



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Εφαρμογής Ασφαλούς Οδήγησης για
Κινητές Συσκευές**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

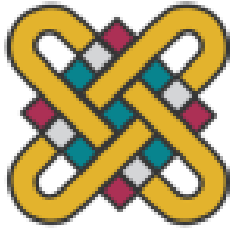
του

ΔΗΜΗΤΡΙΑΔΗΣ ΠΛΑΤΩΝ

(ΑΕΜ: 2368)

Επιβλέπων : **Νικόλαος Δημόκας**
Επίκουρος Καθηγητής

Καστοριά Σεπτέμβριος – 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Εφαρμογής Ασφαλούς Οδήγησης για Κινητές Συσκευές

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΔΗΜΗΤΡΙΑΔΗΣ ΠΛΑΤΩΝ

(ΑΕΜ: 2368)

Επιβλέπων : **Νικόλαος Δημόκας**
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την **ημερομηνία εξέτασης**

.....
Νικόλαος Δημόκας
Επίκουρος Καθηγητής

.....
Δημήτριος Βέργαδος
Επίκουρος Καθηγητής

.....
Σπυρίδων
Νικολάου
Λέκτορας

Καστοριά **Μήνας - Έτος** (παρουσίασης της εργασίας)

Copyright © 2021 – Dimitriadis Platon

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

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της πτυχιακής εργασίας, Δημόκα Νικόλαο, για όλη τη βοήθεια, τις συμβουλές, την υποστήριξη, την καθοδήγηση και την κατανόηση που έδειξε κατά τη διάρκεια συγγραφής της εργασίας.

Ευχαριστώ επίσης όλους του καθηγητές και εργαζόμενους του τμήματος Μηχανικών Πληροφορικής του Πανεπιστημίου Δυτικής Μακεδονίας για τις γνώσεις και τις υπηρεσίες που μου προσέφεραν κατά τη διάρκεια των σπουδών μου στο τμήμα.

Μεγάλο ευχαριστώ στους γονείς μου Δημητριάδη Γεώργιο (καθηγητής του τμήματος Ιατρικής – Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης) και Νίκη Τσιολάκη (καθηγήτρια Δευτεροβάθμιας Εκπαίδευσης) για τη βοήθεια, την υποστήριξη και τη σοφία που μου προσέφεραν όλο το διάστημα των σπουδών μου στην Καστοριά ως γονείς και εκπαιδευτικοί.

Ευχαριστώ πολύ τους κοντινούς μου φίλους Μέγα Φώτιο, Χατζημίση Κωσταντίνο και Χαλούδη Αθανάσιο για την βοήθειά τους σε πολλές πτυχές της ζωής μου κατά τη διάρκεια των σπουδών μου και κατά τη διάρκεια της ανάπτυξης της πτυχιακής εργασίας.

Τέλος ένα μεγάλο ευχαριστώ στον προϊστάμενό μου κατά τη διάρκεια της πρακτικής μου άσκησης, Λαζαρίδη Κωσταντίνο και τον διευθυντή μου Ιατρόπουλο Αθανάσιο για την κατανόηση που δείξαν όσο εργαζόμουν στην Εγνατία Οδό Α.Ε.

Περίληψη

Η ανάπτυξη ηλεκτρονικών παιχνιδιών είναι ένας κλάδος της πληροφορικής που απασχολεί εκατοντάδες χιλιάδες προγραμματιστές στον κόσμο. Η βιομηχανία ηλεκτρονικών παιχνιδιών υπολογίζεται πως σε μερικά χρόνια με τον ρυθμό που αναπτύσσεται θα ξεπεράσει τη μουσική βιομηχανία και θα γίνει η κορυφαία βιομηχανία διασκέδασης. Ο τρόπος ανάπτυξης των παιχνιδιών διαφέρει αρκετά από άλλες εφαρμογές. Παραδείγματος χάριν ο τρόπος που υλοποιούνται τα εμπόδια που θα αντιμετωπίσει ο παίκτης και ο βαθμός δυσκολίας που τα εμπόδια αυτά παρουσιάζουν μπορούν να προκαλέσουν συναισθήματα αλλά και να διδάξουν πολλά. Παρατηρήσαμε πως δεν υπάρχουν πολλά ηλεκτρονικά παιχνίδια που να διδάσκουν στον παίκτη την οδική ασφάλεια, πολύ σημαντική στις μέρες μας. Για αυτό το λόγο αποφασίσαμε να αναπτύξουμε μια εφαρμογή η οποία να εστιάζει ακριβώς στην εκπαίδευση του παίκτη στην οδική συμπεριφορά και στην εξοικείωση με τον κώδικα οδικής κυκλοφορίας. Ταυτόχρονα θα εξηγήσουμε τις τεχνικές που χρησιμοποιήθηκαν, θα κάνουμε σύγκριση των πιο διάσημων μηχανών ανάπτυξης ηλεκτρονικών παιχνιδιών και θα αναλύσουμε τη διαδικασία ανάπτυξης της εφαρμογής. Ένα εγχειρίδιο χρήσης συμπεριλαμβάνεται επίσης στην εργασία, με οδηγίες για τον τρόπο με τον οποίο παίζεται το παιχνίδι.

Λέξεις Κλειδιά: *gamification, κινητή εφαρμογή, ηλεκτρονικά παιχνίδια, ασφαλής οδήγηση, procedural content generation*

Abstract

Video games development is a part of computer science on which hundreds of thousands of programmers work all over the world. It is estimated that the video game industry at its rate of growth will surpass the music industry as the top entertainment industry. The way video games are developed is quite different than how other applications are. For example, the difficulty and the way that game mechanics are designed can invoke different emotions at the player but also educate them. After some research we found that there aren't many video games with the purpose of teaching the player on safe driving, something that is very important nowadays. Therefore, we decided to develop an application that teaches the player about safe driving. We will explain the techniques we used during development, we will compare two of the most famous game engines and we will thoroughly explain the process of developing our video game. There is a user manual also included, which explains how the game is played correctly.

Key Words: gamification, mobile application, video games, safe driving, procedural content generation

Πίνακας Περιεχομένων

Εισαγωγή.....	1
1. Μηχανές Παιχνιδιών.....	2
1.1 Unity 3D.....	2
1.2 Unreal Engine.....	4
1.3 Σύγκριση Unity 3D και Unreal Engine	8
1.4 Γιατί Επιλέξαμε Unity 3D.....	9
2. Βιβλιογραφική Ανασκόπηση	10
2.1 Διαδραστικό quiz e-didaskalia.blogspot.com	10
2.2 Driving in Car παιχνίδι για κινητά	11
2.3 Robok.o.k.....	12
2.4 Safety Driving Simulator: Car	13
3. Υλοποιημένα Σενάρια	14
3.1 Game Flow και Gamification	16
3.2 Το Σύστημα και τα Στοιχεία του	19
3.2.1 Κυρίως Μενού.....	20
3.2.2 Σκηνή Παιχνιδιού	21
3.2.3 Μηχανισμός Φαναριών.....	23
3.2.4 Απαγόρευση Πορείας Δεξιά, Αριστερά ή Ευθεία	26
3.2.5 Όριο Ταχύτητας.....	27
3.2.6 Πινακίδα STOP	27
3.2.7 Υποχρεωτική πορεία	28
3.3 Βελτίωση με το MapBox plug-in και Ανάλυση Δεδομένων Παίκτη	29
4. Εγχειρίδιο Χρήστη.....	30
Συμπεράσματα	34
Βιβλιογραφία.....	36
Παράρτημα Κώδικα	39

Λίστα Εικόνων

Εικόνα 1. Στιγμιότυπο από την πιο παλιά έκδοση του Unity.....	3
Εικόνα 2. Στιγμιότυπο από την πιο σύγχρονη έκδοση της Unreal Engine.....	5
Εικόνα 3. Επίδειξη του global illumination με άμεσο (αριστερά) και έμμεσο (δεξιά) φωτισμό από την Ellie Stone [17]	6
Εικόνα 4. Επίδειξη του real time global illumination χρησιμοποιώντας Lumen [18].....	7
Εικόνα 5. Εργαλείο παρακολούθησης της Unreal Engine [19]	8
Εικόνα 6. Στιγμιότυπο από την εφαρμογή που φαίνονται τα φανάρια	14
Εικόνα 7. Στιγμιότυπο από την εφαρμογή με απαγόρευση αριστερής στροφής	14
Εικόνα 8. Στιγμιότυπο από την εφαρμογή με όριο ταχύτητας	15
Εικόνα 9. Στιγμιότυπο με έργα δρόμο στην αριστερή λωρίδα	15
Εικόνα 10. Στιγμιότυπο με υποχρεωτική αριστερή στροφή	15
Εικόνα 11. Στιγμιότυπο με πινακίδα STOP.....	16
Εικόνα 12. Σύγκριση τροποποίησης συχνότητας εμφάνισης με διαφορετικούς συντελεστές.....	19
Εικόνα 13. UML Class Diagram για το κυρίως μενού.....	21
Εικόνα 14. UML Class Diagram για τη σκηνή του παιχνιδιού	22
Εικόνα 15. Τα φανάρια με εμφανίσιμο το mesh collider ως trigger	23
Εικόνα 16. Τα βασικά στοιχεία του UI κομματιού.....	24
Εικόνα 17. Τα βασικά στοιχεία των φαναριών	26
Εικόνα 18. Η πινακίδα STOP με το trigger collider εμφανίσιμο	28
Εικόνα 19. Σήμα που εμφανίζεται στη σκηνή με τα components του στα δεξιά	29
Εικόνα 20. Το κυρίως μενού της εφαρμογής	30
Εικόνα 21. Μενού τροποποίησης οχήματος	31
Εικόνα 22. Τα κουμπιά χειρισμού του οχήματος	31
Εικόνα 23. Τα σήματα που εμφανίζονται στις διασταυρώσεις	33

Λίστα Πινάκων

Εισαγωγή

Στα σύγχρονα οχήματα έχουν ενσωματώσει προηγμένα συστήματα υποβοήθησης οδηγού (ADAS) [1 – 3] δεδομένου ότι η πλειονότητα των ατυχημάτων οφείλεται σε ανθρώπινο λάθος. Η ADAS προσπαθεί να μειώσει τον αριθμό των ατυχημάτων με τη χρήση καμερών, ραντάρ, αλγορίθμων υπολογιστικής όρασης, ECU υψηλής απόδοσης, συστημάτων σύντηξης αισθητήρων, τεχνικές τεχνητής νοημοσύνης, αλγορίθμων βαθιάς μάθησης κ.λπ. Οι κύριες εφαρμογές των ADAS σχετίζονται με την αναγνώριση σημάτων κυκλοφορίας, την πέδηση έκτακτης ανάγκης, την ανίχνευση πεζών και την ανίχνευση αλλαγής λωρίδας κυκλοφορίας. Παρά τις τεχνολογικές βελτιώσεις και την αυτοματοποίηση που προσφέρουν τα ADAS και τις προσπάθειες που καταβάλλουν οι εθνικοί οργανισμοί για την ασφαλή οδήγηση και την ευαισθητοποίηση των οδηγών, περισσότεροι από ένα εκατομμύριο άνθρωποι σκοτώνονται και μέχρι πενήντα εκατομμύρια τραυματισμοί καταγράφονται κάθε χρόνο σύμφωνα με το [4]. Επιπλέον, οι προβλέψεις σχετικά με τα τροχαία ατυχήματα είναι απαισιόδοξες καθώς προβλέπουν αύξηση κατά 65% [5].

Ο καλύτερος τρόπος για τη μείωση του αριθμού των ατυχημάτων είναι η εκπαίδευση των οδηγών σε ασφαλείς συμπεριφορές οδήγησης. Το gamification έχει κερδίσει σημαντική προσοχή τα τελευταία χρόνια για εκπαιδευτικούς σκοπούς [6, 7]. Σύμφωνα με [8 – 10], το gamification είναι ο σχεδιασμός παιχνιδιών, δηλαδή η αξιοποίηση διαφόρων εννοιών παιχνιδιών, όπως η επίτευξη και η ανταμοιβή μέσω του παιχνιδιού, είτε για την υποστήριξη διαφορετικών συμπεριφορών και δραστηριοτήτων είτε για την εκπαίδευση του χρήστη. Ακολουθώντας αυτόν τον ορισμό, η προτεινόμενη εφαρμογή για κινητά προσπαθεί να ενθαρρύνει την καλή οδική συμπεριφορά, καθώς και να εκπαιδεύσει τον χρήστη σε σημαντικά σενάρια ασφαλούς οδήγησης [11], μέσω ενός παιχνιδιού. Τα υλοποιημένα σενάρια επικεντρώνονται στην εκπαίδευση των χρηστών για το πώς να ενεργούν στα φανάρια, να αποφεύγουν επικίνδυνες περιοχές (όπως εργοτάξια) στους δρόμους, να ακολουθούν τις πινακίδες κυκλοφορίας, να μην υπερβαίνουν τα όρια ταχύτητας, να μην παραβιάζουν την προτεραιότητα των πεζών στη διάβαση και πολλές άλλες περιπτώσεις που μπορεί να συναντήσουν οι χρήστες σε μια καθημερινή διαδρομή. Επιπλέον, η εφαρμογή για κινητά ενθαρρύνει τους χρήστες να κοινωνικοποιηθούν και να μοιραστούν τις επιδόσεις τους στα μέσα κοινωνικής δικτύωσης, δημοσιεύοντας τις επιδόσεις τους στον λογαριασμό τους στο Facebook.

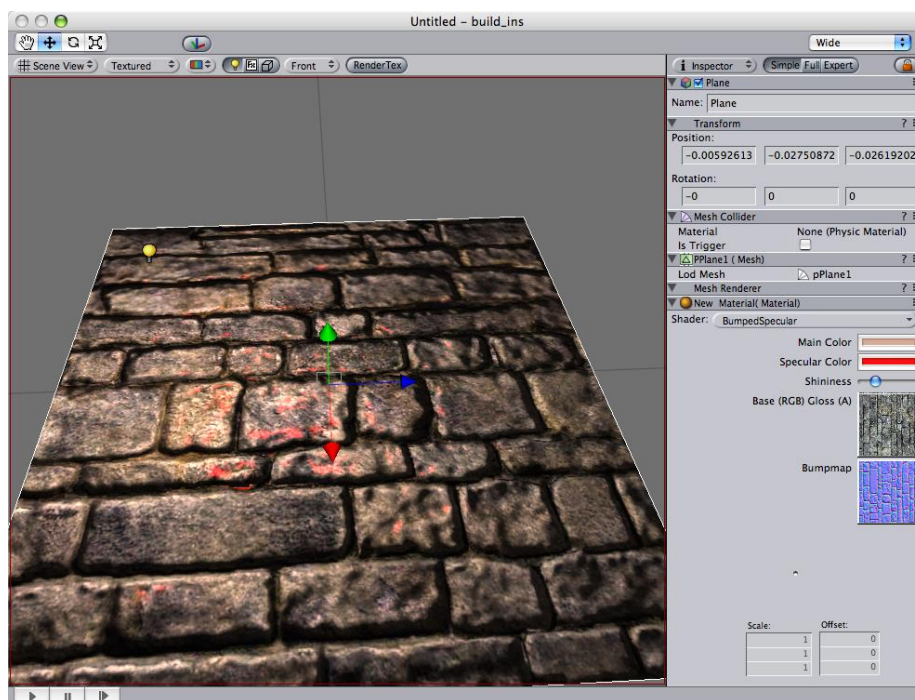
1. Μηχανές Παιχνιδιών

Οι μηχανές παιχνιδιών είναι σύνθετα εργαλεία που χρησιμοποιούνται για τη δημιουργία και ανάπτυξη ενός ηλεκτρονικού παιχνιδιού. Σήμερα, οι μηχανές παιχνιδιών αξιοποιούνται και σε άλλες χρήσεις πέρα από την ανάπτυξη ηλεκτρονικών παιχνιδιών. Χρησιμοποιούνται για τη δημιουργία εκπαιδευτικών διαδραστικών προγραμμάτων και επιστημονικών προσομοιώσεων. Πρόσφατα χρησιμοποιούνται ακόμη και στην τηλεοπτική ψυχαγωγία για τη δημιουργία διαδραστικών και υπερρεαλιστικών οπτικών εφέ. Πλήθος επαγγελματιών τα χρησιμοποιούν για το στάδιο ανάπτυξης αυτών των εφαρμογών, αλλά και προγραμματιστές λογισμικού, μηχανικοί μέχρι και επαγγελματίες σχεδιαστές και καλλιτέχνες. Οι μηχανές παιχνιδιών είναι, συχνά, ισχυρά εργαλεία που ζωντανεύουν ολόκληρους κόσμους και απαιτούν μεγάλη επεξεργαστική ισχύ για να παράγουν ευχάριστα αποτελέσματα. Αυτό οφείλεται στο γεγονός ότι οι μηχανές παιχνιδιών δεν αναλαμβάνουν μόνο την απόδοση της δημιουργίας 2D είτε 3D γραφικών, αλλά χειρίζονται σχεδόν τα πάντα που έχουν να κάνουν με την εφαρμογή. Η είσοδος, τα γραφικά, η προσομοίωση της φυσικών φαινομένων, ο ήχος, η τεχνητή νοημοσύνη και η δικτύωση είναι μερικές από τις κατηγορίες που οι μηχανές παιχνιδιών πρέπει να μπορούν να χειριστούν ταυτόχρονα.

Δύο από τις πιο συχνά χρησιμοποιούμενες μηχανές παιχνιδιών, τόσο εμπορικά, όσο και από ανεξάρτητους προγραμματιστές και φοιτητές, είναι η Unity 3D και η Unreal Engine (που πρόσφατα έκανε διαθέσιμη την 5η έκδοσή της). Και οι δύο αυτές μηχανές εστιάζουν σε διαφορετικά πεδία και κάθε προγραμματιστής και σχεδιαστής έχει τις προτιμήσεις του και ο καθένας για τους λόγους του, καθώς έχουν γίνει συγκρίσεις πολλές φορές μεταξύ τους [12] [13].

1.1 Unity 3D

Το Unity 3D κυκλοφόρησε για πρώτη φορά το 2005. Υποστήριζε μια ποικιλία από πλατφόρμες και ο κατάλογος συνεχίζει να μεγαλώνει μέχρι και σήμερα. Ορισμένες πλατφόρμες και λειτουργικά συστήματα που υποστηρίζει το Unity 3D είναι τα Windows, MacOS, Linux, κινητά, τηλεοράσεις κ.λπ. Το 2013 το Unity κυκλοφόρησε την ενημέρωση 4.3, η οποία έδωσε στους προγραμματιστές τα απαραίτητα εργαλεία για να δημιουργούν και να υποστηρίζουν και εφαρμογές 2D. Η τρέχουσα καθαρή αξία της Unity (Απρίλιος 2022) είναι 26,62B USD και είναι πιθανό να συνεχίσει να αυξάνεται.



