



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και ανάπτυξη διαδραστικής εφαρμογής  
επαυξημένης πραγματικότητας (AR) για έξυπνες  
κινητές συσκευές, για την ενίσχυση της εμπειρίας  
τουριστικής περιήγησης στην Καστοριά**

**Πτυχιακή Εργασία**

ΤΟΥ

**Ντίμτσιου Ευάγγελου Μάριου**

(Α.Μ. 2932)

*Επιβλέπων: Νικολάου Σπυρίδων  
Λέκτορας*

*Καστοριά, Φεβρουάριος 2024*





ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και ανάπτυξη διαδραστικής εφαρμογής  
επαυξημένης πραγματικότητας (AR) για έξυπνες  
κινητές συσκευές, για την ενίσχυση της εμπειρίας  
τουριστικής περιήγησης στην Καστοριά**

**Πτυχιακή Εργασία**

ΤΟΥ

**Ντίμτσιου Ευάγγελου Μάριου**

(Α.Μ. 2932)

**Επιβλέπων: Νικολάου Σπυρίδων  
Λέκτορας**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 8<sup>η</sup> Φεβρουαρίου 2024

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

*Καστοριά, Φεβρουάριος 2024*

Copyright © 2024-Ντίμιτριος Ευάγγελος-Μάριος

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 εκφράσω τις ευχαριστίες μου στον κ. Σπυρίδωνα Νικολάου, Λέκτορα του Τμήματος Πληροφορικής, του Πανεπιστημίου Δυτικής Μακεδονίας, επιβλέποντα καθηγητή της παρούσας πτυχιακής εργασίας για την πολύτιμη βοήθειά του και την καθοδήγησή του.

Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου προς όλους τους καθηγητές του Τμήματος Πληροφορικής, του Πανεπιστημίου Δυτικής Μακεδονίας, που συνέβαλαν στην απόκτηση των γνώσεων και δεξιοτήτων στο επιστημονικό πεδίο της Πληροφορικής, που αξιοποίησα για να δημιουργηθεί το συγκεκριμένο έργο.

Τέλος, ευχαριστώ θερμά την οικογένειά μου για την στήριξή της σε εμένα.

## Περίληψη

---

Η συγκεκριμένη πτυχιακή εργασία έχει ως θέμα την ανάπτυξη μίας εφαρμογής επαυξημένης πραγματικότητας (Augmented Reality - AR) με τίτλο “exploreKastoria”, για έξυπνες κινητές συσκευές σε περιβάλλον Android, για την ενίσχυση της εμπειρίας τουριστικής περιήγησης στα αξιοθέατα της πόλης της Καστοριάς.

Για τη δημιουργία της συγκεκριμένης εφαρμογής χρησιμοποιήθηκε η μηχανή ανάπτυξης παιχνιδιών Unity καθώς και το εργαλείο επαυξημένης πραγματικότητας “AR foundation” το οποίο είναι ενσωματωμένο σε αυτήν. Χρησιμοποιήθηκε επίσης, το εργαλείο Mapbox για την ενσωμάτωση του χάρτη πλοήγησης σε πραγματικό χρόνο, με βάση την τοποθεσία του χρήστη. Τέλος, το περιβάλλον ανάπτυξης της εφαρμογής είναι το Android SDK και το Microsoft Visual Studio.

**Λέξεις Κλειδιά:** *Επαυξημένη Πραγματικότητα (AR), Μικτή Πραγματικότητα, Unity 3d, AR Foundation, Mapbox, SDK, API, Microsoft Visual Studio, App, C#*

## Abstract

---

This thesis focuses on the development of an augmented reality (AR) application entitled "exploreKastoria", for smart mobile devices based on Android OS, to enhance the experience of tourist sightseeing in the city of Kastoria.

For the creation of this application, the game development engine Unity was used as well as the tools that are integrated in it such as AR foundation. The Mapbox tool was also used to integrate the real-time navigation map based on the user's location. Finally, the development environment of the application is the Android SDK and Microsoft Visual Studio.

**Key Words:** *Augmented Reality, Mixed Reality, Unity, AR Foundation, Mapbox, SDK, API, Microsoft Visual Studio, App, C#*

# Πίνακας Περιεχομένων

---

ΕΙΣΑΓΩΓΗ.....	1
1 ΕΠΑΥΞΗΜΕΝΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ (AUGMENTED REALITY).....	2
1.1 Τι είναι η Επαυξημένη Πραγματικότητα.....	2
1.2 Ιστορική Αναδρομή.....	3
1.3 Βασικά χαρακτηριστικά συσκευών Επαυξημένης Πραγματικότητας.....	16
1.3.1 Επιταχυνσιόμετρο.....	16
1.3.2 Γυροσκόπιο.....	17
1.3.3 Μαγνητόμετρο.....	17
1.3.4 Global Positioning System (GPS).....	18
1.4 Τύποι Επαυξημένης Πραγματικότητας.....	19
1.4.1 Marker-based Augmented Reality.....	19
1.4.2 Markerless Augmented Reality.....	20
1.4.3 Location-based Augmented Reality.....	20
1.4.4 Superimposition-based Augmented Reality.....	21
1.4.5 Projection-based Augmented Reality.....	22
1.4.6 Outlining Augmented Reality.....	23
1.5 Μικτή Πραγματικότητα.....	24
1.6 Κατηγορίες εφαρμογών Επαυξημένης & Μικτής Πραγματικότητας.....	26
1.6.1 Ιατρική & Υγειονομική Περιθαλψη.....	30
1.6.2 Κατασκευαστικός τομέας.....	31
1.6.3 Στρατιωτικός τομέας.....	32
1.6.4 Βιομηχανία Ηλεκτρονικών Παιχνιδιών και Ψυχαγωγίας.....	34
1.7 Παραδείγματα εφαρμογών Επαυξημένης & Μικτής Πραγματικότητας για τουριστικές περιηγήσεις σε αξιοθέατα.....	35
1.7.1 Google Arts & Culture.....	35
1.7.2 Natural History Museum: The museum at South Kensington.....	37
1.7.3 The Art gallery of Ontario, Toronto.....	37
2 ΛΟΓΙΣΜΙΚΑ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ....	39
2.1 Διαφορές μεταξύ ενός API και ενός SDK.....	39
2.2 Microsoft Visual Studio.....	40
2.3 Mapbox.....	41
2.4 Unity Game Engine.....	43
2.4.1 Unity Game Engine - Βασικά χαρακτηριστικά διεπαφών.....	45



2.4.2	Unity AR foundation .....	49
3	ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	51
3.1	Software Development Life-Cycle .....	51
3.2	Εργαλεία εφαρμογής.....	51
3.3	Περιγραφή της εφαρμογής .....	52
3.4	Δομή της εφαρμογής.....	52
	Αρχική σκηνή και Χάρτης.....	53
	2 <sup>η</sup> σκηνή: Ομαλή μετάβαση σε AR.....	58
	3 <sup>η</sup> σκηνή: AR και MR .....	58
	Συμπεράσματα.....	62
	Βιβλιογραφία .....	64
	Παράρτημα Α. Κώδικας εφαρμογής.....	68

## Λίστα Εικόνων

---

Εικόνα 1. Mixed Reality .....	2
Εικόνα 2. The sword of Damocles .....	4
Εικόνα 3. Caudell and Mizell coining AR in 1992.....	4
Εικόνα 4. Mobile Augmented Reality Systems (MARS).....	5
Εικόνα 5. Mobile Augmented Reality Systems (MARS).....	6
Εικόνα 6. ARToolKit .....	7
Εικόνα 7. Paths as a UI mechanism to guide outdoor users: same path as in (a) and (b), seen outdoors from above .....	8
Εικόνα 8. A game of AR Chess demonstrates collaboration of the stationary user (left) and the mobile user (right) as seen by the "documentation camera" user .....	8
Εικόνα 9. Archeoguide: Avatar athletes competing in the Stadium. ....	9
Εικόνα 10. A user operating a handheld augmented reality unit tracked in an urban environment.....	9
Εικόνα 11. Mara system by Nokia in 2006 .....	10
Εικόνα 12. Wikitude - Augmented Reality Browser .....	10
Εικόνα 13. MapLens in use with a paper map, overlaying digital information on screen. The red square (centre) is used to select markers.....	11
Εικόνα 14. Vision-based methods for AR content indexing: (a)Design QR-Codes, (b) 4D barcodes, (c) ARToolKit, (d) AR storyboard .....	12
Εικόνα 15. SLAM Technology: How companies are using it in AR .....	12
Εικόνα 16. Apple iPad.....	13
Εικόνα 17. Oculus Rift.....	14
Εικόνα 18. Google Glass .....	14
Εικόνα 19. Microsoft HoloLens 2 Development Edition.....	15
Εικόνα 20. Microsoft HoloLens Augmented Reality application in Medicine .....	15
Εικόνα 21. Επιταχυνσιόμετρο .....	16
Εικόνα 21. Γυροσκόπιο.....	17
Εικόνα 23. Μαγνητόμετρο .....	18
Εικόνα 24. Global Positioning System (GPS) .....	18
Εικόνα 25. Marker-based Augmented Reality.....	19
Εικόνα 26. Markerless Augmented Reality .....	20
Εικόνα 27. Location-based Augmented Reality.....	21
Εικόνα 28. Projection-based Augmented Reality.....	22

Εικόνα 29. Projection-based Augmented Reality.....	22
Εικόνα 30. Projection-based Augmented Reality.....	23
Εικόνα 31. Outlining Augmented Reality .....	24
Εικόνα 32. Mixed Reality spectrum .....	24
Εικόνα 33. AR in Healthcare .....	30
Εικόνα 34. AR in construction .....	32
Εικόνα 35. AR in military .....	33
Εικόνα 36. Halo Recruit .....	34
Εικόνα 37. Google Arts & Culture.....	36
Εικόνα 38. Google Arts & Culture 2.....	36
Εικόνα 39. Natural History Museum: The museum at South Kensington.....	37
Εικόνα 40. Natural History Museum: The museum at South Kensington 2.....	37
Εικόνα 41. The art gallery of Ontario, Toronto .....	38
Εικόνα 42. The Art gallery of Ontario, Toronto 2 .....	38
Εικόνα 43. SDK VS API .....	39
Εικόνα 44. Microsoft Visual Studio Logo.....	40
Εικόνα 45. Microsoft Visual Studio Code IDE .....	41
Εικόνα 46. Mapbox Logo .....	41
Εικόνα 47. Mapbox Navigation .....	42
Εικόνα 48. Unity Game Engine logo .....	43
Εικόνα 49. Unity Asset Store’s website.....	43
Εικόνα 50. Unity Editor.....	44
Εικόνα 51. Unity MARS.....	44
Εικόνα 52. Unity new project .....	45
Εικόνα 53. Unity AR Foundation.....	49
Εικόνα 54. Διάγραμμα ροής εφαρμογής “exploreKastoria” .....	52
Εικόνα 55. Αρχική σκηνή .....	53
Εικόνα 56. Πληροφορίες .....	53
Εικόνα 57. Location provider.....	54
Εικόνα 58. Spawn on map .....	54
Εικόνα 59. Location button .....	55
Εικόνα 60. Directions button.....	55
Εικόνα 61. Directions Factory.....	56
Εικόνα 62. Directions screenshots .....	56

Εικόνα 63. Close button .....	57
Εικόνα 64. Changetoar1 trigger script.....	57
Εικόνα 65. Second Scene detection script for change scene .....	58
Εικόνα 66. Image track and myscript .....	58
Εικόνα 67. Qr scan from app screenshot .....	59
Εικόνα 68. ImageClickHandler script for images .....	59
Εικόνα 69. Images from app screenshot .....	60
Εικόνα 70. Info and zoom from app screenshots.....	60

## ΕΙΣΑΓΩΓΗ

---

Η συγκεκριμένη πτυχιακή εργασία έχει ως θέμα την ανάπτυξη μίας εφαρμογής επαυξημένης πραγματικότητας για έξυπνες κινητές συσκευές σε περιβάλλον Android, με σκοπό την ενίσχυση της εμπειρίας τουριστικής περιήγησης στα αξιοθέατα της πόλης της Καστοριάς.

Στη συγκεκριμένη εφαρμογή ο χρήστης καθοδηγείται, μέσω ψηφιακού χάρτη στην έξυπνη κινητή συσκευή του, προς το επιλεγμένο αξιοθέατο και φτάνοντας σε αυτό, με τη βοήθεια της κάμερας της κινητής συσκευής σκανάρει το QR code που υπάρχει στο αξιοθέατο που επισκέπτεται, για να ανακαλύψει περαιτέρω πληροφορίες.

Η εργασία αποτελείται από τρία κεφάλαια, στα οποία καλύπτεται και το θεωρητικό υπόβαθρο, αλλά και το πρακτικό μέρος για τη δημιουργία μίας εφαρμογής που περιέχει λειτουργίες επαυξημένης πραγματικότητας (Augmented Reality - AR).

Στο πρώτο κεφάλαιο παρουσιάζεται η ιστορική αναδρομή με τα στάδια εξέλιξης της επαυξημένης πραγματικότητας, γίνεται αναφορά στις τεχνολογικές έννοιες, στα συστατικά μέρη και στα κύρια χαρακτηριστικά της επαυξημένης πραγματικότητας, στους τύπους των εφαρμογών επαυξημένης πραγματικότητας και στους διάφορους τομείς όπου χρησιμοποιούνται εφαρμογές επαυξημένης πραγματικότητας. Επίσης, παρουσιάζονται αναλυτικά αρκετές εφαρμογές επαυξημένης πραγματικότητας που χρησιμοποιούνται σε μουσεία και αξιοθέατα, οι οποίες χρησιμοποιούνται για την ενίσχυση της εμπειρίας τουριστικής περιήγησης.

Στο δεύτερο κεφάλαιο παρουσιάζονται τα εργαλεία και τα λογισμικά τα οποία χρειάστηκαν για να αναπτυχθεί η εφαρμογή "exploreKastoria" στα πλαίσια της παρούσας πτυχιακής εργασίας. Αναλύονται τα βασικά συστατικά μέρη τους, καθώς και μερικά σχόλια σχετικά με την λειτουργία τους.

Στο τρίτο κεφάλαιο παρουσιάζεται ο κύκλος ζωής για τη δημιουργία της εφαρμογής "exploreKastoria" στα πλαίσια της πτυχιακής εργασίας. Δίνεται το διάγραμμα ροής της εφαρμογής και αναλύεται βήμα-βήμα η πορεία ανάπτυξής της, με τις δυνατότητες επαυξημένης πραγματικότητας που προσφέρονται μέσω της Unity Game Engine.

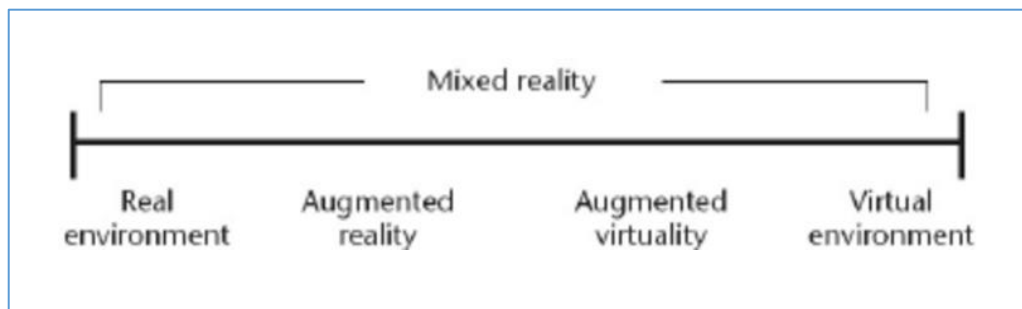
Στο τέλος, αναφέρονται τα συμπεράσματα που προκύπτουν από την εκπόνηση της παρούσας πτυχιακής εργασίας και παρατίθεται η σχετική βιβλιογραφία.

# 1 ΕΠΑΥΞΗΜΕΝΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ (AUGMENTED REALITY)

Με την τεχνολογία να εξελίσσεται οι εφαρμογές Επαυξημένης Πραγματικότητας (Augmented Reality – AR) έχουν μπει στο προσκήνιο για μια νέα και πιο ευχάριστη εμπειρία περιήγησης. Χρησιμοποιώντας τέτοιες εφαρμογές, ο χρήστης έρχεται ένα βήμα πιο κοντά στη Μικτή Πραγματικότητα που συνδυάζει τον πραγματικό κόσμο και τον εικονικό κόσμο, με τη βοήθεια έξυπνων συσκευών.

## 1.1 Τι είναι η Επαυξημένη Πραγματικότητα

Στο συνεχές Πραγματικού και Εικονικού περιβάλλοντος των Milgram and Kishino (1994) (Εικόνα 1), η Επαυξημένη Πραγματικότητα αποτελεί ένα μέρος της γενικής περιοχής της Μικτής Πραγματικότητας. Τόσο τα εικονικά περιβάλλοντα (ή εικονική πραγματικότητα) όσο και η επαυξημένη εικονικότητα, στην οποία τα πραγματικά αντικείμενα προστίθενται στα εικονικά, αντικαθιστούν το περιβάλλον του περιβάλλοντος με ένα εικονικό. Αντίθετα, η Επαυξημένη Πραγματικότητα λαμβάνει χώρα στον πραγματικό κόσμο.



Εικόνα 1. Mixed Reality

Πηγή: [https://www.researchgate.net/figure/Reality-virtuality\\_fig1\\_292150312](https://www.researchgate.net/figure/Reality-virtuality_fig1_292150312)

Σύμφωνα με τους ορισμούς των (Azuma,1997; Azuma et al., 2001), ένα σύστημα Επαυξημένης Πραγματικότητας δεν περιορίζεται σε συγκεκριμένες τεχνολογίες απεικόνισης, όπως είναι η οθόνη που τοποθετείται στο κεφάλι (HMD). Ούτε περιορίζεται στην αίσθηση της όρασης, καθώς η Επαυξημένη Πραγματικότητα μπορεί και ενδεχομένως θα εφαρμόζεται σε όλες τις αισθήσεις, συμπεριλαμβανομένης της ακοής, της αφής και της όσφρησης. Τέλος, η αφαίρεση πραγματικών αντικειμένων με την επικάλυψη εικονικών, προσεγγίσεις γνωστές ως διαμεσολαβημένη ή μειωμένη πραγματικότητα, θεωρείται επίσης Επαυξημένη Πραγματικότητα.

Η Επαυξημένη Πραγματικότητα ενσωματώνει τις πληροφορίες του πραγματικού και του εικονικού κόσμου χρησιμοποιώντας υπολογιστές και άλλες τεχνολογίες για την προσομοίωση των φυσικών πληροφοριών που είναι δύσκολο να βιώσουμε σε ένα συγκεκριμένο χρονικό και χωρικό εύρος του πραγματικού κόσμου, εφαρμόζει τις εικονικές πληροφορίες (σε μορφή κειμένου, ήχου, εικόνων, κινουμένων σχεδίων και

βίντεο) στον πραγματικό κόσμο και γίνεται αντιληπτή από τις ανθρώπινες αισθήσεις, έτσι ώστε να επιτευχθεί η αισθητηριακή εμπειρία πέρα από την πραγματικότητα. [1]

Συνοψίζοντας, ένα σύστημα Επαυξημένης Πραγματικότητας:

- συνδυάζει πραγματικά και εικονικά αντικείμενα σε πραγματικό περιβάλλον
- καταχωρίζει (ευθυγραμμίζει) πραγματικά και εικονικά αντικείμενα μεταξύ τους
- λειτουργεί διαδραστικά, σε τρεις διαστάσεις και σε πραγματικό χρόνο.

Το πραγματικό περιβάλλον και τα εικονικά αντικείμενα τοποθετούνται στην ίδια εικόνα ή χώρο σε πραγματικό χρόνο. Το σύστημα Επαυξημένης Πραγματικότητας έχει αλληλεπίδραση σε πραγματικό χρόνο και οι χρήστες μπορούν να αλληλεπιδράσουν με το σύστημα με πολλές μορφές, δημιουργώντας έτσι μια ισχυρή αίσθηση εμπύθισης. [2]

Κατά τη διερεύνηση των διεπαφών χρήστη, του λογισμικού συστημάτων και των σεναρίων εφαρμογών επαυξημένης πραγματικότητας εστιάζουμε στα ακόλουθα σημεία:

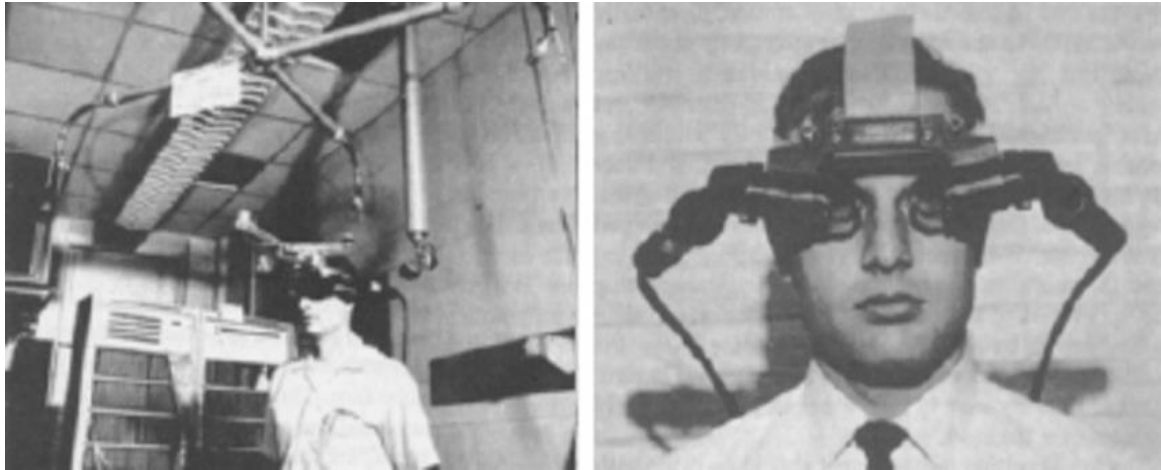
- Προσδιορισμός γενικών εργασιών που ένας χρήστης έξυπνης κινητής συσκευής θα εκτελέσει χρησιμοποιώντας ένα υπολογιστικό σύστημα με επίγνωση περιβάλλοντος
- Καθορισμός ενός ολοκληρωμένου συνόλου επαναχρησιμοποιήσιμων στοιχείων διεπαφής χρήστη για εφαρμογές AR για κινητές συσκευές.
- Συνδυασμένη χρήση διαφορετικών τεχνολογιών οθόνης για την καλύτερη υποστήριξη χρηστών κινητών συσκευών, ταμπλετών ή φορητών (π.χ. γυαλιά AR).

Η Επαυξημένη Πραγματικότητα χρησιμοποιείται ευρέως στους τομείς του στρατού, της ιατρικής, των κατασκευών, της εκπαίδευσης, της μηχανικής, του κινηματογράφου και της τηλεόρασης, της ψυχαγωγίας και σε άλλους τομείς. Με την ανάπτυξη των έξυπνων κινητών τηλεφώνων και των ταμπλετών, καθώς και των φορητών συσκευών, η χρήση των εφαρμογών Επαυξημένης Πραγματικότητας γίνεται όλο και πιο εκτεταμένη.

