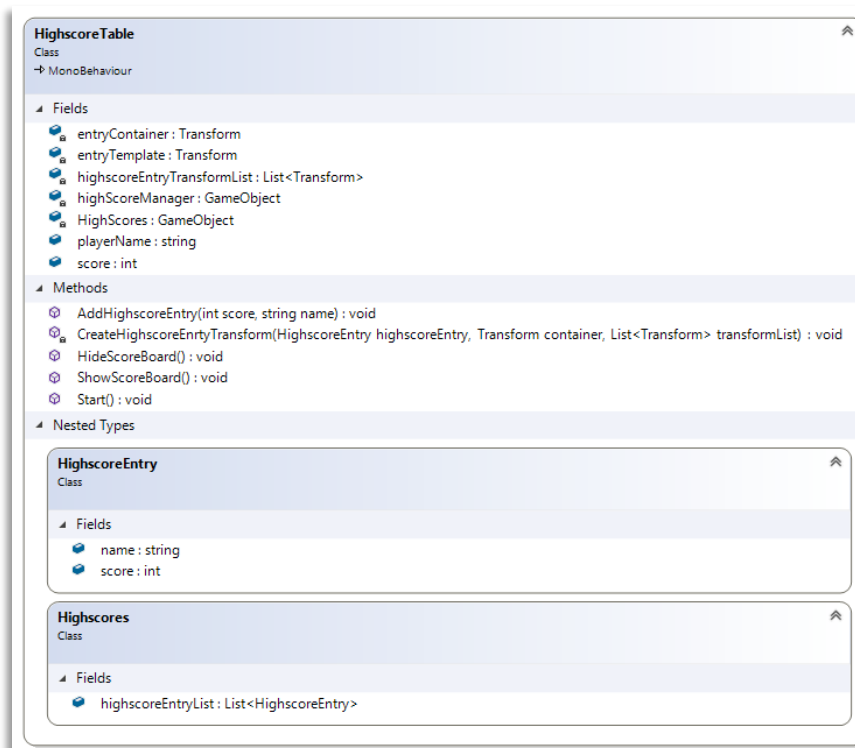
Το **GridPanel** είναι ένα αντικείμενο το οποίο είναι υπεύθυνο για την δημιουργία του πλέγματος (grid) στο παιχνίδι της μνήμης. Το **ImageHidden** είναι ένα αντικείμενο το οποίο

βρίσκεται και αυτό στο παιχνίδι της μνήμης και είναι υπεύθυνο για το game logic του συγκεκριμένου παιχνιδιού.



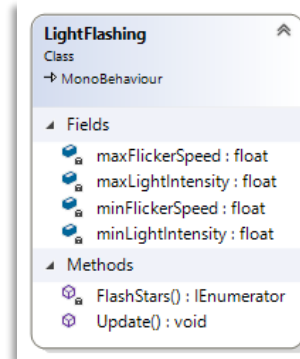
Εικόνα 106. GridPanel και ImageHidden GameObjects

Το **HighScores** αντικείμενο στην εισαγωγική σκηνή είναι αυτό που θα δημιουργήσει τον πίνακα με την βαθμολογία και θα αναλάβει να ενημερώνεται από το HighScoreManager για τυχόν καινούργιες βαθμολογίες που πρέπει να καταγραφούν στον πίνακα.



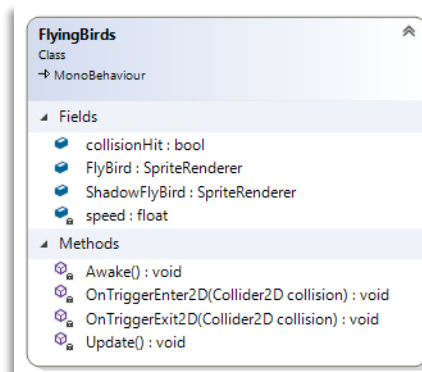
Εικόνα 107. HighScores GameObject Εισαγωγική Σκηνή

Το Star Shine αντικείμενο είναι υπεύθυνο για τα τρεμοπαίγματα που κάνουν τα αστέρια σαν εφέ στο παιχνίδι. Αυτό επιτυγχάνεται σε συνδυασμό με το Light2D script που παρέχει η ίδια η μηχανή Unity.



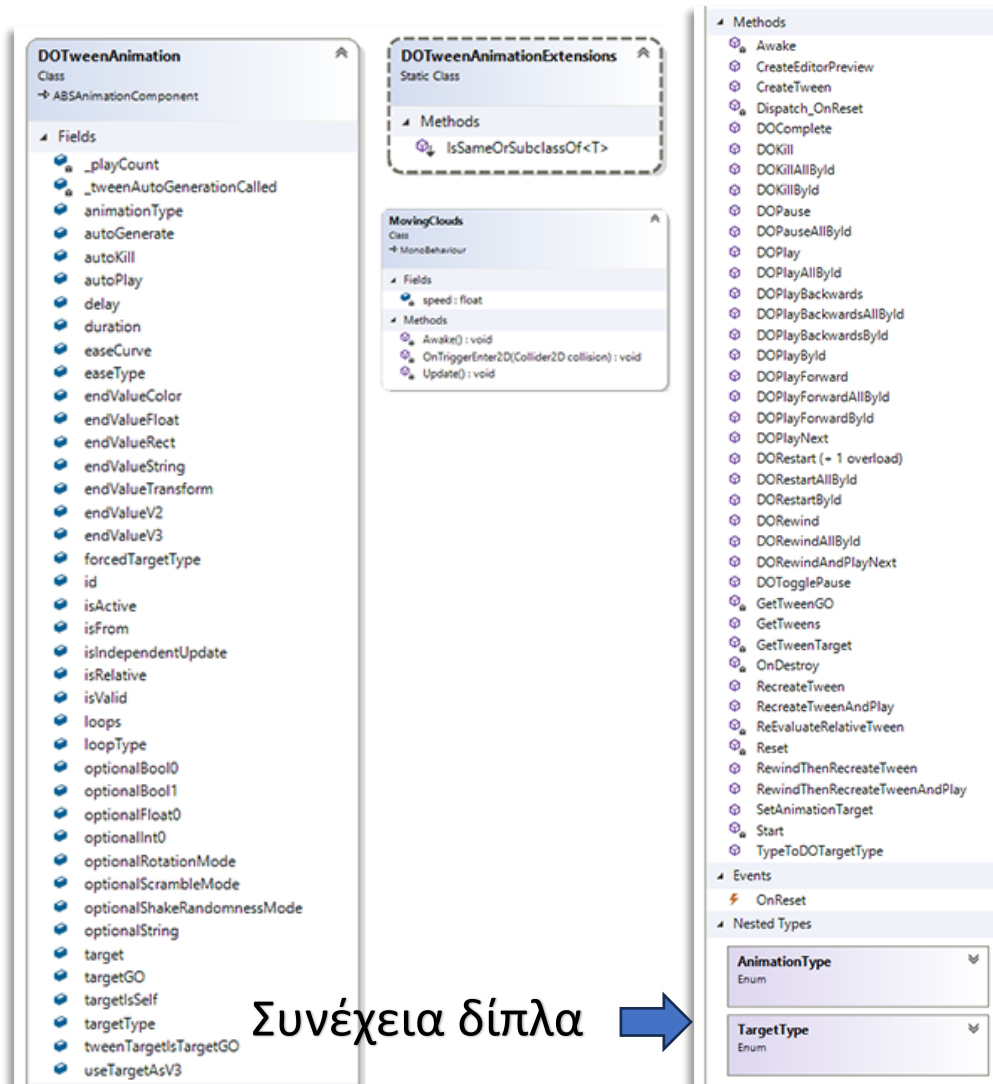
Εικόνα 108. Star Shine GameObject Εισαγωγική Σκηνή

Εδώ το αντικείμενο BirdFlying αναλαμβάνει να κινήσει το ψηφιακό στοιχείο πουλί από το αριστερό άκρο της οθόνης στο δεξί και το αντίθετο. Αυτό το πετυχαίνει ελέγχοντας για συγκρούσεις 2D μεταξύ του ίδιου του αντικειμένου με το αντικείμενο που είναι υπεύθυνο για να του αλλάξει πορεία .



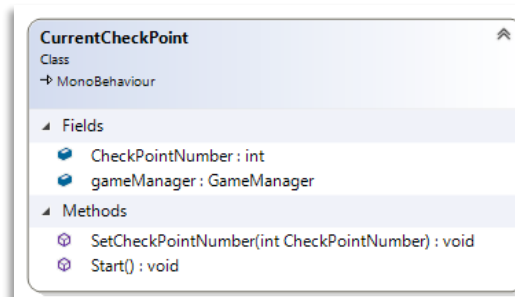
Εικόνα 109. BirdFlying GameObject

Το αντικείμενο Moving Clouds είναι υπεύθυνο για το animation που κάνουν τα σύννεφα στο παιχνίδι. Αυτό επιτυγχάνεται με την χρήση όχι μόνο του animator αλλά και του πακέτου DOTween Animation το οποίο είναι η μέθοδος interpolation μεταξύ δυο σημείων και έτσι το κόστος στην μηχανή είναι πιο μικρό από την χρήση της πρώτης μεθόδου.