Εικόνα 1. Στιγμιότυπο από την πιο παλιά έκδοση του Unity

Ένα χαρακτηριστικό που πρέπει να διαθέτουν οι μηχανές παιχνιδιών, είναι η εξαιρετική απόδοση γραφικών. Η Unity 3D δεν είχε πάντα το απαραίτητο λογισμικό για να παράγει υπερρεαλιστικά και εντυπωσιακά αποτελέσματα, αλλά από το 2019 και μετά κυκλοφόρησαν ενημερώσεις που επικεντρώθηκαν πολύ στην παραγωγή γραφικών. Συγκεκριμένα, κυκλοφόρησαν το High-Definition Rendering Pipeline (HDRP), ένα scriptable render pipeline (SRP) που εστιάζει στην απόδοση γραφικών υψηλών προδιαγραφών και δίνει στους προγραμματιστές τα εργαλεία που χρειάζονται για να το πετύχουν αυτό, όπως έχουν γράψει στη δημοσίευσή τους για την ενημέρωση [14]. Μερικά από αυτά τα εργαλεία περιλαμβάνουν τη δυνατότητα των προγραμματιστών να δημιουργούν τα δικά τους εφέ post-processing για τα έργα τους, τη δυνατότητα εισαγωγής C# και shaders με το Custom Render Pass για τη συλλογή και το χειρισμό διαφόρων πληροφοριών, το Look-Dev που επιτρέπει στους προγραμματιστές να συγκρίνουν assets σε διαφορετικές ρυθμίσεις φωτισμού και τις ρυθμίσεις κλιμάκωσης που δίνουν στους προγραμματιστές τη δυνατότητα να δημιουργούν HDRP assets, το καθένα με διαφορετικές ρυθμίσεις ποιότητας, ώστε να δίνουν στο χρήστη τη δυνατότητα να επιλέξει την ποιότητα γραφικών που του ταιριάζει καλύτερα. Μεταξύ άλλων ενημερώσεων έχουν επίσης υλοποιήσει υποστήριξη ray-tracing και shader graphs που κάνουν ποιοτικές αναβαθμίσεις στο ήδη εξαιρετικό σύστημα του Unity.

Η υποστήριξη της Unity προς τους προγραμματιστές είναι επίσης κάτι για το οποίο η μηχανή είναι πολύ γνωστή. Το asset store της Unity διαθέτει μεγάλη ποικιλία προϊόντων, τόσο δωρεάν όσο και σε προσιτές τιμές, τα οποία κυμαίνονται από 3D models μέχρι ολόκληρα νέα προσαρμοσμένα features και προγράμματα υποστήριξης που δημιουργήθηκαν από ανεξάρτητους προγραμματιστές. Έτσι, με το asset store της Unity είναι αρκετά εύκολο για τον καθένα να βρει έτοιμο περιεχόμενο και να αρχίσει να αναπτύσσει κάτι δικό του, γεγονός που ενισχύει την προτίμηση της Unity προς τους ανεξάρτητους προγραμματιστές. Προσφέρουν επίσης εξαιρετικό εκπαιδευτικό περιεχόμενο που βοηθά τους αρχάριους να μάθουν περισσότερα

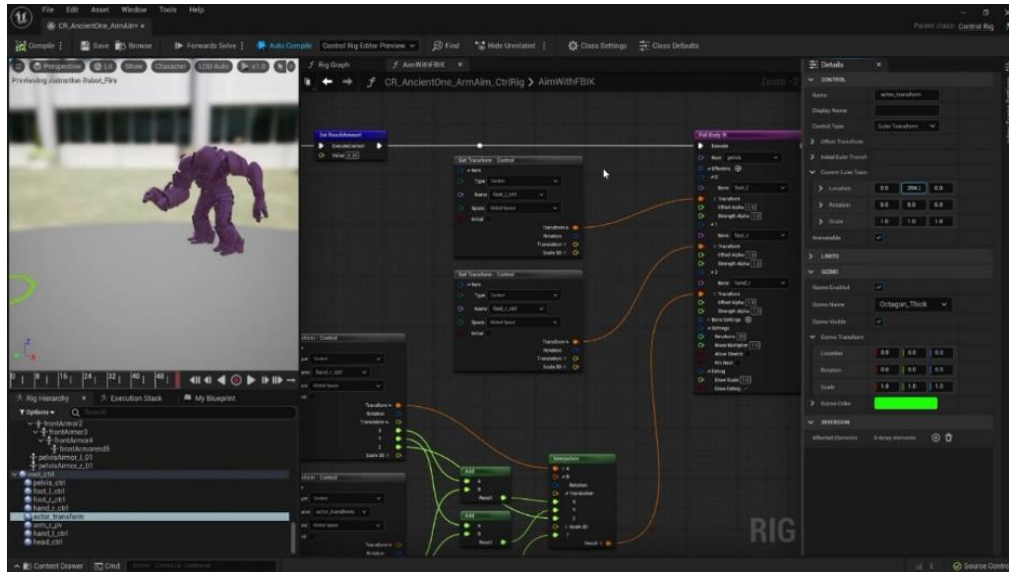
για τα χαρακτηριστικά και τη λειτουργικότητα της μηχανής τους. Για αυτούς τους λόγους, δεν είναι περίεργο που η Unity είναι μια μηχανή παιχνιδιών που κυριαρχεί σε συγκεκριμένες αγορές όπου η δημοσίευση ενός βιντεοπαιχνιδιού είναι απλή, όπως για παράδειγμα οι αγορές εφαρμογών για κινητά.

Ένας από τους λόγους για τους οποίους οι νέοι προγραμματιστές προτιμούν επίσης το Unity 3D είναι η scripting γλώσσα της επιλογής του, η οποία είναι η C#. Αν και η Unity 3D υποστηρίζει άλλες scripting γλώσσες, όπως η Java και η C++, φαίνεται ότι η C# είναι η καταλληλότερη για τις περισσότερες καταστάσεις και σενάρια για αυτή τη μηχανή παιχνιδιών. Η C# είναι μια εύκολη στην εκμάθηση και χρήση γλώσσα προγραμματισμού που είναι κατάλληλη για αρχάριους. Το documentation της Unity σχετικά με τον τρόπο σωστής χρήσης όλων των βιβλιοθηκών για την γραφή κώδικα είναι καλά γραμμένη και διαθέσιμη για όλους δωρεάν στο διαδίκτυο.

Ένα άλλο σπουδαίο χαρακτηριστικό που προσφέρει η Unity είναι τα εργαλεία ανάπτυξης για εφαρμογές VR και AR. Η VR και η AR είναι τεχνολογίες που έχουν γνωρίσει ιδιαίτερη ανάπτυξη τα τελευταία χρόνια, η VR ιδιαίτερα, καθώς όσο αναπτύσσεται η τεχνολογία, τόσο πιο φθηνή και πιο προσιτή γίνεται στο κοινό. Αυτό ανοίγει νέες ευκαιρίες για τις εταιρείες είτε να αναπτύξουν προϊόντα για τους καταναλωτές, είτε για εταιρείες όπως η Unity να παρέχουν στους προγραμματιστές τα εργαλεία για τη δημιουργία των εν λόγω προϊόντων. Σύμφωνα με το επίσημο εγχειρίδιο της Unity [15], στο κεφάλαιο Virtual Reality στην επισκόπηση VR, οι συσκευές VR που υποστηρίζονται από το Unity 3D είναι οι Gear VR, Oculus CV1 και Vive. Επίσης, σύμφωνα με το εγχειρίδιο στο ίδιο κεφάλαιο, στο υποκεφάλαιο συσκευές VR, το Unity υποστηρίζει επίσης την πλατφόρμα του Steam για VR, OpenVR, η οποία είναι πολύ επωφελής για το Unity, καθώς το Steam είναι ο κορυφαίος διανομέας βιντεοπαιχνιδιών στην αγορά. Εκτός όμως από τα βιντεοπαιχνίδια, το VR και το AR είναι τεχνολογίες που μπορούν να διαμορφώσουν το μέλλον της εκπαίδευσης και των κοινωνικών μέσων και η Unity έχοντας εύκολα προσβάσιμα εργαλεία όχι μόνο για μεγάλες εταιρείες, αλλά και για ανεξάρτητους προγραμματιστές, είναι ένα τεράστιο βήμα προς την πρόοδο και την τελειοποίηση αυτών των τεχνολογιών.

1.2 Unreal Engine

Η Unreal Engine ξεκίνησε να αναπτύσσεται το 1995 από τον Tim Sweeney, ο οποίος αργότερα ίδρυσε την εταιρεία Epic Games. Η μηχανή κυκλοφόρησε το 1998 ταυτόχρονα με την κυκλοφορία ενός ηλεκτρονικού παιχνιδιού που είχε αναπτυχθεί με αυτή τη μηχανή και ονομαζόταν Unreal, από το οποίο η μηχανή θα έπαιρνε αργότερα το όνομά της. Με την πάροδο των ετών η Unreal Engine θα κυκλοφορούσε άλλες 4 εκδόσεις, με την τελευταία να είναι η Unreal Engine 5 που ανακοινώθηκε στις 13 Μαΐου 2020 και κυκλοφόρησε εμπορικά στις 5 Απριλίου 2022. Η 3η και η 4η έκδοση της Unreal Engine θα εκτοξεύσουν τη δημοτικότητά τους σε μερικά από τα πιο κερδοφόρα στούντιο και εκδότες ηλεκτρονικών παιχνιδιών μεταξύ 2005 και 2020. Μεγάλες εκδοτικές εταιρείες όπως η Microsoft, η Sony, η Ubisoft, η Activision και η Square Enix θα αναλάμβαναν την πρόκληση να κάνουν πολλά έργα χρησιμοποιώντας την Unreal Engine ως βάση για τη δουλειά τους.



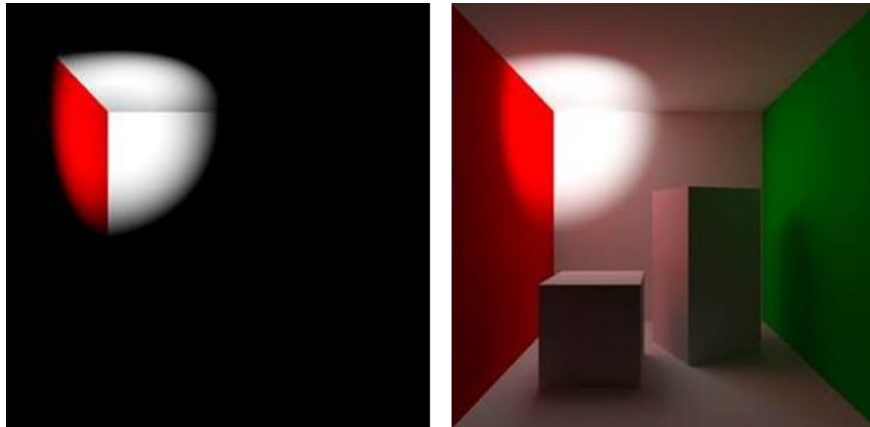
Εικόνα 2. Στιγμιότυπο από την πιο σύγχρονη έκδοση της Unreal Engine

Ο λόγος για τον οποίο οι περισσότερες από αυτές τις εταιρείες χρησιμοποιούν την Unreal Engine, είναι η εξαιρετική παραγωγή γραφικών της. Από τότε που η Unreal Engine έγινε εμπορικά διαθέσιμη, οι προγραμματιστές της συνέχισαν να τη βελτιώνουν. Κάθε αναβάθμιση αυτής της μηχανής έφερε κάτι καινοτόμο. Η Unreal Engine 2 κυκλοφόρησε το 2002 και έφερε χαρακτηριστικά όπως κινηματογραφικά εργαλεία επεξεργασίας, νέα συστήματα σωματιδίων, plug-in εξαγωγής για λογισμικό 3D μοντελοποίησης και νέο σύστημα animation βασισμένο σε σκελετό. Η Unreal Engine 3 που κυκλοφόρησε 2 χρόνια αργότερα θα βελτίωνε τα γραφικά αυξάνοντας την ποιότητα των γραφικών με μεθόδους όπως υπολογισμοί σκιών ανά pixel αντί για χρήση κορυφών, προσομοιώσεις μαλακών σωμάτων, ρυθμιζόμενα καταστρέψιμα στοιχεία περιβάλλοντος και προσομοιώσεις πλήθους. Επίσης, για πρώτη φορά από την αρχική της κυκλοφορία της το 1998, η Unreal Engine θα γινόταν διαθέσιμη στο iOS. Η 4η έκδοση της μηχανής κυκλοφόρησε στις 19 Μαρτίου 2014 και έφερε ακόμη περισσότερες καινοτόμες βελτιώσεις όπως το global illumination σε πραγματικό χρόνο, μια τεχνική που χρησιμοποιείται για την καλύτερη προσομοίωση της συμπεριφοράς του φωτός με τις επακόλουθες αναπηδήσεις του φωτός από τη μια επιφάνεια στην άλλη, η οποία με sub-surface scattering δημιουργεί ρεαλιστικό φωτισμό. Με την 5η έκδοσή της, η Unreal Engine 5 διευρύνει ακόμη περισσότερο τα όρια της απόδοσης γραφικών. Ένα επαναστατικό χαρακτηριστικό της είναι το Nanite, το οποίο επιτρέπει στους καλλιτέχνες να εισάγουν πολύ λεπτομερή meshes στα έργα τους. Μέσω της φωτογραμμετρίας, οι καλλιτέχνες μπορούν να χρησιμοποιούν τρισδιάστατα μοντέλα με ρεαλιστικές υφές και δισεκατομμύρια πολύγωνα με μεγάλη αποτελεσματικότητα. Αυτό το χαρακτηριστικό μπορεί να αυξήσει ή να μειώσει το επίπεδο λεπτομέρειας του μοντέλου ανάλογα με την απόσταση σχεδίασης από την κυρίως κάμερα, γεγονός που επιτρέπει στους υπολογιστές να χειρίζονται υπερλεπτομερή αντικείμενα με μεγάλη απόδοση. Με αυτό και τις μονάδες αποθήκευσης υψηλής ταχύτητας που είναι πλέον ευρέως διαθέσιμες, η Unreal Engine μπορεί να δημιουργήσει εκπληκτικά περιβάλλοντα που χρησιμοποιούνται ακόμη και σε εφέ που δημιουργούνται από υπολογιστή (CGI – Computer Generated Imagery) για σειρές και ταινίες.

Ο κατάλογος των δυνατοτήτων και των χαρακτηριστικών συνεχίζεται. Με την πάροδο των ετών η Unreal Engine έχει διαθέσει στους προγραμματιστές μια μεγάλη ποικιλία

χαρακτηριστικών. Ο ιστότοπος της Unreal [16] διαθέτει έναν κατάλογο με όλα τα features που είναι διαθέσιμα στην μηχανή. Μερικά από τα πιο αξιοσημείωτα χαρακτηριστικά της Unreal Engine είναι τα Lumen, Niagara Particle System, Open Source Code και Profiling and Performance.

Το Lumen είναι το πλήρως δυναμικό σύστημα global illumination της Unreal Engines. Το global illumination είναι η τεχνική κατά την οποία η μηχανή παιχνιδιών προσομοιώνει τον τρόπο με τον οποίο λειτουργεί το φως στην πραγματική ζωή.



Εικόνα 3. Επίδειξη του global illumination με άμεσο (αριστερά) και έμμεσο (δεξιά) φωτισμό από την Ellie Stone [17]

Ένα αντικείμενο μπορεί να φωτίζεται άμεσα από μια πηγή φωτός ή έμμεσα με το φως να ανακλάται από ένα άλλο αντικείμενο και να προσπίπτει έμμεσα σε αυτό το αντικείμενο. Αυτό που κάνουν οι περισσότερες μηχανές τρισδιάστατων παιχνιδιών είναι ότι, αφού γίνει η ενσάρκωση μιας πηγής φωτός, υπολογίζουν τη διαδρομή των αναπηδήσεων που κάνει το φως σε μια σκηνή και ανάλογα το πώς εκτρέπεται το φως. Η μεγαλύτερη διαφορά μεταξύ των περισσότερων μηχανών παιχνιδιών και του συστήματος global illumination Lumen της Unreal είναι ότι το Lumen είναι δυναμικό και λειτουργεί κατά την εκτέλεση ή, με άλλα λόγια, δεν χρειάζεται κάνει “bake” το φως στη σκηνή αλλά υπολογίζει εκατομμύρια αναπηδήσεις ενώ η προσομοίωση εκτελείται. Αυτό καταλήγει στο ότι η Unreal Engine είναι σε θέση να αποδίδει φωτισμένες σκηνές υψηλής ποιότητας πολύ αποτελεσματικά και το γεγονός ότι το επιτυγχάνει αυτό κατά τη διάρκεια εκτέλεσης καθιστά το Lumen μια απίστευτη εφεύρεση.



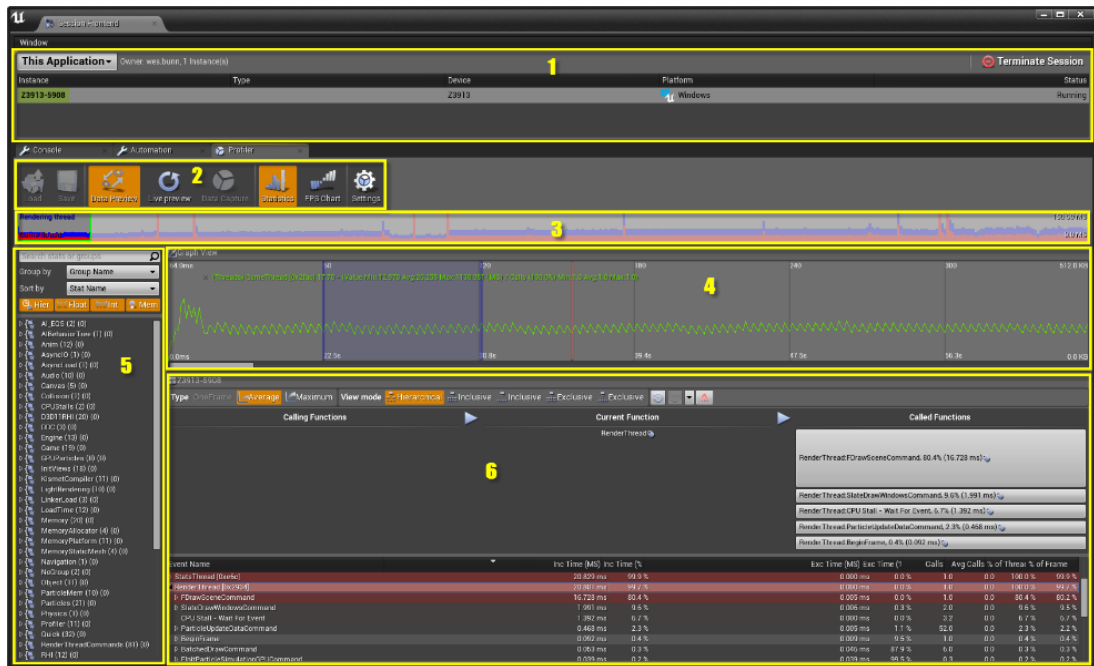
Εικόνα 4. Επίδειξη του real time global illumination χρησιμοποιώντας Lumen [18]

Το Niagara Particle System είναι ένα προηγμένο σύστημα σωματιδίων που μπορεί να χρησιμοποιηθεί για την απόδοση διαφόρων εικονικών εφέ. Έχει επίσης δημιουργηθεί με τέτοιο τρόπο ώστε οι καλλιτέχνες να μπορούν να χειριστούν αυτά τα VFX (Virtual Effects) χωρίς τη βοήθεια προγραμματιστών. Τα τέσσερα βασικά συστατικά του Niagara είναι τα systems, οι emitters, τα modules και οι parameters. Αυτά τα βασικά συστατικά συνδυάζονται σε ένα ιεραρχικό σχήμα όπου το ένα περιέχει το άλλο. Οι parameters είναι αφηρημένα δεδομένα που μπορούν να χρησιμοποιηθούν για τη δημιουργία των βασικών εφέ στο τελικό αποτέλεσμα. Περιέχονται μέσα σε modules που ενθυλακώνουν συμπεριφορές και μπορούν να τροποποιηθούν είτε με αυτοδημιούργητες συναρτήσεις, χρησιμοποιώντας High-Level Shading Language (HLSL) είτε δημιουργούνται με έναν Graph Editor. Τα modules περιέχονται σε emitters που μπορούν να αποδώσουν τα VFX με πολλαπλούς τρόπους. Τέλος, οι emitters περιέχονται μέσα σε ένα system που συνδυάζει τα πάντα στο τελικό αποτέλεσμα.