## 1.2 Ιστορική Αναδρομή

Το πρώτο σύστημα Επαυξημένης Πραγματικότητας δημιουργήθηκε το 1968 από τον επιστήμονα Ivan Sutherland και την ερευνητική ομάδα του στο πανεπιστήμιο του Χάρβαρντ και είχε την ονομασία «The sword of Damocles» [3] (Δαμόκλειος σπάθη).

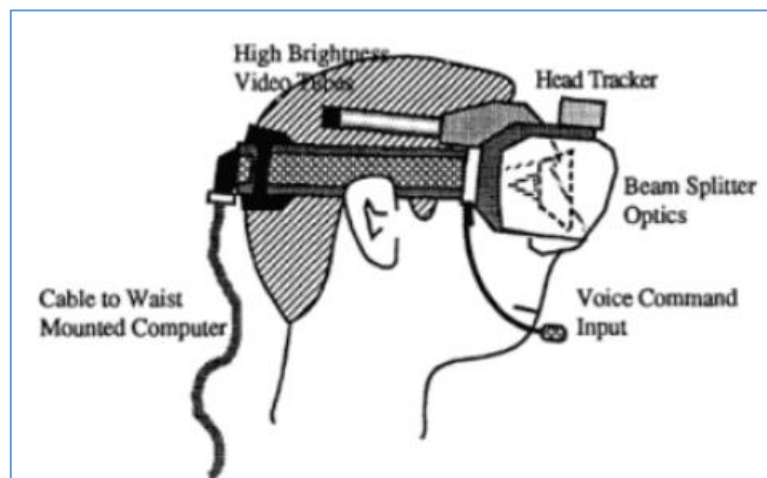
Ήταν η πιο απλή έκδοση της σημερινής συσκευής γυαλιών Εικονικής Πραγματικότητας VR. Χρησιμοποιήθηκε μια οπτική διαφανή οθόνη τοποθετημένη στο κεφάλι του χρήστη που παρακολουθείται από έναν από τους δύο διαφορετικούς ιχνηλάτες έξι βαθμών ελευθερίας (6 Degrees of Freedom - 6DOF): έναν μηχανικό ιχνηλάτη και έναν ιχνηλάτη υπερήχων. Με αυτή τη συσκευή ο χρήστης έβλεπε μια τρισδιάστατη εικονική πραγματικότητα μέσω μίας οθόνης που υπήρχε στο εσωτερικό της. Η πρωτότυπη συσκευή κρεμόταν από ψηλά και ήταν δεμένη στο κεφάλι του χρήστη, ώστε να παρακολουθεί τις κινήσεις του κεφαλιού και να πράττει ανάλογα, δημιουργώντας έτσι την αίσθηση της εναλλακτικής πραγματικότητας. Λόγω της περιορισμένης επεξεργαστικής ισχύος των υπολογιστών εκείνη την εποχή, μόνο πολύ απλά σχέδια wireframe μπορούσαν να εμφανιστούν σε πραγματικό χρόνο.



**Εικόνα 2. The sword of Damocles**

Πηγή: [https://www.researchgate.net/figure/The-worlds-first-head-mounted-display-with-the-Sword-of-Damocles-Sutherland-1968\\_fig2\\_292150312](https://www.researchgate.net/figure/The-worlds-first-head-mounted-display-with-the-Sword-of-Damocles-Sutherland-1968_fig2_292150312)

Το 1992, ο Tom Caudell και ο David Mizell ήταν οι πρώτοι που επινόησαν τον όρο «Augmented Reality» σε μια επιστημονική εργασία [4] για να αναφερθούν στην επικάλυψη υλικού που παρουσιάζεται από υπολογιστή πάνω από τον πραγματικό κόσμο. Οι Caudell και Mizell συζητούν τα πλεονεκτήματα του AR έναντι του VR, όπως η απαίτηση λιγότερης επεξεργαστικής ισχύος, καθώς πρέπει να αποδοθούν λιγότερα pixel. Αναγνωρίζουν επίσης τις αυξημένες απαιτήσεις καταχώρισης προκειμένου να ευθυγραμμιστεί το πραγματικό με το εικονικό. Επιπλέον, ανέπτυξαν την πρώτη εφαρμογή εκπαίδευσης βασισμένη στην Επαυξημένη Πραγματικότητα, για να βοηθήσουν στη συντήρηση στο εργοστάσιο αεροσκαφών της εταιρείας Boeing για τη μείωση του κόστους και τη βελτίωση της αποδοτικότητας. Η πειραματική εφαρμογή AR έδινε στους εκπαιδευόμενους σαφείς οδηγίες μέσω των ειδικών γυαλιών που περιείχαν πληροφορίες για το τι πρέπει να κάνουν σύμφωνα με το σημείο που βρίσκονται και με βελάκια που τους κατεύθυναν για να κάνουν μία συγκεκριμένη εργασία.



**Εικόνα 3. Caudell and Mizell coining AR in 1992**

Πηγή: <https://www.researchgate.net/publication/275974448> The History of Mobile Augmented Reality



Το 1993 οι J. M. Loomis, R. G. Golledge και R. L. Klatzky [5] ανέπτυξαν ένα πρωτότυπο σύστημα πλοήγησης εξωτερικού χώρου με τη βοήθεια φωνητικών εντολών, για άτομα με προβλήματα όρασης. Συνδυάζοντας μια φορητή υπολογιστική συσκευή με έναν διαφορικό δέκτη GPS και μια ηλεκτρονική πυξίδα που φοριέται στο κεφάλι. Η εφαρμογή χρησιμοποιεί δεδομένα από μια βάση δεδομένων GIS (Σύστημα Γεωγραφικών Geographic Information System) και παρέχει βοήθεια πλοήγησης χρησιμοποιώντας μια "ακουστική εικονική οθόνη": οι ετικέτες εκφωνούνται χρησιμοποιώντας έναν συνθέτη ομιλίας και αναπαράγονται σε σωστές θέσεις εντός του ακουστικού χώρου του χρήστη.

Το 1997 δημοσιεύτηκε από τον Steve Feiner και την ερευνητική ομάδα του στο Πανεπιστήμιο Columbia και στο Naval Research Laboratory, η έρευνα για το MARS (Mobile Augmented Reality Systems) [5] με στόχο τη διερεύνηση της συνέργειας δύο πολλά υποσχόμενων πεδίων έρευνας διεπαφής χρήστη:

- Επαυξημένη Πραγματικότητα (AR) στην οποία οι οθόνες 3D χρησιμοποιούνται για την επικάλυψη ενός συνθετικού κόσμου πάνω από τον πραγματικό κόσμο,
- Κινητή Υπολογιστική (Mobile Computing), στην οποία όλο και πιο μικρές και φθηνές υπολογιστικές συσκευές, συνδεδεμένες με ασύρματα δίκτυα, μας επιτρέπουν να χρησιμοποιούμε υπολογιστικές εγκαταστάσεις ενώ περιπλανιόμαστε στον πραγματικό κόσμο.

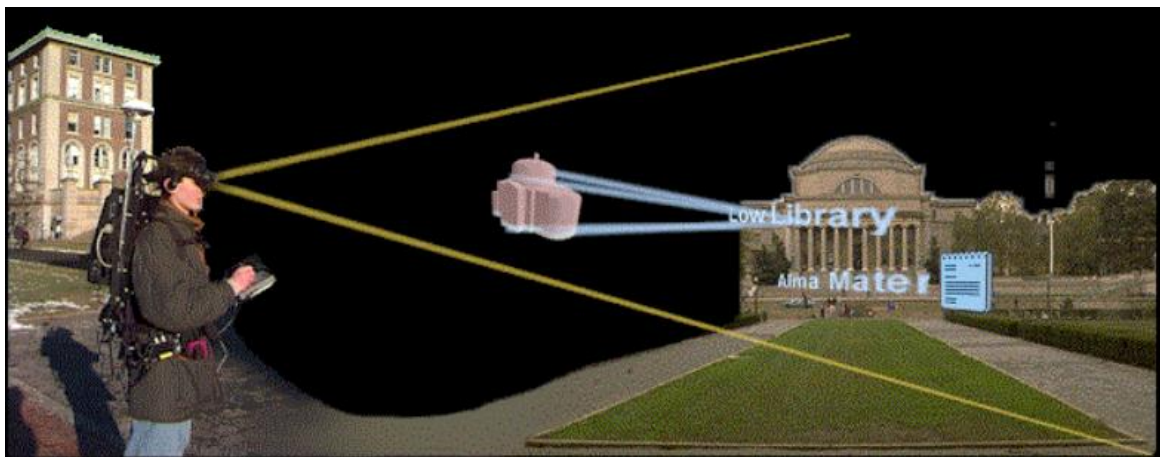
Το MARS αποτέλεσε το πρώτο κινητό σύστημα επαυξημένης πραγματικότητας που μπορούσε να δημιουργήσει έναν διάχυτο 3D χώρο πληροφοριών χρησιμοποιώντας μια διαφανή οθόνη που φοριέται στο κεφάλι με ενσωματωμένο tracker προσανατολισμού διαφορικό GPS και ψηφιακό ραδιοδέκτη για ασύρματη πρόσβαση στο διαδίκτυο κι ένα σακίδιο με έναν φορητό υπολογιστή με γραφίδα και διεπαφή επιφάνειας αφής. Η τεχνολογία παρακολούθησης αναπτύχθηκε από την εταιρεία InterSense και η υποδομή πληροφοριών βασισμένη σε πομποδέκτη υπερύθρων από την εταιρεία Eyeled GmbH.



**Εικόνα 4. Mobile Augmented Reality Systems (MARS)**

Πηγή: <https://history.siggraph.org/experience/mobile-augmented-reality-systems-by-feiner/>

Βασισμένο στις αναδυόμενες τεχνολογίες AR της εποχής, το MARS μπορούσε να δημιουργήσει 3D γραφικά με κτήρια και αξιοθέατα και έδινε πληροφορίες για την καθοδήγηση των τουριστών. Καθώς οι συμμετέχοντες που φορούν το σύστημα MARS περπατούν γύρω και κοντά στο σημείο ενδιαφέροντος, παρακολουθούνται από έναν ιχνηλάτη έξι βαθμών ελευθερίας. Οι πληροφορίες που βλέπουν βρίσκονται σε σχέση με το σύστημα συντεταγμένων 3D του σημείου ενδιαφέροντος. Για παράδειγμα, ένα αξιοθέατο μπορεί να περιβάλλεται από εικονικές αναπαραστάσεις σχετικού υλικού. Σε άλλα μέρη του σημείου ενδιαφέροντος, η παρακολούθηση επιτυγχάνεται μέσω ενός συνδυασμού αδρανειακών ιχνηλατών προσανατολισμού κεφαλής και σώματος και ενός ιχνηλάτη θέσης. Σε αυτές τις περιοχές, οι πληροφορίες βρίσκονται σε σχέση με το 3D σύστημα συντεταγμένων του σώματος του χρήστη, αλλά είναι ευαίσθητες στη θέση του χρήστη. Καθώς οι χρήστες μετακινούνται μεταξύ περιοχών του σημείου ενδιαφέροντος, όπου ισχύουν διαφορετικές τεχνολογίες παρακολούθησης, η διεπαφή χρήστη προσαρμόζεται για να χρησιμοποιήσει την καλύτερη διαθέσιμη. Οι πομποί υπερύθρων επιτρέπουν στους συμμετέχοντες να εξερευνήσουν τμήματα του ίδιου χώρου πληροφοριών με τις δικές τους φορητές συσκευές.



**Εικόνα 5. Mobile Augmented Reality Systems (MARS)**

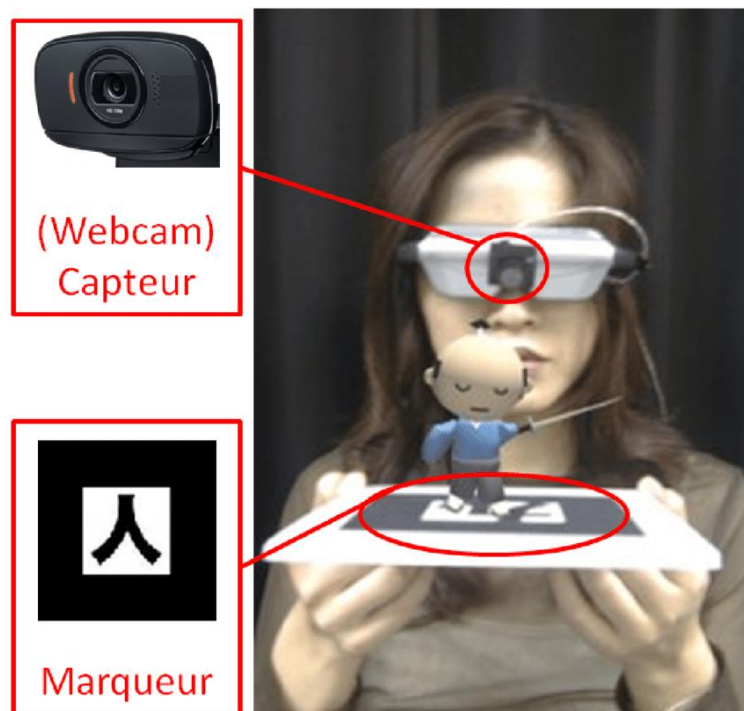
Πηγή: <https://graphics.cs.columbia.edu/projects/mars/mars.gif>

Οι διεπαφές χρήστη MARS ενσωματώνουν τρεις τεχνικές που διερευνούμε για την ανάπτυξη αποτελεσματικών διεπαφών χρήστη επαυξημένης πραγματικότητας: φιλτράρισμα πληροφοριών, σχεδιασμός στοιχείων διεπαφής χρήστη και διαχείριση προβολής. Το φιλτράρισμα πληροφοριών βοηθά στην επιλογή των πιο σχετικών πληροφοριών για παρουσίαση, με βάση δεδομένα σχετικά με το χρήστη, τις εργασίες που εκτελούνται και το περιβάλλον, συμπεριλαμβανομένης της τοποθεσίας του χρήστη.

Ο σχεδιασμός στοιχείων διεπαφής χρήστη καθορίζει τη μορφή με την οποία πρέπει να μεταφερθούν αυτές οι πληροφορίες, με βάση τους διαθέσιμους πόρους προβολής και την ακρίβεια παρακολούθησης. Για παράδειγμα, η απουσία παρακολούθησης θέσης υψηλής ακρίβειας θα ευνοούσε τα σταθεροποιημένα εξαρτήματα σώματος ή οθόνης έναντι των σταθεροποιημένων στον κόσμο που θα έπρεπε να καταχωρηθούν με τα

φυσικά αντικείμενα στα οποία αναφέρονται. Η διαχείριση προβολών προσπαθεί να διασφαλίσει ότι τα εικονικά αντικείμενα που επιλέγονται για εμφάνιση είναι κατάλληλα διατεταγμένα σε σχέση με τις προβολές τους στο επίπεδο προβολής. Για παράδειγμα, τα εικονικά αντικείμενα που δεν περιορίζονται να καταλαμβάνουν μια συγκεκριμένη θέση στον 3D κόσμο θα πρέπει να είναι διατεταγμένα έτσι ώστε να μην εμποδίζουν την προβολή άλλων φυσικών ή εικονικών αντικειμένων στη σκηνή που είναι πιο σημαντικά.

Το 1999, οι Hirokazu Kato και Mark Billinghurst παρουσιάζουν το ARToolKit [6], μια βιβλιοθήκη εντοπισμού θέσεων με έξι βαθμούς ελευθερίας, που χρησιμοποιεί μια προσέγγιση που βασίζεται σε πρότυπα για την αναγνώριση, καθώς και τετραγωνικούς καταπιστευματικούς δείκτες (markers), δηλαδή αντικείμενα που τοποθετούνται στο οπτικό πεδίο ενός συστήματος απεικόνισης που εμφανίζεται στην παραγόμενη εικόνα, για χρήση ως σημείο αναφοράς ή μέτρο. Μπορεί να είναι είτε κάτι τοποθετημένο μέσα ή πάνω στο θέμα της απεικόνισης, είτε ένα σημάδι ή ένα σύνολο σημαδιών στο πλέγμα ενός οπτικού οργάνου. Το ARToolKit είναι διαθέσιμο ως ανοικτού κώδικα με την άδεια GPL και εξακολουθεί να είναι πολύ δημοφιλές στην κοινότητα AR.



Εικόνα 6. ARToolKit

Πηγή: <https://www.researchgate.net/profile/Mahmoud-Haydar/publication/281357543/figure/fig9/AS:997646810484746@1614868943960/La-technologie-ARToolkit.png>

Το 1999, οι Tobias Höllerer et al. [7] παρουσίασαν ένα κινητό σύστημα AR που περιλαμβάνει διεπαφές χρήστη εσωτερικού χώρου (επιφάνεια εργασίας, επιτραπέζια AR και VR στο κεφάλι) για την αλληλεπίδραση με τον εξωτερικό χρήστη [26]. Ενώ οι εξωτερικοί χρήστες βιώνουν μια χωροθετημένη πολυμεσική παρουσίαση πρώτου προσώπου μέσω μιας οθόνης που τοποθετείται στο κεφάλι, οι εσωτερικοί χρήστες μπορούν να αποκτήσουν μια επισκόπηση της εξωτερικής σκηνής. Αυτό ήταν το πρώτο

κινητό σύστημα AR που χρησιμοποίησε RTK GPS και έναν αδρανειακό-μαγνητικό ανιχνευτή προσανατολισμού.



Εικόνα 7. Paths as a UI mechanism to guide outdoor users: same path as in (a) and (b), seen outdoors from above

Πηγή: [https://graphics.cs.columbia.edu/projects/mars/images/AR/CandG/010\\_CarmanPath.jpg](https://graphics.cs.columbia.edu/projects/mars/images/AR/CandG/010_CarmanPath.jpg)

Το 2001, οι Reitmayr και Schmalstieg [8] παρουσιάζουν ένα κινητό, πολυχρηστικό σύστημα AR. Οι ιδέες της κινητής επαυξημένης πραγματικότητας και της συνεργασίας μεταξύ των χρηστών σε επαυξημένο κοινόχρηστο χώρο συνδυάζονται και συγχωνεύονται σε ένα υβριδικό σύστημα. Η επικοινωνία πραγματοποιείται με χρήση LAN και ασύρματου WLAN, όπου κινητοί και σταθεροί χρήστες δρουν σε έναν κοινό επαυξημένο χώρο.



Εικόνα 8. A game of AR Chess demonstrates collaboration of the stationary user (left) and the mobile user (right) as seen by the "documentation camera" user

Πηγή: [https://www.researchgate.net/publication/3928576\\_Mobile\\_Collaborative\\_Augmented\\_Reality](https://www.researchgate.net/publication/3928576_Mobile_Collaborative_Augmented_Reality)

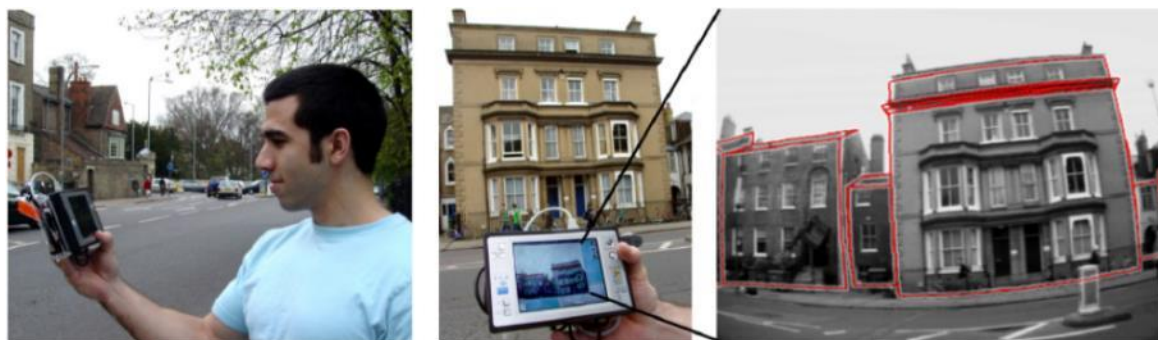
Το 2001, οι Vlahakis et al. [9] παρουσιάζουν το Archeoguide, ένα κινητό σύστημα AR για χώρους πολιτιστικής κληρονομιάς. Το σύστημα περιέχει μια διεπαφή πλοήγησης, τρισδιάστατα μοντέλα αρχαίων ναών και αγαλμάτων, καθώς και άβαταρ που ανταγωνίζονται για τη νίκη στον ιστορικό αγώνα δρόμου στο αρχαίο στάδιο. Ενώ η επικοινωνία βασίζεται στο WLAN, ο ακριβής εντοπισμός πραγματοποιείται με τη χρήση GPS. Στο πλαίσιο του συστήματος μπορεί να χρησιμοποιηθεί μια κλιμακούμενη διάταξη κινητών μονάδων, ξεκινώντας από ένα σύστημα μεγέθους φορητού υπολογιστή με HMD, μέχρι υπολογιστές παλάμης και Pocket PC.



**Εικόνα 9. Archeoguide: Avatar athletes competing in the Stadium.**

Πηγή: [https://www.researchgate.net/publication/220955275\\_ARCHEOGUIDE\\_first\\_results\\_of\\_an\\_augmented\\_reality\\_mobile\\_computing\\_system\\_in\\_cultural\\_heritage\\_sites](https://www.researchgate.net/publication/220955275_ARCHEOGUIDE_first_results_of_an_augmented_reality_mobile_computing_system_in_cultural_heritage_sites)

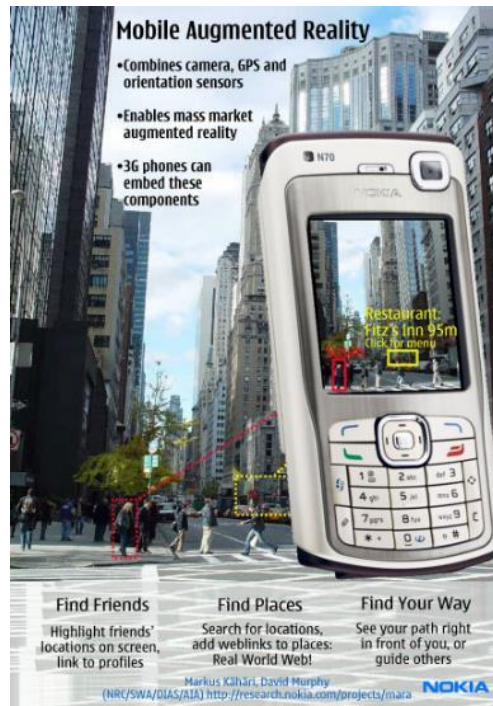
Το 2006, οι Reitmayr και Drummond [10] παρουσιάζουν ένα υβριδικό σύστημα εντοπισμού με βάση το μοντέλο για εξωτερική επαυξημένη πραγματικότητα σε αστικό περιβάλλον, το οποίο επιτρέπει ακριβείς επικαλύψεις σε πραγματικό χρόνο σε φορητή συσκευή. Το σύστημα συνδυάζει έναν ανιχνευτή με βάση τις ακμές για ακριβή εντοπισμό, μετρήσεις γυροσκοπίου για την αντιμετώπιση γρήγορων κινήσεων, μετρήσεις βαρύτητας και μαγνητικού πεδίου για την αποφυγή μετατόπισης, και μια εφεδρική αποθήκη πλαισίων αναφοράς με online επιλογή πλαισίων για αυτόματη επανεκκίνηση μετά από δυναμικές αποκλείσεις ή αποτυχίες.