Εικόνα 110. Moving Clouds GameObject

Τέλος το αντικείμενο CheckPoint ελέγχει ποτέ κάνει αριστερό κλικ ή πατάει με το δάκτυλό του ο παίκτης τις αριθμημένες κουκίδες του παιχνιδιού για να ανοίξει τα μίνι παιχνίδια. Ανάλογα το νούμερο του CheckPoint θα εμφανιστούν τα μίνι παιχνίδια για να παίξει ο χρήστης και θα βαθμολογηθεί για το συγκεκριμένο CheckPoint και κανένα άλλο.



Εικόνα 111. CheckPoint GameObjects



Επειδή στην δωρεάν έκδοση ο κώδικας της μηχανής Unity δεν είναι διαθέσιμος κάποια αντικείμενα είχαν script και κλάσης που τα προσφέρει η μηχανή και δεν υπάρχει άμεση ικανότητα απεικόνισής τους.

6. Συμπεράσματα και Μελλοντική Εξέλιξη

Ο σχεδιασμός της εφαρμογής είναι το πιο δύσκολο κομμάτι γιατί πρέπει να γίνει σωστή μελέτη για το περιβάλλον στο οποίο ο προγραμματιστής θέλει να χρησιμοποιηθεί και με ποιον τρόπο. Επειδή αυτό το περιβάλλον είναι εν μέρει μεταβαλλόμενο, γιατί το επίπεδο των παιδιών είναι διαφορετικό θα πρέπει να μπορεί να δώσει στα παιδιά την δυνατότητα να αλληλοεπιδρούν με την εφαρμογή χωρίς να χρειάζεται η παρέμβαση/εποπτεία κάποιου εκπαιδευτικού. Κάθε παιδί έχει διαφορετικό υπόβαθρο αντίληψης και μάθησης που επηρεάζεται από το περιβάλλον στο οποίο μεγαλώνει. Για να γεφυρωθεί αυτή η διαφορά θα πρέπει η εφαρμογή να παρέχει μια μορφή σεμιναρίου (tutorial) ώστε ο εκάστοτε μαθητής να μπορεί να φτάσει ένα συγκεκριμένο επίπεδο το οποίο θα είναι απαραίτητο για την ορθή χρήση της εφαρμογής.

Η εφαρμογή θα μπορούσε να χρησιμοποιηθεί για συμπληρωματική μάθηση πάνω σε ένα η περισσότερα γνωστικά αντικείμενα η για όξυνση της αντίληψης του χρήστη η απλώς για διασκέδαση μετά από έντονη μελέτη. Η εφαρμογή δεν απαιτεί την παρουσία κάποιου επιβλέποντα χωρίς αυτό να σημαίνει πως η παρουσία του δεν θα βοηθούσε καλύτερα την εξέλιξη του χρήστη .

Μελλοντικά θα ενσωματωθεί μια εξωτερική βάση δεδομένων για να αποθηκεύει τις επιδόσεις των χρηστών ώστε να έχουν κάποια μορφή ιστορικού που θα μπορούν να ανατρέξουν για να παρακολουθούν την εξέλιξή τους. Επίσης θα προστεθεί ένα ξεχωριστό τμήμα που θα εξηγεί την ορθή χρήση της εφαρμογής μέσω ενός επιμορφωτικού σεμιναρίου (tutorial). Για να γίνει πιο ανταγωνιστικό /διασκεδαστικό θα μπορούσε επίσης να αναπτυχθεί και ένα τμήμα που θα υποστηρίζει τη δια δράση πολλών παικτών ταυτόχρονα στην ίδια ή και σε διαφορετικές πλατφόρμες. Για να έχει αντοχή στο χρόνο η εφαρμογή σταδιακά θα μπορούσε να εμπλουτιστεί με περισσότερες δραστηριότητες σε διαφορά γνωστικά αντικείμενα.

Μια από τις μεγαλύτερες προκλήσεις που πιθανόν να συναντήσει ο σχεδιαστής της εφαρμογής είναι η δημιουργία της βάσης δεδομένων. Πρέπει να γίνει σωστή μελέτη για το πώς θα σχεδιαστεί η βάση. Που και πως θα αποθηκεύει τα δεδομένα της εφαρμογής όπως τα στοιχεία των χρηστών, τις ασκήσεις/παιχνίδια, τις βαθμολογίες, το ιστορικό και πολλά

άλλα. Μια μη σωστή υλοποίηση της βάσης θα έχει σοβαρές επιπτώσεις στην όλη εφαρμογή εάν κατά την κατασκευή της εφαρμογής ο προγραμματιστής καταλήξει στο συμπέρασμα παραδείγματος χάριν πως δεν έχει υπολογίσει σωστά τον τρόπο με τον οποίο θα γίνονται οι εγγραφές στην βάση. Εδώ πρέπει ο σχεδιαστής της βάσης να υπολογίσει και τι πληροφορία θέλει να αντλήσει από την βάση ώστε να έχει ένα feedback για την εξέλιξη τις εκπαιδευτικής διαδικασίας.

Μια ακόμα σοβαρή πρόκληση είναι η δυσκολία στην αναβάθμιση της βάσης. Εάν σε μια επόμενη έκδοση της εφαρμογής θα ήθελε ο σχεδιαστής να προσθέσει κάτι ακόμα στις ιδιότητες της εφαρμογής και δεν το είχε προβλέψει κατά την αρχική σχεδίαση της βάσης τότε η αναβάθμιση της εφαρμογής θα ήταν μη υλοποιήσιμη ή η υλοποίησή της θα κόστιζε περισσότερο από το όφελος που προσπαθεί να αποκομίσει. Δηλαδή με λίγα λόγια δεν συμφέρει να προσπαθήσει το εγχείρημα γιατί ίσως το να την φτιάξει από την αρχή θα ήταν πιο σύντομη ή πιο οικονομική διαδικασία.

Όταν είναι πολύ σύνθετη η διαδικασία υλοποίησης της εκάστοτε εφαρμογής πρέπει να σπάσει σε μικρότερα μέρη, για να είναι πιο εύκολα διαχειρίσιμη από τον προγραμματιστή και την ομάδα του. Εάν ο σχεδιαστής θέλει το πρόγραμμα να υλοποιηθεί σε πολλαπλές πλατφόρμες τότε η χρήση του Unity είναι ιδανική .

Σαν μια μελλοντική αναβάθμιση στο πρόγραμμα θα εισαχθούν πιο έντονα στοιχεία gamification (παιχνιδοποίησης) και οπτικοακουστικής πληροφορίας. Ο χρήστης θα έχει πρόσβαση σε εκπαιδευτικά tutorials και σε εξωτερική βάση δεδομένων για την χρήση της εφαρμογής. Θα εφαρμοστεί μια ροή ενημέρωσης για καινούργιες πληροφορίες όπου ο χρήστης σε πραγματικό χρόνο θα ενημερώνεται είτε με εικόνα, είτε με ήχο, είτε με κείμενο ή συνδυασμό αυτών, για τυχόν καινούργια ερωτηματολόγια, ασκήσεις, εργασίες, παιχνίδια κτλ.

Όλο το παιχνίδι θα τοποθετηθεί σε ένα πλαίσιο στο οποίο η όλη διάδραση του χρήστη με την εφαρμογή θα έχει τη μορφή εξιστόρησης ενός παραμυθιού ή μιας περιπέτειας. Όσο προχωράει τα επίπεδα ο παίκτης θα δυσκολεύει η εφαρμογή προσθέτοντας επιπλέον μηχανισμούς και έννοιες που ο παίκτης θα πρέπει να αφομοιώσει και να συνδυάσει για να προχωρήσει παρακάτω την ιστορία. Σαν ανταμοιβή ο χρήστης θα κερδίζει εμπειρία και θα ανεβαίνει Level που θα του δίνουν έξτρα ζωές και δυνάμεις. Με αυτό τον τρόπο δημιουργείται το συναίσθημά στον χρήστη ότι συμμετέχει και αυτός στην διαμόρφωση της

ιστορίας και έτσι πετυχαίνουμε να μετακινήσουμε την βαρύτητα της εκπαιδευτικής διαδικασίας υποσυνείδητα σε δεύτερο χρόνο προβάλλοντας σε πρώτο χρόνο την χαλαρότητα και διασκέδαση ενός παιχνιδιού. Έτσι το παιδί δεν θα κουράζεται εύκολα πιέζοντας τον εαυτό του να μάθει ένα εκπαιδευτικό αντικείμενο γιατί θα έχει την αίσθηση ότι παίζει ένα παιχνίδι. Θα υπάρχει δυνατότητα ο χρήστης να επιλέξει σε ποια γλώσσα πέραν της Ελληνικής θέλει να παίζει αλλάζοντας και την θεματολογία των δραστηριοτήτων όπου αυτό χρειάζεται για να ανταποκρίνεται στην εκάστοτε γλώσσα. Θα προστεθεί φίλτρο για την αντιμετώπιση χρήσης κακής γλώσσας. Τέλος θα γίνει υλοποίηση κάποιων μεθόδων anti cheat ώστε να διασφαλιστεί η ακέραιη χρήση και ορθή λειτουργία της εφαρμογής κάτω από οποιοσδήποτε συνθήκες.