Η Unreal είναι επίσης ανοικτού κώδικα, γεγονός που δίνει στους προγραμματιστές πλήρη έλεγχο του τι μπορεί και τι δεν μπορεί να κάνει η μηχανή. Με αυτόν τον τρόπο, τόσο οι μεγάλες εταιρείες όσο και οι ανεξάρτητοι δημιουργοί μπορούν να αξιοποιήσουν τη μηχανή στο μέγιστο των δυνατοτήτων της, δημιουργώντας νέους τρόπους είτε για αλληλεπιδράσεις φυσικής, όπως για παράδειγμα πως η βαρύτητα μπορεί να επηρεάσει τα αντικείμενα μέσα σε ένα περιβάλλον, είτε ακόμη και να αλλάξουν τον τρόπο με τον οποίο η μηχανή χειρίζεται την απόδοση για να δημιουργήσει στυλιζαρισμένες εικόνες. Όποιος και αν είναι ο λόγος, το γεγονός ότι η μηχανή είναι ανοικτού κώδικα είναι πολύ σημαντικό χαρακτηριστικό, ειδικά για παραγωγές υψηλού προϋπολογισμού και γνωστούς εκδότες και στούντιο.

Επίσης, η Unreal Engine προσφέρει εξαιρετικά εργαλεία για την παρακολούθηση της απόδοσης ενός project όσον αφορά τη διαχείριση των πόρων. Όλα τα εργαλεία του Unreal είναι σπουδαία, αλλά μερικές φορές μπορεί να είναι πολύ επιβαρυντικά για ένα μηχάνημα. Έτσι, προσφέρουν μια εξαιρετική διεπαφή προφίλ και επιδόσεων, με την οποία οι προγραμματιστές

και οι καλλιτέχνες μπορούν να δουν ποια assets καταλαμβάνουν τους περισσότερους πόρους κατά τη διάρκεια της εκτέλεσης.



Εικόνα 5. Εργαλείο παρακολούθησης της Unreal Engine [19]

1.3 Σύγκριση Unity 3D και Unreal Engine

Και οι δύο μηχανές παιχνιδιών μπορούν να έχουν πολλές λειτουργίες και τεχνικές με τις οποίες εφαρμόζουν αυτές τις λειτουργίες. Και οι δύο όμως έχουν πλεονεκτήματα και μειονεκτήματά. Ως εκ τούτου, θα συγκρίνουμε μερικές από τις δυνατότητες αυτών των μηχανών και θα εξετάσουμε γιατί και πώς τα κάνουν αυτά καλύτερα από την άλλη.

Το πρώτο που γίνεται πολύ εμφανές όταν συγκρίνουμε αυτές τις μηχανές μεταξύ τους είναι η γραφική ποιότητα. Ενώ και οι δύο μηχανές έχουν σημαντικές γραφικές ικανότητες, οι καινοτόμες τεχνικές και οι πρόσφατες ενημερώσεις της Unreal Engines μπορούν να δημιουργήσουν εκπληκτικά αποτελέσματα. Το global illumination σε πραγματικό χρόνο και τα υπερλεπτομερή meshes εξακολουθούν να είναι στοιχεία που το Unity 3D δεν μπορεί να χειριστεί τόσο καλά. Το HDRP της Unity 3D μπορεί να παράγει πολύ παρόμοια αποτελέσματα με αυτά που μπορεί να κάνει η Unreal Engine 5, αλλά οι επιδόσεις της Unreal Engine είναι πολύ μεγαλύτερες.

Στη συνέχεια, θα πρέπει να συγκρίνουμε την ευκολία υλοποίησης κώδικα που έχει κάθε πλατφόρμα. Και οι δύο μηχανές έχουν και πάλι τα μειονεκτήματα και πλεονεκτήματα στην επιλογή των βασικών γλωσσών προγραμματισμού και τον τρόπο με τον οποίο χειρίζονται τον πηγαίο τους κώδικα. Η Unity 3D, όπως αναφέρθηκε παραπάνω, μπορεί να αξιοποιηθεί περισσότερο με τη χρήση της C#. Αυτή είναι μια εξαιρετική επιλογή, επειδή η C# είναι μια γλώσσα φιλική προς τους αρχάριους και το αντικειμενοστραφές μοντέλο της είναι πολύ πιο κατάλληλο για την ανάπτυξη παιχνιδιών. Από την άλλη πλευρά, η Unity δεν προσφέρει μεγάλη ελευθερία όσον αφορά τον πηγαίο κώδικά της. Αυτό είναι κάτι που προσφέρει η Unreal Engine. Αυτό δίνει στους προγραμματιστές μεγαλύτερη ελευθερία να τροποποιήσουν τη μηχανή σύμφωνα με τις ανάγκες τους. Ως εκ τούτου, η Unreal Engine προτιμάται περισσότερο από μεγάλες παραγωγές

και η Unity 3D με το πιο σταθερό της σύστημα προτιμάται από νέους και ανεξάρτητους προγραμματιστές.

Επίσης, το animation είναι κάτι που απασχολεί τους περισσότερους σχεδιαστές. Ο χειρισμός των animation στο Unity 3D, παρόλο που το Unity διαθέτει κάποια animation tools για key framing και τις basic interpolation, είναι πιο δύσκολο από ό,τι στο Unreal Engine. Η Unreal προσφέρει σπουδαία animation tools και animation adaptability, όπως η δυνατότητα χρήσης δεδομένων animation σε διαφορετικούς σκελετούς animation, κάτι που και πάλι το Unity 3D δεν είναι απόλυτα ικανό να κάνει.

Η τιμολόγηση του Unity 3D βοηθάει τους ανεξάρτητους και νέους προγραμματιστές. Σύμφωνα με την Evelyn Trainor-Fogleman στο άρθρο του Unity vs Unreal Engine: Game engine comparison guide for 2021 [12], το Unity 3D έχει ένα πρόγραμμα συνδρομής 75\$ το μήνα, ενώ η Unreal Engine δικαιούται το 5% των κερδών του παιχνιδιού σε δικαιώματα, κάτι που δεν θα ωφελούσε απαραίτητα τους νέους προγραμματιστές ή τα νέα στούντιο.

Τέλος, η Unity 3D έχει μεγαλύτερη γενική υποστήριξη για τη μηχανή. Το κατάστημα assets της Unity διαθέτει μεγάλη ποικιλία από μοντέλα, user interface στοιχεία και εργαλεία για την σχεδίαση και υλοποίηση gameplay και κώδικα που μπορεί κανείς είτε να αγοράσει είτε να πραγματοποιήσει κανείς. Επίσης, προσφέρουν μεγάλη υποστήριξη όταν πρόκειται για τεχνικά ζητήματα. Όχι μόνο αυτό, αλλά υπάρχει και τεράστια κοινότητα υποστήριξης για το Unity 3D. Υπάρχουν σπουδαία μαθήματα και επί πληρωμή και δωρεάν που διδάσκουν το Unity 3D και την ανάπτυξη παιχνιδιών σε ανθρώπους σε όλο τον κόσμο.

1.4 Γιατί Επιλέξαμε Unity 3D

Η μηχανή Unity 3D είναι μια από της πιο διαδεδομένες στον κόσμο όπως αναφέρθηκε και σε προηγούμενα υποκεφάλαια. Επίσης χρησιμοποιείται πολύ από προγραμματιστές και δημιουργούς οι οποίοι ξεκινούν να ασχολούνται με τη δημιουργία ηλεκτρονικών παιχνιδιών, χαρακτηριστικό το οποίο ευνόησε την υλοποίηση της εφαρμογής μας. Επίσης, όπως προαναφέρθηκε, η εταιρία Unity έχει πολύ καλό documentation για όλες τις μεθόδους και τα αντικείμενα που χρησιμοποιούνται για την ανάπτυξη κώδικα. Δεδομένων όλων των προηγούμενων και την εξαιρετική υποστήριξη που υπάρχει και από άλλους indie δημιουργούς (που δουλεύουν μόνοι τους ή με μικρή ομάδα χωρίς υποστήριξη και χρηματοδότηση από εκδότες) προς την κοινότητα του Unity, θεωρήσαμε πως ήταν η καλύτερη επιλογή.

2. Βιβλιογραφική Ανασκόπηση

Το παρόν κεφάλαιο μελετά και συγκρίνει εφαρμογές που βρέθηκαν διαδικτυακά, οι οποίες έχουν στόχο σε κάποιο βαθμό την άσκηση, εκπαίδευση σε θέματα οδικής ασφάλειας. Οι τύποι εφαρμογών που κρίθηκαν είναι εφαρμογές διαδικτύου, εφαρμογές desktop και εφαρμογές mobile. Οι δοκιμές και συγκρίσεις πραγματοποιήθηκαν με τη χρήση του υπολογιστή HP omen 17-cbxxx και του smart phone Huawei P20 Pro.

Πληροφορίες συστήματος ηλεκτρονικού υπολογιστή:

Επεξεργαστής	Intel® Core™ i7-9750H CPU @ 2.60GHz 2.59 GHz
Μνήμη RAM	16.0 GB (εκ των οποίων 15.9GB είναι χρησιμοποιήσιμη)
Κάρτα γραφικών	NVIDIA® GeForce® GTX 1660 Ti (αποκλειστική μνήμη GDDR6 6 GB)
Τύπος συστήματος	64-bit operating system, x64-based processor
Λειτουργικό σύστημα	Windows 10
Σκληρός δίσκος	256 GB PCIe® NVMe™ M.2 SSD

Πληροφορίες συστήματος smart phone:

Επεξεργαστής	Octa-core Cortex A-73 (4x2.4GHz)
Μνήμη RAM	6GB
Κάρτα γραφικών	Mali-G72 MP12
Λειτουργικό σύστημα	Android 10

2.1 Διαδραστικό quiz e-didaskalia.blogspot.com

Στην ιστοσελίδα e-didaskalia.blogspot.com υπάρχει ένα διαδραστικό quiz [20] το οποίο έχει σκοπό να εξετάσει πόσο καλά γνωρίζει ο ερωτώμενος τα σήματα οδικής κυκλοφορίας. Δεν υπάρχουν πολλές πληροφορίες για τη δημιουργία του ερωτηματολογίου εκτός από την χρονιά που αναπτύχθηκε, δηλαδή 2017. Το quiz αποτελείται από 20 ερωτήσεις πολλαπλής επιλογής. Οι ερωτήσεις περιέχουν μια εικόνα με το σήμα που πρέπει να αναγνωρίσει ο ερωτώμενος καθώς και 3 με 4 απαντήσεις από τις οποίες πρέπει να επιλέξει μία. Τη στιγμή που επιλέγεται μία απάντηση ο ερωτώμενος ενημερώνεται απευθείας για το αν αυτό που διάλεξε είναι σωστό ή όχι. Ταυτόχρονα υπάρχει και ένας μετρητής στο κάτω μέρος του quiz ο οποίος ενημερώνει τον ερωτώμενο για το πόσες ακόμα ερωτήσεις πρέπει να απαντηθούν. Αφού απαντηθούν όλες οι ερωτήσεις στη θέση που ήταν ο μετρητής θα εμφανιστεί το αποτέλεσμα του quiz ενημερώνοντας τον χρήστη για το πόσες ερωτήσεις απάντησε σωστά από τις 20.

Το quiz προσφέρει μια μικρή γνώση πάνω σε κάποια σήματα που αφορούν κυρίως οδηγούς. Υπάρχει επίσης η δυνατότητα στο χρήστη να κάνει το τεστ από την αρχή αφού έχει δει τα λάθη του. Σε αυτό το βαθμό η εφαρμογή καταφέρνει να διδάξει μέχρι ένα σημείο τον χρήστη. Ωστόσο οι ερωτήσεις δεν αλλάζουν σειρά, γεγονός το οποίο θα προσέφερε ένα βαθμό δυσκολίας και επιπλέον είναι πάντα οι ίδιες 20. Ένας τρόπος για να βελτιωθεί η λειτουργικότητα του quiz θα ήταν να αντλεί ερωτήσεις από μία μικρή βάση δεδομένων με περισσότερες ερωτήσεις. Το quiz είναι όμως καλό και μπορεί να εξετάσει τις γνώσεις ενός οδηγού πάνω στα σήματα του κώδικα οδικής κυκλοφορίας

2.2 Driving in Car παιχνίδι για κινητά

Η εφαρμογή Driving in Car [21] είναι ένα 3D παιχνίδι οδήγησης για κινητά Android. Η εφαρμογή αναπτύχθηκε το 2016 από την εταιρία Doodle Mobile Ltd. Η εφαρμογή είναι μικρή σε μέγεθος, μόλις 20MB, έχει πάνω από 5.000.000 downloads στο Google Play και το τελευταίο update στην έκδοση 1.9 έγινε στις 15 Μαΐου 2019.

Στο Drive in Car ο παίκτης καλείται να φέρει εις πέρας διάφορες αποστολές όπως για παράδειγμα να παραδώσει πολλαπλά πακέτα σε διάφορα σημεία της πόλης όπου οδηγεί γρήγορα ή να δώσει υπηρεσίες μεταφοράς για περαστικούς ανθρώπους. Όταν ο παίκτης φέρνει εις πέρας την αποστολή του παίρνει έξτρα πόντους όταν η οδήγησή του είναι προσεκτική. Αυτό κυρίως μέσα στην εφαρμογή σημαίνει πως δεν πρέπει ο παίκτης να οδηγεί σε αντίστροφη λωρίδα, δεν πρέπει να χτυπάει πάνω σε περαστικά οχήματα και δεν πρέπει να πέφτει πάνω σε πεζούς. Με την επιτυχία κάθε αποστολής ο παίκτης επιβραβεύεται με νομίσματα τα οποία μπορεί να ξοδέψει για την αγορά νέων οχημάτων και νέων χρωμάτων για κάθε όχημα ξεχωριστά. Υπάρχει και η δυνατότητα αγοράς νέων οχημάτων μέσω μικροσυναλλαγών (microtransactions).

Η εφαρμογή εφαρμόζει αρκετά λειτουργικό χειρισμό. Το όχημα του παίκτη κινείται με μια σταθερή επιτάχυνση όσο ο παίκτης πατάει πεπερασμένα το κουμπί για το γκάζι το οχήματος, μέχρι να φτάσει την μέγιστη ταχύτητα το όχημα, και τα κουμπιά ανταποκρίνονται καλά στο χειρισμό που τους αναλογεί.

Ωστόσο τα γραφικά του παιχνιδιού δεν είναι αρκετά καλά. Πιο αναλυτικά το πρόβλημα είναι πως τα textures των μοντέλων δεν έχουν αρκετά μεγάλη ανάλυση. Επίσης δεν βοηθάει το γεγονός πως πάρα πολλά μοντέλα στο παιχνίδι έχουν χαμηλής ποιότητας meshes, δηλαδή τα μοντέλα δεν αποτελούνται από αρκετά πολύγωνα έτσι ώστε να φαίνονται λεπτομέρειες σε μικρές κυρτότητες που έχουν κάποια αμάξια ή κάποιοι NPCs (non-playable characters) μέσα στην εφαρμογή. Αυτό φαίνεται επίσης και στα κτήρια όπου τα meshes αποτελούνται από πολύ λίγα πολύγωνα και δεν απεικονίζονται με λεπτομέρειες. Το γεγονός αυτό σε συνδυασμό με τα χαμηλής ανάλυσης textures αποδίδει ένα μη εμφανίσιμο αποτέλεσμα, ακόμα και για κινητά τα οποία, ειδικά στην εποχή όπου κυκλοφόρησε η εφαρμογή, δε διαθέτουν την ίδια ισχύ όπως ένας ηλεκτρονικός υπολογιστής. Μια άλλη τακτική που δεν εφαρμόζεται πολύ σωστά είναι το Distance-based texture downscaling, όπου τα textures των μοντέλων αλλάζουν σε textures κατώτερης ανάλυσης ανάλογα με το πόσο μακριά από τον παίκτη είναι. Παρόλο που αυτό γίνεται για να αυξηθεί η απόδοση της εφαρμογής ταυτόχρονα δημιουργεί και μία απογοητευτική παράσταση όπου τα textures των μοντέλων απεικονίζονται σε μεγαλύτερη ανάλυση σε μικρή απόσταση από τον παίκτη, με αποτέλεσμα όσο προχωράει το όχημα πολλά μοντέλα, όπως ο δρόμος και τα κτήρια,

να γίνονται πιο οπτικά λεπτομερές. Επίσης δε γίνεται πουθενά χρήση είτε baked είτε real time προσομοίωσης φωτός.

Παρόλα αυτά το παιχνίδι προσφέρει ικανοποιητική ψυχαγωγία στον παίκτη, δεν προσφέρει κάτι παραπάνω όσον αφορά την εξοικείωση με τους κανόνες του κώδικα οδικής κυκλοφορίας. Δεν υπάρχουν επίσης ρεαλιστικά σενάρια, όπως χρήση σημάτων ή προτεραιοτήτων στους δρόμους, δεν υπάρχουν φανάρια. Η χρήση παρόμοιων σεναρίων σίγουρα θα βοηθούσε πολύ στο να εξελιχθεί αυτή η εφαρμογή από ένα απλό παιχνίδι που βασίζεται πάνω στην ιδέα της οδικής ασφάλειας σε μία εκπαιδευτική εφαρμογή.

2.3 Robok.o.k.

Η εφαρμογή Robok.o.k. αναπτύχθηκε από δύο μεταπτυχιακές φοιτήτριες του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (Ελένη Σιαμτανίδου και Αναστασία Γουναρίδου) με τη μηχανή Unreal Engine (άγνωστη η έκδοση που χρησιμοποιήθηκε). Σύμφωνα με συνέντευξη των δύο φοιτητριών το Robok.o.k. συνδυάζει την διασκέδαση και την εκμάθηση του κώδικα οδικής κυκλοφορίας και απευθύνεται σε μικρά παιδιά ηλικίας 8 με 12 χρόνων, όπως αναφέρεται στην ενημερωτική ιστοσελίδα infokids [22]. Έμπνευση για την ανάπτυξη της εφαρμογής ήταν ο υψηλός αριθμός τροχαίων ατυχημάτων στην Ελλάδα.