**Εικόνα 10. A user operating a handheld augmented reality unit tracked in an urban environment.**

Πηγή: [https://www.researchgate.net/publication/221221372\\_Going\\_out\\_Robust\\_model-based\\_tracking\\_for\\_outdoor\\_augmented\\_reality](https://www.researchgate.net/publication/221221372_Going_out_Robust_model-based_tracking_for_outdoor_augmented_reality)

Επίσης το 2006, η NOKIA παρουσιάζει το Mara, μια εφαρμογή AR guidance με πολλούς αισθητήρες για κινητά τηλέφωνα. Η πρωτότυπη εφαρμογή επικαλύπτει τη συνεχή ροή εικόνας που καταγράφει η κάμερα με γραφικά και κείμενο σε πραγματικό χρόνο, σχολιάζοντας το περιβάλλον του χρήστη.



Εικόνα 11. Mara system by Nokia in 2006

Πηγή: <https://techblog.gr/wp-content/uploads/2007/04/Nokia-Mara.jpg>

Το 2008, η Mobilizy εγκαινιάζει το Wikitude, μια εφαρμογή που συνδυάζει δεδομένα GPS και πυξίδας με τα λήμματα της Wikipedia. Το Wikitude World Browser επικαλύπτει πληροφορίες στην προβολή κάμερας σε πραγματικό χρόνο ενός smartphone Android.



Εικόνα 12. Wikitude - Augmented Reality Browser

Πηγή: <https://readyssetrek.com/wp-content/uploads/2013/01/wikitude-browser.jpg>

Επίσης το 2008, με την Επαυξημένη Πραγματικότητα 2.0, οι Schmalstieg κ.ά. παρουσίασαν για πρώτη φορά μια ιδέα που συνδύαζε ιδέες του Web 2.0, όπως τα μέσα κοινωνικής δικτύωσης, το crowd sourcing μέσω της δημόσιας συμμετοχής και μια ανοικτή αρχιτεκτονική για τη σήμανση και τη διανομή περιεχομένου, και την εφάρμοσαν στην επαυξημένη πραγματικότητα για κινητά για να δημιουργήσουν μια κλιμακούμενη εμπειρία AR.

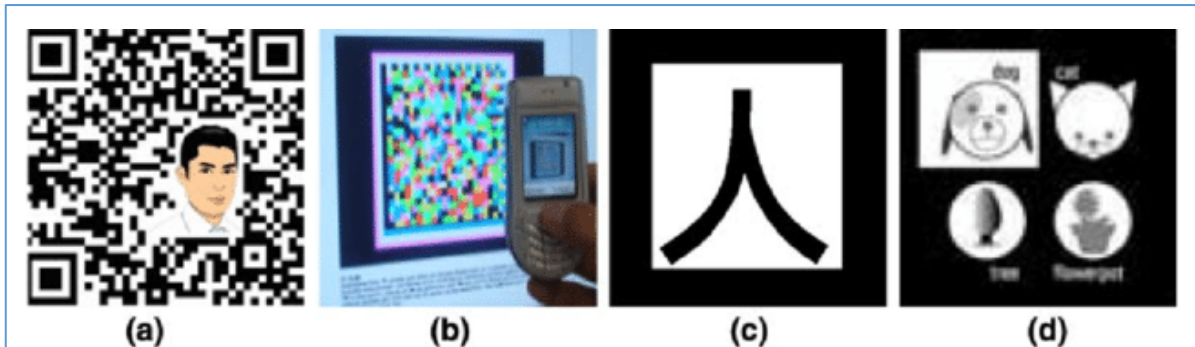
Το 2009, οι Morrison et al. [11] παρουσιάζουν το MapLens, που είναι ένας κινητός χάρτης επαυξημένης πραγματικότητας που επιτρέπει τη διερεύνηση φυσικών χαρτών μέσω της κάμερας του κινητού τηλεφώνου με ψηφιακές πληροφορίες που επικαλύπτονται πάνω από την άποψη. Μέσω του MapLens μπορεί κανείς να δει, για παράδειγμα, φωτογραφίες που συνδέονται με συγκεκριμένη τοποθεσία στο χάρτη ή με τη δική του τοποθεσία. Το MapLens μπορεί να χρησιμοποιηθεί για τη μεταφόρτωση νέου περιεχομένου με γεωγραφική αναφορά στο χάρτη. Κατά την έρευνά τους, πραγματοποίησαν μια ευρεία μελέτη χρηστών με τη μορφή ενός παιχνιδιού που βασίζεται στην τοποθεσία σε εξωτερικούς χώρους. Οι δοκιμές πεδίου δείχνουν ότι οι κύριες δυνατότητες των χαρτών AR έγκεινται στη χρήση τους ως συνεργατικό εργαλείο.



**Εικόνα 13. MapLens in use with a paper map, overlaying digital information on screen. The red square (centre) is used to select markers.**

Πηγή: [https://www.researchgate.net/publication/221519396\\_Like\\_Bees\\_Around\\_the\\_Hive\\_A\\_Comparative\\_Study\\_of\\_a\\_Mobile\\_Augmented\\_Reality\\_Map](https://www.researchgate.net/publication/221519396_Like_Bees_Around_the_Hive_A_Comparative_Study_of_a_Mobile_Augmented_Reality_Map)

Επίσης το 2009, οι Hagbi et al. [12] παρουσίασαν μια προσέγγιση που επιτρέπει την παρακολούθηση της στάσης της κινητής συσκευής στρέφοντάς την προς τα αντικείμενα [6] (βλ. Σχ. 14(β)). Σε αντίθεση με τα υπάρχοντα συστήματα, η προσέγγιση επιτρέπει την παρακολούθηση ενός ευρέος συνόλου επίπεδων σχημάτων, ενώ ο χρήστης μπορεί να διδάξει στο σύστημα νέα σχήματα κατά τη διάρκεια της λειτουργίας δείχνοντάς τα στην κάμερα. Τα μαθημένα σχήματα διατηρούνται στη συνέχεια από το σύστημα σε μια βιβλιοθήκη σχημάτων που επιτρέπει νέα σενάρια εφαρμογών AR όσον αφορά την αλληλεπίδραση με τη σκηνή αλλά και το σχεδιασμό.

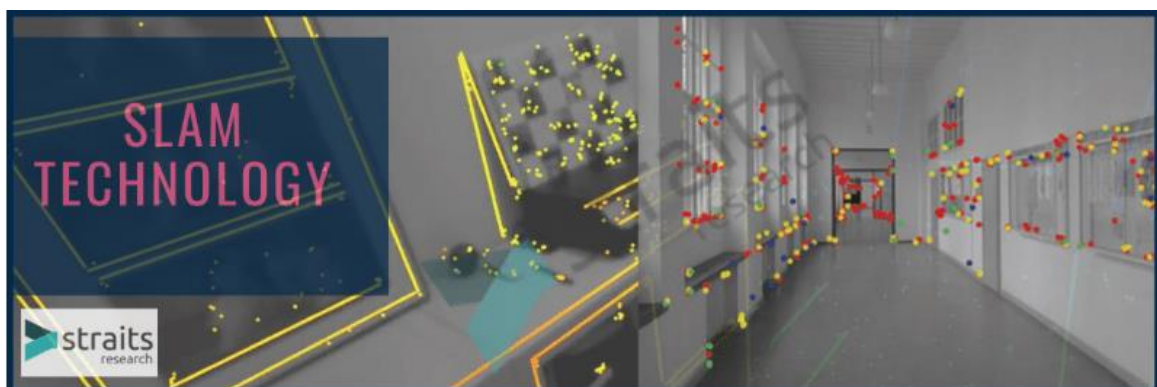


**Εικόνα 14. Vision-based methods for AR content indexing:  
(a)Design QR-Codes, (b) 4D barcodes, (c) ARToolKit, (d) AR storyboard**

Πηγή: [https://www.researchgate.net/publication/221221139\\_In-Place\\_3D\\_Sketching\\_for\\_Authoring\\_and\\_Augmenting\\_Mechanical\\_Systems](https://www.researchgate.net/publication/221221139_In-Place_3D_Sketching_for_Authoring_and_Augmenting_Mechanical_Systems)

Από τα τέλη του 2009 και μετά, η έρευνα και η ανάπτυξη της AR καθοδηγείται γενικά από υψηλές προσδοκίες και τεράστιες επενδύσεις από κορυφαίες εταιρείες παγκοσμίως, όπως η Microsoft, η Google, το Facebook, η Qualcomm και άλλες. Σε γενικές γραμμές, η πρόοδος στις δυνατότητες των κινητών συσκευών εισάγει μια ισχυρή ώθηση προς την κινητή πληροφορική, και η διαθεσιμότητα της επεξεργασίας στο νέφος υποστηρίζει περαιτέρω την πρόταση και την ανάπτυξη λύσεων διακομιστή-πελάτη για σκοπούς AR.

Μια σημαντική τάση που ξεκίνησε γύρω στο 2010, προερχόμενη από την εργασία του Davison το 2003 [13] και αργότερα διερευνήθηκε περαιτέρω από τους Klein και Murray [14], είναι η έντονη χρήση της τεχνολογίας **SLAM (Simultaneous Localization and Mapping)** στην AR, που κατανοεί τον φυσικό κόσμο μέσω χαρακτηριστικών σημείων. Η τεχνολογία SLAM χρησιμοποιείται στις τεχνολογίες όρασης για να λαμβάνονται τα οπτικά δεδομένα από το φυσικό περιβάλλον με τη μορφή σημείων και κουκκίδων για να τροφοδοτούνται τα δεδομένα στις μηχανές. Η SLAM παρέχει οπτική είσοδο για συσκευές και υπολογιστές, κάνοντάς τους να κατανοούν τι συμβαίνει στον φυσικό κόσμο. Τα δεδομένα αυτά βοηθούν επίσης τους προγραμματιστές AR να δημιουργήσουν διαδραστικές και ρεαλιστικές εμπειρίες για το κοινό. Η απλούστερη μορφή της τεχνολογίας SLAM είναι η κατανόηση του δαπέδου, των εμποδίων και των τοίχων.



**Εικόνα 15. SLAM Technology: How companies are using it in AR**

Πηγή: <https://geospatialmedia.s3.amazonaws.com/wp-content/uploads/2020/07/SLAM-Technology.png>



Η τεχνολογία SLAM εξακολουθεί να κυριαρχεί σε μεγάλο μέρος της έρευνας και της ανάπτυξης της AR. Σήμερα, οι τεχνολογικοί κολοσσοί, όπως η Apple, η Google και το Facebook, επενδύουν σημαντικά στην τεχνολογία SLAM για να την χρησιμοποιήσουν με τον καλύτερο δυνατό τρόπο για να προηγηθούν στο τεχνολογικό τους παιχνίδι. Οι προηγμένες τεχνολογίες SLAM, όπως το ARKit της Apple, ή το Google Tango, δημιουργούν έναν ιστό του περιβάλλοντος σε πραγματικό χρόνο και μας ενημερώνουν για το δάπεδο, τους τοίχους και τα αντικείμενα στο περιβάλλον επιτρέποντας σε όλα γύρω μας να λειτουργούν ως ένα στοιχείο που μπορεί να αλληλεπιδράσει. Η τεχνολογία SLAM μπορεί να χρησιμοποιηθεί σε διάφορα σενάρια, όπως αυτοκινούμενα αυτοκίνητα, παιχνίδια, ρομποτική, τεχνητή νοημοσύνη και επαυξημένη πραγματικότητα.επιτρέπει στις εφαρμογές AR να αναγνωρίζουν τρισδιάστατα αντικείμενα και σκηνές, καθώς και να παρακολουθούν άμεσα τον κόσμο και να επικαλύπτουν ψηφιακές διαδραστικές επαυξήσεις.

Το 2010, η Apple κυκλοφορεί το **iPad**, το οποίο γίνεται ο πρώτος υπολογιστής-ταμπλέτα που υιοθετείται από το ευρύ κοινό. Το iPad διαθέτει υποβοηθούμενο GPS, επιταχυνσιόμετρα, μαγνητόμετρα, προηγμένο chipset γραφικών (PowerVR SGX535), επιτρέποντας τις δυνατότητες δημιουργίας εφαρμογών AR σε υπολογιστή-ταμπλέτα.



**Εικόνα 16. Apple iPad**

<https://www.idevice.ro/wp-content/uploads/2017/09/iOS-11-iPhone-iPad-Realitate-Augmentata.jpg>

Το 2011, η Qualcomm ανακοινώνει την κυκλοφορία του SDK της πλατφόρμας AR. Εκείνη τη στιγμή ονομάζεται QCAR, το οποίο αργότερα θα ονομαστεί **Vuforia**.

Το 2012, κυκλοφόρησε το κιτ ανάπτυξης **Oculus Rift**, μια σειρά γυαλιών εικονικής πραγματικότητας ανεπτυγμένα και κατασκευασμένα από την Oculus VR. Αυτό ξεκίνησε μια νέα διαφημιστική εκστρατεία στην εικονική πραγματικότητα και στην ανάπτυξη περισσότερων head-mounted displays, κυρίως για σκοπούς gaming. Τον Μάρτιο του 2014, η Oculus αγοράστηκε από το Facebook για 2 δισεκατομμύρια δολάρια.



Εικόνα 17. Oculus Rift

Πηγή: <https://static.techspot.com/images/products/2019/audio-video/org/2019-05-02-product-4.jpg>

Το 2013, η Google κυκλοφορεί το **Google Glass** (επίσης γνωστό ως Google Project Glass). Το Google Glass είναι ένα οπτικό Head-Mounted Display (HMD) που μπορεί να ελεγχθεί με έναν ενσωματωμένο αισθητήρα αφής ή με εντολές φυσικής γλώσσας. Μετά τη δημόσια ανακοίνωσή του, το Google Glass είχε μεγάλο αντίκτυπο στην έρευνα, αλλά ακόμη περισσότερο στην αντίληψη του κοινού για την τεχνολογία μικτής πραγματικότητας.



Εικόνα 18. Google Glass

Πηγή: <https://www.androidauthority.com/wp-content/uploads/2022/05/Google-Glass-Enterprise-Edition-2-official-1000w-563h.jpg.webp>

Το 2016 η Microsoft κυκλοφορεί στην αγορά το κιτ ανάπτυξης μικτής πραγματικότητας **Microsoft HoloLens**, [15] ένα σετ κεφαλής που συνδυάζει AR και VR που εκτελούν την πλατφόρμα Windows Mixed Reality στο πλαίσιο του λειτουργικού συστήματος Windows 10 και γρήγορα κέρδισε δημοτικότητα. Το HoloLens μπορεί να εντοπίσει την καταγωγή του στο Kinect, ένα πρόσθετο για την κονσόλα παιχνιδιών Xbox της Microsoft που εισήχθη το 2010. Ουσιαστικά είναι γυαλιά που εφάπτονται σε όλο το κεφάλι για να αναγνωρίζει την κίνηση του κεφαλιού, αλλά περιέχει και εξειδικευμένα ακουστικά για μία πιο ρεαλιστική εμπειρία.

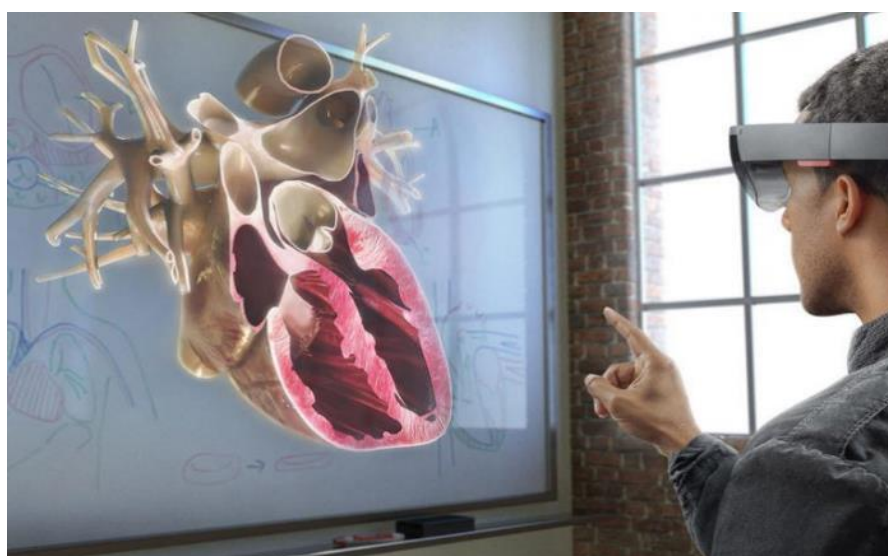
Το Microsoft HoloLens διαθέτει πληθώρα οπτικών αισθητήρων, δύο σε κάθε πλευρά για την περιφερειακή ανίχνευση της "κατανόησης του περιβάλλοντος", μια κύρια κάμερα βάθους με κατεύθυνση προς τα κάτω για τη λήψη των κινήσεων των χεριών και εξειδικευμένα ηχεία που προσομοιώνουν τον ήχο από οποιοδήποτε σημείο του

δωματίου. Το HoloLens διαθέτει επίσης αρκετά μικρόφωνα, μια κάμερα HD, έναν αισθητήρα φωτός περιβάλλοντος και μια προσαρμοσμένη από τη Microsoft "μονάδα ολογραφικής επεξεργασίας", η οποία ισχυρίζεται ότι έχει μεγαλύτερη επεξεργαστική ισχύ από τον μέσο φορητό υπολογιστή. Όλα αυτά μαζί ανιχνεύουν τον χωρικό προσανατολισμό της μονάδας στο δωμάτιο, παρακολουθούν τους τοίχους και τα αντικείμενα στο δωμάτιο και συνδυάζουν τα ολογράμματα με το περιβάλλον. Χρησιμοποιώντας πολλαπλούς αισθητήρες, προηγμένα οπτικά συστήματα και ολογραφική επεξεργασία που συνδυάζεται άψογα με το περιβάλλον του, αυτά τα ολογράμματα μπορούν να χρησιμοποιηθούν για την προβολή πληροφοριών, την ανάμειξη με τον πραγματικό κόσμο ή ακόμη και την προσομοίωση ενός εικονικού κόσμου.



**Εικόνα 19. Microsoft HoloLens 2 Development Edition**

Πηγή: <https://www.geeky-gadgets.com/wp-content/uploads/2019/05/Microsoft-HoloLens-2-Development-Edition.jpg>



**Εικόνα 20. Microsoft HoloLens Augmented Reality application in Medicine**

Πηγή: <https://fowmedia.com/wp-content/uploads/2017/04/Microoft-augmented-reality-768x556.jpg>

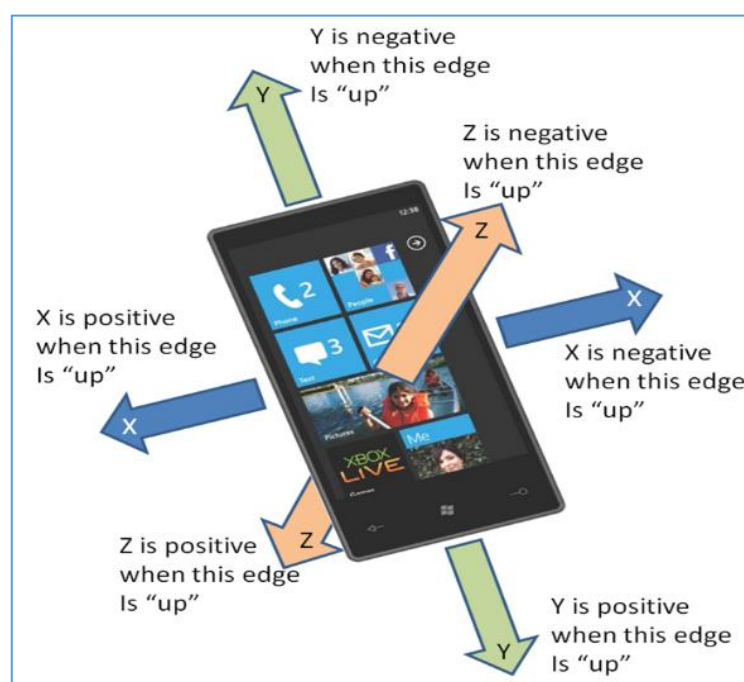
### 1.3 Βασικά χαρακτηριστικά συσκευών Επαυξημένης Πραγματικότητας

Τα βασικά χαρακτηριστικά που πρέπει να έχει μία συσκευή για να λειτουργήσει εφαρμογές επαυξημένης πραγματικότητας είναι η κάμερα, οι αισθητήρες κίνησης, το GPS, καινοτόμο λογισμικό και καλό επεξεργαστή, κάτι που διακρίνεται σε όλες τις καινούργιες συσκευές πλέον.

#### 1.3.1 Επιταχυνσιόμετρο

Το Επιταχυνσιόμετρο (Accelerometer) [16] είναι μία συσκευή μέτρησης της επιτάχυνσης (το ρυθμό μεταβολής της ταχύτητας). Όταν το επιταχυνσιόμετρο εντοπίζει επιτάχυνση, η μάζα υπόκειται σε μια δύναμη που την κάνει να κινηθεί ανάλογα με το περίβλημα μέτρησης. Στη συνέχεια, αυτή η κίνηση μετρείται και μετατρέπεται σε ένα ηλεκτρικό σήμα που μπορεί να ερμηνευθεί από ηλεκτρονικά κυκλώματα. Επίσης, το επιταχυνσιόμετρο μπορεί να μετρήσει και στατικές δυνάμεις, όπως είναι η βαρύτητα. Τα επιταχυνσιόμετρα χρησιμοποιούνται στις έξυπνες κινητές συσκευές για πολλούς και διάφορους λόγους, όπως να ανιχνεύει αλλαγές στον προσανατολισμό και να κάνει την οθόνη να περιστρέφεται.

Αξιοποιείται από πολλές εφαρμογές έξυπνων κινητών συσκευών, όπως για παράδειγμα σε έναν δυναμικό χάρτη, διότι λόγω των αισθητήρων του καταλαβαίνει που βρισκόμαστε την κάθε στιγμή. Παρά την τακτική χρήση του επιταχυνσιόμετρου για παιχνίδια, βίντεο και άλλες δραστηριότητες του smartphone, λίγοι γνωρίζουν πώς λειτουργεί στην πραγματικότητα το gadget ή πώς οι μηχανικοί κατάφεραν να στριμώξουν ένα τόσο μικρό αλλά σημαντικό κομμάτι τεχνολογίας, το οποίο μπορεί να ανιχνεύσει την κίνηση προς τρεις κατευθύνσεις, σε ένα smartphone πάχους χιλιοστών.