Βιβλιογραφία

- [1] “Discover, share, Play anywhere,” *SIMMER.io*, Sep-2017. [Online]. Available: <https://simmer.io/>. [Accessed: 15-Apr-2023].
- [2] “Free and open source 2D and 3D game engine,” *Godot Engine*, Feb-2014. [Online]. Available: <https://godotengine.org/>.
- [3] “Git”, <https://git-scm.com/> (accessed May 16, 2023).
- [4] “Imagine create grow do More imagine with Unity,” *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*, 08-Jun-2005. [Online]. Available: <https://unity.com/>.
- [5] “Let's build from here,” *GitHub*. [Online]. Available: <https://github.com/>. [Accessed: 05-May-2023].
- [6] “Solve the hardest devops challenges with the power of perforce,” *Perforce Software | Development Tools For Innovation at Scale*, 24-May-2023. [Online]. Available: <https://www.perforce.com/>. [Accessed: 05-May-2023].
- [7] “The complete solution for next-generation game development by Crytek,” *CRYENGINE*, 2004. [Online]. Available: <https://www.cryengine.com/>.
- [8] “The DEVSECOPS platform,” *GitLab*. [Online]. Available: <https://about.gitlab.com/>. [Accessed: 05-May-2023].
- [9] “The most powerful real-time 3D creation tool,” *Unreal Engine*, 1998. [Online]. Available: <https://www.unrealengine.com/en-US>.
- [10] “Visual Studio: IDE and Code Editor for Software Developers and Teams,” Visual Studio, May 23, 2023. <https://visualstudio.microsoft.com/>
- [11] Appetize.io. (n.d.). Run native mobile apps in your browser. Retrieved December 13, 2020, from <https://appetize.io/>
- [12] Atlassian, “Git solution for teams using jira,” *Bitbucket*. [Online]. Available: <https://bitbucket.org/>. [Accessed: 05-May-2023].
- [13] E. Christopoulou and S. Xinogalos, “Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices,” *International Journal of Serious Games*, vol. 4, no. 4, Dec. 2017, doi: 10.17083/ijsg.v4i4.194. From https://www.researchgate.net/publication/322027338_Overview_and_Comparative_Analysis_of_Game_Engines_for_Desktop_and_Mobile_Devices

-
- [14] F. Lanzinger, 2D Game Development with Unity. CRC Press, 2020.
- [15] itch.io, "itch.io," itch.io. <https://itch.io/>
- [16] J. Hocking, Unity in Action, Third Edition: Multiplatform Game Development in C#. Simon and Schuster, 2022.
- [17] N. A. Borromeo, Hands-On Unity 2021 Game Development - Second Edition: Create, Customize, and Optimize Your Own Professional Games from Scratch with Unity 2021. 2021.
- [18] P. Borysov, "Developing Mobile Apps for Educational Purposes," *IDAP Blog*, 02-Jan-2022. From <https://idapgroup.com/blog/educational-app-development/>
- [19] P. Buttfield-Addison, J. Manning, and T. Nugent, Unity Game Development Cookbook: Essentials for Every Game. O'Reilly Media, 2019.
- [20] R. Damm, "Introducing the new Input System," *Unity Blog*, 14-Oct-2019. From <https://blog.unity.com/technology/introducing-the-new-input-system>
- [21] R. Rothman and J. Spain, "11 Best Apps for Kids and Students of 2023," *Good Housekeeping*, Feb. 06, 2023. [Online]. Available: <https://www.goodhousekeeping.com/electronics/g28212386/best-apps-for-kids/>
- [22] R. Salama and M. Elsayed, "A live comparison between [Unity](#) and [Unreal](#) Game Engines," *Global Journal of Information Technology: Emerging Technologies*, vol. 11, no. 1, pp. 01–07, 2021. From https://www.researchgate.net/publication/354892247_A_live_comparison_between_Unity_and_Unreal_game_engines
- [23] S. Tykoski, Mastering Game Design with Unity 2021: Immersive Workflows, Visual Scripting, Physics Engine, GameObjects, Player Progression, Publishing, and a Lot More (English Edition). BPB Publications, 2022.
- [24] T. K. Mohd, F. Bravo-Garcia, L. Love, M. Gujadhur, and J. Nyadu, "Analyzing strengths and weaknesses of Modern Game Engines," *International Journal of Computer Theory and Engineering*, vol. 15, no. 1, pp. 54–60, 2023. From https://www.researchgate.net/publication/368590412_Analyzing_Strengths_and_Weaknesses_of_Modern_Game_Engines

Παράρτημα Α.

GameManager.cs

```

using UnityEngine;
using DG.Tweening;
using TMPro;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class GameManager : MonoBehaviour
{
    public static GameManager instance;
    public Canvas mainCanvas;
    public GameObject contentLevel1;
    public GameObject contentLevel2;
    public RectTransform level1Panel;
    public RectTransform level2Panel;
    public GameObject level1Path;
    public GameObject level1Props;
    public GameObject level2Path;
    public GameObject level2Props;
    public RectTransform miniGamePanel;
    public RectTransform MathPanel;
    public RectTransform MemoryPanel;
    public RectTransform MusicPanel;
    //option panel -----
    public RectTransform OptionMenuPanel;
    public RectTransform SoundMenu;
    public RectTransform OptionMenu;
    //end of option panel -----
    public GameObject infoPanel;
    public GameObject btnNextLevel;
    public RectTransform fogOfWar;
    public GameObject contentLevel2Reached;
    public GameObject contentLevel1Reached;
    public int hearts = 4;
    public GameObject HeartFirst;
    public GameObject HeartSecond;
    public GameObject HeartThird;
    public GameObject HeartFourth;
    public AudioManager audioManager;
    public GameObject[] checkPoint;
    public GameObject[] Stars;
    public string difficulty = "Normal";
    public RectTransform leftEnd;
    public RectTransform rightEnd;
    public Transform clouds;
    public int currentPlace = 0;
    public Button startMiniGameBtn;
    public Button startMiniGameBtn2;
    public Button startMiniGameBtn3;
    public SpriteRenderer Level6checkpointStar1;
    public SpriteRenderer Level6checkpointStar2;
    public SpriteRenderer Level6checkpointStar3;
    public SpriteRenderer Level13checkpointStar1;
    public SpriteRenderer Level13checkpointStar2;
    public SpriteRenderer Level13checkpointStar3;
    public GameObject levelSignAnimation;
    public GameObject level2SignAnimation;
    public TextMeshProUGUI totalStars;
    public GameObject TotalStars;//shake effect
    public int CountWinStars;
    //public Button resumeButton;
    //public bool isPaused = false;
    //public CanvasGroup pauseMenu;
    public bool debugcheck;
    public GameObject Player;
    public GameObject Highscore;
    public GameObject HandButtonDownMiddle;

```

```

public GameObject HandButtonDownRight;
public GameObject HandButtonDownLeft;
public GameObject GameOverMessage;

private void Awake()
{
    if (instance != null)
    {
        Destroy(gameObject);
    }
    else
    {
        instance = this;
    }
}

#if !(DEVELOPMENT_BUILD || UNITY_EDITOR)
// Disable Unity logger in non-development builds to improve performance
    Debug.unityLogger.logEnabled = false;
#endif
}

private void Start()
{
    levelSignAnimation.SetActive(true);
    //resumeButton.onClick.AddListener(TogglePause);
}

private void Update()
{
    checkPoint[currentPlace].SetActive(true);

    if (Level6checkpointStar1.enabled || Level6checkpointStar2.enabled || Level6checkpointStar3.enabled)
    {
        fogOfWar.DOAnchorPos(new Vector2(3000f, 4f), 9f);
        btnNextLevel.SetActive(true);
    }
    if(Level13checkpointStar1.enabled || Level13checkpointStar2.enabled || Level13checkpointStar3.enabled)
    {
        Invoke(nameof(Win),3f);
    }

    if (debugcheck)//only for debug to check win condition
    {
        Level13checkpointStar1.enabled = true;
        NextLevel();
    }

    //if (Input.GetKeyDown(KeyCode.Escape))
    //{
    //    //TogglePause();
    //    //resumeButton.enabled = true;
    //}

    #region Heart Display
    if (hearts == 4)
    {
        //Debug.Log("Full life");
    }
    else if (hearts == 3)
    {
        HeartFourth.SetActive(false);
    }
    else if (hearts == 2)
    {
        //HeartFourth.SetActive(false);
        HeartThird.SetActive(false);
    }
    else if (hearts == 1)
    {
        //HeartFourth.SetActive(false);
        //HeartThird.SetActive(false);
        HeartSecond.SetActive(false);
    }
}

```

```

    }
    else
    {
        //HeartFourth.SetActive(false);
        //HeartThird.SetActive(false);
        //HeartSecond.SetActive(false);
        HeartFirst.SetActive(false);
        Debug.Log("GameOver !");
        Lose();
        audioManager.Play("SFX_Lose");
    }
#endregion Heart Display
}

public void NextLevel()
{
    contentLevel1Reached.SetActive(false);
    contentLevel2Reached.SetActive(true);
    level2SignAnimation.SetActive(true);
}

#region Mini Game Panel

public void ShowMiniGamesPanel()
{
    miniGamePanel.DOAnchorPos(Vector2.zero, 1f);
    CheckDifficulty();
    HandButtonDownMiddle.SetActive(true);
    HandButtonDownRight.SetActive(true);
    HandButtonDownLeft.SetActive(true);
}

public void HideMiniGamesPanel()
{
    miniGamePanel.DOAnchorPos(new Vector2(-2000f, 0f), 1f);
    HandButtonDownMiddle.SetActive(false);
    HandButtonDownRight.SetActive(false);
    HandButtonDownLeft.SetActive(false);
    Debug.Log("hide mini game panel");
}

#endregion Mini Game Panel

#region Math Mini Game

public void ShowMathPanel()
{
    //hide the mini game panel in order not to overlap with the mini game panel
    miniGamePanel.DOAnchorPos(new Vector2(-2000f, 0f), 1f);

    MathPanel.DOAnchorPos(Vector2.zero, 1f);
    startMiniGameBtn.enabled = true;
}

public void HideMathPanel()
{
    MathPanel.DOAnchorPos(new Vector2(0f, 2000f), 1f);
    startMiniGameBtn.enabled = false;
    gameObject.GetComponent<MathProblems>().ResetBtn();
}

#endregion Math Mini Game

#region Memory Mini Game

public void ShowMemoryPanel()
{
    //hide the mini game panel in order not to overlap with the mini game panel
    miniGamePanel.DOAnchorPos(new Vector2(-2000f, 0f), 1f);

    MemoryPanel.DOAnchorPos(Vector2.zero, 1f);
    GetComponent<MemoryGameLogic>().StartMemoryGame();
    startMiniGameBtn2.enabled = true;
}