Το παιχνίδι έχει 4 σενάρια τα οποία μπορούν να ολοκληρωθούν με όχημα καθώς και με τον παίκτη να είναι πεζός. Μέσα στο παιχνίδι ο παίκτης καλείται να ολοκληρώσει μια αποστολή που του δίνεται στην αρχή της πίστας είτε οδηγώντας είτε περπατώντας με προσοχή. Όταν ο παίκτης οδηγήσει και μπει σε αντίθετο ρεύμα ή οδηγήσει μέσα από το πεζοδρόμο χάνει μία από τις 3 ευκαιρίες που έχει σε κάθε περίπτωση. Αν ο παίκτης περάσει με λάθος χρώμα σηματοδότη ή χτυπήσει πάνω στα γύρω οχήματα τότε το παιχνίδι τερματίζει κατευθείαν και πρέπει να επαναληφθεί το σενάριο από την αρχή. Όταν πεζός ο παίκτης πρέπει να περπατήσει προς τον προορισμό του. Περνώντας το δρόμο από σημεία όπου δεν υπάρχει διάβαση πεζών ή με λάθος σηματοδότη τερματίζει το παιχνίδι αυτόματα και το σενάριο επίσης πρέπει να επαναληφθεί από την αρχή. Σε κάθε περίπτωση σφάλματος παρέχεται στον παίκτη και ένα μικρό επεξηγηματικό κείμενο όπου εξηγείται το σφάλμα του. Σε διάφορα σημεία κατά τη διάρκεια του παιχνιδιού ο παίκτης πρέπει να απαντήσει μερικές ερωτήσεις που αφορούν την οδική ασφάλεια όπως για παράδειγμα την χρήση ζώνης ή κινητών τηλεφώνων κατά τη διάρκεια της οδήγησης. Οι ιδέες και τα σενάρια που υλοποιούνται στην εφαρμογή είναι όσο περίπλοκα και δύσκολα θα θεωρούσε κάποιος ως δεδομένο πως το κοινό της είναι παιδιά μικρής ηλικίας. Είναι διασκεδαστικό και έχει μια μεγάλη ποικιλία από πίστες και διάφορα σενάρια περιβάλλοντος οδήγησης. Τα σενάρια στο δρόμο ωστόσο είναι λίγα αλλά αυτό οφείλεται λογικά σε χρονικούς περιορισμούς.

Η Unreal Engine η οποία χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής Robok.o.k είναι μία πολύ γνωστή μηχανή ανάπτυξης ηλεκτρονικών παιχνιδιών και προσφέρει πληθώρα features και εργαλείων τα οποία μπορούν να χρησιμοποιήσουν οι προγραμματιστές. Ένα άλλο θετικό χαρακτηριστικό της είναι και η ποιότητα graphics rendering που προσφέρει. Μπορεί να κάνει χρήση μοντέλων με δεκάδες ή και εκατοντάδες χιλιάδες πολύγωνα και προσφέρει πολύ καλές προσομοιώσεις φωτός, σκιών και after effects. Δεδομένων όλων αυτών τα γραφικά στο παιχνίδι φαίνονται ικανοποιητικά αλλά σίγουρα μπορεί να υπάρξει βελτίωση. Συγκεκριμένα θα μπορούσε να γίνει χρήση normal maps σε συνδυασμό με height maps ή bump maps σε δρόμους και κτήρια για να φαίνεται περισσότερο η λεπτομέρεια πάνω στις επιφάνειες. Στην περίπτωση αυτή δεν θα

υπήρχε μεγάλη διαφορά στην απόδοση καθώς η χρήση των normal maps και bump maps δεν είναι απαιτητική στον επεξεργαστή και στην κάρτα γραφικών.

2.4 Safety Driving Simulator: Car

Το Safety Driving Simulator: Car [23] δημιουργήθηκε και εκδόθηκε από την εταιρία United Independent Entertainment GmbH στις 4 Μαΐου 2016. Δεν υπάρχουν πολλές πληροφορίες για την ανάπτυξη του παιχνιδιού, όπως για παράδειγμα η μηχανή που χρησιμοποιείται για rendering των γραφικών. Σύμφωνα με τη σελίδα τους στην πλατφόρμα διανομής ηλεκτρονικών παιχνιδιών Steam το παιχνίδι έχει σκοπό να βάλει τον παίκτη σε ένα προσομοιωμένο περιβάλλον όπου μπορεί να οδηγήσει σε διαφορετικές πόλεις ακολουθώντας ταυτόχρονα κανόνες οδικής ασφάλειας. Υπάρχει σύστημα που ελέγχει αν οδηγεί ο παίκτης στη σωστή λωρίδα και άλλες παραβάσεις, προσομοίωση επήρειας αλκοόλ και ναρκωτικών ουσιών, ρεαλιστική βιωσιμότητα όπως πινακίδες στους δρόμους, πεζοί και φανάρια. Επίσης παρέχεται πληθώρα επιλογών στα οχήματα.

Σύμφωνα όμως με πολλές κριτικές χρηστών το παιχνίδι έχει πολλαπλά προβλήματα τα οποία φαίνεται να έχουν ρίζα τον σχεδιασμό και τον χειρισμό του. Πολλοί παίκτες παραπονέθηκαν για προβλήματα όπως μη λειτουργική προσομοίωση physics και διάφορα bugs και glitches που προέκυψαν ενώ χρησιμοποιούσαν την εφαρμογή, όπως για παράδειγμα δυσκολία στις στροφές των αυτοκινήτων που αναφέρθηκε από μερικούς χρήστες. Επίσης προβλήματα που αναφέρονταν πολλοί παίκτες είναι στα γραφικά του παιχνιδιού. Όντως και από το φωτογραφικό υλικό της εφαρμογής μπορούμε να δούμε μερικά πράγματα που θα χρειαζόνταν σίγουρα βελτίωση όπως η χρήση προσομοίωσης φωτός πραγματικού χρόνου για αντικείμενα που κουνιούνται, ambient φως σε μικρές γωνίες για τον τονισμό της λεπτομέρειας, μοντέλα χαμηλών πολυγώνων με χαμηλής ποιότητας textures. Κάτι ενδιαφέρον που παρατηρήθηκε είναι και η έλλειψη anisotropic φιλτραρίσματος, τεχνική που χρησιμοποιείται σε γραφικά υπολογιστών ώστε τα textures πάνω σε αντικείμενα να μη φαίνονται θολά όταν τα κοιτάμε από μία λοξή γωνία και ταυτόχρονα όταν είναι πολύ μακριά. Αυτό μπορεί να οφείλεται στην αδυναμία της συσκευής με την οποία πάρθηκαν οι συγκεκριμένες εικόνες να χρησιμοποιήσει την εν λόγω τεχνική ή όντως η μηχανή με την οποία δημιουργήθηκε το παιχνίδι να μην υποστηρίζει anisotropic φιλτράρισμα (κάτι το οποίο πλέον πολύς κόσμος μπορεί να εφαρμόσει πολύ απλά μέσω drivers για κάρτες γραφικών).

3. Υλοποιημένα Σενάρια

Για τους σκοπούς της παρούσας εφαρμογής, υλοποιήθηκαν διάφορα σενάρια. Τα σενάρια αυτά προσομοιάζουν σε κάποιο βαθμό σενάρια της πραγματικής ζωής που θα συνέβαιναν κατά τη διάρκεια μιας διαδρομής. Τα σενάρια αυτά, τα οποία θα εξηγηθούν επίσης σε μεγαλύτερο βάθος παρακάτω, είναι τα ακόλουθα:

- Φανάρια



Εικόνα 6. Στιγμιότυπο από την εφαρμογή που φαίνονται τα φανάρια

- Απαγόρευση στροφής δεξιά, αριστερά και ευθείας πορείας



Εικόνα 7. Στιγμιότυπο από την εφαρμογή με απαγόρευση αριστερής στροφής

- Όριο ταχύτητας



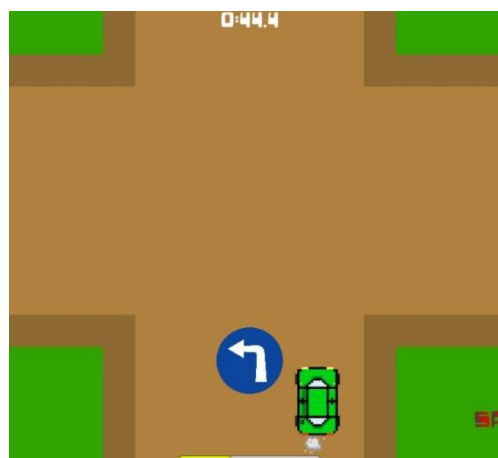
Εικόνα 8. Στιγμιότυπο από την εφαρμογή με όριο ταχύτητας

- Έργα δρόμου



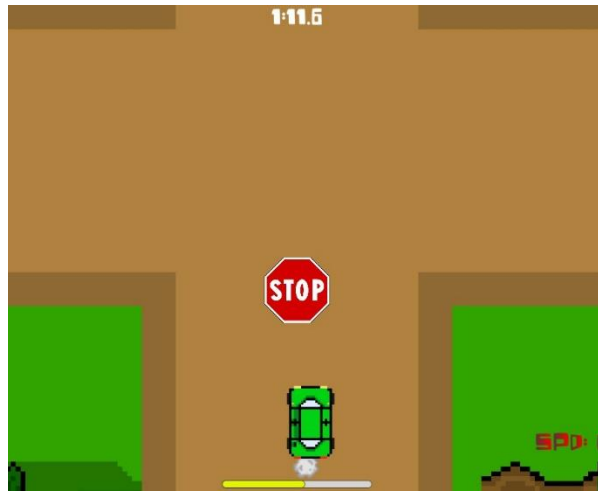
Εικόνα 9. Στιγμιότυπο με έργα δρόμο στην αριστερή λωρίδα

- Υποχρεωτική πορεία δεξιά, αριστερά ή ευθεία



Εικόνα 10. Στιγμιότυπο με υποχρεωτική αριστερή στροφή

- Πινακίδα STOP



Εικόνα 11. Στιγμιότυπο με πινακίδα STOP

Καθ' όλη τη διάρκεια του παιχνιδιού, ολόκληρο το επίπεδο δημιουργείται διαδικαστικά. Αυτό σημαίνει ότι κάθε πλακίδιο φόντου στο επίπεδο που παίζει ο παίκτης τοποθετείται τυχαία. Κάθε επίπεδο από το οποίο ξεκινάει ο παίκτης είναι μοναδικό και τα σενάρια εμφανίζονται με τυχαίο τρόπο στα πλακίδια. Αυτό συμβαίνει σύμφωνα με ορισμένους κανόνες, όπως το γεγονός ότι το σενάριο που δεν αφήνει τον παίκτη να στρίψει δεξιά ή αριστερά, δεν θα εμφανιστεί σε έναν ευθύ δρόμο ή πως οι μηχανισμοί της υποχρεωτικής πορείας δε μπορούν να εμφανιστούν μαζί με μια απαγόρευση πορείας. Για κάθε μηχανισμό υπάρχει ένα μοναδικό ποσοστό πιθανοτήτων να εμφανιστεί σε κάθε πλακίδιο. Ακόμα και το μήκος του επιπέδου μπορεί να καθοριστεί τυχαία αν θέλουμε, αφού όλα αυτά είναι απλά παράμετροι στην εφαρμογή.

3.1 Game Flow και Gamification

Στην προκειμένη περίπτωση, η ροή του παιχνιδιού (Game Flow) αναφέρεται στη μετάβαση από το ένα επίπεδο στο επόμενο. Σε αυτό το project έχουμε διαφορετικά επίπεδα που μπορούν να επαναληφθούν. Ο τρόπος με τον οποίο λειτουργεί αυτό είναι με το να δημιουργούνται διαδικαστικά διαφορετικοί μηχανισμοί του παιχνιδιού. Για παράδειγμα, τα πλακίδια του χάρτη εμφανίζονται από μια λίστα πλακιδίων με κάποιες προδιαγραφές. Στις διασταυρώσεις τα πλακίδια που εμφανίζονται σε όλες τις πλευρές δεν μπορούν να είναι επίσης τύπου διασταυρώσεων. Οι μηχανισμοί γεννιούνται επίσης τυχαία μέσα σε όλα τα πλακίδια. Δεν μπορούν να εμφανίζονται περισσότεροι από ένας μηχανισμός ταυτόχρονα, αλλά αυτό θα μπορούσε να εφαρμοστεί και να ρυθμιστεί για να αυξηθεί η δυσκολία του παιχνιδιού.

Κατά τη διάρκεια ενός επιπέδου, ο παίκτης κερδίζει πόντους για την επιτυχή ολοκλήρωση σεναρίων που εμφανίζονται τυχαία. Ο παίκτης κερδίζει επίσης πόντους καθώς προχωράει μπροστά, οπότε όσο μεγαλύτερη είναι η διάρκεια του επιπέδου, τόσο περισσότερους πόντους κερδίζει. Κάθε φορά που ο παίκτης αποτυγχάνει να ολοκληρώσει ένα σενάριο, αφαιρούνται πόντοι από τους συσσωρευμένους πόντους. Ο παίκτης επιτρέπεται να κάνει 3 λάθη κατά τη διάρκεια ενός επιπέδου, διαφορετικά το επίπεδο τελειώνει και χάνει όλους τους

συσσωρευμένους πόντους του. Μετά την ολοκλήρωση του επιπέδου, οι συσσωρευμένοι πόντοι μετατρέπονται σε νόμισμα εντός του παιχνιδιού. Ο παίκτης μπορεί στη συνέχεια να χρησιμοποιήσει αυτό το νόμισμα για να ξεκλειδώσει νέα οχήματα και skins οχημάτων που μπορεί να χρησιμοποιήσει.

Για την ανάπτυξη αυτού του παιχνιδιού, προσπαθήσαμε να ακολουθήσουμε την αρχή του gamification. Σύμφωνα με τους Jenni Majuri, Joanna Koivisto και Juho Hamari στην εργασία τους στα πρακτικά του συνεδρίου GameFin [6] και σύμφωνα με το άρθρο "An Approach for Modularizing Gamification Concerns" [8], το gamification είναι η αναπαράσταση ενός σεναρίου ή καταστάσεων με τη χρήση μηχανισμών παιχνιδιού, δηλαδή με τη χρήση διαφόρων εννοιών παιχνιδιού. Στοιχεία παιχνιδιού είναι η επίτευξη και η ανταμοιβή μέσω αυτού και η υποστήριξη διαφορετικών συμπεριφορών και δραστηριοτήτων οι οποίες προσπαθούν να εκπαιδεύσουν τον παίκτη. Αυτές οι έννοιες και οι μηχανισμοί μπορεί να είναι εξαιρετικά περίπλοκοι ή πολύ απλοί. Μπορεί να είναι δύσκολο να πληρούνται όλες οι προϋποθέσεις του gamification, καθώς μια τέτοια επιδίωξη μπορεί να οδηγήσει σε ένα δυσλειτουργικό προϊόν. Η συντήρηση της εφαρμογής και ο σχεδιασμός μπορεί επίσης να είναι πολύ δύσκολο να επιτύχουν ικανοποιητικό αποτέλεσμα, δεδομένου ότι τα εναύσματα για τους μηχανισμούς του παιχνιδιού μπορεί να είναι πολύ περίπλοκα, όπως αναφέρθηκε προηγουμένως.

Ακολουθώντας τις γενικές αυτές αρχές αυτό που προσπαθούμε να επιτύχουμε με αυτό το παιχνίδι είναι να ενθαρρύνουμε την σωστή οδική συμπεριφορά, καθώς και να εκπαιδεύσουμε τον παίκτη σε βασικές έννοιες της οδήγησης, μέσω ενός διασκεδαστικού παιχνιδιού. Μέσω των σεναρίων που υλοποιούμε προσπαθούμε να δείξουμε στους παίκτες πώς να συμπεριφέρονται στα φανάρια, να αποφεύγουν επικίνδυνες περιοχές στους δρόμους και κάποια απλά σήματα που θα μπορούσαν να συναντήσουν σε μια καθημερινή διαδρομή. Ενθαρρύνουμε επίσης τους παίκτες μας να κοινωνικοποιηθούν και να διαδώσουν τις επιδόσεις τους στα μέσα κοινωνικής δικτύωσης, δημοσιεύοντας τις επιδόσεις τους μέσω του παιχνιδιού μας στο λογαριασμό τους στο Facebook.

Παρόλο που ορισμένοι ερευνητές, σύμφωνα με τον Mazarakis A. στο άρθρο του σχετικά με το gamification [10], δεν έχουν βρει μόνιμα πλεονεκτήματα στη χρήση του gamification για τη μάθηση και την ενθάρρυνση θετικής συμπεριφοράς, φαίνεται ότι υπάρχουν και οφέλη από την εφαρμογή του gamification στα παιχνίδια, όπως η μεγαλύτερη σύνδεση με τον χρήστη μέσω του παιχνιδιού. Φυσικά η gamification εφαρμόζεται έντονα σε ό,τι αφορά τις εμπειρίες AR και VR, όπου η εμπύθιση μπορεί να είναι πολύ μεγαλύτερη από μια εφαρμογή για κινητά.