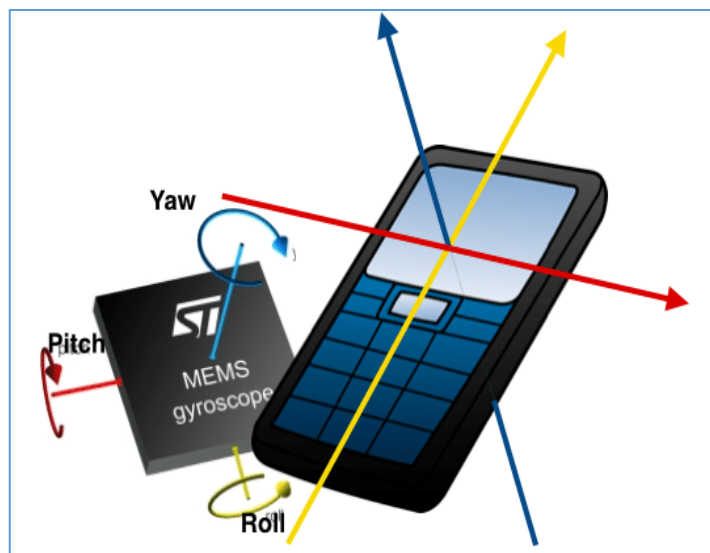
Εικόνα 21. Επιταχυνσιόμετρο

Πηγή: <https://blogs.windows.com/windowsdeveloper/2010/09/08/using-the-accelerometer-on-windows-phone-7/>

### 1.3.2 Γυροσκόπιο

Το Γυροσκόπιο (Gyroscope) [17] χρησιμεύει στην αναφορά μίας κατεύθυνσης, στην σταθερότητα της πλοήγησης και γενικότερα σε ότι αφορά τις κατευθύνσεις. Τα γυροσκόπια σχεδιάζονται με έναν περιστρεφόμενο δείκτη που έχει μια καλά καθορισμένη αξονική κατεύθυνση. Ο δείκτης τοποθετείται με τρόπο που του επιτρέπει να περιστρέφεται ελεύθερα, όταν το γυροσκόπιο βιώνει περιστροφή.

Η εφαρμογή του γυροσκοπίου στις έξυπνες κινητές συσκευές μπορεί να χρησιμοποιηθεί είτε για μία πιο εύκολη πλοήγηση στο λειτουργικό σύστημα, όπως να μετακινήσουμε το κινητό ή το τάμπλετ μας για να κλειδώσουμε την οθόνη, είτε για παιχνίδια που είναι σχεδιασμένα να λειτουργούν με αισθητήρες κίνησης. Για παράδειγμα, ένα παιχνίδι με αυτοκίνητο που όταν μετακινείται η συσκευή κινείται και αυτό αναλόγως. Τέλος, μπορεί να χρησιμοποιηθεί για πλοήγηση σε χάρτες, αντιλαμβανόμενο τότε αλλάζουμε κατεύθυνση.

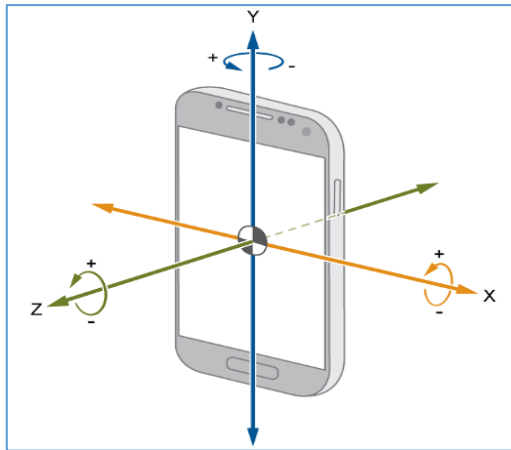


Εικόνα 22. Γυροσκόπιο

Πηγή: [https://www.researchgate.net/figure/Three-axes-of-gyroscope-on-smartphones\\_fig4\\_263285563](https://www.researchgate.net/figure/Three-axes-of-gyroscope-on-smartphones_fig4_263285563)

### 1.3.3 Μαγνητόμετρο

Το Μαγνητόμετρο (Magnetometer) [18] είναι ένας αισθητήρας που μετρά την ισχύ και την κατεύθυνση ενός μαγνητικού πεδίου στους τρεις άξονες (X, Y, Z). Όταν ένα μαγνητόμετρο τοποθετείται σε ένα μαγνητικό πεδίο, οι μαγνητικές δυνάμεις που ασκούνται από αυτό το πεδίο αλληλεπιδρούν με τα εσωτερικά στοιχεία του αισθητήρα. Αυτές οι αλληλεπιδράσεις μπορούν να προκαλέσουν αλλαγές στις ιδιότητες του αισθητήρα που είναι ανιχνεύσιμες και μετρήσιμες. Το Μαγνητόμετρο διαβάζει την ένταση του μαγνητικού πεδίου γύρω από μια έξυπνη κινητή συσκευή. Ο ενσωματωμένος αισθητήρας μαγνητόμετρου στη συσκευή μετρά το μαγνητικό πεδίο κατά μήκος των αξόνων X, Y και Z. Το μαγνητόμετρο στις έξυπνες κινητές συσκευές μπορεί να χρησιμοποιηθεί για εφαρμογές συστημάτων πλοήγησης, των πυξίδων, των ανιχνευτών μετάλλων και πολλών άλλων.

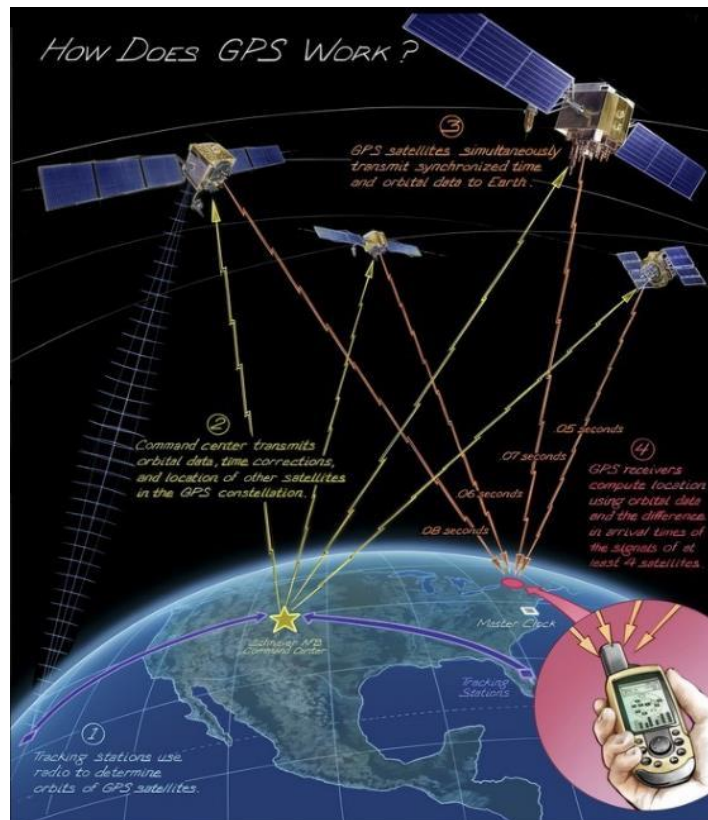


**Εικόνα 23. Μαγνητόμετρο**

Πηγή: <https://www.mathworks.com/help/supportpkg/android/ref/magnetometer.html>

### 1.3.4 Global Positioning System (GPS)

Το Global Positioning System (GPS) [19], ή αλλιώς «Παγκόσμιο σύστημα εντοπισμού θέσης» είναι ένα δίκτυο από δορυφόρους οι οποίοι εκπέμπουν ραδιοκύματα και προσδιορίζουν την ακριβή τοποθεσία ενός αντικειμένου στη Γη. Χρησιμοποιείται κυρίως για συστήματα πλοήγησης. Ορισμένοι δέκτες GPS είναι τόσο ακριβείς που μπορούν να προσδιορίσουν τη θέση τους με ακρίβεια ενός εκατοστού. Οι δέκτες GPS παρέχουν την τοποθεσία σε γεωγραφικό πλάτος, γεωγραφικό μήκος και υψόμετρο. Παρέχουν επίσης την ακριβή ώρα.



**Εικόνα 24. Global Positioning System (GPS)**

Πηγή: <https://timeandnavigation.si.edu/multimedia-asset/how-does-gps-work>

## 1.4 Τύποι Επαυξημένης Πραγματικότητας

Υπάρχουν πολλοί τύποι εφαρμογών Επαυξημένης Πραγματικότητας, κι ο καθένας από αυτούς εξυπηρετεί διαφορετικό σκοπό. Πολλές εταιρείες ανάπτυξης εφαρμογών επαυξημένης πραγματικότητας προωθούν συνεχώς τις εμπειρίες των χρηστών, υιοθετώντας πλούσιους και καινοτόμους τρόπους αναπαράστασης ψηφιακού περιεχομένου. Και, αυτό καθίσταται εφικτό με τη βοήθεια της χρήσης διαφορετικών τύπων επαυξημένης πραγματικότητας.

### 1.4.1 Marker-based Augmented Reality

Η επαυξημένη πραγματικότητα που βασίζεται σε δείκτες (Marker-based AR) αναγνωρίζει τις φυσικές εικόνες, τους "δείκτες" και τα τρισδιάστατα μοντέλα με τη βοήθεια των εφαρμογών AR. Όταν ανοίγει η εφαρμογή επαυξημένης πραγματικότητας, η εφαρμογή χρησιμοποιεί την οπίσθια κάμερα της κινητής συσκευής σας για να εντοπίσει τέτοιους δείκτες.

Αυτός ο τύπος τεχνολογίας επαυξημένης πραγματικότητας είναι επίσης γνωστός ως επαυξημένη πραγματικότητα με βάση την αναγνώριση (Recognition based AR) ή αναγνώριση εικόνας (Image Recognition AR), όπως ένα τοπίο, ή ένα QR code, που σκοπός τους είναι η αναγνώριση της αναφερόμενης εικόνας μέσω της κάμερας και η προσομοίωση αυτής ή η δημιουργία ενός διαφορετικού γραφικού στην οθόνη της συσκευής μας. Με βάση την ανίχνευση των δεικτών, η AR προβάλλει επάλληλο περιεχόμενο με τη μορφή βίντεο, εικόνας, τρισδιάστατων μοντέλων, κλιπ κινουμένων σχεδίων ή σκηνών. Με τη βοήθεια μιας εφαρμογής, οι χρήστες μπορούν να αλληλεπιδρούν με αυτό το περιεχόμενο.

Πρόκειται για έναν από τους πιο δημοφιλείς τύπους επαυξημένης πραγματικότητας που βοηθά τους χρήστες να δουν το αντικείμενο/εικόνα με πιο λεπτομερή τρόπο και από διαφορετικές γωνίες. Επιπλέον, όταν οι χρήστες περιστρέφουν τον δείκτη, το τρισδιάστατο φανταστικό περιστρέφεται ταυτόχρονα.



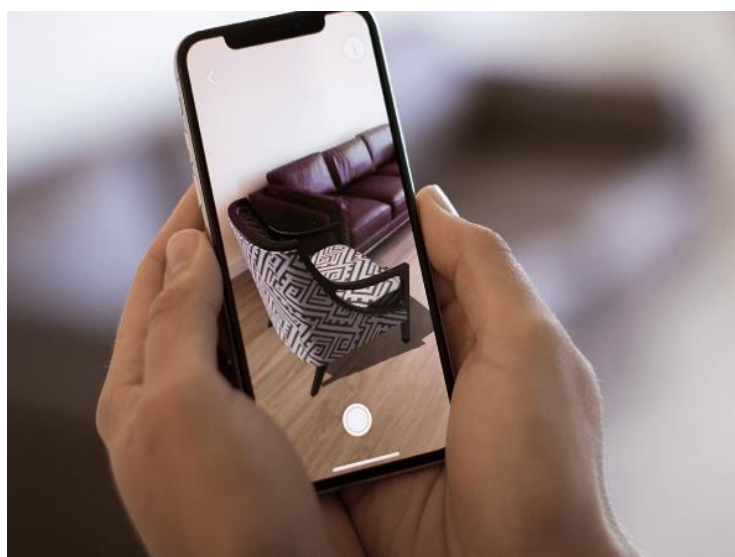
Εικόνα 25. Marker-based Augmented Reality

Πηγή: [https://www.researchgate.net/figure/example-of-marker-based-AR\\_fig1\\_332543647](https://www.researchgate.net/figure/example-of-marker-based-AR_fig1_332543647)

### 1.4.2 Markerless Augmented Reality

Σε αντίθεση με τις εφαρμογές Marker-based AR η επαυξημένη πραγματικότητα χωρίς δείκτες (Markerless AR) λειτουργεί χωρίς κανένα δείκτη, ούτε χρησιμοποιούν εικόνες του πραγματικού κόσμου, αλλά σαρώνουν το περιβάλλον κι αναγνωρίζουν την τοποθεσία μέσω του δέκτη GPS ή του επιταχυνσιόμετρου κι έτσι μπορούν να προσανατολιστούν σχετικά με το χώρο που βρισκόμαστε και να συσχετίσουν τα διάφορα αντικείμενα που υπάρχουν τριγύρω.

Είναι ένας από τους σημαντικότερους τύπους εικονικής πραγματικότητας που βασίζεται στη θέση της συσκευής, την ψηφιακή πυξίδα, την κάμερα και το επιταχυνσιόμετρο για τη συλλογή των πληροφοριών θέσης. Οι εφαρμογές για κινητά που περιέχουν τέτοιες λειτουργίες συνήθως ζητούν από τον χρήστη μια επίπεδη επιφάνεια ή δάπεδο για την τοποθέτηση των αντικειμένων AR, ώστε να μην αιωρούνται πάντα στον αέρα. Η εφαρμογή AR χωρίς δείκτες εντοπίζει αυτή την επίπεδη επιφάνεια και εκτελεί το υπερκείμενο περιεχόμενο πάνω της. Αυτός ο τρόπος είναι πιο χρονοβόρος ως προς την υλοποίηση και με πιο υψηλό κόστος, αλλά είναι και ο πιο δημοφιλής λόγω των πολλών δυνατοτήτων που παρέχει και της εύκολης χρήσης.



**Εικόνα 26. Markerless Augmented Reality**

Πηγή: <https://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/>

### 1.4.3 Location-based Augmented Reality

Αποτελεί έναν από τους πιο εφαρμοσμένους τύπους επαυξημένης πραγματικότητας που χρησιμοποιείται από τους χρήστες. Εξαρτάται κυρίως από το δέκτη GPS, την ψηφιακή πυξίδα, τις κάμερες των smartphone και άλλες τεχνολογίες για τον εντοπισμό της τοποθεσίας. Σε σύγκριση με την AR που βασίζεται σε δείκτες, δεν απαιτεί ειδικούς δείκτες για τον εντοπισμό του τόπου όπου τοποθετείται το εικονικό αντικείμενο. [12] Το ψηφιακό περιεχόμενο χαρτογραφείται σε μια συγκεκριμένη τοποθεσία. Έτσι, όταν ο χρήστης εισέρχεται σε μια προκαθορισμένη τοποθεσία, τα αντικείμενα εμφανίζονται στην οθόνη.



Το AR που βασίζεται στην τοποθεσία δεν χρειάζεται καμία ένδειξη από το αντικείμενο για να αναπτυχθεί, καθώς μπορεί να προβλέψει από την εστίαση του χρήστη να αντιστοιχίσει τα δεδομένα σε πραγματικό χρόνο με την τρέχουσα τοποθεσία. Επιτρέπει επίσης στους προγραμματιστές να προβάλλουν δημιουργικό, διαδραστικό και χρήσιμο ψηφιακό περιεχόμενο σε γεωγραφικά σημεία ενδιαφέροντος στην εφαρμογή. Προσθέτει επίσης οφέλη στους ταξιδιώτες για να γνωρίζουν την τοποθεσία οποιασδήποτε συγκεκριμένης περιοχής χρησιμοποιώντας εικονικά τρισδιάστατα αντικείμενα, βίντεο, κείμενα, συνδέσμους και ήχο.



**Εικόνα 27. Location-based Augmented Reality**

Πηγή: <https://vakoms.com/blog/everything-you-need-to-know-to-build-location-based-ar-app/>

#### **1.4.4 Superimposition-based Augmented Reality**

Με την AR που βασίζεται στην επικάλυψη (Superimposition AR), δημιουργείται μια εναλλακτική προβολή ενός αντικειμένου και μπορεί να χρησιμοποιηθεί για μερική ή πλήρη αντικατάσταση της αρχικής προβολής του αντικειμένου. Αυτό σημαίνει ότι η τεχνολογία αυτή αντικαθιστά είτε ολόκληρη την προβολή ενός υπάρχοντος φυσικού αντικειμένου, είτε ενός τμήματός του, με μια ψηφιακή επαυξημένη προβολή. Με άλλα λόγια, εντοπίζονται συγκεκριμένα αντικείμενα ή χαρακτηριστικά στην προβολή του χρήστη -ίσως ένα εξώφυλλο βιβλίου, μια ετικέτα προϊόντος ή ένα ορόσημο- και στη συνέχεια επικαλύπτεται το σχετικό ψηφιακό περιεχόμενο πάνω στο αντικείμενο ή το χαρακτηριστικό. Σε αυτόν τον τύπο επαυξημένης πραγματικότητας, η αναγνώριση αντικειμένων είναι εξαιρετικά σημαντική. Σε περίπτωση που η εφαρμογή δεν είναι σε θέση να αναγνωρίσει αυτό που βλέπει, θα αποτύχει να αντικαταστήσει την αρχική προβολή με την εικονική. Η AR με επικάλυψη παρέχει πολλαπλές προβολές ενός αντικειμένου-στόχου με επιλογή επισήμανσης των επιπλέον σχετικών πληροφοριών για το εν λόγω αντικείμενο.



**Εικόνα 28. Projection-based Augmented Reality**

Πηγή: <https://www.researchgate.net/publication/331181344/figure/fig4/AS:727773790343169@1550526201051/An-example-of-Superimposition-based-AR-in-a-History-Field-trip.jpg>

Σε φυσικά καταστήματα, η AR που βασίζεται στην επικάλυψη μπορεί να δώσει στους πελάτες οδηγίες και καθοδήγηση. Με την επικάλυψη εικονικών βελών στο περιβάλλον, οι αγοραστές μπορούν να βρουν το δρόμο τους προς τα προϊόντα που αναζητούν. Η AR που βασίζεται στην επικάλυψη μπορεί επίσης να παρέχει στους πελάτες λεπτομέρειες του προϊόντος. Στρέφοντας την κάμερα του smartphone τους σε ένα προϊόν, οι αγοραστές μπορούν να δουν εικονικές επικαλύψεις με λεπτομέρειες όπως η τιμή, τα χαρακτηριστικά και οι κριτικές.

#### **1.4.5 Projection-based Augmented Reality**

Αυτός ο τύπος τεχνολογίας δεν χρησιμοποιείται από τον χρήστη. Η AR με βάση την προβολή (Projection-based AR) είναι μια τεχνική προβολής βίντεο που μπορεί να επεκτείνει ή να αποδώσει ψηφιακά δεδομένα προβάλλοντας εικόνες στην επιφάνεια τρισδιάστατων αντικειμένων ή στο φυσικό χώρο του χρήστη.

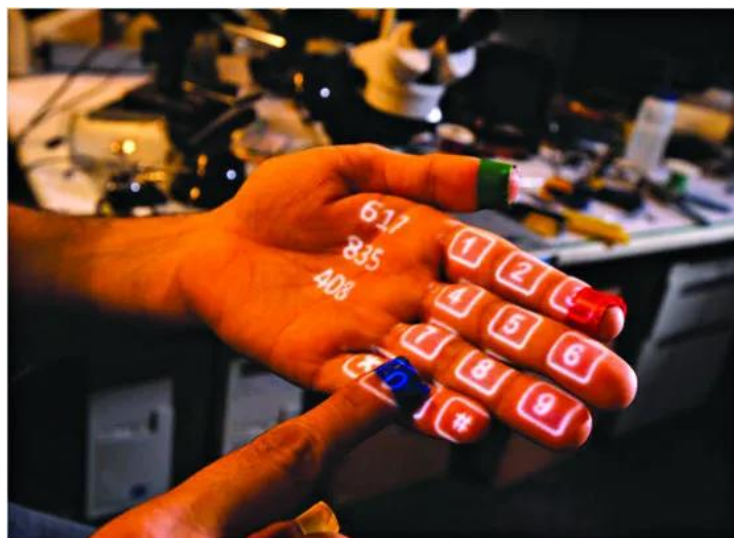


**Εικόνα 29. Projection-based Augmented Reality**

Πηγή: <https://i.ytimg.com/vi/qMGVCMucrsc/maxresdefault.jpg>

Πρόκειται για ένα λογισμικό επαυξημένης πραγματικότητας που δημιουργεί ψευδαισθήσεις με βάση το βάθος, τη θέση και τον προσανατολισμό ενός αντικειμένου, ρίχνοντας τεχνητό φως στις πραγματικές επίπεδες επιφάνειες υπό κατάλληλη καθοδήγηση για την απλοποίηση των πολύπλοκων χειρωνακτικών εργασιών στις επιχειρήσεις. Το καλύτερο είναι ότι εξαφανίζει την ανάγκη για υπολογιστές και οθόνες, καθώς οι οδηγίες τοποθετούνται σε έναν συγκεκριμένο χώρο εργασίας.

Βασίζεται σε προβολείς για την προβολή τρισδιάστατων εικόνων ή ψηφιακού περιεχομένου σε μια επίπεδη δισδιάστατη επιφάνεια, όπως ένας τοίχος, ένα δάπεδο ή ένα αντικείμενο. Δεν δημιουργεί πλήρως καθηλωτικά περιβάλλοντα, αλλά κυρίως ολογράμματα για εκδηλώσεις και ταινίες.



**Εικόνα 30. Projection-based Augmented Reality**

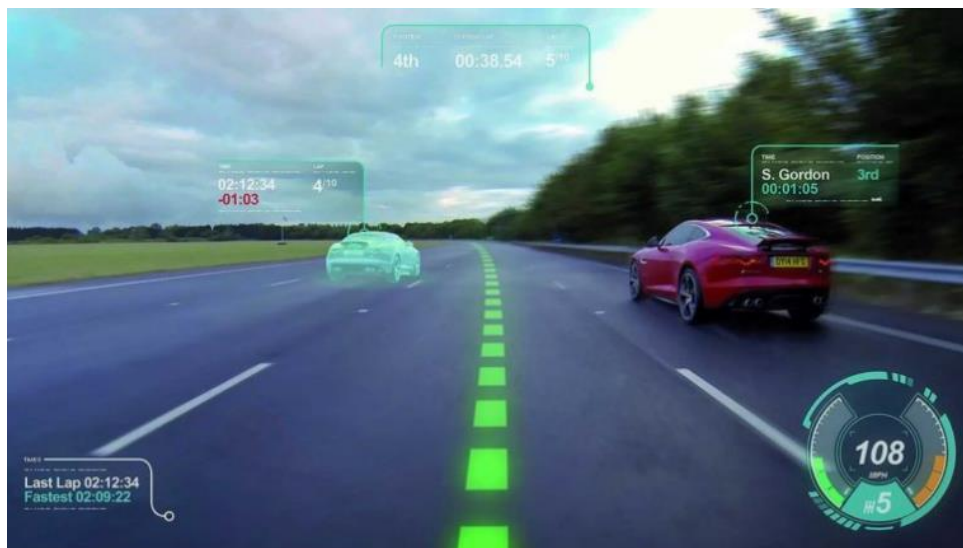
<http://static.digit.in/default/fe396991ff4d4c1f1a5cad8d6941ed17b84b9574.jpeg>

Επιτρέπει στους χρήστες να κινούνται ελεύθερα στο περιβάλλον εντός ενός συγκεκριμένου χώρου όπου τοποθετούνται ο προβολέας και η κάμερα. Με αυτή την τεχνολογία, είναι εύκολο να εκτελεστούν γραφικές αναπαραστάσεις με εικόνες ή βίντεο υψηλής ευκρίνειας που οι συνήθεις τεχνικές φωτισμού δεν μπορούν να δείξουν. Μπορεί επίσης να αλλάξει το οπτικό σχήμα του αντικειμένου με τη ροή του χρόνου.