```



```

}

public void HideMemoryPanel()
{
    MemoryPanel.DOAnchorPos(new Vector2(0f, -2000f), 1f);
    startMiniGameBtn2.enabled = false;
    gameObject.GetComponent<MemoryGameLogic>().ResetMemoryBtn();
}

#endregion Memory Mini Game

#region Music Game Panel

public void ShowMusicPanel()
{
    //hide the mini game panel in order not to overlap with the mini game panel
    miniGamePanel.DOAnchorPos(new Vector2(-2000f, 0f), 1f);
    gameObject.GetComponent<MusicGameLogic>().RandomList();
    MusicPanel.DOAnchorPos(Vector2.zero, 1f);
    startMiniGameBtn3.enabled = true;
}

public void HideMusicPanel()
{
    MusicPanel.DOAnchorPos(new Vector2(3000f, 0f), 1f);
    startMiniGameBtn3.enabled = false;
    gameObject.GetComponent<MusicGameLogic>().ResetMusicBtn();
}

#endregion Music Game Panel

#region Option Menu Panel

public void ShowOptionPanel() {
    OptionMenuPanel.DOAnchorPos(Vector2.zero, 1f);
    Debug.Log("show Option panel");
}

public void HideOptionPanel()
{
    OptionMenuPanel.DOAnchorPos(new Vector2(-2000f, -2000f), 1f);
    Debug.Log("hide Option panel");
}
#endregion Option Menu Panel

//Update is called once per frame

public void SetDifficultyToHard()
{
    difficulty = "Hard";
    Debug.Log("Hard difficulty");
}

public void SetDifficultyToEasy()
{
    difficulty = "Easy";
    Debug.Log("Easy difficulty");
}

public void SetDifficultyToNormal()
{
    difficulty = "Normal";
    Debug.Log("Normal difficulty");
}

public int GetCheckPointNumber()
{
    if (currentPlace == 13)
    {
        return currentPlace;
    }
    else if (currentPlace >= 0 || currentPlace <= 12)
    {
        checkPoint[currentPlace + 1].SetActive(true);
    }
}

```

```

    }
    return currentPlace;
}

public void CheckDifficulty()
{
    switch (difficulty)
    {
        case "Normal":
            GetComponent<MusicGameLogic>().TimeLeftMusic = 45f;
            GetComponent<MemoryGameLogic>().TimeLeft = 30f;
            GetComponent<MathProblems>().life = 3;
            GetComponent<MathProblems>().tries = 3;
            break;
        case "Easy":
            GetComponent<MusicGameLogic>().TimeLeftMusic = 60f;
            GetComponent<MemoryGameLogic>().TimeLeft = 60f;
            GetComponent<MathProblems>().life = 6;
            GetComponent<MathProblems>().tries = 6;
            break;
        case "Hard":
            GetComponent<MusicGameLogic>().TimeLeftMusic = 30f;
            GetComponent<MemoryGameLogic>().TimeLeft = 15f;
            GetComponent<MathProblems>().life = 1;
            GetComponent<MathProblems>().tries = 3;
            break;
    }
}

public void CountTotalStars()
{
    CountWinStars = 0;
    for(int i = 0; i <= 13; i++)
    {
        for(int k = 3; k <= 5; k++)
        {
            if (Stars[i].GetComponentInChildren<SpriteRenderer>()[k].enabled == true)
                CountWinStars ++;
        }
    }
    totalStars.text = CountWinStars.ToString();
    TotalStars.transform.DOShakeScale(10f);
}

public void CheckMath()// called when the player answers all the Math problems
{
    if (gameObject.GetComponent<MathProblems>().number == 3)
    {
        for (int i = 3; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MathProblems>().number == 2)
    {
        for (int i = 4; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MathProblems>().number == 1)
    {
        for (int i = 5; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    CountTotalStars();
}

public void CheckMemory()// called when the player answers all the Math problems

```

```

{
    if (gameObject.GetComponent<MemoryGameLogic>().stars == 3)
    {
        for (int i = 3; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MemoryGameLogic>().stars == 2)
    {
        for (int i = 4; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MemoryGameLogic>().stars == 1)
    {
        for (int i = 5; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    CountTotalStars();
}

public void CheckMusic()// called when the player answers all the Math problems
{
    if (gameObject.GetComponent<MusicGameLogic>().count == 3)
    {
        for (int i = 3; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MusicGameLogic>().count == 2)
    {
        for (int i = 4; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    else if (gameObject.GetComponent<MusicGameLogic>().count == 1)
    {
        for (int i = 5; i <= 5; i++)
        {
            Stars[currentPlace].GetComponentsInChildren<SpriteRenderer>()[i].enabled = true;
            Debug.Log($"Happy Stars {i}");
        }
    }
    CountTotalStars();
}

public void Restart()
{
    Time.timeScale = 1.0f;
    // set UI text
}

//public void TogglePause()
//{
//    isPaused = Time.timeScale == 0;
//    Time.timeScale = isPaused ? 1 : 0;
//    AudioListener.pause = !isPaused;
//    //pauseMenu.alpha = isPaused ? 0 : 1;
//    pauseMenu.interactable = !isPaused;
//    pauseMenu.blocksRaycasts = !isPaused;
//}

public void MovePlayerToNextCheckPoint()

```

```

    {
        Player.GetComponent<DOTweenPath>().DOGoto(5, true);
    }

    public void Win()
    {
        Highscore.SetActive(true);
        totalStars.text = CountWinStars.ToString();
        AudioManager.Stop("Music");
        //Time.timeScale = 0.0f;
        // set UI text
        GameObject.Find("Star Yellow").GetComponentInChildren<TextMeshProUGUI>().text = total-
Stars.text;
    }

    // called if the remaining time on a problem reaches 0
    public void Lose()
    {
        //Time.timeScale = 0.0f;
        // set UI text
        GameOverMessage.SetActive(true);
    }

    public void MainMenuBtn()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
    }

    public void ReturnToMainMenuAfterATime()
    {
        Invoke(nameof(MainMenuBtn), 4.0f);
    }
}

```

MathProblems.cs

```

using UnityEngine;
using TMPro;
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;

public class MathProblems : MonoBehaviour
{
    private int firstNumber, secondNumber, temporaryNumber, finalNumber;
    //private string Addition, Subtraction, Multiplication, Division;
    public TextMeshProUGUI firstNumberText;
    public TextMeshProUGUI secondNumberText;
    public TextMeshProUGUI firstResult;
    public TextMeshProUGUI secondResult;
    public TextMeshProUGUI thirdResult;
    public List<Image> imageListOperation = new List<Image>();
    public Button firstBtn;
    public Button secondBtn;
    public Button thirdBtn;
    public Button startMiniGameBtn;
    public bool allowChoice = false;
    public TextMeshProUGUI lifeLeft;
    public TextMeshProUGUI wintext;
    public int number;
    public int choice = 0;
    public bool correct = true;
    public int life;
    public int tries;
    public GameObject winImage;
    public GameObject loseImage;
    public GameObject starImage1;
    public GameObject starImage2;
    public GameObject starImage3;
    public AudioManager audioManager;
    public int pickedOperation;
}

```

```

[SerializeField]
private string[] calculations = { "Addition", "Subtraction", "Multiplication", "Division" };
[SerializeField]
private Image correctAnswer1;
[SerializeField]
private Image correctAnswer2;
[SerializeField]
private Image correctAnswer3;
[SerializeField]
private Image wrongAnswer1;
[SerializeField]
private Image wrongAnswer2;
[SerializeField]
private Image wrongAnswer3;
[SerializeField]
private GameObject nextQuestion;
public Canvas CanvasTopBar;
public TextMeshProUGUI TimerText;
public bool TimerOn = false;
public float TimeLeft;
public TextMeshProUGUI TriesLeft;

public void Update()
{
    if (TimerOn)
    {
        if (TimeLeft > 0)
        {
            TimeLeft -= Time.deltaTime;
            UpdateTimer(TimeLeft);
        }
        else
        {
            //LoseImage.SetActive(true);
            life--;
            Debug.Log("Time is UP!");
            TimeLeft = 0;
            TimerOn = false;
            StartCoroutine(StartResetMiniGame());
            loseImage.SetActive(true);
            gameObject.GetComponent<GameManager>().hearts--;
            audioManager.Play("SFX_Lose");
        }
    }
}

public void CheckDifficulty()
{
    if (gameObject.GetComponent<GameManager>().difficulty == "Normal")
    {
        life = 3;
        tries = 3;
        audioManager.Play("Vocals_3Lives_3Tries");
        TriesLeft.text = "Προσπάθειες " + tries.ToString();
        //TimeLeft = 20f;
    }
    else if (gameObject.GetComponent<GameManager>().difficulty == "Easy")
    {
        life = 6;
        tries = 6;
        audioManager.Play("Vocals_6Lives_6Tries");
        TriesLeft.text = "Προσπάθειες " + tries.ToString();
        //TimeLeft = 30f;
    }
    else if (gameObject.GetComponent<GameManager>().difficulty == "Hard")
    {
        life = 1;
        tries = 3;
        audioManager.Play("Vocals_1Life_3Tries");
        TriesLeft.text = "Προσπάθειες " + tries.ToString();
        //TimeLeft = 10f;
    }
}