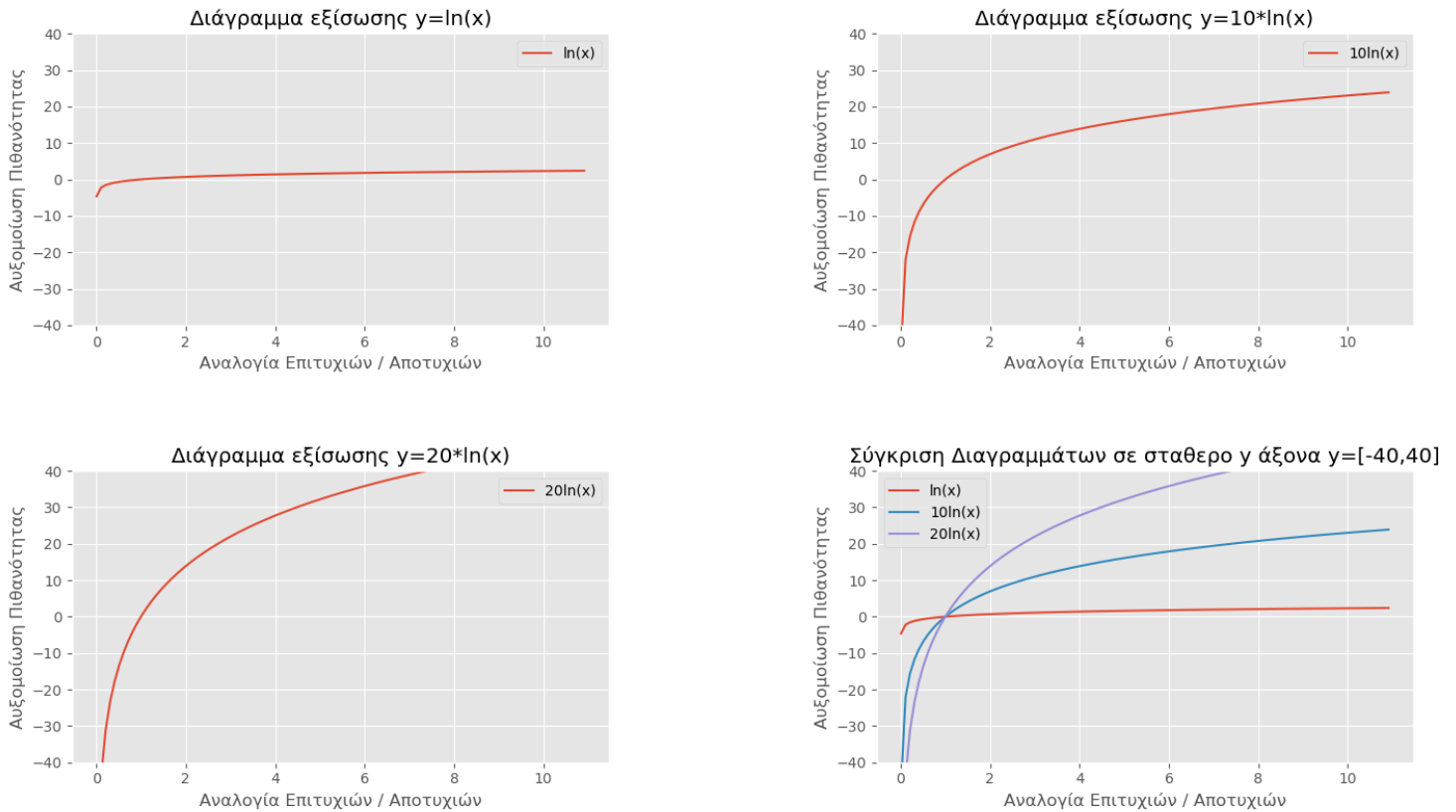
Μια τεχνική εφαρμογή που εξυπηρετεί τον σκοπό του παιχνιδιού μας και μπορεί να αυξήσει περαιτέρω τις γνώσεις μας σχετικά με τις δυσκολίες που αντιμετωπίζουν οι παίκτες μας κατά τη διάρκεια του παιχνιδιού, είναι να κρατάμε στατιστικά στοιχεία σε μια βάση δεδομένων με πληροφορίες στην αποτυχία ή στην επιτυχία που έχουν οι παίκτες κατά τη διάρκεια μιας πίστας. Αξιοποιώντας αυτά τα δεδομένα, μπορούμε στη συνέχεια να καθορίσουμε, σε ποιους μηχανισμούς οι παίκτες αποτυγχάνουν περισσότερο, και έτσι μπορούμε να καθορίσουμε τη συχνότητα με την οποία θα πρέπει να εμφανίζονται αυτοί οι μηχανισμοί. Για να κάνουμε τα επίπεδα πιο δύσκολα, το παιχνίδι θα εμφανίζει πιο συχνά τους πιο δύσκολους μηχανισμούς και για να κάνουμε το παιχνίδι πιο εύκολο θα εμφανίζει πιο συχνά τους μηχανισμούς στους οποίους οι παίκτες αποτυγχάνουν λιγότερο. Χρησιμοποιώντας αυτά τα δεδομένα, θα μπορούσαμε επίσης

να εξισορροπήσουμε περισσότερο το παιχνίδι μας και να διορθώσουμε συγκεκριμένους αριθμούς που επηρεάζουν τους μηχανισμούς του παιχνιδιού και να διορθώσουμε ζητήματα που μπορεί να προκύψουν κατά τη διάρκεια του playthrough των παικτών.

Η λήψη των δεδομένων των παικτών και ο τρόπος επεξεργασίας που προτάθηκε παραπάνω ακολουθεί ορισμένες από τις βασικές αρχές της Δυναμικής Προσαρμογής (Dynamic Adjustment), όπως αναφέρεται στο άρθρο ανασκόπησης του Mohammad Zohaib [24]. Η DDA (Dynamic Difficulty Adjustment) είναι μια μέθοδος σχεδιασμού παιχνιδιών που προσαρμόζει αυτόματα τη δυσκολία του παιχνιδιού ώστε να αντισταθμίζει τις δεξιότητες των παικτών. Όσο καλύτερος είναι ο παίκτης στο παιχνίδι, τόσο πιο δύσκολο γίνεται το παιχνίδι σε πραγματικό χρόνο, προκειμένου να διασφαλιστεί ότι ο παίκτης δεν θα βαρεθεί ή δε θα απογοητευτεί, σε περίπτωση που το παιχνίδι είναι πολύ δύσκολο. Αυτό μπορεί να γίνει με διάφορους τρόπους, όπως η προσαρμογή της τυχαιότητας και της συχνότητας των εμποδίων που αντιμετωπίζει ο παίκτης κατά τη διάρκεια του παιχνιδιού. Με την ανώνυμη ανάλυση των συνολικών δεδομένων των παικτών, μπορούμε να προσδιορίσουμε ποιους μηχανισμούς οι παίκτες θεωρούν δύσκολους και η εφαρμογή, υπολογίζοντας τη δυσκολία ορισμένων μηχανισμών, μπορεί να καθορίσει τη συχνότητα εμφάνισης των μηχανισμών ώστε να επιτευχθεί μεγαλύτερη ισορροπία.

Στην εφαρμογή μας αυτό επιτυγχάνεται αρχικά με την συλλογή των απαραίτητων δεδομένων η οποία γίνεται ανώνυμα. Πριν ξεκινήσει ο παίκτης την πίστα, συλλέγουμε δεδομένα για κάθε έναν από τους μηχανισμούς που εμφανίζονται τυχαία στο παιχνίδι. Μέσω ενός SQL query το οποίο τρέχει σε ένα PHP script συλλέγουμε δεδομένα για ένα μηχανισμό χρησιμοποιώντας σαν παραμέτρους το ID του χρήστη, το οποίο δημιουργείται αυτόματα εξετάζοντας την διεύθυνση MAC της συσκευής του, και το όνομα του μηχανισμού που θέλουμε να εξετάσουμε. Έπειτα μετράμε πόσες φορές έχει αποτύχει σε αυτόν τον μηχανισμό και πόσες φορές έχει τον έχει περάσει με επιτυχία. Το επόμενο βήμα είναι να φτιάξουμε μια αναλογία με τα δεδομένα του μηχανισμού: $\frac{\text{αριθμός επιτυχιών}}{\text{αριθμός αποτυχιών}}$. Αυτό το αποτέλεσμα περνάει μετά στο κυρίως C# script όπου θα εφαρμόσουμε τον φυσικό λογάριθμο και θα πολλαπλασιάσουμε το αποτέλεσμα με έναν συντελεστή προσαρμογής αποτελέσματος a . Ο συντελεστής αυτός χρησιμοποιείται έτσι ώστε να προσαρμόσουμε το αποτέλεσμα του φυσικού λογαρίθμου, ώστε να εξυπηρετεί καλύτερα το σκοπό μας. Στην εφαρμογή μας επιλέξαμε να δώσουμε στο συντελεστή a την τιμή 15, διότι μετά από δοκιμές πιστεύουμε πως δίνει τη καλύτερη μείωση και αύξηση πιθανοτήτων.

$$\text{τροποποίηση συχνότητας εμφάνισης} = a * \log_e\left(\frac{\text{αριθμός επιτυχιών}}{\text{αριθμός αποτυχιών}}\right)$$



Εικόνα 12. Σύγκριση τροποποίησης συχνότητας εμφάνισης με διαφορετικούς συντελεστές

Το αποτέλεσμα αυτής της πράξης προστίθεται στη συχνότητα εμφάνισης του αντίστοιχου μηχανισμού. Εάν η αναλογία είναι μικρότερη του 1, δηλαδή ο παίκτης δυσκολεύεται σε κάποιο μηχανισμό, τότε το αποτέλεσμα είναι ένας αριθμός που αφαιρείται από τη συχνότητα εμφάνισης του μηχανισμού με αποτέλεσμα ο μηχανισμός αυτός να εμφανίζεται λιγότερο. Στη περίπτωση που το αποτέλεσμα είναι πάνω από 1 τότε η συχνότητα εμφάνισης του μηχανισμού αυξάνεται.

Ο λόγος για την προσαρμογή της DDA είναι να διατηρηθεί σταθερή η ροή του παιχνιδιού. Σύμφωνα με τον καθηγητή Mihaly Csikszentmihaly στο άρθρο του Flow: The Psychology of Optimal Experience [25], η διατήρηση της ροής κατά τη διάρκεια ορισμένων στιγμών μπορεί να αυξήσει τις μαθησιακές εμπειρίες. Γι' αυτό θέλουμε να επιτύχουμε μια ικανοποιητική ροή για τον παίκτη, αφού ο στόχος του παιχνιδιού μας είναι η εκπαίδευση. Συμπερασματικά λοιπόν, σταθεροποιώντας τη ροή ενός ατόμου με την εμπειρία του παιχνιδιού μας, μπορούμε να μεγιστοποιήσουμε την ικανότητά του να μάθει από το παιχνίδι μας.

3.2 Το Σύστημα και τα Στοιχεία του

Όπως αναφέρθηκε, το παιχνίδι αναπτύχθηκε σε Unity 3D. Το Unity χρησιμοποιεί ένα πολύ αυστηρό αντικειμενοστραφές μοντέλο για τη δημιουργία ακόμη και των πιο βασικών γεωμετρικών και χαρακτηριστικών.

Το παιχνίδι χωρίζεται σε σκηνές. Υπάρχουν οι σκηνές παιχνιδιού, όπου τελούνται οι περισσότερες από τις βασικές λειτουργίες του παιχνιδιού, και η σκηνή του κύριου μενού, όπου ο παίκτης μπορεί να αλλάξει τις επιλογές του παιχνιδιού, να προσαρμόσει το όχημά του και να επιλέξει ένα επίπεδο για να παίξει.

3.2.1 Κυρίως Μενού

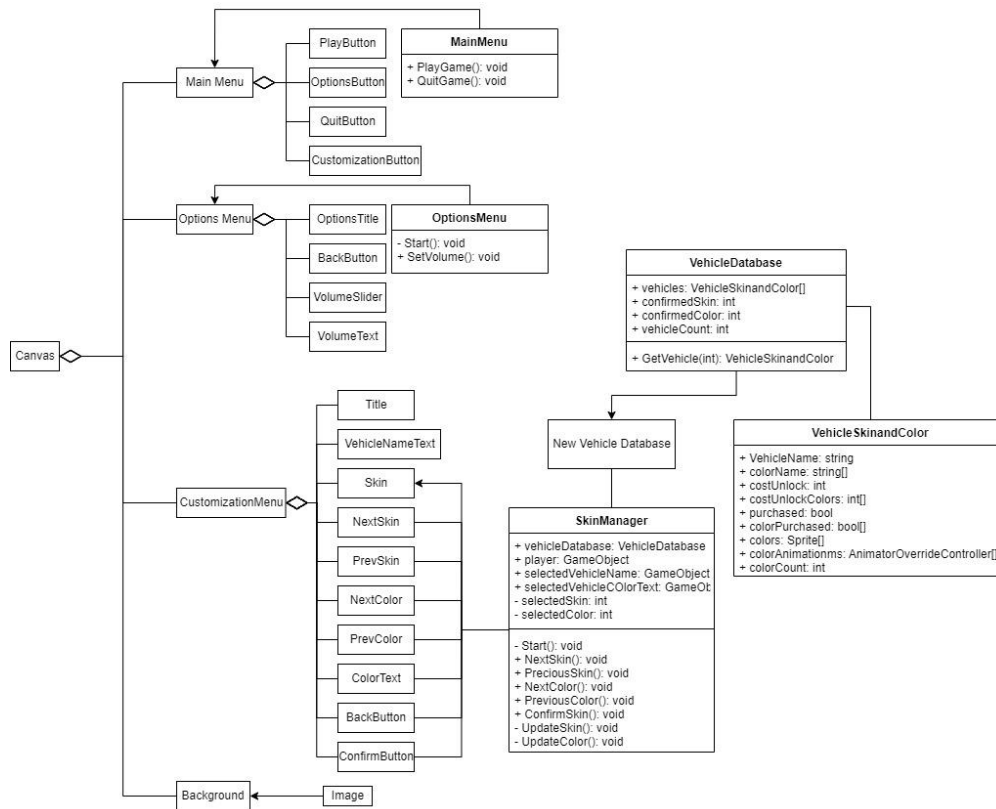
Στο κύριο μενού έχουμε τον βασικό καμβά. Σε αυτόν τον καμβά τοποθετούμε όλα τα στοιχεία της διεπαφής χρήστη, όπως κουμπιά, διαδραστικά κείμενα, ετικέτες, sliders και πολλά άλλα. Όταν το παιχνίδι φορτώνεται για πρώτη φορά, μας παρουσιάζεται το κύριο μενού. Στον καμβά του κύριου μενού υπάρχουν τέσσερα κουμπιά με τις ετικέτες play, customize, options και quit.

Το κουμπί play μεταφέρει τον παίκτη σε μια οθόνη όπου θα μπορεί να επιλέξει ένα επίπεδο για να παίξει και στη συνέχεια το παιχνίδι φορτώνει μια συγκεκριμένη σκηνή που αντιστοιχεί σε αυτό το επίπεδο. Το Unity περιλαμβάνει μια απλή βιβλιοθήκη διαχείρισης σκηνών, με την οποία ο προγραμματιστής αναθέτει αριθμούς id στις σκηνές και μπορεί να τις τοποθετήσει σε μια λίστα ουράς φόρτωσης. Από εκεί μπορεί είτε να φορτώσει την επόμενη σκηνή στην ουρά είτε να φορτώσει μια σκηνή με βάση τον αριθμό id της.

Το κουμπί customization μεταφέρει τον παίκτη σε μια σκηνή όπου μπορεί να επιλέξει ένα όχημα για να παίξει το επίπεδο. Προς το παρόν υπάρχει μόνο ένα όχημα για να διαλέξει, αλλά αυτό το όχημα έχει επίσης διαφορετικές χρωματικές παραλλαγές. Τα δεδομένα για το ποιο όχημα φορτώνεται κάθε φορά στο μοντέλο του παίκτη τα διαχειρίζεται ο κώδικας. Για τον σκοπό αυτό δημιουργήθηκε μια μικρή δομή που αποθηκεύει sprites οχημάτων σε συγκεκριμένους τύπους οχημάτων με διαφορετικά χρώματα και κινούμενα σχέδια. Το μενού προσαρμογής επιτρέπει στον παίκτη να περάσει από αυτές τις επιλογές και αργότερα οι κλειδωμένες επιλογές μπορούν να ξεκλειδωθούν μέσω του νομίσματος του παιχνιδιού.

Το κουμπί options μεταφέρει τον παίκτη στην οθόνη επιλογών όπου μπορεί να αλλάξει την ένταση του παιχνιδιού. Υπάρχει ένα αντικείμενο στο παιχνίδι που χειρίζεται όλους τους ήχους του. Η οθόνη αλληλεπιδρά με αυτό το αντικείμενο και ορίζει έναν αριθμό float ως ποσοστό έντασης των ήχων. Το slider έχει 5 στάδια που ρυθμίζουν την ένταση του ήχου. Το αντικείμενο που χειρίζεται τον ήχο είναι ένα από τα λίγα αντικείμενα που δεν καταστρέφονται μεταξύ των μεταβάσεων σκηνών. Η Unity επιτρέπει στον προγραμματιστή να έχει αντικείμενα που μπορούν να μεταφερθούν σε άλλες σκηνές είτε υπάρχουν ήδη εκεί είτε όχι.

Και τέλος, το κουμπί quit έχει την πολύ απλή αποστολή να τερματίζει το παιχνίδι. Πρόκειται για μια απλή λειτουργία που αποκτά πρόσβαση στα στοιχεία ελέγχου των εφαρμογών και απλά τις τερματίζει.



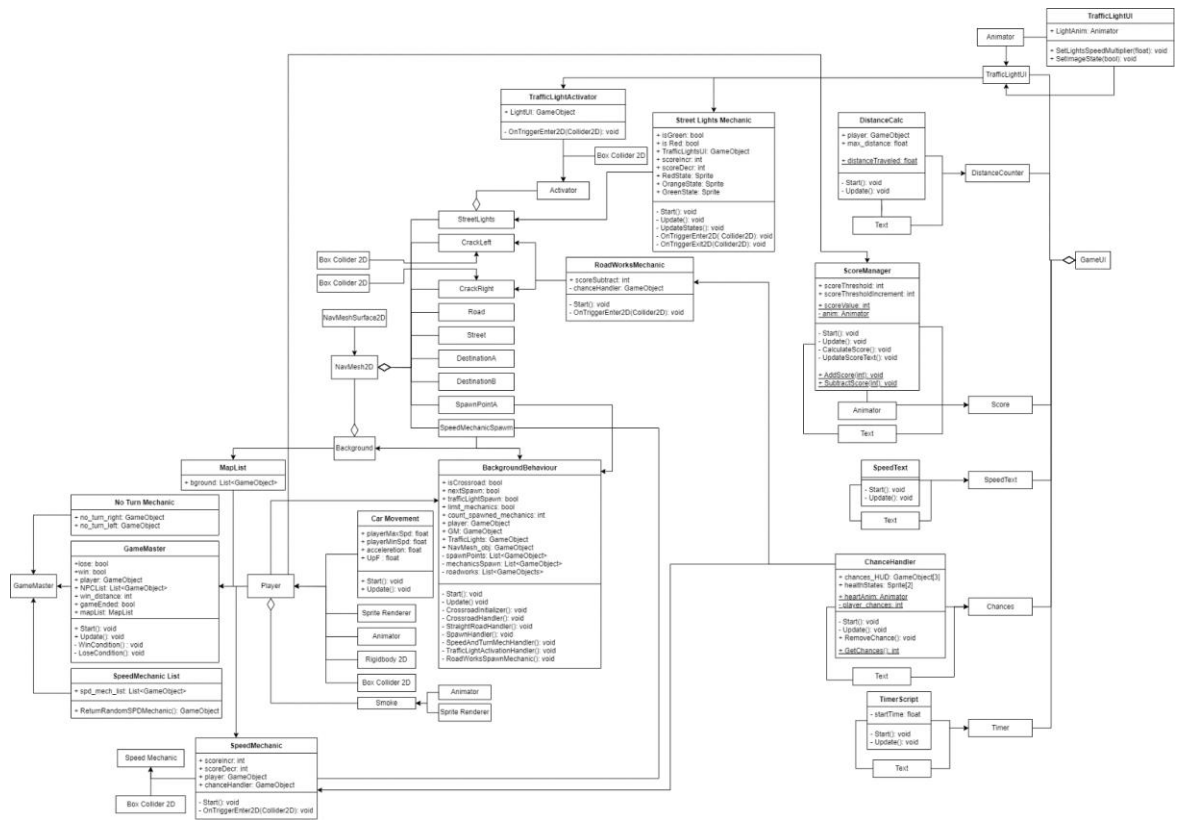
Εικόνα 13. UML Class Diagram για το κυρίως μενού

Όλα τα κουμπιά διασυνδέονται με τα αντίστοιχα πάνελ στον καμβά. Για παράδειγμα, αν ο παίκτης βρίσκεται στο κύριο μενού και θέλει να μεταβεί στο μενού επιλογών, το μόνο που έχουμε να κάνουμε είναι απλά να ορίσουμε τη συνάρτηση κλήσης κουμπιού `SetActive(true)` για το αντικείμενο `OptionsMenu` στον καμβά και να καλέσουμε την ίδια συνάρτηση για το κύριο μενού, αλλά αυτή τη φορά περνώντας την τιμή `false` για να το απενεργοποιήσουμε, ώστε να μην έχουμε και τα δύο πάνελ ενεργά ταυτόχρονα. Αντίστροφα για να επιστρέψουμε από το μενού επιλογών στο κύριο μενού, όταν ο παίκτης πατήσει το κουμπί `Back`, το μόνο που κάνουμε είναι να επικαλεστούμε τη συνάρτηση `SetActive` με την τιμή `false` για το πάνελ επιλογών και να κάνουμε το ίδιο με τη συνάρτηση `true` για το αντικείμενο `MainMenu`. Αυτό είναι κάτι που μπορεί να γίνει χωρίς ούτε μια γραμμή κώδικα χάρη στη διεπαφή χρήστη της Unity. Κάποιες βασικές λειτουργίες όπως αυτή έχουν ήδη υλοποιηθεί σε όλα τα κουμπιά. Για πιο ενδιαφέρουσες συναρτήσεις ο προγραμματιστής πρέπει να γράψει τις δικές του και στη συνέχεια να τις εφαρμόσει στο κουμπί στο συμβάν `Click()`.