#### **1.4.6 Outlining Augmented Reality**

Πρόκειται για έναν άλλο τύπο επαυξημένης πραγματικότητας που αναγνωρίζει τις γραμμές και τα όρια με μια ενημερωμένη επαυξημένη προβολή ενός αντικειμένου που τα ανθρώπινα μάτια δεν μπορούν να αναγνωρίσουν. Αν και το ανθρώπινο μάτι είναι γνωστό ότι είναι η καλύτερη κάμερα στον κόσμο, υπάρχουν περιορισμοί. Δεν μπορούμε να κοιτάμε τα πράγματα για πολύ ώρα. Δεν μπορούμε να δούμε καλά σε συνθήκες χαμηλού φωτισμού και το πιο σίγουρο είναι ότι δεν μπορούμε να δούμε τις υπέρυθρες ακτίνες. Οι εφαρμογές επαυξημένης πραγματικότητας που εκτελούν σκιαγράφιση περιγράμματος (Outlining AR) χρησιμοποιούν ειδικές κάμερες για την εκτέλεση τέτοιων ενεργειών σε ορισμένες καταστάσεις.

Αυτή η τεχνολογία χρησιμοποιεί την αναγνώριση αντικειμένων για την ευρύτερη κατανόηση του παρόντος περιβάλλοντος και μπορεί να μοιάζει λίγο με την projection-based AR. Για παράδειγμα, κάθε φορά που παρκάρετε το σύγχρονο αυτοκίνητό σας στο σκοτάδι, η Outlining AR αναγνωρίζει τα όρια του δρόμου και τα περιγράφει για εσάς. Αυτή η μέθοδος μπορεί επίσης να χρησιμοποιηθεί στην αρχιτεκτονική και τη μηχανική για να περιγράψει τα κτίρια και τους πυλώνες στήριξής τους.



**Εικόνα 31. Outlining Augmented Reality**

Πηγή: <http://www.arreverie.com/blogs/wp-content/uploads/2017/05/outline-ar-arreverie-768x432.jpg>

## 1.5 Μικτή Πραγματικότητα

Η Μικτή Πραγματικότητα (Mixed Reality είναι ένας όρος που χρησιμοποιείται για να περιγράψει ένα περιβάλλον χρήστη στο οποίο η φυσική πραγματικότητα και το ψηφιακό περιεχόμενο συνδυάζονται με τρόπο που επιτρέπει την αλληλεπίδραση με και μεταξύ πραγματικών και εικονικών αντικειμένων. Φυσικά και εικονικά αντικείμενα μπορεί να συνυπάρχουν σε περιβάλλοντα μικτής πραγματικότητας και να αλληλεπιδρούν σε πραγματικό χρόνο. Η μικτή πραγματικότητα (MR) είναι Σε αντίθεση με την εικονική πραγματικότητα (VR), η οποία βυθίζει τον τελικό χρήστη σε ένα εντελώς ψηφιακό περιβάλλον, ή την επαυξημένη πραγματικότητα (AR), η οποία τοποθετεί ψηφιακό περιεχόμενο πάνω σε ένα φυσικό περιβάλλον, η μικτή πραγματικότητα συνδυάζει ψηφιακά και πραγματικά περιβάλλοντα.



**Εικόνα 32. Mixed Reality spectrum**

Πηγή: <https://www.researchgate.net/publication/366879982/figure/fig4/AS:1143128111179015@1672924207967/Mixed-Reality-continuum-spectrum.png>

Ο όρος "μικτή πραγματικότητα" εισήχθη σε μια εργασία των Paul Milgram και Fumio Kishino το 1994, "A Taxonomy of Mixed reality Visual Displays". [20] Η εργασία τους διερεύνησε την έννοια της συνέχειας της εικονικότητας και την ταξινόμηση των οπτικών οθονών. Έκτοτε, η εφαρμογή της μικτής πραγματικότητας έχει ξεπεράσει τις οθόνες και περιλαμβάνει:

- Περιβαλλοντική κατανόηση: χωρική χαρτογράφηση και άγκυρες.
- Ανθρώπινη κατανόηση: παρακολούθηση χεριών, παρακολούθηση ματιών και εισαγωγή ομιλίας.
- Χωρικός ήχος.
- Θέσεις και τοποθέτηση σε φυσικούς και εικονικούς χώρους.
- Συνεργασία σε τρισδιάστατα περιουσιακά στοιχεία σε χώρους μικτής πραγματικότητας.

Η μικτή πραγματικότητα αναφέρεται μερικές φορές και ως υβριδική πραγματικότητα ή εκτεταμένη πραγματικότητα (Extended Reality - XR). Μερικές φορές, θεωρείται εσφαλμένα ως ένα είδος επαυξημένης πραγματικότητας, αλλά η ικανότητά της για διαδραστικότητα μεταξύ πραγματικών και ψηφιακών στοιχείων την τοποθετεί πιο μακριά στο συνεχές της εικονικότητας, το οποίο έχει τη φυσική πραγματικότητα στο ένα άκρο και την καθηλωτική εικονική πραγματικότητα στο άλλο.

Αποτελεί λοιπόν, ένα μείγμα φυσικού και ψηφιακού κόσμου, που ξεκλειδώνει φυσικές και διαισθητικές τρισδιάστατες αλληλεπιδράσεις μεταξύ ανθρώπου, υπολογιστή και περιβάλλοντος. Αυτή η νέα πραγματικότητα βασίζεται σε εξελίξεις στην όραση υπολογιστών, την επεξεργασία γραφικών, τις τεχνολογίες απεικόνισης, τα συστήματα εισόδου και το υπολογιστικό νέφος. Ένα σετ κεφαλής ακολουθεί το βλέμμα του χρήστη και χαρτογραφεί το φυσικό περιβάλλον του χρήστη και το λογισμικό χρησιμοποιεί αλγόριθμους βαθιάς μάθησης για να ευθυγραμμίσει το ψηφιακό περιεχόμενο με συγκεκριμένες περιοχές του χάρτη. Ο προγραμματισμός MR επιτρέπει στα ψηφιακά αντικείμενα να αλληλεπιδρούν με τα φυσικά αντικείμενα και στους ανθρώπους να αλληλεπιδρούν με τα ψηφιακά αντικείμενα σαν να είναι φυσικά.

Η μικτή πραγματικότητα γίνεται ολοένα και πιο δημοφιλής για τους καταναλωτές και τις επιχειρήσεις. Μας απελευθερώνει από τις εμπειρίες που είναι συνδεδεμένες με την οθόνη, προσφέροντας ενστικτώδεις αλληλεπιδράσεις με δεδομένα στους χώρους διαβίωσης και με τους φίλους μας. Εκατοντάδες εκατομμύρια διαδικτυακοί χρήστες σε όλο τον κόσμο έχουν βιώσει τη μικτή πραγματικότητα μέσω των φορητών συσκευών τους. Η Mobile AR προσφέρει σήμερα τις πιο διαδεδομένες λύσεις μικτής πραγματικότητας στα μέσα κοινωνικής δικτύωσης. Οι άνθρωποι μπορεί να μην συνειδητοποιούν καν ότι τα φίλτρα AR που χρησιμοποιούν είναι εμπειρίες μικτής πραγματικότητας με εκπληκτικές ολογραφικές αναπαραστάσεις ανθρώπων, ολογραφικά μοντέλα 3D υψηλής πιστότητας και τον πραγματικό κόσμο γύρω τους.

## 1.6 Κατηγορίες εφαρμογών Επαυξημένης & Μικτής Πραγματικότητας

Η επαυξημένη πραγματικότητα (AR) έχει κεντρίσει την προσοχή όλων τα τελευταία χρόνια, από επιχειρηματικούς κολοσσούς μέχρι απλούς καταναλωτές. Υπάρχουν πολλοί τομείς της καθημερινής ζωής όπου χρησιμοποιούνται ποικίλες εφαρμογές Επαυξημένης, Εικονικής ή Μικτής Πραγματικότητας, λόγω των διαφορετικών χαρακτηριστικών και δυνατοτήτων που παρέχουν. Με τη δυνατότητά της να συνδυάζει τον εικονικό κόσμο με δεδομένα του πραγματικού κόσμου, η επαυξημένη πραγματικότητα είναι μια ισχυρή τεχνολογία όταν αξιοποιείται κατάλληλα. Οι πιθανές χρήσεις της επαυξημένης πραγματικότητας είναι ατελείωτες και εφαρμόζονται σχεδόν σε κάθε κλάδο όπως: εκπαίδευση και κατάρτιση, υγειονομική περίθαλψη, κατασκευαστική και κτηματομεσιτική βιομηχανία, ηλεκτρονικά παιχνίδια και ψυχαγωγία, στρατιωτικός τομέας, παραγωγή και μεταποίηση, εφοδιαστική αλυσίδα, λιανικό εμπόριο, διαφήμιση, αθλητισμός, κλπ.

Πιο συγκεκριμένα, η Επαυξημένη Πραγματικότητα χρησιμοποιείται ευρέως από διάφορους τομείς και βιομηχανίες όπως:

- 1. Εκπαίδευση και Κατάρτιση:** Η εκπαίδευση είναι ένας από τους σημαντικότερους τομείς που θα επωφεληθεί από τις τεχνολογίες AR και VR. Η Επαυξημένη Πραγματικότητα παρέχει μια εξαιρετική εμπειρία τόσο για τους μαθητές όσο και για τους εκπαιδευτικούς, που μπορούν να επωφεληθούν από αυτή την τεχνολογία για την ανάπτυξη νέων δεξιοτήτων. Μπορεί να χρησιμοποιηθεί για να επιτρέψει στους εκπαιδευόμενους να αλληλεπιδράσουν με ασφάλεια με πράγματα στα οποία δεν θα είχαν πρόσβαση διαφορετικά, ενώ παραμένουν σε ένα οικείο περιβάλλον, είτε πρόκειται για μια τάξη είτε για εκπαίδευση στην εργασία. Επιπλέον, λόγω της δυνατότητας της επαυξημένης πραγματικότητας να επηρεάσει πολλές πτυχές της εκπαίδευσης, οι εκπαιδευτικοί έχουν ήδη αρχίσει να την χρησιμοποιούν σε αίθουσες διδασκαλίας παγκοσμίως και αναμένεται ότι αυτή η τάση θα συνεχιστεί. Στο επαγγελματικό τοπίο, οι εργοδότες μπορούν να χρησιμοποιήσουν επαυξημένες και εικονικές πραγματικότητες για να βελτιώσουν τις ευκαιρίες εισαγωγής και κατάρτισης των εργαζομένων. Ένα AR Headset μπορεί να καθοδηγήσει έναν εργαζόμενο κατά τη διαδικασία εξέτασης ενός εξοπλισμού, διδάσκοντάς τον για κάθε στοιχείο καθώς προχωράει. Περιβάλλοντα εικονικής πραγματικότητας μπορούν να διδάξουν στους υπαλλήλους πώς να αντιμετωπίζουν μια αγχωτική εμπειρία πελάτη ή να ολοκληρώνουν ορισμένες εργασίες.
- 2. Υγειονομική Περίθαλψη:** Ο τομέας της υγειονομικής περίθαλψης μπορεί να μεταμορφωθεί δραστικά με τη βοήθεια της τεχνολογίας AR και της VR. Η Επαυξημένη Πραγματικότητα έχει πλήθος χρήσεων σε αυτόν τον τομέα, συμπεριλαμβανομένης της υποβοήθησης των γιατρών στην εκτέλεση χειρουργικών επεμβάσεων, στην ιατρική εκπαίδευση, στην περίθαλψη, και πολλά άλλα. Μέσω των κατάλληλων εφαρμογών, οι χειρουργοί και οι εξειδικευμένοι ειδικοί μπορούν να εξασκούν πολύπλοκες διαδικασίες χωρίς να διακινδυνεύουν ακριβούς πόρους ή την άνεση των

ασθενών. Οι φοιτητές που πρέπει να μάθουν πώς λειτουργούν οι διαδικασίες μπορούν επίσης να παρακολουθήσουν έναν χειρουργό να εκτελεί μέσα από ένα AR / VR Headset - το οποίο δίνει πολύ περισσότερες λεπτομέρειες από ό,τι θα έπαιρνες αν στεκόσουν πάνω από τον ώμο κάποιου.

Η εικονική και επαυξημένη πραγματικότητα στην υγειονομική περίθαλψη έχει επίσης αντίκτυπο στο είδος της υποστήριξης των ασθενών που μπορούν να προσφέρουν οι γιατροί και το νοσηλευτικό προσωπικό. Στην εποχή της τηλεϊατρικής, όπου οι ασθενείς πρέπει να αλληλεπιδρούν με τους ειδικούς από απόσταση, η VR και η AR ξεπερνούν τις τηλεδιασκέψεις. Οι γιατροί μπορούν να δουν τους ασθενείς τους και να διαγνώσουν προβλήματα από απόσταση. Οι νοσηλευτές μπορούν να διδάξουν στους ασθενείς πώς να εκτελούν δραστηριότητες αυτοφροντίδας χρησιμοποιώντας επικαλύψεις και γραφικά επαυξημένης πραγματικότητας.

- 3. Κατασκευαστικός τομέας:** Η επαυξημένη πραγματικότητα είναι ένα κρίσιμο εργαλείο στον κατασκευαστικό κλάδο από το στάδιο του σχεδιασμού μέχρι την πραγματική κατασκευαστική διαδικασία. Η AR και η VR μπορούν να κάνουν τον κλάδο αυτό πιο αποτελεσματικό και παραγωγικό. Οι αρχιτέκτονες, οι μηχανικοί και οι εργολάβοι κατασκευαστικών έργων είναι ενθουσιασμένοι με τις δυνατότητες της Επαυξημένης Πραγματικότητας για σκοπούς ψηφιακής μοντελοποίησης και πολλά άλλα. Πολλά αρχιτεκτονικά εργαλεία βοηθούν στην οπτικοποίηση του χώρου, επιτρέποντας στους σχεδιαστές να δημιουργούν πολύ πιο εύκολα τρισδιάστατα μοντέλα σπιτιών, κτιρίων ή έργων υποδομής, τόσο στην εικονική όσο και στην επαυξημένη πραγματικότητα. Στην κατασκευαστική πλευρά της εξίσωσης, η τεχνολογία AR προσφέρει εφαρμογές που κυμαίνονται από την εκπαίδευση των εργαζομένων σχετικά με την ασφάλεια έως την τεχνολογία καταγραφής και παρακολούθησης της προόδου που συγκρίνει άμεσα τα πραγματικά εργοτάξια με τα εικονικά μοντέλα σε πραγματικό χρόνο για να διασφαλίσει ότι δεν αποκλίνουν.
- 4. Κτηματομεσιτική αγορά ακινήτων (Real-Estate):** Ο τομέας των αγοραπωλησιών ακινήτων (Real Estate) βρίσκεται στη λίστα των κορυφαίων τομέων που είναι πιθανότερο να προσελκύσουν τις μεγαλύτερες επενδύσεις σε AR και VR. Με τη βοήθεια της Επαυξημένης Πραγματικότητας ενισχύεται η εμπειρία περιήγησης του υποψήφιου αγοραστή στο φυσικό χώρο του ακινήτου, ενώ η Εικονική Πραγματικότητα μπορεί ακόμη και παρέχει σε έναν υποψήφιο αγοραστή μια εικονική περιήγηση σε ένα ακίνητο χωρίς να απαιτείται να το επισκεφθεί αυτοπροσώπως. Επίσης, με τη χρήση εφαρμογών Επαυξημένης Πραγματικότητας όπως ο κατάλογος AR της IKEA, οι πελάτες που αγοράζουν σπίτια είναι σε θέση να μετρήσουν το εμβαδόν του σπιτιού, να επιλέξουν τα κατάλληλα έπιπλα και οικιακά αντικείμενα χωρίς καν να τα έχουν αγοράσει.
- 5. Βιομηχανία Ηλεκτρονικών Παιχνιδιών:** Μια άλλη αγαπημένη χρήση της Επαυξημένης Πραγματικότητας είναι τα ηλεκτρονικά παιχνίδια. Το AR gaming ενσωματώνει το παιχνίδι με το περιβάλλον του χρήστη σε πραγματικό χρόνο. Πριν

από μερικά χρόνια, το Pokemon Go ήταν μια τεράστια επιτυχία, με πάνω από 250 εκατομμύρια παίκτες το μήνα να μεταφέρονται σε πραγματικές τοποθεσίες από τη σειρά βιντεοπαιχνιδιών. Αυτό έδειξε τις δυνατότητες της AR στους τομείς των παιχνιδιών και της ψυχαγωγίας. Ιδιαίτερα, η επαυξημένη πραγματικότητα έχει βρει μια φυσική στέγη στον τομέα των παιχνιδιών, όπου η τεχνολογία έχει τροφοδοτήσει αρκετές τεράστιες επιτυχίες παιχνιδιών για κινητά.

- 6. Αυτοκινητοβιομηχανία & Μεταφορές:** Η Επαυξημένη Πραγματικότητα στην αυτοκινητοβιομηχανία βοηθά στη βελτίωση της οδηγικής εμπειρίας όσον αφορά τις πληροφορίες για την κυκλοφορία, τη συντομότερη διαδρομή, τα τυφλά σημεία και πολλά άλλα. Χρησιμοποιείται επίσης εκτενώς σε επιδείξεις προϊόντων και εκπαιδευτικές ενότητες, όπως επικαλύψεις εικόνων AR κάτω από το καπό ενός αυτοκινήτου για την εκπαίδευση στην επισκευή του κινητήρα. Για παράδειγμα, η Audi δημιούργησε μια εφαρμογή AR που επιτρέπει στους χρήστες να βλέπουν αυτοκίνητα σχεδόν οπουδήποτε και να δημιουργούν εξατομικευμένες πίστες δοκιμών που δείχνουν πώς μπορεί να λειτουργεί το όχημα. Η διαδικασία αγοράς αυτοκινήτου γίνεται αμέσως πολύ πιο καθηλωτική όταν οι πελάτες έχουν VR και AR για να πειραματιστούν.
- 7. Τουριστική βιομηχανία:** Η τεχνολογία AR μπορεί να κάνει τα ταξίδια πιο ευχάριστα. Ξεκινώντας από την επιλογή ενός ξενοδοχείου μέχρι την κράτηση δωματίου και τον προγραμματισμό εκδρομών, η Επαυξημένη Πραγματικότητα βοηθά την τουριστική βιομηχανία με αμέτρητους τρόπους. Αναπτύσσεται για να προσφέρει στους πελάτες αξέχαστες και χωρίς προβλήματα εμπειρίες κατά τον προγραμματισμό των διακοπών τους. Για παράδειγμα, τα ξενοδοχεία μπορούν να προσφέρουν στους πελάτες εικονικές περιηγήσεις στα δωμάτια πριν από την επίσκεψή τους. Οι αεροπορικές εταιρείες θα μπορούσαν να δώσουν στους πελάτες την ευκαιρία να χρησιμοποιήσουν ένα σετ ακουστικών VR που τους κάνει να αισθάνονται ότι βρίσκονται στον προορισμό τους, ενώ βρίσκονται ακόμα στη μέση του ταξιδιού. Επιπλέον, οι λειτουργίες που βασίζονται στην τοποθεσία επιτρέπουν την παροχή μιας καλύτερης εμπειρίας που είναι προσαρμοσμένη στα ενδιαφέροντα και τις απαιτήσεις του χρήστη. Υπάρχουν επίσης μεμονωμένες τοποθεσίες που έχουν πειραματιστεί με την επαυξημένη πραγματικότητα για την καθοδήγηση των επισκεπτών.
- 8. Βιομηχανία Παραγωγής, Μεταποίησης και Εφοδιαστική Αλυσίδα:** Η Επαυξημένη Πραγματικότητα στη μεταποιητική βιομηχανία χρησιμοποιείται κυρίως για τη συντήρηση και την υποστήριξη. Για παράδειγμα, η AR βοηθά στην ταχύτερη συναρμολόγηση εξελιγμένων μηχανών με ακρίβεια, γεγονός που συμβάλλει στην εξοικονόμηση κόστους και στις ταχύτερες παραδόσεις. Οι επαγγελματίες της παραγωγής και της εφοδιαστικής μπορούν να χρησιμοποιήσουν ολογραφικές εικόνες για να βελτιώσουν την παραγωγικότητα και την απόδοση. Επίσης, μπορεί να χρησιμοποιηθεί για την εκπαίδευση νέων επαγγελματιών της μεταποίησης πώς να χρησιμοποιούν εξειδικευμένα μηχανήματα με την επικάλυψη οδηγιών μέσω AR



Headsets, καθώς εργάζονται. Στον τομέα των logistics και της εφοδιαστικής αλυσίδας η χρήση της Επαυξημένης Πραγματικότητας θα μπορούσε να κάνει θαύματα όταν πρόκειται για την ενίσχυση της αποδοτικότητας των εργαζομένων και στην καλύτερη διαχείριση αποθεμάτων.