```

```

public void OnBtnStart()
{
    if (tries > 0)
    {
        //reset the numbers to 0 AND the list of operators to empty
        firstNumber = 0; secondNumber = 0; temporaryNumber = 0; finalNumber = 0; pickedOperation =
0; choice = 0;
        firstNumberText.text = "";
        secondNumberText.text = "";
        firstResult.text = "";
        secondResult.text = "";
        thirdResult.text = "";
        startMiniGameBtn.enabled = true;
        correctAnswer1.GetComponent<Image>().enabled = false;
        correctAnswer2.GetComponent<Image>().enabled = false;
        correctAnswer3.GetComponent<Image>().enabled = false;
        wrongAnswer1.GetComponent<Image>().enabled = false;
        wrongAnswer2.GetComponent<Image>().enabled = false;
        wrongAnswer3.GetComponent<Image>().enabled = false;
        firstBtn.enabled = true;
        secondBtn.enabled = true;
        thirdBtn.enabled = true;
        allowChoice = true;
        correct = false;
        number = 0;
        lifeLeft.text = life.ToString();
        CheckOperations();
        audioManager.Play("SFX_ButtonHigh");
        nextQuestion.SetActive(false);
        TimeLeft = 30f;
        TimerOn = true;
        TimerText.text = TimeLeft.ToString();
        TriesLeft.text = "Προσπάθειες " + tries.ToString();
    }
}

public void ResetBtn()
{
    firstNumber = 0; secondNumber = 0; temporaryNumber = 0; finalNumber = 0; pickedOperation = 0;
choice = 0;
    firstNumberText.text = "";
    secondNumberText.text = "";
    firstResult.text = "";
    secondResult.text = "";
    thirdResult.text = "";
    startMiniGameBtn.enabled = true;
    for (int i = 0; i <= imageListOperation.Count - 1; i++)//here we reset all the sprite opera-
tors to false in order to start all over again
    {
        imageListOperation[i].enabled = false;
    }
    correctAnswer1.GetComponent<Image>().enabled = false;
    correctAnswer2.GetComponent<Image>().enabled = false;
    correctAnswer3.GetComponent<Image>().enabled = false;
    wrongAnswer1.GetComponent<Image>().enabled = false;
    wrongAnswer2.GetComponent<Image>().enabled = false;
    wrongAnswer3.GetComponent<Image>().enabled = false;
    firstBtn.enabled = true;
    secondBtn.enabled = true;
    thirdBtn.enabled = true;
    allowChoice = true;
    correct = false;
    number = 0;
    nextQuestion.SetActive(true);
    TimeLeft = 30f;
    TimerOn = false;
    TimerText.text = TimeLeft.ToString();
    audioManager.Play("SFX_Close");
    lifeLeft.text = "";
    loseImage.SetActive(false);
    winImage.SetActive(false);
    starImage1.SetActive(false);
    starImage2.SetActive(false);
    starImage3.SetActive(false);
}

```

```

    TriesLeft.text = "Προσπάθειες " + tries.ToString();
}

public void UpdateTimer(float currentTime)
{
    currentTime++; //overwrite in the game manager script
    float minutes = Mathf.FloorToInt(currentTime / 60);
    float seconds = Mathf.FloorToInt((currentTime % 60) - 1);
    TimerText.text = string.Format("{0:00} : {1:00}", minutes, seconds);
}

public int RandomNumberCalc() //choose a random number to use with the operation picker method(checkOperations())
{
    int randomIndex = Random.Range(0, 4);
    Debug.Log("randomNumberCalc : " + randomIndex);

    return randomIndex;
}

public void CheckOperations()
{
    for (int i = 0; i <= imageListOperation.Count-1; i++) //here we reset all the sprite operators to false in order to start all over again
    {
        imageListOperation[i].enabled = false;
        Debug.Log("checkOperations - Sprite list count : " + imageListOperation[i]);
    }
    pickedOperation = RandomNumberCalc(); //we use the previous method to take a random number and compare it with the saved calculation operators

    if (pickedOperation == 0)
    {
        imageListOperation[0].enabled = true;
        CalcRandomNumbers(calculations[0]); //here is addition
    }
    else if (pickedOperation == 1)
    {
        imageListOperation[1].enabled = true;
        CalcRandomNumbers(calculations[1]); //here is subtraction
    }
    else if (pickedOperation == 2)
    {
        imageListOperation[2].enabled = true;
        CalcRandomNumbers(calculations[2]); //here is multiplication
    }
    else if (pickedOperation == 3)
    {
        imageListOperation[3].enabled = true;
        CalcRandomNumbers(calculations[3]); //here is division
    }
}

public void CalcRandomNumbers(string pickedOperation) //use the chosen operation with random numbers
{
    firstNumber = Random.Range(1, 10);
    //firstNumberText.text = firstNumber.ToString();
    secondNumber = Random.Range(1, 10);
    //secondNumberText.text = secondNumber.ToString();

    //we want always to have a positive number so... we change places the number if the first is smaller than the second to achieve that
    if (firstNumber - secondNumber <= 0)
    {
        temporaryNumber = secondNumber;
        secondNumber = firstNumber;
        firstNumber = temporaryNumber;
        firstNumberText.text = firstNumber.ToString();
        secondNumberText.text = secondNumber.ToString();
    }
    else
    {
        firstNumberText.text = firstNumber.ToString();
        secondNumberText.text = secondNumber.ToString();
    }
}

```

```

    }

    switch (pickedOperation)//implementing the math operation with the chosen case
    {
        case "Addition":
            finalNumber = firstNumber + secondNumber;
            //firstResult.text = finalNumber.ToString();
            break;

        case "Subtraction":
            finalNumber = firstNumber - secondNumber;
            //firstResult.text = finalNumber.ToString();
            break;

        case "Multiplication":
            finalNumber = firstNumber * secondNumber;
            //firstResult.text = finalNumber.ToString();
            break;

        case "Division":
            finalNumber = firstNumber / secondNumber;
            //firstResult.text = finalNumber.ToString();
            break;
    }
    int showAnswers = Random.Range(0, 3);//random logic to pick a new spot to show the correct
    answer each time

    if (showAnswers == 0)//where to show the correct answer with random logic
    {
        firstResult.text = finalNumber.ToString();
        secondResult.text = Random.Range(1, 82).ToString();
        thirdResult.text = Random.Range(1, 82).ToString();
        if (secondResult.text == thirdResult.text || secondResult.text == firstResult.text ||
        thirdResult.text == firstResult.text)
        {
            secondResult.text = Random.Range(1, 82).ToString();
            thirdResult.text = Random.Range(1, 82).ToString();
        }
    }
    else if(showAnswers == 1)
    {
        firstResult.text = Random.Range(1, 82).ToString();
        secondResult.text = finalNumber.ToString();
        thirdResult.text = Random.Range(1, 82).ToString();
        if (firstResult.text == thirdResult.text || firstResult.text == secondResult.text || thir-
        dResult.text == secondResult.text)
        {
            firstResult.text = Random.Range(1, 82).ToString();
            thirdResult.text = Random.Range(1, 82).ToString();
        }
    }
    else if(showAnswers == 2)
    {
        firstResult.text = Random.Range(1, 82).ToString();
        secondResult.text = Random.Range(1, 82).ToString();
        thirdResult.text = finalNumber.ToString();
        if (firstResult.text == secondResult.text || firstResult.text == thirdResult.text ||
        secondResult.text == thirdResult.text)
        {
            firstResult.text = Random.Range(1, 82).ToString();
            secondResult.text = Random.Range(1, 82).ToString();
        }
    }
    Debug.Log("The first number was : " + firstNumber + " The operation was "+ pickedOperation + "
    and the second number was : " + secondNumber + " THE FINAL RESULT IS : " + finalNumber);
}

public int PlayersChoice(int choice)
{
    if (tries > 0)
    {
        if (finalNumber.ToString() == firstResult.text)
        {
            if (choice == 1)