3.2.2 Σκηνή Παιχνιδιού

Στη σκηνή του παιχνιδιού υπάρχουν πολλά στοιχεία που αλληλοεπιδρούν μεταξύ τους. Το πιο σημαντικό που αλληλοεπιδρά με τα περισσότερα στοιχεία κατά τη διάρκεια του παιχνιδιού είναι το αντικείμενο παίκτη. Η κύρια λειτουργία αυτού του αντικειμένου είναι να αντιπροσωπεύει τον παίκτη στον κόσμο του παιχνιδιού. Περιέχει ένα `sprite` για τα γραφικά, ένα `script` που χειρίζεται τα `controls` του παίκτη και ένα `2D Box Collider` που χρησιμοποιείται για την ενεργοποίηση των συγκρούσεων με τους μηχανισμούς.

Το επόμενο πιο σημαντικό στοιχείο είναι το αντικείμενο GameManager. Το αντικείμενο GameManager περιέχει scripts και λειτουργίες που είναι πιο σφαιρικά διαθέσιμα σε όλα τα αντικείμενα της σκηνής, καθώς και χρήσιμα δεδομένα στα οποία μπορούν να έχουν πρόσβαση όλα τα αντικείμενα της σκηνής. Για παράδειγμα, το GameManager περιέχει μια αναφορά στο αντικείμενο Player, ώστε να μη χρειάζεται να ψάχνουμε κάθε φορά ολόκληρη τη σκηνή για να βρούμε το αντικείμενο Player. Περιέχει επίσης λίστες και πληροφορίες με μηχανισμούς, όπως μια λίστα με όλους τους μηχανισμούς ταχύτητας που μπορούν να αναπαραχθούν στο χάρτη και αναφορά στους μηχανισμούς που δεν επιτρέπεται η στροφή. Περιλαμβάνει ακόμα μια λίστα με όλα τα διαθέσιμα φόντα από τα οποία μπορεί να επιλέξει το script που χειρίζεται τα φόντα. Τέλος περιέχει πληροφορίες σχετικά με την απόσταση που έχει διανύσει ο παίκτης στο επίπεδο και λειτουργίες που αποφασίζουν πότε τελειώνει το επίπεδο και αν ο παίκτης έχει κερδίσει το επίπεδο ή όχι.



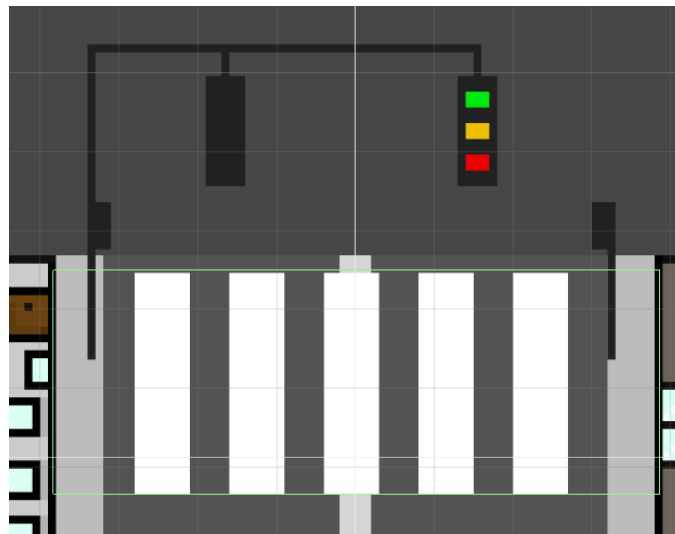
Εικόνα 14. UML Class Diagram για τη σκηνή του παιχνιδιού

Το φόντο είναι το κύριο πεδίο όπου εμφανίζονται όλοι οι μηχανισμοί και αλληλοεπιδρούν με τον παίκτη. Για να αποφευχθεί η ενασχόληση με πολύ μεγάλους αριθμούς, το παιχνίδι έχει σχεδιαστεί για να δίνει στον παίκτη την εντύπωση ότι το όχημα κινείται, βάζοντας το φόντο να γλιστράει προς τα κάτω όταν ο παίκτης κινείται προς τα εμπρός. Όσο πιο γρήγορα οδηγεί ο παίκτης τόσο πιο γρήγορα γλιστράει το φόντο. Αυτό γίνεται με το script BackgroundBehaviour, το οποίο χειρίζεται και άλλες λειτουργίες όπως την εμφάνιση των μηχανισμών στο πεδίο, οπότε περιέχει μια αναφορά στους περισσότερους μηχανισμούς του παιχνιδιού. Για να γίνει αυτό, κάθε φόντο είναι το ίδιο. Έχουν το ίδιο μέγεθος και το μόνο που αλλάζει είναι το sprite του καθενός.

Τα φόντα που αποτελούν μια μορφή σταυροδρομιού χαρακτηρίζονται επίσης ως σταυροδρόμια μέσα στα scripts επειδή αντιμετωπίζονται διαφορετικά από τους ευθείς δρόμους. Ορισμένοι μηχανισμοί δεν μπορούν να εμφανιστούν σε ευθείς δρόμους και ορισμένοι μηχανισμοί επίσης δεν μπορούν να εμφανιστούν σε σταυροδρόμια.

3.2.3 Μηχανισμός Φαναριών

Οι φωτεινοί σηματοδότες είναι ένα από τα πιο προφανή σενάρια που μπορούν να εφαρμοστούν σε ένα βιντεοπαιχνίδι σχετικά με την ασφαλή οδήγηση. Σε αυτό το σενάριο ο παίκτης πρέπει να σταματήσει τουλάχιστον πλήρως ενώ το φανάρι είναι κόκκινο και ταυτόχρονα δεν πρέπει να μπει στη γραμμή διέλευσης πεζών. Μόλις το φανάρι γίνει πράσινο, τότε ο παίκτης μπορεί να συνεχίσει να οδηγεί προς τα εμπρός.



Εικόνα 15. Τα φανάρια με εμφανισμό το mesh collider ως trigger

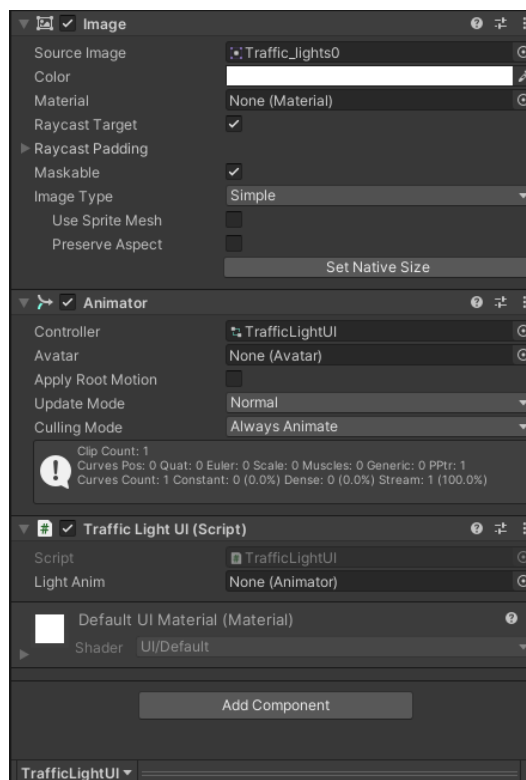
Ο τρόπος με τον οποίο λειτουργεί αυτό το σενάριο είναι ότι το φανάρι εμφανίζεται στη διεπαφή χρήστη στην κάτω αριστερή γωνία της οθόνης. Αυτό συμβαίνει επειδή η κλίμακα του κόσμου δεν προσφέρει μεγάλη ορατότητα των φωτεινών σηματοδοτών στον παίκτη. Τα φανάρια θα έπρεπε να είναι πολύ μεγαλύτερα σε σύγκριση με το περιβάλλον γύρω τους. Τα φανάρια έχουν 3 καταστάσεις: πράσινο, κίτρινο και κόκκινο. Στην πράσινη κατάσταση ο παίκτης επιτρέπεται να συνεχίσει να περνάει, ενώ στην κίτρινη και κόκκινη κατάσταση ο παίκτης πρέπει να σταματήσει πριν από τη διάβαση πεζών.

Το σενάριο λειτουργεί σε δύο διαφορετικά επίπεδα:

- το επίπεδο διεπαφής χρήστη,
- και το επίπεδο παιχνιδιού ή μηχανισμού.

Στο επίπεδο διεπαφής χρήστη, έχουμε ένα αντικείμενο GameObject με ένα απλό component μιας εικόνας, έναν animator και το script που χειρίζεται αυτό το τμήμα της διεπαφής χρήστη. Το animator περιέχει ένα απλό κλιπ κίνησης όπου η εικόνα του GameObject αλλάζει από κόκκινο σε κίτρινο και τελικά σε πράσινο. Έχουμε επίσης ορίσει μια παράμετρο στο animator έτσι ώστε αυτή η παράμετρος να χειρίζεται την ταχύτητα του animation. Αυτό συμβαίνει επειδή με αυτόν τον τρόπο μπορούμε να προσθέσουμε ποικιλία στο μηχανισμό μας, αφού στον πραγματικό

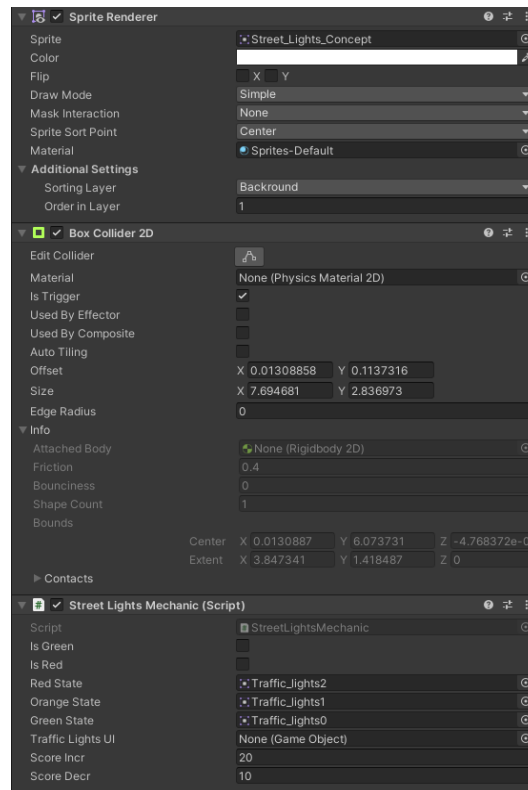
κόσμο ο χρόνος που τα φώτα παραμένουν αναμμένα ποικίλλει. Χαμηλώνοντας την ταχύτητα animation, ουσιαστικά κάνουμε τις καταστάσεις μεγαλύτερες και αυξάνοντάς την τις κάνουμε μικρότερες. Το script είναι μικρό αφού δεν χειρίζεται μηχανισμούς. Η μόνη δουλειά που έχει αυτό το αντικείμενο είναι να πληροφορεί τον παίκτη σχετικά με το φανάρι. Το script έχει ένα public attribute με τύπο Animator. Στη μέθοδο Start() αναθέτουμε στο attribute this το συστατικό Animator αυτού του αντικειμένου. Με αυτόν τον τρόπο έχουμε πρόσβαση στο animator σε περιβάλλον προγραμματισμού και μπορούμε να κάνουμε αλλαγές στις παραμέτρους του animator. Στη συνέχεια έχουμε δύο μεθόδους, public void SetLightsSpeedMultiplier(float m) και public void SetImageState(bool b). Η πρώτη μέθοδος έχει την πολύ απλή δουλειά να περάσει τον αριθμό float m στην παράμετρο του animator που θα πολλαπλασιάσει την ταχύτητα του κλιπ animation με τον εν λόγω αριθμό. Η δεύτερη μέθοδος θα είναι υπεύθυνη για την ενεργοποίηση και την απενεργοποίηση της διεπαφής του φωτεινού σηματοδότη όποτε χρειάζεται. Αυτό το κάνει απλά θέτοντας το enabled stat του συστατικού image του αντικειμένου σε b οπότε είτε true είτε false. Το κάνουμε αυτό επειδή η απενεργοποίηση ολόκληρου του αντικειμένου θα καταστήσει το script του άχρηστο και το script θα σταματήσει να λειτουργεί στο παρασκήνιο. Επίσης, αν το b είναι true αυτό σημαίνει ότι το σενάριο φορτώνει. Έτσι, πρέπει επίσης να ξεκινήσουμε το κλιπ κίνησης του φωτεινού σηματοδότη. Για τον σκοπό αυτόν έχουμε ορίσει ένα trigger ως συνθήκη για τη μετάβαση του animation από την αδράνεια στην ενεργοποίηση των φώτων. Για να το κάνουμε αυτό, απλά λέμε στη σκανδάλη να ενεργοποιηθεί και το animation παίζει επίσης.



Εικόνα 16. Τα βασικά στοιχεία του UI κομματιού

Στο επίπεδο του παιχνιδιού ή του μηχανισμού τα πράγματα περιπλέκονται περισσότερο. Εδώ χειριζόμαστε το μεγαλύτερο μέρος του σεναρίου. Οι περισσότεροι χάρτες περιέχουν το GameObject που είναι ουσιαστικά τα φανάρια. Αυτό το GameObject περιέχει όλα τα στοιχεία

που χρειάζονται τα φανάρια για να λειτουργήσουν. Το αντικείμενο γονέα περιέχει τον Sprite Renderer που αναπαράγει τα γραφικά των φωτεινών σηματοδοτών, ένα Box Collider 2D που έχει οριστεί σε isTrigger και το script που χειρίζεται το μεγαλύτερο μέρος του σεναρίου. Το script περιέχει μερικά χαρακτηριστικά. Συγκεκριμένα, έχουμε δύο bools που αντιπροσωπεύουν τις καταστάσεις των φωτεινών σηματοδοτών (είτε κόκκινα είτε πράσινα), τρία αντικείμενα Sprite που αντιπροσωπεύουν και πάλι τις διαφορετικές καταστάσεις στις οποίες βρίσκονται οι φωτεινοί σηματοδότες. Χρησιμοποιούμε αυτά τα αντικείμενα Sprite για λόγους σύγκρισης. Στη συνέχεια έχουμε μια αναφορά στο στοιχείο διεπαφής χρήστη του μηχανικού. Το χρειαζόμαστε αυτό επειδή αυτά τα δύο αντικείμενα πρέπει να επικοινωνούν, αφού ανάλογα με τις εικόνες που δείχνει το animation στον παίκτη, θα πρέπει να ενεργούμε διαφορετικά. Τέλος, έχουμε δύο παραμέτρους που δείχνουν πόσους πόντους βαθμολογίας παίρνει ο παίκτης για την επιτυχή ολοκλήρωση του μηχανισμού καθώς και πόσους χάνει είτε για τη διέλευση με κόκκινο φανάρι, είτε για τη μη στάση πριν από τη διάβαση. Στη μέθοδο Start() βρίσκουμε το αντικείμενο της σκηνής στο οποίο πρέπει να κάνουμε αναφορά για τη διεπαφή των φωτεινών σηματοδοτών. Στη συνέχεια ορίζουμε έναν τυχαίο πολλαπλασιαστή ταχύτητας κίνησης στο εύρος 0,2 έως 1 και στη συνέχεια περνάμε αυτή την τιμή στο στοιχείο διεπαφής χρήστη που αναφέραμε νωρίτερα. Έπειτα καλούμε τη συνάρτηση UpdateStates(). Αυτή η συνάρτηση απλώς ενημερώνει τα bool attributes ανάλογα με το ποιο sprite δείχνει το animation της διεπαφής στον παίκτη. Επίσης, καλούμε αυτή τη συνάρτηση στη συνάρτηση Update() επειδή θέλουμε να ελέγχουμε τις καταστάσεις σε κάθε καρέ κατά τη διάρκεια του χρόνου εκτέλεσης. Οι επόμενες δύο μέθοδοι είναι η ενσωματωμένη συνάρτηση της Unity στις OnTriggerEnter2D και OnTriggerExit2D. Με αυτές τις μεθόδους μπορούμε να χειριστούμε τον συγκρουστή που αναθέσαμε στο αντικείμενο που δείχνει που βρίσκεται η διάβαση στο πλακίδιο του χάρτη. Στο OnTriggerEnter2D ελέγχουμε αν συμβαίνει σύγκρουση με κάποιο άλλο αντικείμενο που έχει την ετικέτα "Player". Αν αυτό συμβαίνει και το φως είναι πράσινο, τότε απονέμουμε πόντους στον παίκτη. Διαφορετικά, αφαιρούμε πόντους επειδή το φως είναι κόκκινο ή πορτοκαλί. Στη συνέχεια, στο OnTriggerExit2D απλά απενεργοποιούμε το φανάρι της διεπαφής χρήστη καλώντας την προαναφερθείσα μέθοδο SetImageState(bool b) από το αντικείμενο που αναφέρεται στη διεπαφή χρήστη. Περνάμε την τιμή false για να το απενεργοποιήσουμε. Το τελευταίο πράγμα που ολοκληρώνει το σενάριο ως επί το πλείστον είναι το αντικείμενο-παιδί που έχει αυτό το αντικείμενο με την ονομασία Activator. Αυτό το αντικείμενο-παιδί έχει μόνο δύο συστατικά, ένα Box Collider 2D που έχει οριστεί σε isTrigger και ένα μικρό script. Αυτό το αντικείμενο υπάρχει για το σκοπό της ύπαρξης περισσότερων του ενός box collider σε ένα prefab αντικείμενο, καθώς η Unity δεν μπορεί να χειριστεί δύο συστατικά collider στο ίδιο GameObject. Σκοπός αυτού του αντικειμένου είναι η ενεργοποίηση της διεπαφής χρήστη. Έτσι και πάλι στο σενάριο αναφερόμαστε στο GameObject που είναι η διεπαφή χρήστη για τα φανάρια και του αναθέτουμε αυτό το αντικείμενο στη μέθοδο Start(). Στη συνέχεια στη μέθοδο OnTriggerEnter2D καλούμε απλά τη μέθοδο SetImageState(bool b) αλλά αυτή τη φορά περνάμε την τιμή true για να ενεργοποιήσουμε τη διεπαφή χρήστη.