- 9. Λιανικό Εμπόριο και Ηλεκτρονικό Εμπόριο:** Ο τομέας του λιανικού εμπορίου προσφέρει μια από τις πιο καινοτόμες και ενδιαφέρουσες εφαρμογές για την Επαυξημένη Πραγματικότητα. Οι λιανοπωλητές, τόσο online όσο και offline, μπορούν να αξιοποιήσουν τις εφαρμογές επαυξημένης πραγματικότητας προς όφελός τους για να αλληλεπιδρούν με τους πελάτες πιο αποτελεσματικά. Και τώρα που η τεχνολογία είναι ευρύτερα διαθέσιμη και σε πιο λογικές τιμές για όλες τις επιχειρήσεις, όλοι μπορούν να τη χρησιμοποιήσουν. Η ζήτηση για αυτές τις λύσεις αυξήθηκε ακόμη περισσότερο από το 2020 με την άφιξη της παγκόσμιας πανδημίας Covid. Με τα καταστήματα να μην μπορούν να ανοίξουν, οι εταιρείες έπρεπε να αναζητήσουν νέους τρόπους για να προσφέρουν νέους σύγχρονους τρόπους αγορών στους καταναλωτές, όπως η μέτρηση αντικειμένων χρησιμοποιώντας τις κάμερες των έξυπνων smartphone ή tablets ή να φανταστούν τα έπιπλα στα δωμάτιό τους, προτού τα αγοράσουν. Βοηθά τους καταναλωτές να βλέπουν και να επιλέγουν τα προϊόντα που είναι διαθέσιμα στο κατάστημα και να αναζητούν γρήγορα πληροφορίες για τα προϊόντα. Η εικονική δοκιμή με τη χρήση επαυξημένης πραγματικότητας ενθαρρύνει πολλούς περισσότερους πελάτες να κάνουν αγορές απ' ό,τι είχαν προγραμματίσει. Πολλές διαφορετικές δοκιμαστικές εφαρμογές έχουν αποκτήσει δημοτικότητα, όπως η δοκιμαστική εφαρμογή γυαλιών με βάση την επαυξημένη πραγματικότητα, η δοκιμαστική εφαρμογή μακιγιάζ, η δοκιμαστική εφαρμογή κοσμημάτων, η δοκιμαστική εφαρμογή επίπλων, η δοκιμαστική εφαρμογή ηλεκτρικών συσκευών και η δοκιμαστική εφαρμογή αξεσουάρ αυτοκινήτου, μεταξύ άλλων. Χρησιμοποιείται επίσης σε οδηγίες χρήσης προϊόντων, προωθητικές ενέργειες, μεθόδους προετοιμασίας κ.λπ.
- Επίσης, μπορεί να προσφέρει μια εξατομικευμένη εμπειρία στους καταναλωτές, βελτιώνοντας έτσι την ευκολία εξυπηρέτησής τους. Για παράδειγμα, στη βιομηχανία μόδας η Επαυξημένη Πραγματικότητα μπορεί να χρησιμοποιηθεί ως εικονικός καθρέφτης που βοηθά τους καταναλωτές να επιλέγουν το επιθυμητό ντύσιμο από το σπίτι και να ψωνίζουν σε περιβάλλον εικονικής πραγματικότητας, με εικονικά δοκιμαστήρια που τους επιτρέπουν να δοκιμάζουν τα ρούχα τους και τα αξεσουάρ της επιλογής τους, για να δουν πώς μπορεί να φαίνονται.

- 10. Στρατιωτικός τομέας:** Οι τεχνολογίες τόσο της επαυξημένης όσο και της εικονικής πραγματικότητας, μπορεί να χρησιμοποιηθούν για εκπαιδευτικούς σκοπούς, καθώς και για πραγματικές επιχειρήσεις στο πεδίο της μάχης. Η μικτή πραγματικότητα μπορεί να χρησιμοποιηθεί για την εκπαίδευση στρατιωτών στη χρήση όπλων, τακτικών και διαδικασιών χωρίς να τους θέτει σε κίνδυνο. Αυτό συμβάλλει στη μείωση των τραυματισμών, ενώ παράλληλα αυξάνει την αποτελεσματικότητα,

επιτρέποντας στους στρατιώτες να αποκτήσουν τις απαραίτητες δεξιότητες με τον δικό τους ρυθμό. Στην πραγματικότητα, τα στρατεύματα του αμερικανικού στρατού χρησιμοποιούν AR headsets κατά τη διάρκεια ασκήσεων εκπαίδευσης μάχης τουλάχιστον από το 2017 - και οι συσκευές αυτές αναμένεται να γίνουν ακόμη πιο διαδεδομένες τα επόμενα χρόνια.

### 1.6.1 Ιατρική & Υγειονομική Περίθαλψη

Η επαυξημένη πραγματικότητα και η εικονική πραγματικότητα είναι μια από τις πιο υποσχόμενες ψηφιακές τεχνολογίες σήμερα και έχει τη δυνατότητα να φέρει επανάσταση στην υγειονομική περίθαλψη και την ιατρική βιομηχανία για τους γιατρούς και τους ασθενείς. Ωστόσο, η Επαυξημένη Πραγματικότητα διαφέρει από τον πιο γνωστό "συγγενή" της, την Εικονική Πραγματικότητα (VR), καθώς η τελευταία δημιουργεί έναν τρισδιάστατο κόσμο αποσυνδέοντας πλήρως τον χρήστη από την πραγματικότητα μεταφέροντάς τον μέσω κλειστών VR Headsets, ενώ η πρώτη, η Επαυξημένη Πραγματικότητα, διατηρεί τον χρήστη στην τρέχουσα ύπαρξή του και απλώς την επαυξάνει με εικονικές πληροφορίες. Με έναν απλούστερο τρόπο, η Επαυξημένη Πραγματικότητα φέρνει μια νέα διάσταση στην υγειονομική περίθαλψη και την ιατρική. Υπάρχουν δύο πτυχές στις οποίες η Επαυξημένη Πραγματικότητα είναι μοναδική: οι χρήστες δεν χάνουν την επαφή με την πραγματικότητα και μεταφέρει τις πληροφορίες το συντομότερο δυνατό. Αυτά τα διακριτικά χαρακτηριστικά επιτρέπουν στην Επαυξημένη Πραγματικότητα να γίνει κινητήρια δύναμη στο μέλλον της ιατρικής. Στην περίπτωση της Επαυξημένης Πραγματικότητας, η χρήση της τεχνολογίας στην ιατρική και την υγειονομική περίθαλψη αποτελεί φυσική συνέπεια της τεχνολογικής ανάπτυξης και της έκρηξης των δεδομένων. [21] [22]



Εικόνα 33. AR in Healthcare

Πηγή: <https://medium.com/swevens/augmented-reality-ar-in-healthcare-3c12bdf86a8e>

Τα πιο σημαντικά οφέλη της Επαυξημένης Πραγματικότητας στο τομέα της ιατρικής και την υγειονομικής περίθαλψης είναι τα εξής:

- Οι νέες καινοτομίες της Επαυξημένης Πραγματικότητας μπορούν να βοηθήσουν στην ενίσχυση της ικανότητας των γιατρών και των χειρουργών να διαγνώσουν, να θεραπεύσουν και να χειρουργήσουν τους ασθενείς τους με μεγαλύτερη ακρίβεια, δίνοντάς τους πρόσβαση σε δεδομένα και πληροφορίες για τους ασθενείς σε πραγματικό χρόνο, ταχύτερα και ακριβέστερα από ποτέ.
- Η τεχνολογία επαυξημένης πραγματικότητας έχει καταστήσει δυνατή την επικάλυψη τρισδιάστατων ανατομικών πληροφοριών πάνω στον ασθενή.
- Μια οθόνη AR που ενσωματώνει όλα τα δεδομένα απεικόνισης και ασθενούς και επιτρέπει στους γιατρούς να διατηρούν τα μάτια τους στον ασθενή έχει τη δυνατότητα να βελτιώσει την ποιότητα, την ασφάλεια και να μειώσει το κόστος μειώνοντας τις επιπλοκές που σχετίζονται με τη διαδικασία.
- Η εκπαίδευση χωρίς κίνδυνο είναι η μεγάλη δυνατότητα χρήσης της επαυξημένης πραγματικότητας. Η τεχνολογία AR καθιστά την ιατρική εκπαίδευση πιο διαδραστική και βοηθά τους σπουδαστές να χρησιμοποιήσουν τις θεωρητικές γνώσεις στον πρακτικό κόσμο.
- Άλλη μία βοήθεια των εφαρμογών AR και MR είναι στις φυσικοθεραπείες που μας δείχνει την πορεία του ασθενή και πως να επισπεύσουμε την διαδικασία ανάρρωσης.
- Τέλος, η χρήση της επαυξημένης πραγματικότητας μπορεί να φανεί χρήσιμη σε άτομα με ειδικές ανάγκες, όπως σε άτομα με αδυναμία όρασης με τη βοήθεια φωνητικών εντολών μπορεί να τους κατευθύνει πιο εύκολα.

### **1.6.2 Κατασκευαστικός τομέας**

Η τεχνολογία AR έχει μεταμορφώσει τον τρόπο με τον οποίο κατασκευάζουμε και σχεδιάζουμε δομές και έχει γίνει απαραίτητο εργαλείο για τους αρχιτέκτονες, τους μηχανικούς και τους επαγγελματίες του κλάδου των κατασκευών. Ο κατασκευαστικός κλάδος έχει αγκαλιάσει την τεχνολογία AR για να βελτιώσει τις διαδικασίες σχεδιασμού, προγραμματισμού και κατασκευής. Ακολουθούν μερικοί από τους τρόπους με τους οποίους χρησιμοποιείται η AR στις κατασκευές: [23] [24]

- Σχεδιασμός και οπτικοποίηση: Η τεχνολογία AR επιτρέπει στους αρχιτέκτονες και τους σχεδιαστές να απεικονίζουν τρισδιάστατα μοντέλα των σχεδίων τους σε ένα πραγματικό περιβάλλον. Χρησιμοποιώντας AR, μπορούν να περιηγηθούν σε ένα ψηφιακό μοντέλο και να κάνουν τροποποιήσεις στο σχέδιο σε πραγματικό χρόνο. Αυτό διευκολύνει τον εντοπισμό πιθανών σχεδιαστικών ατελειών και τη βελτίωση του σχεδίου πριν από την έναρξη της κατασκευής.
- Κατασκευή επί τόπου: Η τεχνολογία AR μπορεί να χρησιμοποιηθεί στο εργοτάξιο για τη βελτίωση της ακρίβειας και της αποτελεσματικότητας της κατασκευής. Με την επικάλυψη ψηφιακών μοντέλων στον πραγματικό κόσμο, οι εργαζόμενοι στις κατασκευές μπορούν να διασφαλίσουν ότι τα δομικά στοιχεία του κτιρίου

τοποθετούνται στις σωστές θέσεις και στις σωστές γωνίες. Αυτό μπορεί να συμβάλει στη μείωση των λαθών και των επανεργασιών, εξοικονομώντας χρόνο και χρήμα.

- Εκπαίδευση για την ασφάλεια: Η τεχνολογία AR μπορεί να χρησιμοποιηθεί για την εκπαίδευση των εργαζομένων στις κατασκευές σχετικά με τις διαδικασίες και τα πρωτόκολλα ασφαλείας. Με την AR, οι εργαζόμενοι μπορούν να εξασκούνται στις διαδικασίες ασφαλείας σε ένα προσομοιωμένο περιβάλλον, διευκολύνοντας τον εντοπισμό πιθανών κινδύνων και μαθαίνοντας πώς να ανταποκρίνονται σε αυτούς.
- Συντήρηση και επισκευή: Η τεχνολογία AR μπορεί επίσης να χρησιμοποιηθεί για τη συντήρηση και την επισκευή κτιρίων. Με την επικάλυψη ψηφιακών πληροφοριών σε φυσικά αντικείμενα, οι εργαζόμενοι στη συντήρηση μπορούν να αναγνωρίζουν γρήγορα τον εξοπλισμό και να έχουν πρόσβαση σε πληροφορίες σχετικά με τις διαδικασίες συντήρησης. Αυτό μπορεί να συμβάλει στη μείωση του χρόνου διακοπής λειτουργίας και στη βελτίωση της αποτελεσματικότητας της συντήρησης.



**Εικόνα 34. AR in construction**

Πηγή: <https://smarttek.solutions/blog/augmented-reality-in-construction/>

### **1.6.3 Στρατιωτικός τομέας**

Η μικτή πραγματικότητα (MR) είναι μια αναδυόμενη τεχνολογία που έχει τη δυνατότητα να αλλάξει τον τρόπο με τον οποίο αλληλεπιδρούμε με τον κόσμο γύρω μας. Αυτή η τεχνολογία, η οποία συνδυάζει πτυχές τόσο της επαυξημένης όσο και της εικονικής πραγματικότητας (AR), μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς, καθώς και για πραγματικές επιχειρήσεις στο πεδίο της μάχης. [25]

Η μικτή πραγματικότητα μπορεί να χρησιμοποιηθεί για την εκπαίδευση στρατιωτών στη χρήση όπλων, τακτικών και διαδικασιών χωρίς να τους θέτει σε κίνδυνο. Αυτό συμβάλλει στη μείωση των τραυματισμών, ενώ παράλληλα αυξάνει την αποτελεσματικότητα, επιτρέποντας στους στρατιώτες να αποκτήσουν τις απαραίτητες δεξιότητες με τον δικό τους ρυθμό. Στην πραγματικότητα, τα στρατεύματα του αμερικανικού στρατού χρησιμοποιούν AR headsets κατά τη διάρκεια ασκήσεων

εκπαίδευσης μάχης τουλάχιστον από το 2017 - και οι συσκευές αυτές αναμένεται να γίνουν ακόμη πιο διαδεδομένες τα επόμενα χρόνια.

Ο στρατός των ΗΠΑ χρησιμοποιεί σήμερα εκδόσεις των Microsoft HoloLens AR ως μέρος του ολοκληρωμένου συστήματος οπτικής ενίσχυσης (Integrated Visual Augmentation System – IVAS), το οποίο επιτρέπει στους στρατιώτες να οπτικοποιούν δεδομένα, να βλέπουν γύρω από γωνίες και μέσα από τοίχους και να παρέχουν επικοινωνία σε πραγματικό χρόνο με άλλες ομάδες. Ο στρατός χρησιμοποιεί το IVAS σε εκπαιδευτικά σενάρια για να βοηθήσει τους στρατιώτες να μάθουν πώς να χειρίζονται διάφορα κομμάτια εξοπλισμού, από άρματα μάχης και αεροσκάφη μέχρι μη επανδρωμένα αεροσκάφη και πυροβολικό. Για παράδειγμα, το IVAS μπορεί να χρησιμοποιηθεί για να εκπαιδεύσει τους στρατιώτες στο πώς να χρησιμοποιούν το οπλικό σύστημα ενός άρματος ή πού είναι οι καλύτερες θέσεις βολής για μια συγκεκριμένη αποστολή. Εκτός από την παροχή αυτού του είδους της πρακτικής εκπαίδευσης, το IVAS μπορεί επίσης να παρέχει πληροφορίες σε πραγματικό χρόνο σχετικά με τις θέσεις του εχθρού, ώστε οι στρατιώτες να είναι καλύτερα προετοιμασμένοι για καταστάσεις μάχης.



**Εικόνα 35. AR in military**

Πηγή: <https://www.techspot.com/news/77642-microsoft-wins-480-million-hololens-contract-us-army.html>

Η τεχνολογία επιτρέπει στους στρατιώτες να εκπαιδεύονται σε πολύπλοκα περιβάλλοντα όπου υπάρχουν πολλές μεταβλητές, όπως το πώς να χειριστούν το όπλο, ώστε να είναι πιο ακριβείς στο στόχο τους, η οδήγηση ενός τεθωρακισμένου οχήματος μεταφοράς προσωπικού σε μια ζώνη μάχης ή η διεξαγωγή μιας επιχείρησης τη νύχτα και σε κακές καιρικές συνθήκες. Τα προτερήματα που προκύπτουν χάρη στη χρήση αυτής της τεχνολογίας είναι αρχικά, ότι τα γυαλιά αυτά, μπορούν να εντοπίσουν για τυχόν εχθρούς ανάμεσα σε τοίχους, καθώς επίσης βοηθούν στον εντοπισμό βόμβας ή νάρκης.

Μια άλλη σημαντική εφαρμογή της μικτής πραγματικότητας είναι στον τομέα των συστημάτων διοίκησης και ελέγχου (Command-and-Control systems (C2)). Οι πλατφόρμες C2 επιτρέπουν στους διοικητές όλων των επιπέδων να έχουν πρόσβαση σε λεπτομέρειες της αποστολής, δεδομένα σε πραγματικό χρόνο σχετικά με τις φίλιες θέσεις ή τις εχθρικές δυνάμεις -ακόμη και αν βρίσκονται χιλιάδες χιλιόμετρα μακριά η μία από την άλλη- και ακριβέστερες εκτιμήσεις σχετικά με τις τρέχουσες επιχειρήσεις από ό,τι ήταν ποτέ άλλοτε δυνατόν. Ωστόσο, επειδή αυτά τα συστήματα είναι τόσο πολύπλοκα, απαιτούν προσεκτικό σχεδιασμό πριν από την εφαρμογή τους.

#### 1.6.4 Βιομηχανία Ηλεκτρονικών Παιχνιδιών και Ψυχαγωγίας

Η Επαυξημένη Πραγματικότητα υπόσχεται πολλά για τη βιομηχανία ηλεκτρονικών παιχνιδιών και ψυχαγωγίας, η οποία έχει αναπτυχθεί γρήγορα τα τελευταία χρόνια. Στα παιχνίδια επαυξημένης πραγματικότητας (AR), τα ψηφιακά δεδομένα τοποθετούνται πάνω στον πραγματικό κόσμο, ώστε οι παίκτες να μπορούν να αλληλεπιδρούν με εικονικά πράγματα εκεί. Οι παίκτες μπορούν πλέον να βιώσουν τα παιχνίδια σε ένα νέο επίπεδο εμπύθισης χάρη στην τεχνολογία επαυξημένης πραγματικότητας (AR). Τα όρια μεταξύ του εικονικού και του πραγματικού κόσμου θα μπορούσαν να θολώσουν ως αποτέλεσμα, καθιστώντας τα παιχνίδια πιο ρεαλιστικά και ελκυστικά.

Για παράδειγμα, η τεχνολογία AR χρησιμοποιείται σε παιχνίδια όπως το Pokemon Go για τη δημιουργία ενός εικονικού κόσμου που εμφανίζεται μέσα στο πραγματικό περιβάλλον του παίκτη. Επίσης, Το Halo Recruit προσφέρει μια εμπειρία μικτής πραγματικότητας για τα Windows 10 της Microsoft και δημιουργήθηκε από την εταιρεία ανάπτυξης VR Endeavor One σε συνεργασία με 343 άλλες εταιρείες Industries. Το Halo Recruit απαιτεί ένα σετ Windows MR Headsets για να μπορέσουν οι παίκτες να παίξουν.



Εικόνα 36. Halo Recruit

Πηγή: <https://www.purexb.com/news/2022/07/343-developer-shares-halo-vr-concepts-from-2017>

Τα παιχνίδια AR έχουν τη δυνατότητα να προωθήσουν νέες μορφές διαπροσωπικής αλληλεπίδρασης. Τα παιχνίδια AR για πολλούς παίκτες μπορούν να προωθήσουν μια κοινή εμπειρία που συνδέει τον εικονικό και τον φυσικό κόσμο. Καθώς οι παίκτες πρέπει να περπατούν για να αλληλεπιδράσουν με τα εικονικά πράγματα στο περιβάλλον τους, τα παιχνίδια AR μπορούν επίσης να προωθήσουν την κίνηση και την εξερεύνηση.

Η τεχνολογία AR έχει διάφορες επιπτώσεις στη βιομηχανία των παιχνιδιών κι όχι μόνο να χρησιμεύει ως πλατφόρμα ανάπτυξης για παιχνίδια. Για παράδειγμα, η AR μπορεί να χρησιμοποιηθεί για τη βελτίωση του μάρκετινγκ και της διαφήμισης των παιχνιδιών, δίνοντας στους παίκτες μια πιο καθηλωτική εμπειρία με τα παιχνίδια, ακόμη και πριν αυτά δημοσιευτούν επίσημα. Η διαφήμιση εντός του παιχνιδιού και η πώληση εικονικών αγαθών είναι μόνο δύο παραδείγματα των πρόσθετων πηγών εσόδων που μπορεί να δημιουργήσει η AR για τους παραγωγούς παιχνιδιών.

Ωστόσο, υπάρχουν και ορισμένες δυσκολίες με τα παιχνίδια AR. Για παράδειγμα, τα παιχνίδια AR με μεγάλη κατανάλωση υπολογιστικών πόρων μπορεί να απαιτούν ισχυρό υλικό και σταθερή σύνδεση στο διαδίκτυο. Επειδή πρέπει να συνδυάζουν ομαλά τις πτυχές του εικονικού και του πραγματικού κόσμου, τα παιχνίδια AR μπορεί να είναι δύσκολο να κατασκευαστούν. Οι δυνατότητες των παιχνιδιών επαυξημένης πραγματικότητας είναι σημαντικές παρά τις δυσκολίες αυτές. Έχει τη δυνατότητα να αλλάξει την αγορά παιχνιδιών και να προσφέρει στους χρήστες νέες εμπειρίες. Καθώς αναζητούν νέες πηγές εσόδων και μεθόδους επικοινωνίας με τους παίκτες, τα παιχνίδια AR μπορούν επίσης να ανοίξουν νέες προοπτικές για τους εκδότες και τους προγραμματιστές παιχνιδιών.

Εν κατακλείδι, οι δυνατότητες της επαυξημένης πραγματικότητας για τα παιχνίδια είναι συναρπαστικές και μπορεί να έχουν μεγάλο αντίκτυπο στον τομέα. Το AR gaming έχει τη δυνατότητα να καινοτομήσει και να βελτιώσει την εμπειρία παιχνιδιού, να προωθήσει την κοινωνική αλληλεπίδραση και να αλλάξει τον τρόπο με τον οποίο παίζουμε και ασχολούμαστε με τα παιχνίδια. Τα μελλοντικά παιχνίδια AR είναι πιθανό να είναι ακόμη πιο δημιουργικά και ελκυστικά καθώς η τεχνολογία εξελίσσεται.

## **1.7 Παραδείγματα εφαρμογών Επαυξημένης & Μικτής Πραγματικότητας για τουριστικές περιηγήσεις σε αξιοθέατα**

Με την εξέλιξη της τεχνολογίας η δημιουργία εφαρμογών επαυξημένης και μικτής πραγματικότητας για περιηγήσεις σε αξιοθέατα αποτελεί ένα σημαντικό κομμάτι για την ενίσχυση του τουρισμού και να κάνουν πιο ενδιαφέρουσα την εμπειρία της ξενάγησης. Πλέον υπάρχουν πολλές εφαρμογές που κάνουν ξεχωριστή μια προβολή εκθεμάτων.

### **1.7.1 Google Arts & Culture**

Το Google Arts & Culture [26] είναι μία από τις πιο δημοφιλείς εφαρμογές της Google LLC, με περισσότερες από 5 εκατομμύρια λήψεις, η οποία χρησιμοποιείται σε περιηγήσεις και ξεναγήσεις από διάφορους οργανισμούς σε όλο τον κόσμο. Μέχρι και

σήμερα υποστηρίζει πάνω από 2000 μνημεία 80 περίπου χωρών. Ένα από τα θετικά της προτερήματα είναι ότι η περιήγηση των μνημείων αυτών μπορεί να συμβεί σε οποιοδήποτε σημείο κι αν βρισκόμαστε.