```



```

    {
        //correct = true;
        tries --;
        Debug.Log("Bravo the correct choice is 1! tries left : " + tries + " life : " +
life);

        //checkLifeAndTries(tries, life);
        correctAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer2.GetComponent<Image>().enabled = true;
        wrongAnswer3.GetComponent<Image>().enabled = true;
        audioManager.Play("SFX_Win");
    }
    else
    {
        //correct = false;
        tries --;
        life--;
        lifeLeft.text = life.ToString();
        Debug.Log("sorry wrong choice! tries left : " + tries + " life : " + life);
        correctAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer2.GetComponent<Image>().enabled = true;
        wrongAnswer3.GetComponent<Image>().enabled = true;
        audioManager.Play("SFX_Stun1");
    }
}
else if (finalNumber.ToString() == secondResult.text)
{
    if (choice == 2)
    {
        //correct = true;
        tries--;
        Debug.Log("Bravo the correct choice is 2! tries left : " + tries + " life : " +
life);

        //checkLifeAndTries(tries, life);
        correctAnswer2.GetComponent<Image>().enabled = true;
        wrongAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer3.GetComponent<Image>().enabled = true;
        audioManager.Play("SFX_Win");
    }
    else
    {
        //correct = false;
        tries --;
        life--;
        lifeLeft.text = life.ToString();
        Debug.Log("sorry wrong choice! tries left : " + tries + " life : " + life);
        correctAnswer2.GetComponent<Image>().enabled = true;
        wrongAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer3.GetComponent<Image>().enabled = true;
        audioManager.Play("SFX_Stun1");
    }
}
else if (finalNumber.ToString() == thirdResult.text)
{
    if (choice == 3)
    {
        //correct = true;
        tries --;
        Debug.Log("Bravo the correct choice is 3! tries left : " + tries + " life : " +
life);

        correctAnswer3.GetComponent<Image>().enabled = true;
        wrongAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer2.GetComponent<Image>().enabled = true;
        audioManager.Play("SFX_Win");
    }
    else
    {
        //correct = false;
        tries --;
        life--;
        lifeLeft.text = life.ToString();
        Debug.Log("sorry wrong choice! tries left : " + tries + " life : " + life);
        correctAnswer3.GetComponent<Image>().enabled = true;
        wrongAnswer1.GetComponent<Image>().enabled = true;
        wrongAnswer2.GetComponent<Image>().enabled = true;

```

```

        audioManager.Play("SFX_Stun1");
    }
}
return tries;
}

public void BtnOneChoice()// 3 possible answers 3 buttons from 1 to 3 the players choice will go
to the playersChoice(int choice) method above
{
    //check the graphics panel for the method check the buttons under result panel 16
    if (allowChoice)
    {
        choice = 1;
        Debug.Log("Player Button choice :" + 1);
        PlayersChoice(choice);
        firstBtn.enabled = false;
        secondBtn.enabled = false;
        thirdBtn.enabled = false;
        nextQuestion.SetActive(true);
        if (tries == 0 || life == 0)
        {
            startMiniGameBtn.enabled = false;
            number = SetStars(CheckLifeAndTries(life));
        }
    }
}

public void BtnTwoChoice()// 3 possible answers 3 buttons from 1 to 3 the players choice will go
to the playersChoice(int choice) method above
{
    //check the graphics panel for the method check the buttons under result panel 16
    if (allowChoice)
    {
        choice = 2;
        Debug.Log("Player Button choice :" + 2);
        PlayersChoice(choice);
        firstBtn.enabled = false;
        secondBtn.enabled = false;
        thirdBtn.enabled = false;
        nextQuestion.SetActive(true);
        if (tries == 0 || life == 0)
        {
            startMiniGameBtn.enabled = false;
            number = SetStars(CheckLifeAndTries(life));
        }
    }
}

public void BtnThreeChoice()// 3 possible answers 3 buttons from 1 to 3 the players choice will go
to the playersChoice(int choice) method above
{
    //check the graphics panel for the method check the buttons under result panel 16
    if (allowChoice)
    {
        choice = 3;
        Debug.Log("Player Button choice :" + 3);
        PlayersChoice(choice);
        firstBtn.enabled = false;
        secondBtn.enabled = false;
        thirdBtn.enabled = false;
        nextQuestion.SetActive(true);
        if (tries == 0 || life == 0)
        {
            startMiniGameBtn.enabled = false;
            number = SetStars(CheckLifeAndTries(life));
        }
    }
}

public string CheckLifeAndTries(int life)
{
    TimerOn = false;
    string result = "";
    Debug.Log("Tries left : " + tries);
    Debug.Log("Life left : " + life);
    if (gameObject.GetComponent<GameManager>().difficulty == "Normal")

```

```
{
    if (life == 3)
    {
        Debug.Log("You win!!Perfect score !3 Stars");
        result = "Άριστα";
    }
    else if (life == 2)
    {
        Debug.Log("You win!!2 Stars");
        result = "Πολύ καλά";
    }
    else if (life == 1)
    {
        Debug.Log("You win!!1 Stars");
        result = "Καλά";
    }
    else if (life == 0)
    {
        Debug.Log("Sorry you lose!");
        correct = false;
        result = "Προσπάθησε ξανά!";
    }
}

if (gameObject.GetComponent<GameManager>().difficulty == "Hard")
{
    if (life == 1)
    {
        Debug.Log("You win!!3 Stars");
        result = "Άριστα";
    }
    else if (life == 0)
    {
        Debug.Log("Sorry you lose!");
        correct = false;
        result = "Προσπάθησε ξανά!";
    }
}

if (gameObject.GetComponent<GameManager>().difficulty == "Easy")
{
    if (life == 6)
    {
        Debug.Log("You win!!Perfect score !3 Stars");
        result = "Άριστα";
    }
    else if (life == 5)
    {
        Debug.Log("You win!!2 Stars");
        result = "Πολύ καλά";
    }
    else if (life == 4)
    {
        Debug.Log("You win!!2 Stars");
        result = "Πολύ καλά";
    }
    else if (life == 3)
    {
        Debug.Log("You win!!2 Stars");
        result = "Πολύ καλά";
    }
    else if (life == 2)
    {
        Debug.Log("You win!!1 Stars");
        result = "Καλά";
    }
    else if (life == 1)
    {
        Debug.Log("You win!!1 Stars");
        result = "Καλά";
    }
    else if (life == 0)
    {
        Debug.Log("Sorry you lose!");
    }
}
```

```

        correct = false;
        result = "Προσπάθησε ξανά!";
    }
}

return result;
}

public int SetStars(string result)
{
    StartCoroutine(StarAnimations(result));

    GetComponent<GameManager>().CheckMath();
    //resetBtn();
    //GetComponent<GameManager>().hideMathPanel();
    winText.text = result;
    return number;
}

public IEnumerator StartResetMiniGame()
{
    yield return new WaitForSeconds(2.0f);
    ResetBtn();
    GetComponent<GameManager>().HideMathPanel();
    CanvasTopBar.enabled = true;
}

public IEnumerator StarAnimations(string result)
{
    if (result == "Άριστα")
    {
        winImage.SetActive(true);
        number = 3;
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        starImage2.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.5f);
        starImage3.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else if (result == "Πολύ καλά")
    {
        number = 2;
        winImage.SetActive(true);
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        starImage2.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else if (result == "Καλά")
    {
        number = 1;
        winImage.SetActive(true);
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        AudioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else
    {
        number = 0;
        loseImage.SetActive(true);
        gameObject.GetComponent<GameManager>().hearts--;
        AudioManager.Play("SFX_Lose");
    }
}

StartCoroutine(StartResetMiniGame());

```

```

    }
}

```

MemoryGameLogic.cs

```

using System.Collections;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class MemoryGameLogic : MonoBehaviour
{
    public int[] IntRandomArray = { 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5 };
    public int[] IntRandomArray2 = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    public GameObject[] CoverImageArray;
    public Image[] HiddenImageArray;
    public Sprite[] PairImages;
    public GameObject[] WinningStars;
    public AudioManager audioManager;
    public float TimeLeft;
    public bool TimerOn = false;
    public int Moves;
    public TextMeshProUGUI TimerText;
    public TextMeshProUGUI MovesText;
    public GameObject StartButton;
    public Image PauseImage;
    public GameObject PauseButton;
    public GameObject GridPanel;
    public MainImageScript startObject;
    public MainImageScript firstOpen;
    public MainImageScript secondOpen;
    public bool GameFinished = false;
    public bool CanOpen { get { return secondOpen == null; } }
    public GameObject WinImage;
    public GameObject LoseImage;
    public int count = 0;
    public int stars = 0;
    public Canvas CanvasTopBar;

    public void Update()
    {
        if (TimerOn)
        {
            if (TimeLeft > 0)
            {
                TimeLeft -= Time.deltaTime;
                UpdateTimer(TimeLeft);
            }
            else
            {
                LoseImage.SetActive(true);
                Debug.Log("Time is UP!");
                TimeLeft = 0;
                TimerOn = false;
                gameObject.GetComponent<GameManager>().hearts--;
                StartCoroutine(StartResetMiniGame());
                audioManager.Play("SFX_Lose");
            }
        }
    }

    public void CheckGameDifficulty()
    {
        if (gameObject.GetComponent<GameManager>().difficulty == "Normal")
        {
            TimeLeft = 30f;
            UpdateTimer(TimeLeft);
            audioManager.Play("Vocals_30_Seconds");
        }
        else if (gameObject.GetComponent<GameManager>().difficulty == "Easy")
        {