Εικόνα 17. Τα βασικά στοιχεία των φαναριών

Τέλος, επειδή θέλουμε τα φανάρια να εμφανίζονται τυχαία κατά τη διάρκεια της εκτέλεσης, το script που χειρίζεται τη συμπεριφορά του φόντου, αποφασίζει αν θα εμφανίζονται τα φανάρια στο χάρτη. Αυτό αποφασίζεται με την τυχαία δημιουργία ενός αριθμού float μεταξύ 1 και 100. Στη συνέχεια, θέτοντας ένα όριο, ελέγχουμε αν ο αριθμός που παράγεται είναι μεγαλύτερος από το όριο. Σε αυτή την περίπτωση ενεργοποιούμε τα φανάρια και ο παίκτης πρέπει να περάσει από αυτό το σενάριο.

3.2.4 Απαγόρευση Πορείας Δεξιά, Αριστερά ή Ευθεία

Ένα σενάριο που θεωρήσαμε επίσης πολύ σημαντικό είναι η απαγόρευση στροφής. Στην καθημερινότητα κανείς μπορεί να παρατηρήσει πως σε πολλές διασταυρώσεις στους δρόμους υπάρχουν απαγορευτικά σήματα πορείας. Αυτά τα σήματα απαγορεύουν στους οδηγούς την αριστερή και δεξιά στροφή, ή πολλές φορές και την ευθεία πορεία.

Κατά τη διάρκεια που ο παίκτης οδηγεί σε ένα επίπεδο, υπάρχει πιθανότητα να εμφανιστούν διασταυρώσεις. Οι διασταυρώσεις στο παιχνίδι έχουν ειδικά spawn points που τοποθετούνται σε αυτά μηχανισμοί απαγόρευσης πορείας. Εάν ενεργοποιηθεί ο μηχανισμός, κάτι που συμβαίνει τυχαία με ένα ποσοστό πιθανότητας που έχει οριστεί από τον προγραμματιστή, τότε ο μηχανισμός εμφανίζεται στο φόντο με μία εικόνα του σήματος που επιλέχθηκε. Το script συμπεριφοράς του φόντου χειρίζεται τα ποσοστά εμφάνισης απαγόρευσης πορείας, εάν το φόντο είναι μια διασταύρωση. GameMaster script υπάρχει μια λίστα με όλους τους μηχανισμούς που μπορούν να εμφανιστούν πάνω στη διασταύρωση, σε περίπτωση που ο μηχανισμός απαγόρευσης πορείας έχει επιλεγθεί. Όλα τα σήματα που μπορούν να εμφανιστούν, υπάρχουν επίσης μέσα στην εφαρμογή ως prefabs, ώστε να γίνεται η εύκολη κλήση τους όσες φορές θέλουμε. Αυτά τα prefabs των σημάτων περιέχουν μόνο μια εικόνα του σήματος ως Sprite

Renderer component. Έτσι η μηχανή Unity μπορεί να τα εμφανίζει στη σκηνή. Στη συνέχεια, το prefab της διασταύρωσης έχει πλαίσια σύγκρουσης που επιτρέπουν στον παίκτη να στρίψει είτε αριστερά είτε δεξιά. Εάν το πλαίσιο σύγκρουσης ανιχνεύσει την είσοδο του παίκτη σε αυτό και ο αντίστοιχος μηχανισμός μη στροφής είναι ενεργοποιημένος, τότε αυτό σημαίνει ότι ο παίκτης δεν ακολούθησε την πινακίδα και αφαιρούνται 10 πόντοι και ταυτόχρονα ο παίκτης χάνει μια ευκαιρία λάθους. Σε κάθε άλλη περίπτωση ο παίκτης ανταμείβεται με 10 πόντους.

3.2.5 Όριο Ταχύτητας

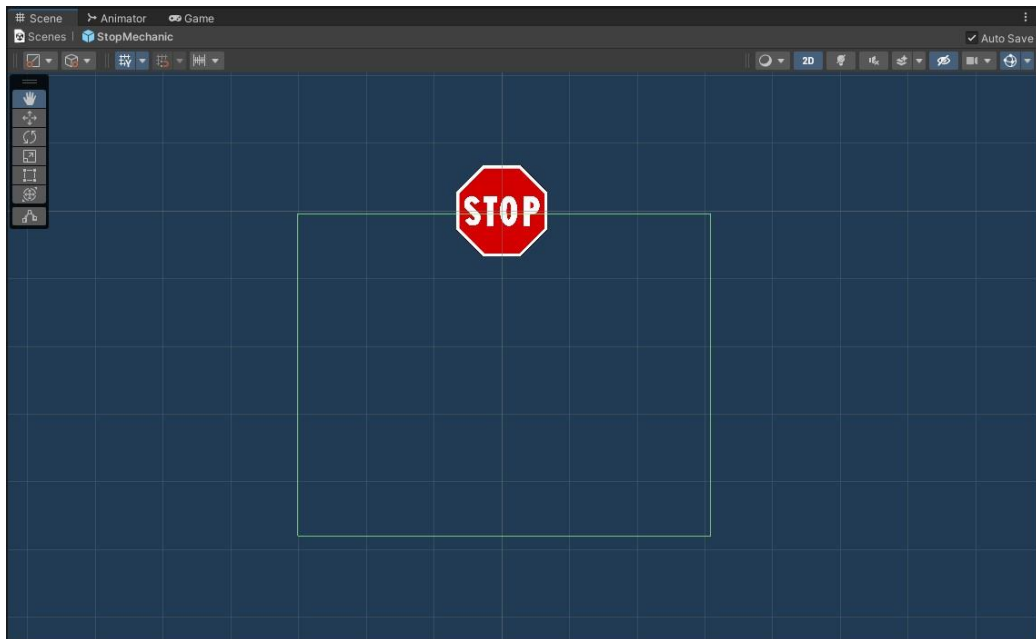
Τα όρια ταχύτητας είναι πολύ σημαντικά στην καθημερινή οδήγηση καθώς υπάρχουν παντού και μέσα και εκτός πόλεων. Η εφαρμογή τους στο παιχνίδι μας είναι επίσης πάρα πολύ απλή.

Το αντικείμενο στη σκηνή εμφανίζεται αφού το BackgroundBehavior αποφασίσει να εμφανίσει όριο ταχύτητας με μια προκαθορισμένη πιθανότητα. Το prefab του ορίου ταχύτητας αποτελείται από το component του Sprite Renderer για να εμφανίζεται η εικόνα στη σκηνή, ένα BoxCollider 2D για τις συγκρούσεις με τον παίκτη, το οποίο βρίσκεται λίγο πιο ψηλά από την πινακίδα, έτσι ώστε ο παίκτης να έχει χρόνο να αντιδράσει, και τέλος ένα script για τη συμπεριφορά του μηχανισμού. Το script περιέχει κάποιες χρήσιμες πληροφορίες για την σωστή λειτουργία του μηχανισμού, όπως οι πόντοι που κερδίζει και χάνει αναλόγως ο παίκτης, αναφορές στο αντικείμενο GameManager, που περιέχει μια λίστα με όλα τα πιθανά όρια ταχύτητας που μπορούν να εμφανιστούν, και ChanceHandler, ώστε να μπορούμε να βγάλουμε ευκαιρίες λάθους που έχει ο παίκτης, σε περίπτωση που δεν περάσει με επιτυχία το όριο ταχύτητας. Το script περιέχει και μια μεταβλητή bool επισημάνει εάν ο παίκτης κατάφερε να περάσει το μηχανισμό αυτόν επιτυχώς. Στη μέθοδο Start() θέτουμε τις αναφορές στα αντικείμενα που χρειαζόμαστε. Έπειτα το script έχει τη μέθοδο OnTriggerEnter2D(Collider2D collision) για να ελέγχουμε τα αντικείμενα που βρίσκει το trigger κατά τη σύγκρουση. Εάν το αντικείμενο είναι ο παίκτης, ελέγχοντας πάντα το tag του αντικειμένου που συγκρούεται με το trigger, τότε ελέγχουμε την ταχύτητα που έχει τη στιγμή που ξεκινά η σύγκρουση. Εάν αυτή η ταχύτητα είναι μεγαλύτερη από αυτήν που έχει δηλωθεί στο prefab του μηχανισμού τότε ο παίκτης χάνει 10 πόντους, αλλιώς ο μηχανισμός έχει περαστεί με επιτυχία και ο παίκτης ανταμείβεται με 15 πόντους. Η συχνότητα με την οποία μπορεί να εμφανίζεται το όριο ταχύτητας ρυθμίζεται από το script συμπεριφοράς του φόντου.

3.2.6 Πινακίδα STOP

Η πινακίδα STOP δουλεύει πολύ παρόμοια με όλους τους παραπάνω μηχανισμούς. Μέσα στην εφαρμογή υπάρχει ένα prefab το οποίο εμφανίζεται στη σκηνή κάθε φορά που το BackgroundBehavior script αποφασίζει τυχαία να εμφανίσει την πινακίδα. Μέσα στο prefab έχουν προκαθοριστεί έναν Sprite Render για τα γραφικά, έναν BoxCollider2D για τον εντοπισμό συγκρούσεων με τον παίκτη και ένα script για τη διαχείριση της συμπεριφοράς του. Το script έχει μια αναφορά στο GameManager και μια μεταβλητή bool που μας δείχνει εάν ο μηχανισμός περάστηκε με επιτυχία από τον παίκτη. Η μέθοδος OnTriggerStay2D(Collider2D collision) ελέγχει σε κάθε καρέ σε πραγματικό χρόνο εάν υπάρχει κάποια σύγκρουση μέσα στο BoxCollider που έχουμε στο αντικείμενο. Ελέγχουμε αν η σύγκρουση γίνεται με τον παίκτη. Σε αυτή τη περίπτωση ελέγχουμε την ταχύτητα του παίκτη. Εάν η ταχύτητα του παίκτη είναι ίση με το μηδέν τότε θεωρούμε πως ο μηχανισμός περάστηκε με επιτυχία. Αλλιώς τη στιγμή που ο παίκτης περάσει το

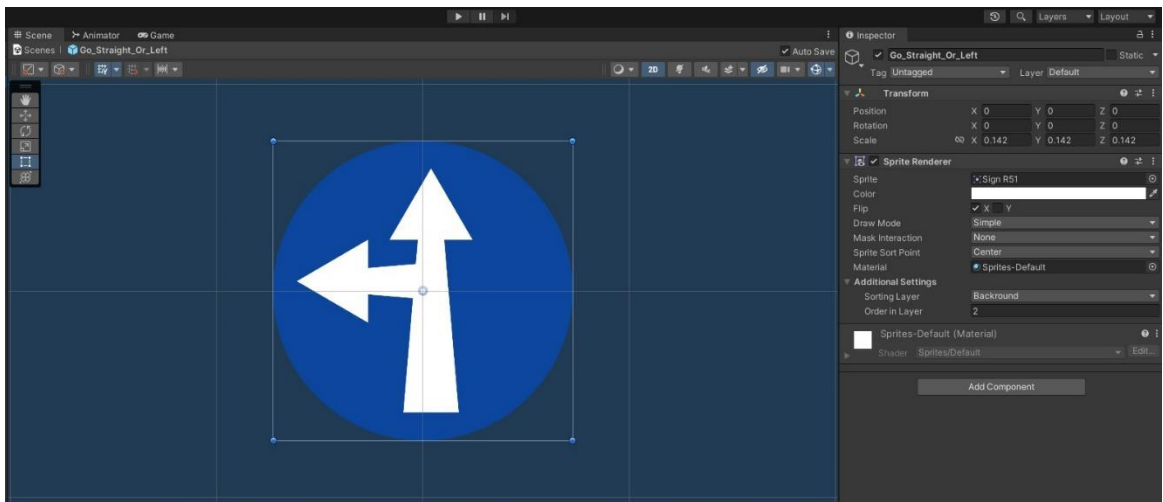
BoxCollider χωρίς να ισχύει η συνθήκη, η μεταβλητή bool δείχνει εξαρχής false οπότε θα θεωρηθεί πως ο παίκτης απέτυχε.



Εικόνα 18. Η πινακίδα STOP με το trigger collider εμφανισμό

3.2.7 Υποχρεωτική πορεία

Η υποχρεωτική πορεία δουλεύει παρόμοια με την απαγόρευση πορείας. Μέσα στην εφαρμογή υπάρχει μια λίστα με όλα τα prefabs που έχουν οριστεί. Συγκεκριμένα υπάρχει ένα prefab για κάθε πιθανό σήμα που συμπεριλάβαμε στην εφαρμογή. Τα σήματα αυτά είναι στροφή αριστερά, στροφή δεξιά, στροφή αριστερά και ευθεία, στροφή δεξιά και ευθεία και ευθεία πορεία. Κάθε prefab δε περιέχει κανένα script. Το μόνο που έχουν είναι η εικόνα του σήματος μέσα σε ένα Sprite Renderer component. Η λίστα που περιέχει όλα τα prefabs έχει οριστεί ως ScriptableObject. Έτσι μπορούμε να χρησιμοποιήσουμε το script κατευθείαν ως αντικείμενο, το οποίο μπορούμε να το περάσουμε στο BackgroundBehaviour. Έπειτα εάν ο δρόμος είναι διασταύρωση και το BackgroundBehaviour αποφασίσει τυχαία να εμφανίσει μια υποχρεωτική διαδρομή, τότε το BackgroundBehaviour διαλέγει τυχαία ένα σήμα υποχρεωτικής διαδρομής και το εμφανίζει στη σκηνή, ως παιδί του αντικειμένου του δρόμου. Η σύγκριση στη στροφή ή στην ευθεία για να ελέγξουμε εάν ο παίκτης έχει πάρει όντως τη σωστή διαδρομή γίνεται ελέγχοντας το όνομα του αντικειμένου του μηχανισμού. Στη περίπτωση που το αντικείμενο του μηχανισμού της διασταύρωσης έχει ένα από τα επιτρεπτά ονόματα της υποχρεωτικής πορείας τότε αναλόγως με τη διαδρομή που πήρε ο παίκτης, μπορούμε να προσθέσουμε ή να αφαιρέσουμε πόντους. Στη περίπτωση που ο παίκτης πάρει τη σωστή διαδρομή, ανταμείβεται με 10 πόντους, αλλιώς χάνει 10 πόντους.



Εικόνα 19. Σήμα που εμφανίζεται στη σκηνή με τα components του στα δεξιά

3.3 Βελτίωση με το MapBox plug-in

Το MapBox είναι μια διαδικτυακή υπηρεσία που παρέχει πολλά ενδιαφέροντα εργαλεία γεωγραφικού εντοπισμού μαζί με άλλες υπηρεσίες όπως πλοήγηση, χάρτες, εργαλεία επεξεργασίας χαρτών και δεδομένα χαρτών. Μεταξύ αυτών των πραγμάτων, μια ακόμη υπηρεσία που παρέχει το MapBox είναι ένα plug-in για το Unity3D, όπου ο προγραμματιστής μπορεί να προσθέσει πλακίδια από χάρτες από όλο τον κόσμο.

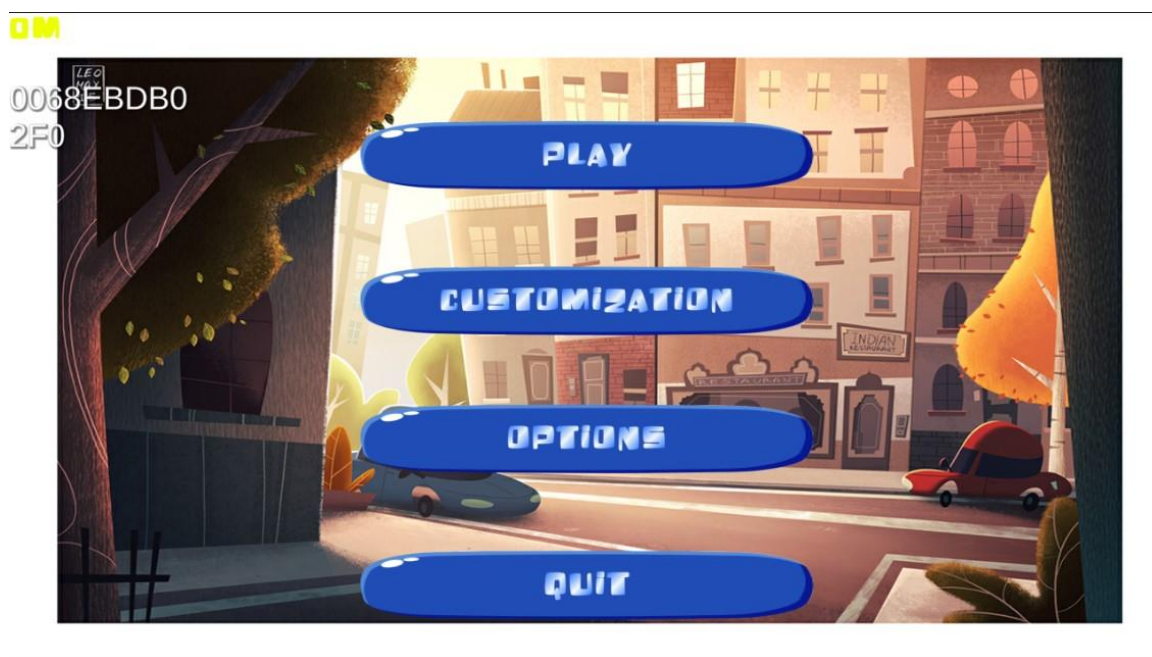
Μια ενδιαφέρουσα βελτίωση που μπορεί να γίνει στο παιχνίδι μας, είναι η προσθήκη πραγματικών τοποθεσιών ως playable maps από πραγματικές πόλεις. Ο παίκτης θα μπορούσε να είναι σε θέση να επιλέξει την πόλη του, και το παιχνίδι θα φόρτωνε ένα τυχαίο ή/και συγκεκριμένο μέρος από την τοποθεσία του και θα το φόρτωνε στο επίπεδο. Στη συνέχεια, ο παίκτης θα μπορούσε να κυκλοφορήσει στην πόλη. Αυτή είναι μια διασκεδαστική και πιο ενδιαφέρουσα εναλλακτική λύση από το να οδηγείται μέσα από τον ίδιο αριθμό προκατασκευασμένων πλακιδίων δρόμων που δημιουργήθηκαν για αυτό το παιχνίδι.

4. Εγχειρίδιο Χρήστη

Πολύ συχνά σήμερα στα ηλεκτρονικά παιχνίδια υπάρχει επιλογή μέσα από την οποία γίνεται ξεκάθαρος στον χρήστη ο τρόπος με τον οποίο παίζεται το παιχνίδι και οι βασικές λειτουργίες του. Η τακτική που χρησιμοποιείται συνήθως στα περισσότερα παιχνίδια είναι να υπάρχει ένα επίπεδο όπου ο παίκτης πειραματίζεται με το παιχνίδι δοκιμάζοντας το και μέσω κειμένου ταυτόχρονα γίνεται μια σύντομη επεξήγηση για το πώς λειτουργεί η εφαρμογή. Ταυτόχρονα υπάρχουν συχνά και παραδείγματα των εμποδίων που θα αντιμετωπίσει ο παίκτης στο μέλλον στο παιχνίδι. Επίσης μπορεί να εξηγούνται την ώρα της δοκιμαστικής χρήσης του παιχνιδιού μόνο αυτά που ο παίκτης βλέπει για πρώτη φορά.