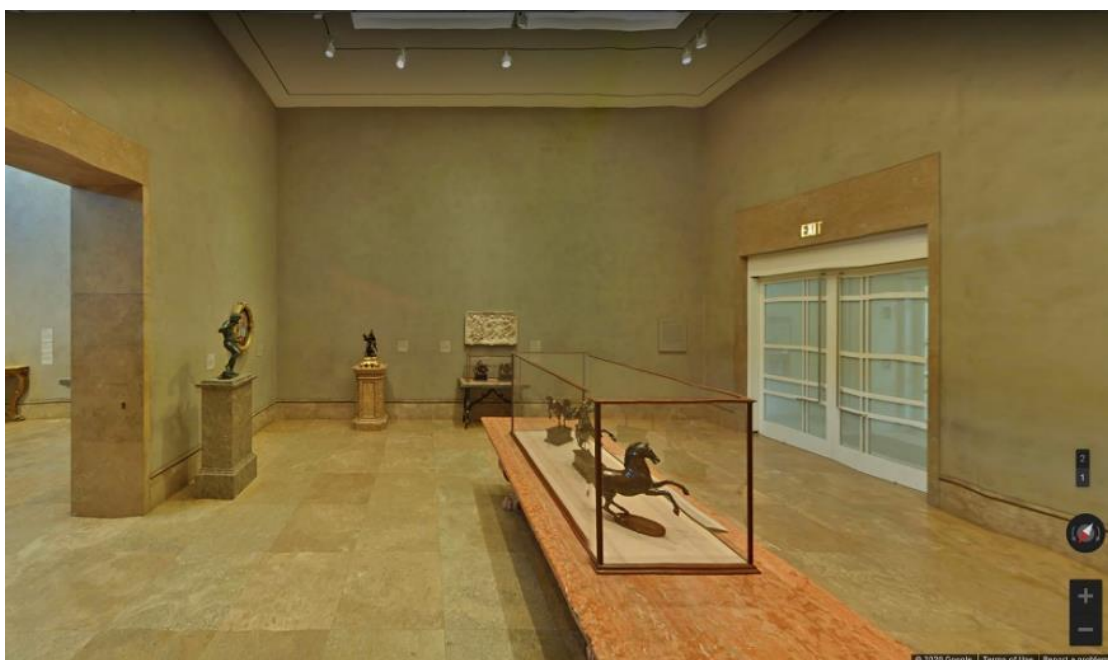
Κάποιες από τις δυνατότητες που προσφέρει είναι:

- Art Projector: Προβολή των έργων σε πραγματικά μεγέθη
- Virtual Reality Tours: Ξεναγήσεις σε γνωστά μουσεία
- Art Recogniser: Αναγνώριση έργων τέχνης σε πραγματικό χρόνο μέσω της κάμερας του κινητού
- Translate: Ανάγνωση πληροφοριών εκθεμάτων σε όλες τις γλώσσες
- Favorites: Αποθήκευση αγαπημένων έργων τέχνης



Εικόνα 37. Google Arts & Culture

Πηγή: <https://lucidea.com/blog/google-arts-culture/>



Εικόνα 38. Google Arts & Culture 2

Πηγή: <https://www.archpaper.com/2020/03/google-arts-culture-over-500-virtual-museums/>



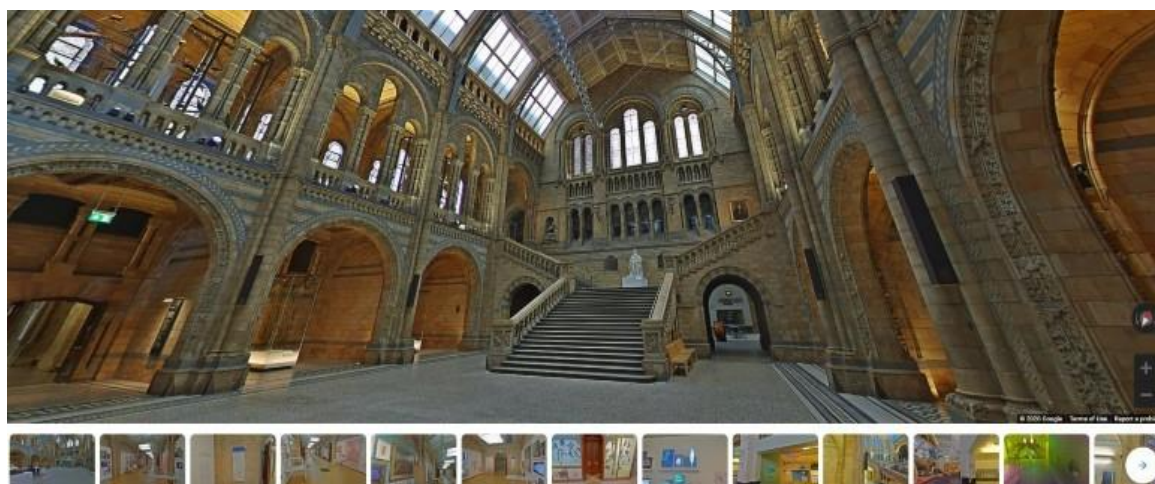
### 1.7.2 Natural History Museum: The museum at South Kensington

Το συγκεκριμένο μουσείο περιέχει μεικτής πραγματικότητας ξενάγηση, αλλά και φωνητικό οδηγό με πληροφορίες για τα αξιοθέατα του. [27]



Εικόνα 39. Natural History Museum: The museum at South Kensington

Πηγή: <https://peakd.com/hive-163772/@mhel/natural-history-museum-london--an-exhibit-of-our-world>



Εικόνα 40. Natural History Museum: The museum at South Kensington 2

Πηγή: <https://www.nhm.ac.uk/visit/virtual-museum.html>

### 1.7.3 The Art gallery of Ontario, Toronto

Η συγκεκριμένη εφαρμογή δημιουργήθηκε για να κεντρίσει περισσότερο το ενδιαφέρον του επισκέπτη, διότι ο μέσος όρος της προσοχής του σε ένα έργο τέχνης

είναι μόλις 17 δευτερόλεπτα. Εγκαθιστώντας την εφαρμογή αυτή για το συγκεκριμένο μουσείο και σκανάροντας κάθε έργο τέχνης ξεχωριστά, αυτό ζωντανεύει στην οθόνη του κινητού μεταφέροντάς το στο σήμερα. [28]



**Εικόνα 41. The art gallery of Ontario, Toronto**

Πηγή: <https://ago.ca/exhibitions/reblink>



**Εικόνα 42. The Art gallery of Ontario, Toronto 2**

Πηγή: [https://www.researchgate.net/figure/Alex-Mayhew-ReBlink-2017-Art-Gallery-of-Ontario-Retrieved-from\\_fig3\\_323620099](https://www.researchgate.net/figure/Alex-Mayhew-ReBlink-2017-Art-Gallery-of-Ontario-Retrieved-from_fig3_323620099)

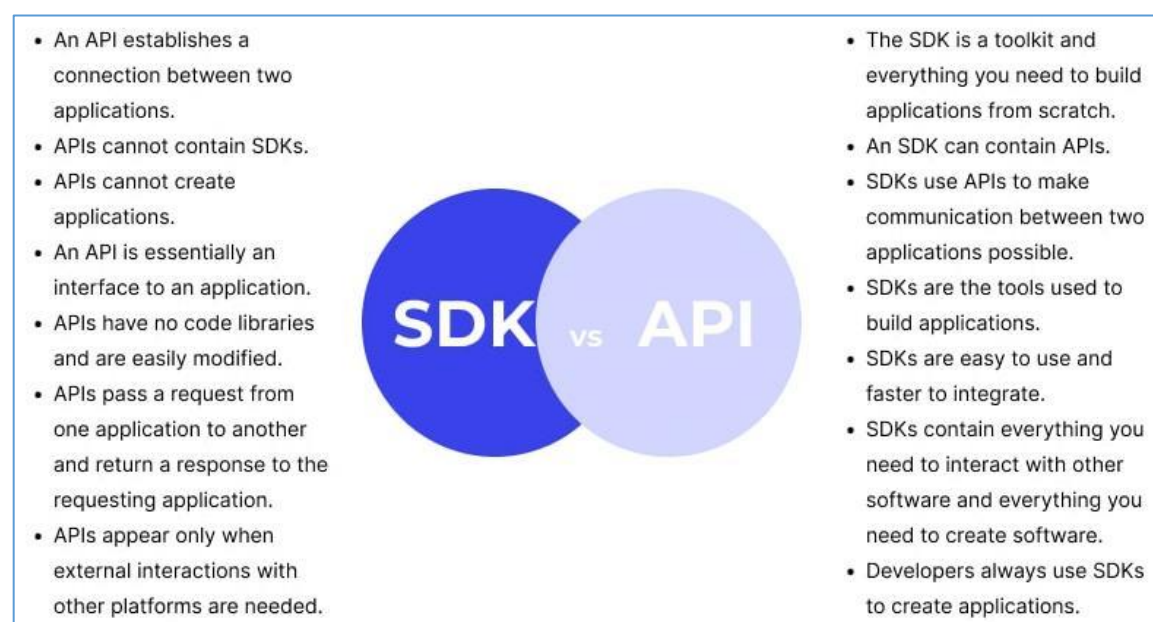
## 2 ΛΟΓΙΣΜΙΚΑ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ

Για τη δημιουργία μίας εφαρμογής επαυξημένης πραγματικότητας για κινητές συσκευές χρειαζόμαστε API, μία γλώσσα μηχανής, ένα SDK για μία πλατφόρμα και φυσικά λογισμικά δημιουργίας εφαρμογών, αλλά και διάφορα εργαλεία αυτών.

### 2.1 Διαφορές μεταξύ ενός API και ενός SDK

Ένα Software Development Kit (SDK) είναι ένα πακέτο που περιέχει διάφορα εργαλεία για την δημιουργία μίας εφαρμογής σε ένα συγκεκριμένο λογισμικό. Κάθε λογισμικό έχει το δικό του SDK. Αναλυτικά σε ένα SDK περιέχονται βιβλιοθήκες για την λειτουργία κώδικα, ένα Integrated Development Environment (IDE) το οποίο χρησιμοποιείται για την εγγραφή του κώδικα και την μετάφραση μίας υψηλής γλώσσας προγραμματισμού σε μία πιο χαμηλή που χρησιμοποιούνται για την δημιουργία εφαρμογών μέσω του compiler. Επίσης, ένα SDK περιέχει δείγματα γραφής κώδικα για να βοηθήσει τον προγραμματιστή, έναν εντοπιστή σφαλμάτων για τον εντοπισμό τυχών λαθών και τέλος ένα Application Programming Interface (API). [29]

Σε αντίθεση, με το SDK ένα API είναι ένα σύνολο κανόνων που προσδιορίζουν πως να επικοινωνούν οι εφαρμογές με τους υπολογιστές. Το API είναι ένας μεσολαβητής ενός πελάτη και ενός διακομιστή, ο πελάτης στέλνει το αίτημα στον API αυτός το προωθεί στο διακομιστή και αυτός με την σειρά του στέλνει την απάντηση στον πελάτη μέσω του API. Ένα API περιλαμβάνει ένα λεγόμενο συμβόλαιο το οποίο στην ουσία είναι οι οδηγίες χρήσεις που έχει δώσει ο προγραμματιστής για το ποια θα είναι η συγκεκριμένη δουλειά που πρέπει να κάνει το συγκεκριμένο API.



Εικόνα 43. SDK VS API

Πηγή: <https://www.wallarm.com/what/sdk-vs-api-whats-the-difference>

Η διαφορά ενός API και ενός SDK είναι ότι το SDK περιέχει API σε αυτό, ενώ το API όχι. Επομένως, αν χρειαστεί ο προγραμματιστής να προσθέσει κάτι σε μία υπάρχουσα εφαρμογή μπορεί να το κάνει μέσω API, ενώ για την δημιουργία μίας εφαρμογής σε μία συγκεκριμένη πλατφόρμα ο προγραμματιστής μέσω του SDK μπορεί να δουλέψει με βιβλιοθήκες και βοηθήματα σχετικά με τον κώδικα που θα χρησιμοποιήσει.

## 2.2 Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών, καθώς και ιστοσελίδων, εφαρμογών και υπηρεσιών ιστού και εφαρμογών για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft όπως τα Windows API, τα Windows Forms, το Windows Presentation Foundation, το Windows Store και το Microsoft Silverlight. Το Visual Studio περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα το οποίο υποστηρίζει το IntelliSense (το στοιχείο ολοκλήρωσης κώδικα) καθώς και το refactoring κώδικα. [30]



Εικόνα 44. Microsoft Visual Studio Logo

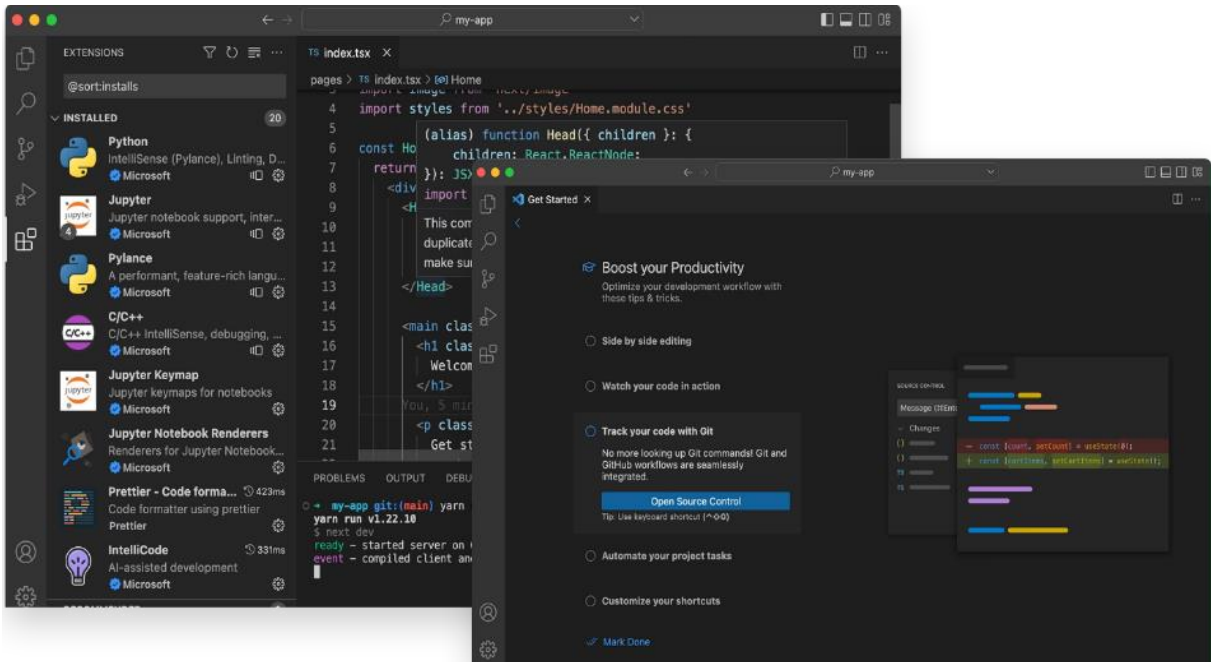
Πηγή: <https://logowik.com/microsoft-visual-studio-vector-logo-4965.html>

Το ολοκληρωμένο πρόγραμμα εντοπισμού σφαλμάτων λειτουργεί τόσο ως πρόγραμμα εντοπισμού αυτών σε επίπεδο πηγής όσο και ως εργαλείο εντοπισμού σφαλμάτων σε επίπεδο μηχανής. Άλλα ενσωματωμένα εργαλεία περιλαμβάνουν ένα προφίλ κώδικα, σχεδιαστή μορφών για την κατασκευή εφαρμογών GUI, σχεδιαστή ιστοσελίδων, σχεδιαστή τάξεων και σχεδιαστή σχήματος βάσης δεδομένων.

Αποδέχεται plugins που βελτιώνουν τη λειτουργικότητα σχεδόν σε όλα τα επίπεδα - συμπεριλαμβανομένης της προσθήκης υποστήριξης για συστήματα ελέγχου πηγής (όπως το Subversion και το Git) και την προσθήκη νέων εργαλείων όπως editors και visual designers για συγκεκριμένες γλώσσες ή ομάδες εργαλείων.

Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού. Επιτρέπει στον Code Editor και στον Debugger να αναγνωρίζει (σε διαφορετικούς βαθμούς) σχεδόν οποιαδήποτε γλώσσα προγραμματισμού, υπό την προϋπόθεση ότι υπάρχει μια συγκεκριμένη γλώσσα. Οι ενσωματωμένες γλώσσες περιλαμβάνουν την C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Υποστηρίζει

για άλλες γλώσσες όπως Python, Ruby, Node.js και M μεταξύ άλλων είναι διαθέσιμη μέσω plugins. Η Java (και η J #) υποστηρίχθηκαν στο παρελθόν.



Εικόνα 45. Microsoft Visual Studio Code IDE

Πηγή: <https://visualstudio.microsoft.com/wp-content/uploads/2022/11/VSCode.webp>

## 2.3 Mapbox

Χάρη στην προηγμένη τεχνολογία του Mapbox ο προγραμματιστής έχει την δυνατότητα να ζωγραφίσει έναν χάρτη δυναμικά στο διαδίκτυο ή σε μία εφαρμογή. Οι προγραμματιστές του Mapbox με την απόδοση των 3D γραφικών και των διανυσματικών πλακιδίων έχουν καταφέρει να κάνουν την εφαρμογή πιο ελκυστική. Η ευκολία και η απλότητα της δημιουργίας ενός χάρτη, η ανάλυσή του, η εύκολη χρήση του σε κινητές συσκευές, οι χαμηλές τιμές και τα ελεύθερα προς χρήση εργαλεία του είναι λίγα από τα θετικά που το κάνουν τόσο δημοφιλή. [31]



Εικόνα 46. Mapbox Logo

Πηγή: <https://www.prnewswire.com/news-releases/mapbox-announces-164-million-series-c-financing-led-by-the-softbank-vision-fund-650230793.html>

Μερικές από τις κυριότερες τεχνολογίες του Mapbox είναι:

- **Δεδομένα:** Η τεχνολογία των δεδομένων της εφαρμογής είναι ένα από τα κύρια κομμάτια και γίνεται μέσω των αισθητήρων του κινητού, της κάμερας ενός φορητού υπολογιστή και από εναέριες εικόνες.

- Όραση: Η τεχνολογία της όρασης δίνει την δυνατότητα στην εφαρμογή να δημιουργήσει δρόμους, χρησιμοποιώντας νευρωνικό δίκτυο. Μετατρέπει τις κάμερες και τα κινητά σε αισθητήρες κίνησης και έτσι έχει την δυνατότητα να καταλάβει τις συνθήκες του δρόμου την δεδομένη χρονική στιγμή.
- Δεδομένα κίνησης: Στην συγκεκριμένη τεχνολογία παρέχεται η δυνατότητα ανίχνευσης κίνησης και κατάστασης ενός δρόμου για δισεκατομμύρια δρόμους, μέσα σε λίγα λεπτά. Χάρη σε αυτήν μπορεί να μας δώσει οδηγίες για το προορισμό μας για μία ομαλή διαδρομή.
- Διανυσματικά πλακίδια: Το Mapbox χρησιμοποιεί τα πλακίδια, για να είναι πιο οργανωμένη και αποδοτική η εφαρμογή. Εκεί αποθηκεύει τα περισσότερα δεδομένα του χάρτη και με αυτά μπορεί να κάνει πιο εύκολα εργασίες, όπως την κλιμάκωση του χάρτη, αλλά και τον σχεδιασμό του. Στα πλακίδια περιέχονται δεδομένα και γεωμετρίες τα οποία κάνουν τον χάρτη διαδραστικό και δυναμικό.
- Γραφικά: Μέσω των γραφικών η εφαρμογή δίνει μία σαφώς καλύτερη εμπειρία στον χρήστη. Χάρη στις βιβλιοθήκες που περιέχονται σε αυτήν δίνει την δυνατότητα σε αυτόν να δημιουργήσει έναν χάρτη με πολύ καλά γραφικά.
- Εργαλεία: Η τεχνολογία των εργαλείων περιλαμβάνει APIs, SDKs , αναζήτηση και πλοήγηση. Παρέχει στο χρήστη την δυνατότητα να δημιουργήσει το χάρτη για εφαρμογή διαδικτύου, κινητά, ή άλλη συσκευή, ανάλογα με τις προτιμήσεις του.

Τέλος,, μερικά από τα μειονεκτήματα που επισημαίνονται από τη χρήση του Mapbox είναι: η χρέωση για τις υπηρεσίες του, ο μικρός χώρος αποθήκευσης για τους δωρεάν εγγεγραμμένους, η αδυναμία φόρτωσης μεγάλων δεδομένων (ορισμένες φορές) και δυσκολία στην χρήση από προγραμματιστές που έχουν συνηθίσει να δουλεύουν σε διαφορετικά API.



**Εικόνα 47. Mapbox Navigation**

Πηγή: [https://assets-global.website-files.com/6050a76fa6a633d5d54ae714/650469400e6eca12e09a8949\\_Transparent%20iPad%20Pro%2011%E2%80%B3%20Space%20Gray%20Mockup%20\(Mockuups%20Studio\).webp](https://assets-global.website-files.com/6050a76fa6a633d5d54ae714/650469400e6eca12e09a8949_Transparent%20iPad%20Pro%2011%E2%80%B3%20Space%20Gray%20Mockup%20(Mockuups%20Studio).webp)

## 2.4 Unity Game Engine

Η μηχανή Unity Game Engine (<https://unity.com/>) χρησιμοποιείται για τη δημιουργία 2D και 3D εφαρμογών χάρη στα εργαλεία SDKs και APIs που διαθέτει, τα οποία παρέχουν στον προγραμματιστή ευκολία στη δημιουργία μίας εφαρμογής. Μπορεί να δημιουργήσει το οτιδήποτε για κινητά IOS και Android, φορητούς και επιτραπέζιους υπολογιστές και tablets. Πολλοί προγραμματιστές το προτιμούν λόγω του API της γλώσσας C# που διαθέτει μέσω του Visual studio. Επίσης, υποστηρίζει και τη γλώσσα Javascript σε διαφορετικό IDE (MonoDeveloper) για όσους επιθυμούν μία εναλλακτική εκτός του Visual Studio. [32]

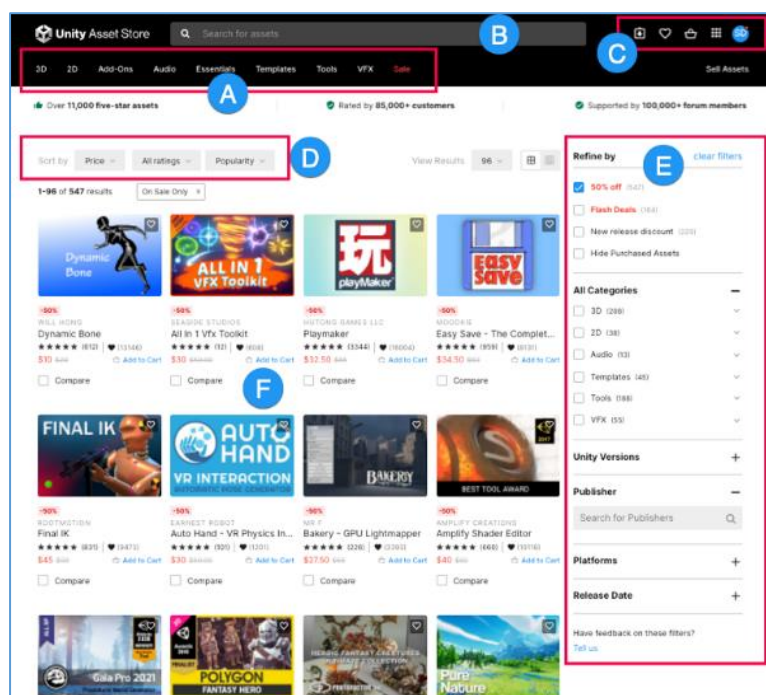


Εικόνα 48. Unity Game Engine logo

Πηγή: <https://brand.unity.com/>

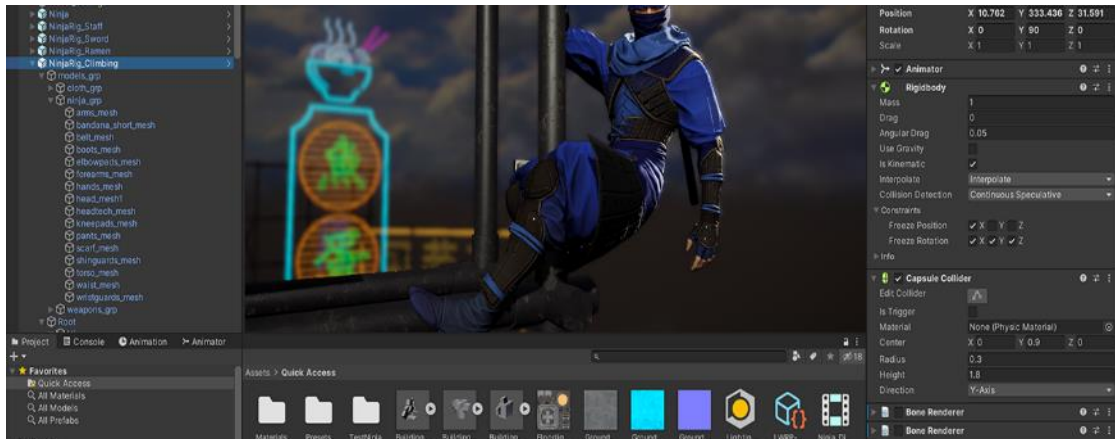
Μερικά από τα κυριότερα εργαλεία που διαθέτει είναι τα εξής:

**Asset store:** Το Unity Asset Store περιέχει μια βιβλιοθήκη με δωρεάν και εμπορικά στοιχεία που δημιουργούν οι Unity Technologies και τα μέλη της κοινότητας. Διατίθεται μεγάλη ποικιλία στοιχείων, όπως υφές, μοντέλα, 2D και 3D γραφικά, κινούμενα σχέδια, ολόκληρα παραδείγματα έργων, διάφορα scripts και επεκτάσεις του Editor. Ξεκινώντας από το Unity 2020.1, το ειδικό παράθυρο του Asset Store δεν φιλοξενείται πλέον μέσα στον Unity Editor. Ωστόσο, στον ιστότοπο του Asset Store <https://assetstore.unity.com/> μπορείτε να αναζητήσετε πακέτα που έχετε αγοράσει και κατεβάσει από το Asset Store και να τα εισάγετε και να τα κατεβάσετε απευθείας στο παράθυρο Package Manager.