```

```

        TimeLeft = 45f;
        UpdateTimer(TimeLeft);
        AudioManager.Play("Vocals_45_Seconds");
    }
    else if (gameObject.GetComponent<GameManager>().difficulty == "Hard")
    {
        TimeLeft = 15f;
        UpdateTimer(TimeLeft);
        AudioManager.Play("Vocals_15_Seconds");
    }
}

public void PauseMemoryGame()
{
    if (TimerOn)
    {
        TimerOn = false;
        StartButton.SetActive(false);
        PauseImage.enabled = false;
        //GridPanel.SetActive(false);
        PauseButton.SetActive(false);
    }
    else
    {
        //TimerOn = true;
        //StartButton.SetActive(true);
        PauseImage.enabled = true;
        PauseButton.SetActive(true);
        //GridPanel.SetActive(true);
    }
}

public void UpdateTimer(float currentTime)
{
    currentTime++; //overwrite in the game manager script

    float minutes = Mathf.FloorToInt(currentTime / 60);
    float seconds = Mathf.FloorToInt((currentTime % 60)-1);

    TimerText.text = string.Format("{0:00} : {1:00}", minutes, seconds);
}

public void ResetMemoryBtn()
{
    stars = 0;
    count = 0;
    Moves = 0;
    MovesText.text = "Moves : " + Moves.ToString();
    //TimeLeft = 30f;
    TimerOn = false;
    firstOpen = null;
    secondOpen = null;
    UpdateTimer(TimeLeft); //update time text
    PauseImage.enabled = false;
    GridPanel.SetActive(true);
    PauseButton.SetActive(false);
    LoseImage.SetActive(false);
    WinImage.SetActive(false);
    WinningStars[3].SetActive(false);
    WinningStars[4].SetActive(false);
    WinningStars[5].SetActive(false);
    //StartButton.SetActive(true);
    AudioManager.Play("SFX_Close");
    for (int j = 0; j <= IntRandomArray2.Length - 1; j++)
    {
        HideImages(j);
    }
}

public void StartMemoryGame()
{
    stars = 0;
    count = 0;
    Moves = 0;

```

```

MovesText.text = "Moves : " + Moves.ToString();
//TimeLeft = 30f;
TimerOn = false;
firstOpen = null;
secondOpen = null;
UpdateTimer(TimeLeft);//update time text
PauseImage.enabled = false;
GridPanel.SetActive(true);
PauseButton.SetActive(false);
//StartButton.SetActive(true);
LoseImage.SetActive(false);
WinImage.SetActive(false);
WinningStars[3].SetActive(false);
WinningStars[4].SetActive(false);
WinningStars[5].SetActive(false);
for (int j = 0; j <= IntRandomArray2.Length - 1; j++)//make a random array to keep the pairs
{
    int item = IntRandomArray[j];
    int randomizeArray = Random.Range(0, j);
    IntRandomArray2[j] = IntRandomArray[randomizeArray];
    IntRandomArray2[randomizeArray] = item;

    HideImages(j);//reset all image cover to be hidden
}

for (int i = 0; i <= IntRandomArray.Length - 1; i++)
{
    if (IntRandomArray2[i] == 0)
    {
        HiddenImageArray[i].sprite = PairImages[0];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 0;
    }
    else if (IntRandomArray2[i] == 1)
    {
        HiddenImageArray[i].sprite = PairImages[1];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 1;
    }
    else if (IntRandomArray2[i] == 2)
    {
        HiddenImageArray[i].sprite = PairImages[2];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 2;
    }
    else if (IntRandomArray2[i] == 3)
    {
        HiddenImageArray[i].sprite = PairImages[3];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 3;
    }
    else if (IntRandomArray2[i] == 4)
    {
        HiddenImageArray[i].sprite = PairImages[4];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 4;
    }
    else if (IntRandomArray2[i] == 5)
    {
        HiddenImageArray[i].sprite = PairImages[5];
        HiddenImageArray[i].GetComponent<MainImageScript>()._spriteId = 5;
    }
}

}

public void HideImages(int ImageToCover)
{
    CoverImageArray[ImageToCover].SetActive(true);//hide images
}

public void ImageOpened(MainImageScript startObject)
{
    TimerOn = true;
    PauseButton.SetActive(true);
    PauseImage.enabled = true;
    if (firstOpen == null)
    {
        firstOpen = startObject;
        audioManager.Play("SFX_Mystic");
    }
}

```

```

    }
    else
    {
        secondOpen = startObject;
        audioManager.Play("SFX_Mystic");
        StartCoroutine(nameof(CheckGuessed));
    }
}

public IEnumerator CheckGuessed()
{
    if (firstOpen.SpriteId == secondOpen.SpriteId)//compare the two objects
    {
        //score++;
        //scoreText.text = score + "/4";
        Debug.Log("true");
        count += 2;
        CheckVictory();
        audioManager.Play("SFX_Target");
    }
    else
    {
        yield return new WaitForSeconds(0.5f);//start timer
        firstOpen.Close();
        secondOpen.Close();
        audioManager.Play("SFX_Stun1");
        Debug.Log("false");
    }
    Moves++;
    MovesText.text = "Moves : " + Moves.ToString();
    firstOpen = null;//reset choice 1
    secondOpen = null;//reset choice 2
}

public void CheckVictory()
{
    if(count != 12)
    {
        GameFinished = false;
        Debug.Log("still covered images");
    }
    else if(count == 12)
    {
        GameFinished = true;
        PauseMemoryGame();
        Debug.Log("end of game you win");
        WinImage.SetActive(true);
        GetComponent<GameManager>().GetCheckPointNumber();
        StartCoroutine(StarAnimations(Moves));
        GetComponent<GameManager>().CheckMemory();
    }
}

public IEnumerator StartResetMiniGame()
{
    yield return new WaitForSeconds(2.0f);
    ResetMemoryBtn();
    GetComponent<GameManager>().HideMemoryPanel();
    CanvasTopBar.enabled = true;
}

public IEnumerator StarAnimations(int Moves)
{
    if (Moves <= 12)
    {
        stars = 3;
        yield return new WaitForSeconds(0.5f);
        WinningStars[3].SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        WinningStars[4].SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.5f);
        WinningStars[5].SetActive(true);
    }
}

```



```

        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else if (Moves > 12 && Moves <= 16)
    {
        stars = 2;
        yield return new WaitForSeconds(0.5f);
        WinningStars[3].SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        WinningStars[4].SetActive(true);
        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else
    {
        stars = 1;
        yield return new WaitForSeconds(0.5f);
        WinningStars[3].SetActive(true);
        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    StartCoroutine(StartResetMiniGame());
}
}
}

```

MusicGameLogic.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPPro;

public class MusicGameLogic : MonoBehaviour
{
    public List<Sprite> SongListSprites = new List<Sprite>();
    public List<Sprite> CurrentSongSprites = new List<Sprite>();
    public List<int> SongNotesKounelaki1 = new List<int>{ 8, 8, 8, 6, 6, 7, 7, 5, 5, 6, 6, 5, 6, 7,
8, 8, 8, 6, 6, 7, 7, 7, 5, 5, 6, 5, 6, 8, 4};
    public List<int> SongNotesPazari2 = new List<int>{ 8, 8, 8, 11, 11, 11, 12, 11, 10, 9, 10, 10, 8,
8, 8, 10, 10, 11, 10, 9, 10, 9, 8, 8, 6, 7, 8, 11, 13, 13, 12, 13, 14, 13, 12, 11, 10, 8, 8, 13, 12,
11};
    public List<int> CurrentSong = new List<int>();
    public List<GameObject> ListOfSongs = new List<GameObject>();
    public List<GameObject> correctAnswersAxKounelaki = new List<GameObject>();
    public List<GameObject> wrongAnswersAxKounelaki = new List<GameObject>();
    public List<GameObject> correctAnswersPazari = new List<GameObject>();
    public List<GameObject> wrongAnswersPazari = new List<GameObject>();
    public List<int> userInputs;
    public GameObject[] pianoKeys;
    public GameObject[] pianoKeyNumbers;
    public bool gamefinished = false;
    public int choice;
    public GameObject startMiniGameBtn;
    public int count;
    public float TimeLeftMusic;
    public bool TimerOnMusic = false;
    public TextMeshProUGUI TimerTextMusicMM;
    public GameObject HorizontalProgressBarTimer;
    public GameObject WinImage;
    public GameObject LoseImage;
    public GameObject starImage1;
    public GameObject starImage2;
    public GameObject starImage3;
    public AudioManager audioManager;
    public int songNoteCurrentChoice;
    public int pianoButtonCurrentChoice;
    public int k = 0;
    public bool GameStarted = false;
}

```

```

public Canvas CanvasTopBar;
public TextMeshProUGUI lifes_tries;
public GameObject PauseButton;

public void Update()
{
    if (TimerOnMusic)
    {
        if (TimeLeftMusic > 0)
        {
            TimeLeftMusic -= Time.deltaTime;
            UpdateTimerMusic(TimeLeftMusic);
            CheckProgressOfSong();

            if (gamefinished)
            {
                CheckForErrors();
            }
        }
        else
        {
            Debug.Log("Time is UP!");
            //LoseImage.SetActive(true);
            TimeLeftMusic = 0;
            //TimerOnMusic = false;
            Lose();
            gameObject.GetComponent<GameManager>().hearts --;
        }
    }
}

public void PauseMusicGame()
{
    if (TimerOnMusic)
    {
        TimerOnMusic = false;
        //startMiniGameBtn.SetActive(false);
        //ResumeImage.enabled = true;
        //PauseImage.sprite = ResumeImage.sprite;
        HorizontalProgressBarTimer.SetActive(false);
    }
    else
    {
        TimerOnMusic = true;
        //startMiniGameBtn.SetActive(true);
        //PauseImage.sprite = PauseImage.sprite;
        //ResumeImage.enabled = false;
        HorizontalProgressBarTimer.SetActive(true);
    }
}

public void PianoBtnChoice(int number)
{
    if (GameStarted)
    {
        if (!gamefinished)
        {
            choice = number;
            Debug.Log("Player Button choice :" + number);
            AddToList(choice);
            ShowProgress();
            if (userInputs[k] != CurrentSong[k])
            {
                count--;
                lifes_tries.text = count.ToString();
                if (count <= 0)
                {
                    Lose();
                }
            }
            k++;
        }
    }
}
}

```