Στην εφαρμογή μας δεν έχει γίνει κάτι παρόμοιο και θα ακολουθήσουμε τον πιο απλό τρόπο. Θα εξηγήσουμε μέσω ενός εγχειριδίου τι θα αντιμετωπίσει ο παίκτης πριν ανοίξει την εφαρμογή.

Αρχικά το πρώτο πράγμα που θα δει ο παίκτης όταν ανοίγει την εφαρμογή είναι το κυρίως μενού με τέσσερις επιλογές: Play, Customization, Options και Quit.



Εικόνα 20. Το κυρίως μενού της εφαρμογής

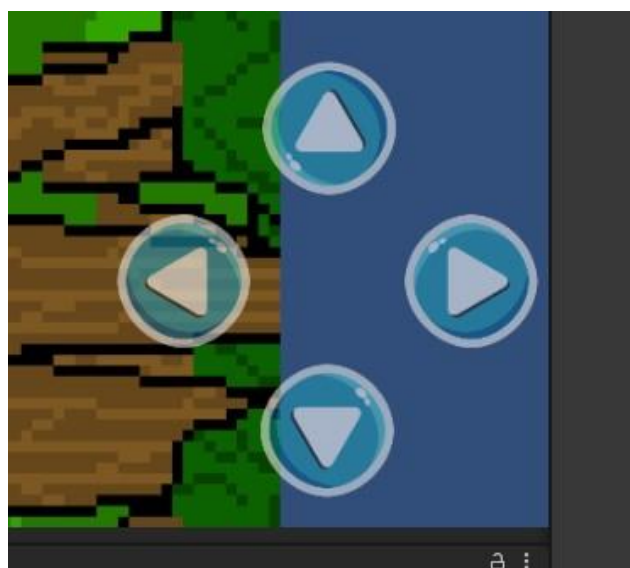
Το κουμπί Play θα μεταφέρει το χρήστη σε μια οθόνη όπου μπορεί να επιλέξει το επίπεδο που θέλει να παίξει. Υπάρχουν 2 επίπεδα μέχρι στιγμής στο παιχνίδι μας: η πόλη και το δάσος. Πατώντας Customization, ο χρήστης μεταφέρεται σε μια οθόνη όπου μπορεί να επιλέξει διάφορα skins για το όχημά του καθώς και διάφορα χρώματα για κάθε όχημα. Αφού, χρησιμοποιώντας τα κουμπάκια στις άκρες, επιλέξει το όχημα και το χρώμα που θέλει, ο χρήστης μπορεί να πατήσει το κουμπί με το σύμβολο tik για να αποθηκεύσει την επιλογή του. Σε κάθε άλλη περίπτωση η επιλογή του χρήστη δεν αποθηκεύεται. Στην επιλογή Options, ο χρήστης μπορεί να αλλάξει τις ρυθμίσεις της εφαρμογής. Για την ώρα αυτό που μπορεί να αλλάξει είναι η ένταση του ήχου. Φυσικά αυτό μπορεί στο μέλλον να επεκταθεί για να γίνεται η αλλαγή περισσότερων ρυθμίσεων.

Με το κουμπί Quit τερματίζεται η εφαρμογή και ο παίκτης μεταφέρετε στην αρχική οθόνη της συσκευής του.



Εικόνα 21. Μενού τροποποίησης οχήματος

Αφού ξεκινήσει ένα επίπεδο ο παίκτης μπορεί να κουνηθεί μέσω τη διεπαφή χρήση που υπάρχει κάτω δεξιά στην οθόνη. Μπορεί να κουνηθεί δεξιά και αριστερά με τα αντίστοιχα βελάκια. Με το πάνω βελάκι ο χρήστης μπορεί να αυξήσει την ταχύτητα με την οποία κινείται το όχημα μέχρι αυτή να φτάσει τη μέγιστη ταχύτητα που έχει το όχημα. Με το κάτω βελάκι ο χρήστης μειώνει την ταχύτητα με την οποία κινείται το όχημα μέχρι αυτό να φτάσει στο μηδέν. Στη περίπτωση που ο παίκτης βρει μια διασταύρωση, μπορεί να κάνει δεξιά και αριστερή στροφή απλά με το να μετακινήσει το όχημά του προς την κατεύθυνση που θέλει, και η στροφή θα εκτελεσθεί.



Εικόνα 22. Τα κουμπιά χειρισμού του οχήματος

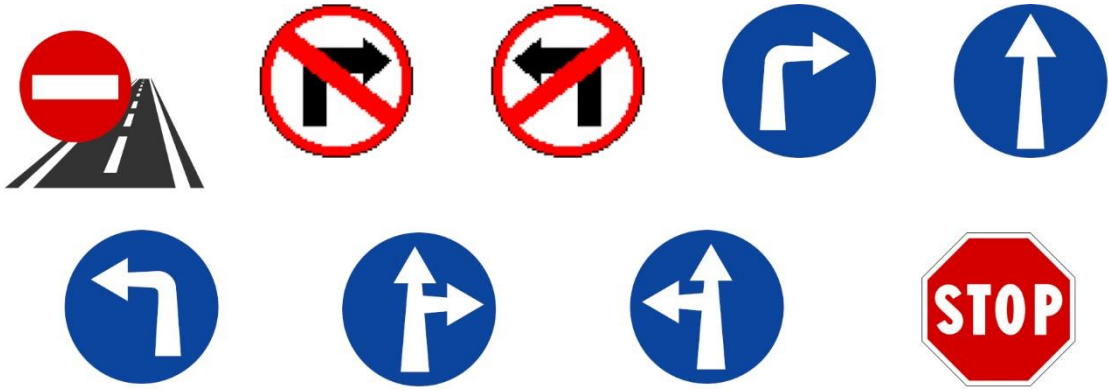
Η μπάρα στο κάτω μέρος της οθόνης γεμίζει όσο ο παίκτης κινείται ευθεία. Η ένδειξη δείχνει το ποσοστό του επιπέδου που έχει ολοκληρώσει ο παίκτης. Σκοπός του παιχνιδιού είναι ο παίκτης να φτάσει μέχρι το τέλος του επιπέδου περνώντας από όλα τα εμπόδια που θα εμφανίζονται στο δρόμο του. Ταυτόχρονα όσο οδηγεί και ξεπερνάει με επιτυχία τα εμπόδια ο παίκτης θα ανταμείβεται με πόντους. Αυτοί οι πόντοι θα μετατρέπονται σε νόμισμα το οποίο μπορεί να αξιοποιηθεί αργότερα. Ο παίκτης επιτρέπεται να κάνει 3 λάθη κατά τη διάρκεια του επιπέδου. Με το 3^ο λάθος το επίπεδο τερματίζει και ο παίκτης χάνει όλους τους πόντους που έχει μαζέψει.

Αρχικά ο παίκτης θα πρέπει να προσέχει να μην οδηγεί με το όχημά του στο πεζοδρόμιο. Κάθε φορά που ο παίκτης ακουμπάει το πεζοδρόμιο με το όχημά του θα χάνει 5 πόντους. Επίσης ο παίκτης θα χάνει 5 πόντους κάθε 3 δευτερόλεπτα που οδηγεί πάνω στο πεζοδρόμιο. Εάν εμφανιστεί το σήμα ορίου ταχύτητας [Εικόνα 8] τότε ο παίκτης θα πρέπει να φροντίσει την ώρα που πλησιάζει το σήμα η ταχύτητά του να είναι μικρότερη ή ίση με την ταχύτητα που δείχνει το σήμα. Σε αυτή τη περίπτωση ανταμείβεται με 10 πόντους, αλλιώς ο παίκτης χάνει 10 πόντους. Υπάρχει η περίπτωση επίσης να υπάρχουν έργα και ρωγμές πάνω στον δρόμο [Εικόνα 9]. Οι ρωγμές μπορεί να εμφανιστούν είτε στη δεξιά είτε στην αριστερή μεριά του δρόμου. Σε αυτή τη περίπτωση ο παίκτης πρέπει να τα αποφύγει. Δεν ανταμείβεται με πόντους αλλά αν αποτύχει ο παίκτης χάνει 10 πόντους και χάνει επίσης και μια ευκαιρία λάθους.

Στην ευθεία μπορεί να υπάρχουν και φανάρια [Εικόνα 6]. Ο παίκτης θα πρέπει να μείνει πίσω από τη διάβαση πεζών μέχρι ο σηματοδότης να δείξει πράσινο. Προσοχή όμως διότι μπορεί να περνάνε και πεζοί. Οι πεζοί όμως εμφανίζονται μόνο μέσα στην πόλη. Σε αυτή τη περίπτωση ο παίκτης θα πρέπει να προσέχει να περάσει ο πεζός πρώτα και μετά να συνεχίσει. Εάν περάσει με πράσινο το φανάρι τότε ανταμείβεται με 15 πόντους. Αλλιώς αν ακουμπήσει τον πεζό ή περάσει με κόκκινο χάνει 10 πόντους και στις δύο περιπτώσεις και χάνει και μια ευκαιρία.

Στις διασταυρώσεις μπορεί να εμφανιστούν οι εξής μηχανισμοί: απαγόρευση πορείας, υποχρεωτική πορεία και πινακίδα STOP.

Όταν εμφανίζεται μια πινακίδα για την απαγόρευση δεξιάς/ αριστερής στροφής ή ευθείας πορείας [Εικόνα 7] τότε ο παίκτης θα πρέπει να πάρει ένα μονοπάτι που δεν έχει αυτό το σήμα. Επιπλέον υπάρχει και η περίπτωση να εμφανιστούν παραπάνω από ένα σήμα στο δρόμο. Τα απαγορευτικά σήματα που μπορούν να εμφανιστούν είναι μέχρι και 2. Αν ο παίκτης πάρει μια λάθος στροφή τότε χάνει 10 πόντους αλλιώς ανταμείβεται με 10 πόντους. Αντίστροφα όταν εμφανίζεται ένα σήμα υποχρεωτικής πορείας [Εικόνα 10] τότε ο παίκτης θα πρέπει αναγκαστικά να πάρει το μονοπάτι που να φαίνεται πάνω στο σήμα. Τα σήματα που μπορεί να εμφανιστούν είναι υποχρεωτική στροφή δεξιά, υποχρεωτική στροφή δεξιά, υποχρεωτική ευθεία διαδρομή, στροφή μόνο δεξιά και ευθεία διαδρομή και στροφή αριστερά και ευθεία. Στη περίπτωση που ο παίκτης δεν ακολουθήσει την υποχρεωτική διαδρομή τότε χάνει 10 πόντους αλλιώς ανταμείβεται με 10 πόντους. Τέλος στη διασταύρωση μπορεί να εμφανιστεί μια πινακίδα STOP [Εικόνα 11]. Τότε ο παίκτης θα πρέπει να σταματήσει πλήρως το όχημά του πριν αποφασίσει πια διαδρομή να πάρει. Τότε ανταμείβεται με 10 πόντους αλλιώς ο παίκτης χάνει 10 πόντους.



Εικόνα 23. Τα σήματα που εμφανίζονται στις διασταυρώσεις

Ο παίκτης μπορεί ανά πάσα στιγμή να κάνει παύση το παιχνίδι και να ανοίξει το μενού παύσης. Από εκεί μπορεί να αλλάξει τις ρυθμίσεις η να τερματίσει το επίπεδο όποτε θέλει. Τερματίζοντας το επίπεδο χωρίς να το τελειώσει κανονικά δε του παρέχει πόντους.

Συμπεράσματα

Οι πρόσφατες εξελίξεις στην αυτοκινητοβιομηχανία και πιο συγκεκριμένα τα προηγμένα συστήματα υποβοήθησης οδηγού προσπαθούν να μειώσουν τα ατυχήματα. Ωστόσο, η πλειονότητα των ατυχημάτων οφείλεται σε ανθρώπινα λάθη. Η εργασία αυτή παρουσιάζει μια εφαρμογή για κινητά τηλέφωνα που επικεντρώνεται σε εκπαιδευτικές πτυχές σχετικά με τις έννοιες της ασφαλούς οδήγησης. Η εφαρμογή προσπαθεί να προωθήσει τις ορθές πρακτικές οδήγησης μέσω του παιχνιδιού υλοποιώντας τα σημαντικότερα σενάρια τροχαίων ατυχημάτων και με τη χρήση του gamification παρέχει ανταμοιβές για την επιτυχή ολοκλήρωση.

Είναι σίγουρο ότι η συγκεκριμένη εφαρμογή θα μπορούσε να επεκταθεί πολύ περισσότερο. Θα μπορούσε να περιλαμβάνει περισσότερες πίστες, πιο πολύ υλικό και περισσότερα σενάρια τα οποία αν ενσωματωνόταν θα διεύρυναν το εκπαιδευτικό εύρος που καλύπτει. Η μελλοντική έρευνα θα ήταν αναμενόμενο να περιλαμβάνει την αξιολόγηση της εφαρμογής με μεγάλο αριθμό χρηστών διαφορετικών ηλικιών. Επιπλέον, οι μελλοντικές μελέτες θα μπορούσαν να διερευνήσουν τη βαθιά μάθηση διαδικαστικής παραγωγής προκειμένου να δημιουργήσουμε δικές μας τρισδιάστατες πόλεις. Τέλος, η μελλοντική έρευνα θα πρέπει να εξετάσει την προσθήκη πραγματικών τοποθεσιών από πραγματικές πόλεις με τη χρήση της διαδικτυακής υπηρεσίας MapBox που παρέχει δεδομένα χαρτών και πινακίδων κυκλοφορίας. Ο χρήστης θα μπορεί να επιλέξει μια πόλη και το παιχνίδι θα φορτώνει ένα τυχαίο ή/και συγκεκριμένο μέρος στο επίπεδο.

Βιβλιογραφία

- [1] T. Rachad, A. Idri και A. Zellou, «Gamified Mobile Applications for Improving Driving Behavior: A Systematic Mapping Study,» *Mobile Information Systems*, τόμ. 2021, pp. 1-24, 2021.
- [2] M. Narkhede και N. Chopade, «Review of Advanced Driver Assistance Systems and Their Applications for Collision Avoidance in Urban Driving Scenario,» σε *International Conference on Machine Learning and Big Data Analytics 2021, 2022*.
- [3] J. F. González-Saavedra , M. Figueroa , S. Céspedes και S. Montejo-Sánchez, «Survey of Cooperative Advanced Driver Assistance Systems: From a Holistic and Systemic Vision,» τόμ. 22, αρ. 8, 2022.
- [4] World Health Organization, «Global Status Report on Road Safety 2018,» World Health Organization, Geneva, 2018.
- [5] M. Sangare, S. Gupta, S. Bouzefrane, S. Banerjee και P. Muhlethaler, «Exploring the forecasting approach for road accidents: Analytical measures with hybrid machine learning,» *Expert Systems with Applications*, τόμ. 167, 2021.
- [6] J. Koivisto, J. Majuri and J. Hamari, “Gamification of education and learning: A review of Empirical Literature,“ in *GamiFIN Conference*, Pori, 2018.
- [7] K. Joanna και H. Juho, «The rise of motivational information systems: A review of gamification research,» *International Journal of Information Management*, τόμ. 45, pp. 191-210, 2019.
- [8] E. Guerra, G. Fornari, W. S. Costa, S. M. Porto, M. P. L. Candia and T. S. Da Silva, “An Approach for Modularizing Gamification Concerns,“ in *International Conference on Computational Science and Its Applications*, Trieste, 2017.
- [9] S. Katie και I. F. Deborah, «Gamification in theory and action: A survey,» *International Journal of Human-Computer Studies*, τόμ. 74, pp. 14-31, 2015.
- [10] A. Mazarakis, *Gamification Reloaded: Current and Future Trends in Gamification Science*, I-com, 2021.
- [11] Hellenic Statistical Authority, «Road Traffic Accidents,» 2020. [Ηλεκτρονικό]. Available: <https://www.statistics.gr/en/statistics/-/publication/SDT04/>.
- [12] E. Trainor-Fogleman, «Unity vs Unreal Engine: Game engine comparison guide for 2021 | Evercast Blog,» 12 July 2021. [Ηλεκτρονικό]. Available: <https://www.evercast.us/blog/unity-vs-unreal-engine>. [Πρόσβαση 25 April 2022].

- [13] H. Singh, «NineHertz.com,» 22 September 2021. [Ηλεκτρονικό]. Available: https://theninehertz.com/blog/unreal-engine-vs-unity-3d-games-development#Whats_Unity. [Πρόσβαση 25 April 2022].
- [14] Unity Technologies, «Unity 2019.3: New features and updates for graphics | Unity,» Unity Technologies, 2019. [Ηλεκτρονικό]. Available: <https://unity.com/releases/2019-3/graphics#high-definition-render-pipeline-hdrp>. [Πρόσβαση 25 April 2022].
- [15] Unity Technologies, «Unity User Manual 2022.1,» Unity Technologies, 2022. [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/2022.1/Documentation/Manual/UnityManual.html>. [Πρόσβαση 25 April 2022].
- [16] Unreal Engine, «Unreal Engine | Features,» Epic Games, [Ηλεκτρονικό]. Available: <https://www.unrealengine.com/en-US/features>. [Πρόσβαση 25 April 2022].
- [17] E. Stone, «community.arm.com,» 18 December 2015. [Ηλεκτρονικό]. Available: <https://community.arm.com/arm-community-blogs/b/graphics-gaming-and-vr-blog/posts/shedding-light-on-global-illumination-techniques>. [Πρόσβαση 16 May 16].
- [18] Epic Games, «Lumen Global Illumination and Reflections,» Epic Games, 2022. [Ηλεκτρονικό]. Available: <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/#:~:text=Lumen%20is%20Unreal%20Engine%205%27s,global%20illum>. [Πρόσβαση 16 May 2022].
- [19] Epic Games, «Profiler Tool Reference,» 2022. [Ηλεκτρονικό]. Available: <https://docs.unrealengine.com/4.27/en-US/TestingAndOptimization/PerformanceAndProfiling/Profiler/>. [Πρόσβαση 16 May 2022].
- [20] Zimbragakis Apostolis, «e-didaskalia.blogspot.com,» March 2017. [Ηλεκτρονικό]. Available: https://e-didaskalia.blogspot.com/2017/03/blog-post_392.html.
- [21] *Driving in Car*, Grand Cayman: Doodle Mobile Ltd., 2016.
- [22] N. Diamantopoulou, «infokids.gr,» 2 October 2020. [Ηλεκτρονικό]. Available: <https://www.infokids.gr/robok-o-k-foititries-eftiaksan-efarmogi-po/>.
- [23] *Safety Driving Simulator: Car*, United Independent Entertainment, 2016.
- [24] M. Zohaib, “Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review,” *Advances in Human-Computer Interaction*, vol. 2018, p. 12, 2018.
- [25] M. Csikszentmihaly, *Flow: The Psychology of Optimal Experience*, Harper & Row, 1990.

- [26] M.-V. Aponte, G. Levieux and S. Natkin, “Scaling the Level of Difficulty in Single Player Video Games,” in *Entertainment Computing - ICEC 2009*, vol. 5709, Berlin, Heidelberg, 2009.

Παράρτημα Κώδικα

Σε περίπτωση που η διατριβή σας περιέχει οποιουδήποτε είδους κώδικα να παρατεθεί εδώ.