```

public void ShowProgress()
{
    if(choice != CurrentSong[k])
    {
        if(CurrentSong == SongNotesKounelaki1)
        {
            wrongAnswersAxKounelaki[k].SetActive(true);
        }
        else
        {
            wrongAnswersPazari[k].SetActive(true);
        }
    }
    else
    {
        if (CurrentSong == SongNotesKounelaki1)
        {
            correctAnswersAxKounelaki[k].SetActive(true);
        }
        else
        {
            correctAnswersPazari[k].SetActive(true);
        }
    }
}

public List<int> RandomList()
{
    CurrentSongSprites[0] = SongListSprites[Random.Range(0,2)];

    if (CurrentSongSprites[0] == SongListSprites[0])
    {
        ListOfSongs[0].GetComponent<Image>().sprite = SongListSprites[0];
        ListOfSongs[0].SetActive(true);
        ListOfSongs[1].SetActive(false);
        CurrentSong = SongNotesKounelaki1;
    }
    else if(CurrentSongSprites[0] == SongListSprites[1])
    {
        ListOfSongs[1].GetComponent<Image>().sprite = SongListSprites[1];
        ListOfSongs[0].SetActive(false);
        ListOfSongs[1].SetActive(true);
        CurrentSong = SongNotesPazari2;
    }
    return CurrentSong;
}

public bool CheckProgressOfSong()
{
    if (userInputs.Count != CurrentSong.Count)//check to see if the player has played all the
notes
    {
        //gamefinished = false;
        Debug.Log("gameNotFinished");
        return false;
    }
    return gamefinished = true;
}

public void CheckGameDifficulty()
{
    if (gameObject.GetComponent<GameManager>().difficulty == "Normal")
    {
        count = 3;
        lifes_tries.text = count.ToString();
        audioManager.Play("Vocals_45_Seconds_3Lives");
    }
    else if (gameObject.GetComponent<GameManager>().difficulty == "Easy")
    {
        count = 6;
        lifes_tries.text = count.ToString();
        audioManager.Play("Vocals_45_Seconds_6Lives");
    }
}

```

```

else if (gameObject.GetComponent<GameManager>().difficulty == "Hard")
{
    count = 1;
    lifes_tries.text = count.ToString();
    audioManager.Play("Vocals_45_Seconds_1Lives");
}
}

public void CheckForErrors()
{
    for (int i = 0; i < CurrentSong[i]; i++)//check for wrong input notes during play
    {
        if (userInputs[i] != CurrentSong[i])
        {
            count--;
            lifes_tries.text = count.ToString();
        }
    }
    if (count <= 0)
    {
        Lose();
    }
    else
    {
        Win();
        GetComponent<GameManager>().GetCheckPointNumber();
    }
}

public void AddToList(int choice)
{
    userInputs.Add(choice);
}

public void UpdateTimerMusic(float currentTime)
{
    currentTime++;//overwrite in the game manager script
    float minutes = Mathf.FloorToInt(currentTime / 60);
    float seconds = Mathf.FloorToInt((currentTime % 60) - 1);
    TimerTextMusicMM.text = string.Format("{0:00} : {1:00}", minutes, seconds);
}

public void ResetMusicBtn()
{
    for (int i = 0; i < correctAnswersAxKounelaki.Count; i++)
    {
        correctAnswersAxKounelaki[i].SetActive(false);
        wrongAnswersAxKounelaki[i].SetActive(false);
    }
    for (int i = 0; i < correctAnswersPazari.Count; i++)
    {
        correctAnswersPazari[i].SetActive(false);
        wrongAnswersPazari[i].SetActive(false);
    }
    PauseButton.SetActive(false);
    k = 0;
    gamefinished = false;
    //count = 3;
    TimeLeftMusic = 45f;
    TimerOnMusic = false;
    HorizontalProgressBarTimer.SetActive(false);
    UpdateTimerMusic(TimeLeftMusic);//update time text
    WinImage.SetActive(false);
    LoseImage.SetActive(false);
    userInputs.Clear();
    startMiniGameBtn.SetActive(true);
    audioManager.Play("SFX_Close");
    audioManager.Play("Music");
    GameStarted = false;
    lifes_tries.text = count.ToString();
}

public void StartMusicGame()
{

```

```

for (int i = 0; i < correctAnswersAxKounelaki.Count; i++)
{
    correctAnswersAxKounelaki[i].SetActive(false);
    wrongAnswersAxKounelaki[i].SetActive(false);
}
for (int i = 0; i < correctAnswersPazari.Count; i++)
{
    correctAnswersPazari[i].SetActive(false);
    wrongAnswersPazari[i].SetActive(false);
}

k = 0;
//count = 3;
TimeLeftMusic = 45f;
TimerOnMusic = true;
HorizontalProgressBarTimer.SetActive(true);
UpdateTimerMusic(TimeLeftMusic);//update time text
//PauseImage.enabled = false;
//PauseButton.SetActive(false);
//StartButton.SetActive(true);
WinImage.SetActive(false);
LoseImage.SetActive(false);
userInputs.Clear();
startMiniGameBtn.SetActive(false);
audioManager.Play("SFX_ButtonHigh");
audioManager.Stop("Music");
GameStarted = true;
lifes_tries.text = count.ToString();
}

public IEnumerator StartResetMiniGame()
{
    yield return new WaitForSeconds(2.0f);
    ResetMusicBtn();
    GetComponent<GameManager>().HideMusicPanel();
    CanvasTopBar.enabled = true;
}

public void Lose()
{
    LoseImage.SetActive(true);
    TimerOnMusic = false;
    HorizontalProgressBarTimer.SetActive(false);
    gameObject.GetComponent<GameManager>().hearts--;
    Debug.Log("Game Lost -- Errors : " + count);
    StartCoroutine(StartResetMiniGame());
    audioManager.Play("SFX_Lose");
}

public void Win()
{
    WinImage.SetActive(true);

    StartCoroutine(StarAnimations(count));
    GetComponent<GameManager>().CheckMusic();
    TimerOnMusic = false;
    HorizontalProgressBarTimer.SetActive(false);
    Debug.Log("Game Won -- Errors : " + count);
}

public IEnumerator StarAnimations(int count)
{
    if (count == 3)
    {
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        starImage2.SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.5f);
        starImage3.SetActive(true);
        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
}

```

```

    }
    else if (count == 2)
    {
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        audioManager.Play("SFX_Mystic");
        yield return new WaitForSeconds(1.0f);
        starImage2.SetActive(true);
        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    else if (count == 1)
    {
        yield return new WaitForSeconds(0.5f);
        starImage1.SetActive(true);
        audioManager.Play("SFX_Mystic");
        GetComponent<GameManager>().GetCheckPointNumber();
    }
    StartCoroutine(StartResetMiniGame());
}
}
}

```

MainMenu.cs

```

using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour
{
    public TMP_Dropdown resolutionDropdown; // Dropdown UI element for selecting resolutions

    public Resolution[] resolutions; // Array to store available screen resolutions

    public void PlayGame()
    {
        // Load the next scene in the build index
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void QuitPlay()
    {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
        Debug.Log("quit");
#endif
        Application.Quit();
    }

    public void DelayApplicationQuit()
    {
        // Invoke the QuitPlay method with a delay of 4 seconds
        Invoke(nameof(QuitPlay), 4.0f);
    }

    private void Start()
    {
        // Get available screen resolutions
        resolutions = Screen.resolutions;

        resolutionDropdown.ClearOptions(); // Clear the options in the resolution dropdown

        List<string> options = new List<string>(); // Create a list to store resolution options

        int currentResolutionIndex = 0; // Variable to store the index of the current screen resolution

        for (int i = 0; i < resolutions.Length; i++)
        {

```

```
string option = resolutions[i].width + " x " + resolutions[i].height;
options.Add(option); // Add the resolution option to the list

// Check if the current resolution matches the screen's current resolution
if (resolutions[i].width == Screen.currentResolution.width && resolutions[i].height ==
Screen.currentResolution.height)
{
    // Set the current resolution index
    currentResolutionIndex = i;
}

resolutionDropdown.AddOptions(options); // Add the resolution options to the dropdown
resolutionDropdown.value = currentResolutionIndex; // Set the dropdown value to the current
resolution index
resolutionDropdown.RefreshShownValue(); // Refresh the displayed value of the dropdown
}

public void SetResolution(int resolutionIndex)
{
    Resolution resolution = resolutions[resolutionIndex];
    // Set the screen resolution using the selected resolution index
    Screen.SetResolution(resolution.width, resolution.height, Screen.fullScreen);
}

public void SetQuality(int qualityIndex)
{
    // Set the overall quality level of the game using the selected quality index
    QualitySettings.SetQualityLevel(qualityIndex);
}

public void SetFullscreen(bool isFullscreen)
{
    // Set the game to fullscreen or windowed mode based on the passed boolean value
    Screen.fullScreen = isFullscreen;
}
}
```