Εικόνα 49. Unity Asset Store's website

- **Cloud Code:** Οι δυνατότητες που παρέχει είναι πολλές. Μέσω Cloud Code μπορεί να δημιουργηθεί μία εφαρμογή πολλαπλών χρηστών, η οποία μπορεί να λειτουργήσει μέσω server ή να δημιουργηθεί μία εφαρμογή με άλλους συνεργάτες αποθηκεύοντας για παράδειγμα μια εργασία στον χώρο αυτόν.
- **Editor:** Μπορούν να δημιουργηθούν εργαλεία και scripts με υποστηριζόμενα APIs τα οποία μπορούν να χρησιμοποιηθούν στην πλατφόρμα της Unity.



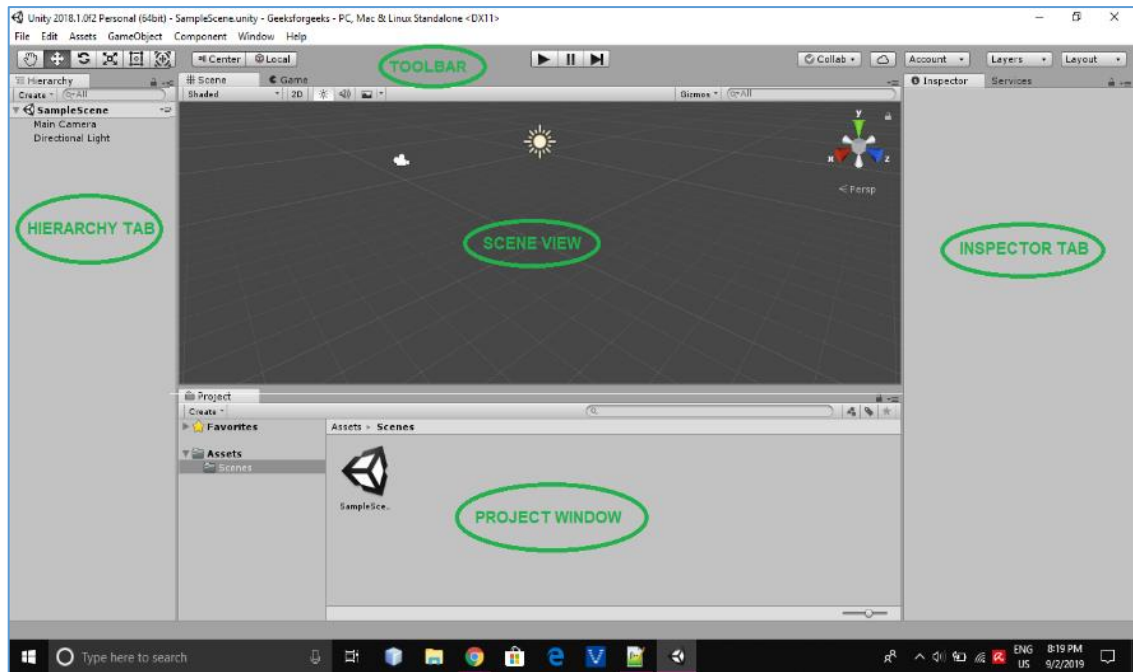
Εικόνα 50. Unity Editor

- **MARS:** Το Unity Mars βοηθά στην επίλυση των δύσκολων προβλημάτων της δημιουργίας εφαρμογών εφαρμογών επαυξημένης πραγματικότητας (AR) για δυναμικά φυσικά περιβάλλοντα, με καλύτερες ροές εργασίας, μείωση του χρόνου ανάπτυξης και εργαλεία συγγραφής σε απλή γλώσσα και πολλαπλές πλατφόρμες με υποστήριξη για iOS, Android και HoloLens.



Εικόνα 51. Unity MARS



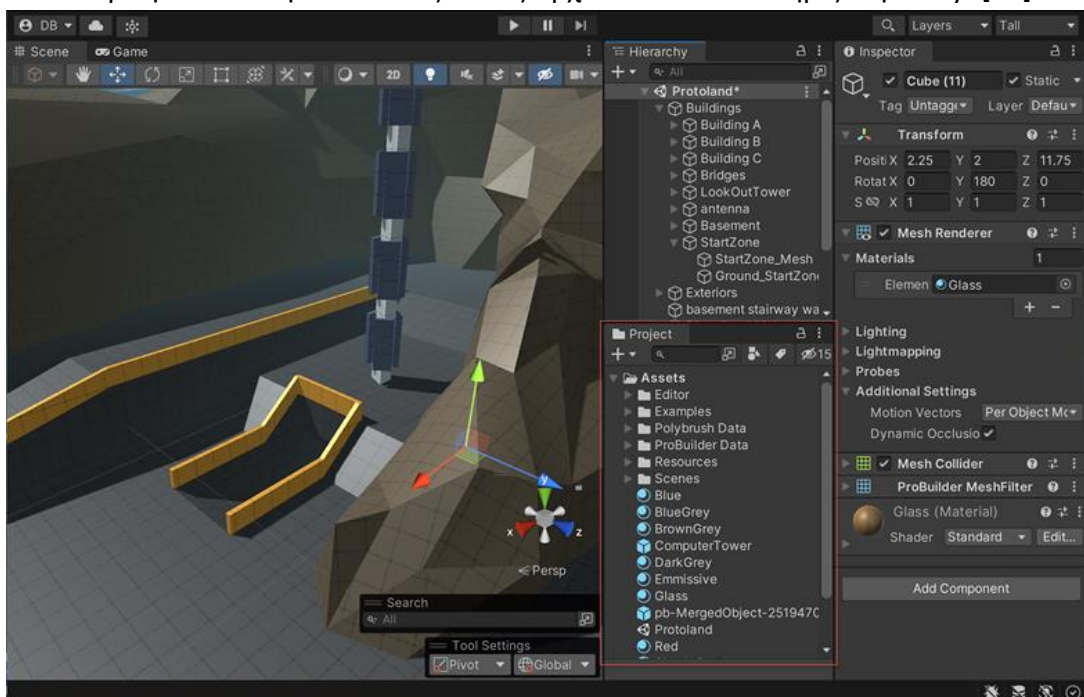


Εικόνα 52. Unity new project

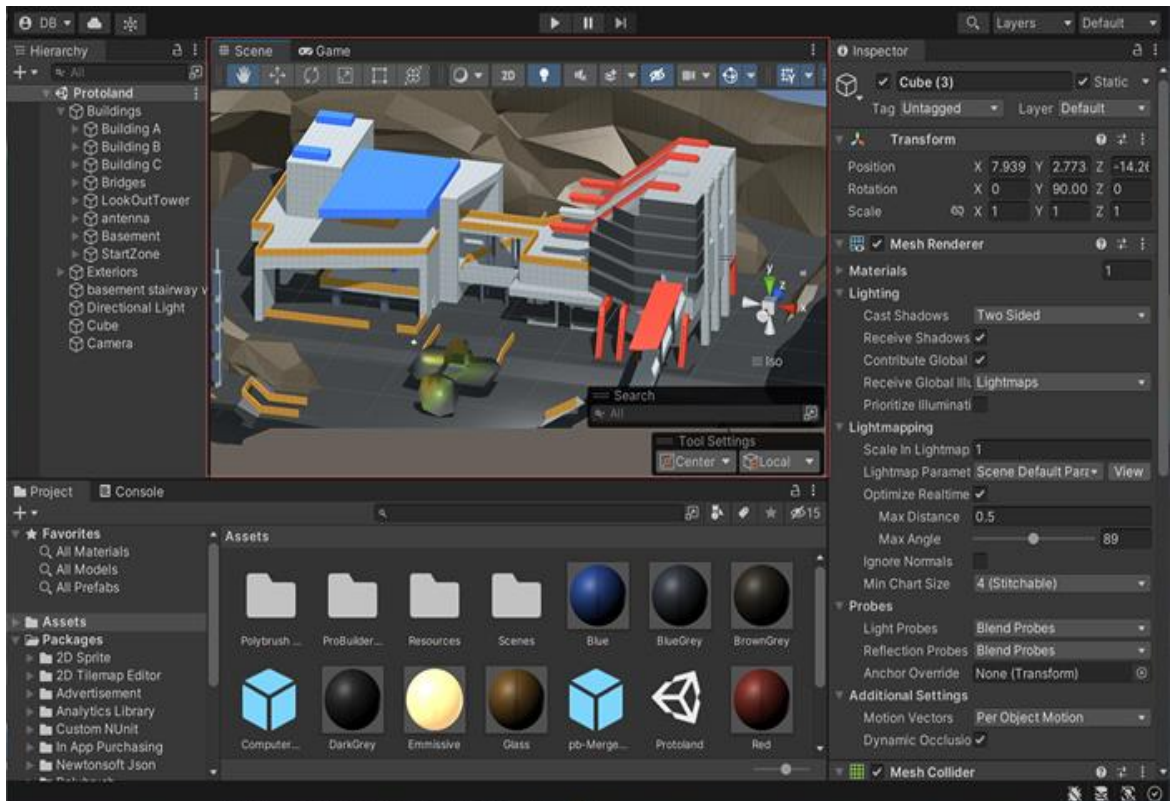
Πηγή: <https://www.geeksforgeeks.org/unity-introduction-to-interface/>

### 2.4.1 Unity Game Engine - Βασικά χαρακτηριστικά διεπαφών

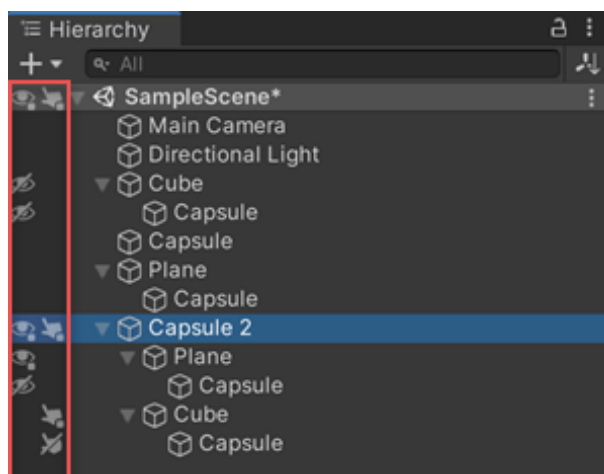
- **Project window:** Είναι ο χώρος που αποθηκεύονται τα Assets. Τα Assets αντιπροσωπεύουν οποιοδήποτε στοιχείο που μπορεί να χρησιμοποιηθεί στο παιχνίδι ή στο Project. Ένα Asset μπορεί να προέρχεται από ένα αρχείο που δημιουργήθηκε εκτός της Unity, όπως ένα 3D μοντέλο, ένα αρχείο ήχου, μια εικόνα, ένα Script ή οποιοσδήποτε άλλος τύπος αρχείου που υποστηρίζει η Unity. [33]



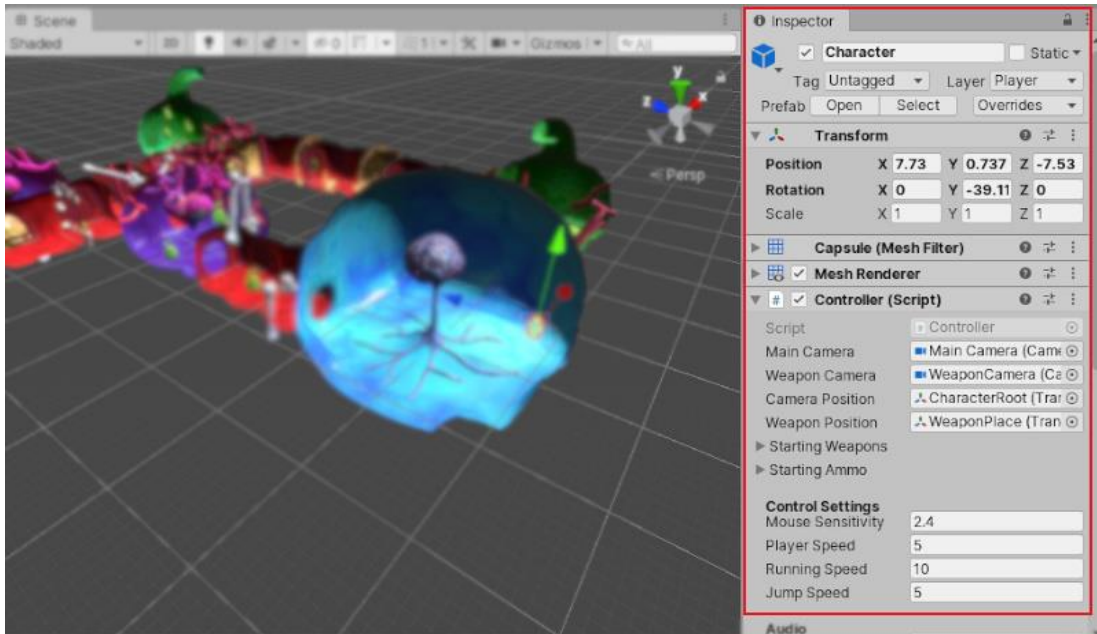
- **Scene view:** Το παράθυρο Scene είναι η διαδραστική προβολή του χρήστη στο Project που δημιουργεί. Γίνεται η χρήση του Scene View για να επιλεγθούν και να χρησιμοποιηθούν τοπία, χαρακτήρες, κάμερες, φώτα και όλων των άλλων τύπων από Game Objects. [34]



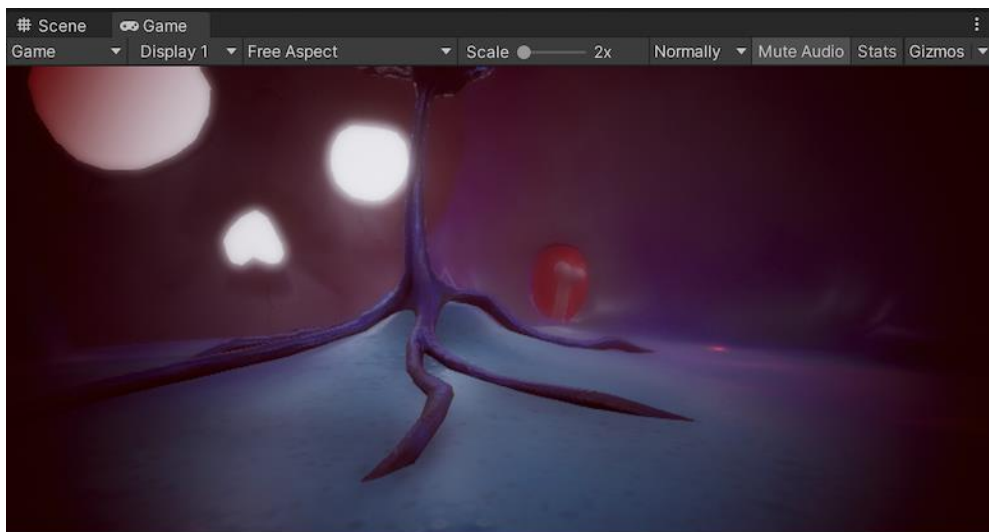
- **Hierarchy window:** Το σύνολο όλων των αντικειμένων που χρησιμοποιούνται και έχουν τοποθετηθεί στην τρέχουσα σκηνή. [35]



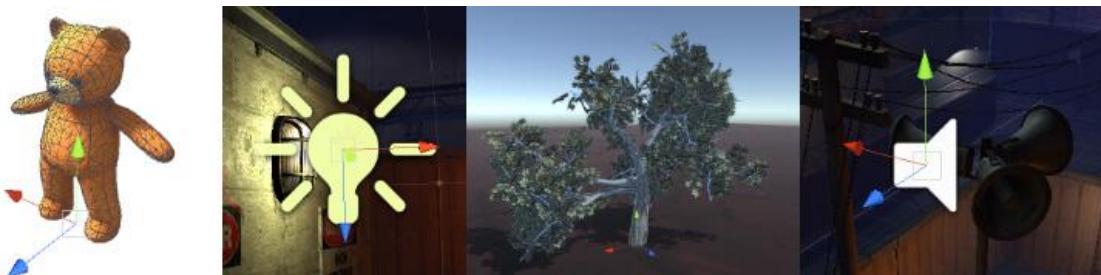
- **Inspector window:** Στο παράθυρο αυτό εμφανίζονται τα χαρακτηριστικά των αντικειμένων που βρίσκονται στο Hierarchy ή στο Project/Assets ανάλογα με το είδος του αντικειμένου. Η θέση, η περιστροφή και το μέγεθος είναι βασικά χαρακτηριστικά για όλα τα αντικείμενα. Ακόμη από εδώ γίνεται η εισαγωγή όλων των ειδών Components όπως ήχο, animations, C# Scripts, Rigid Body, Sprites και άλλα χαρακτηριστικά για το Game Object. [36]



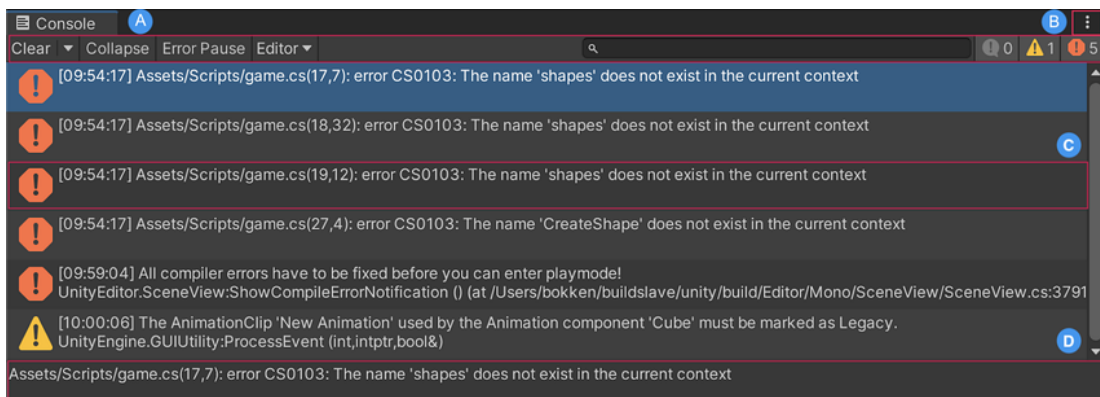
- **Game view:** Το παράθυρο Game View παρέχει τη δυνατότητα στον χρήστη να βλέπει και να τρέχει το τελικό αποτέλεσμα της εφαρμογής ή του παιχνιδιού που δημιουργεί. Είναι ένα χρήσιμο εργαλείο για Debugging καθώς παρέχει στον προγραμματιστή την δυνατότητα να βλέπει την κονσόλα, τον Editor και να δοκιμάζει την εφαρμογή. [37]



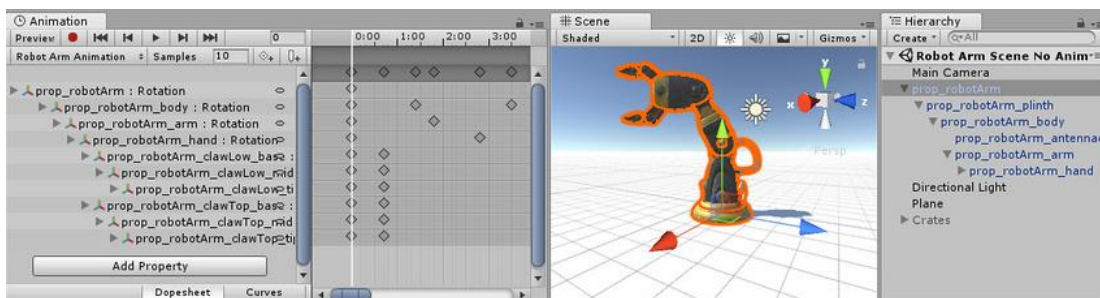
- **GameObjects:** Τρισδιάστατα μοντέλα, φώτα, κάμερα, πηγή ήχου, με τα οποία μπορούμε να χρησιμοποιήσουμε και να προσθέσουμε ιδιότητες μέσω του hierarchy window ανάλογα με τις επιθυμίες μας. [38]



- **Lighting window:** Το παράθυρο Lighting είναι το κύριο σημείο ελέγχου για τις λειτουργίες φωτισμού του Unity. Μπορείτε να χρησιμοποιήσετε το παράθυρο Φωτισμός για να προσαρμόσετε τις ρυθμίσεις που σχετίζονται με το φωτισμό στο Σκηνικό σας και να βελτιστοποιήσετε τα προ-υπολογισμένα δεδομένα φωτισμού για την ποιότητα, το χρόνο ψησίματος και το χώρο αποθήκευσης. [39]
- **Console window:** Αυτό το παράθυρο είναι για Debugging και χρησιμοποιείται για να εκτυπώνονται μηνύματα και τυχόν σφάλματα. Στη συγκεκριμένη περίπτωση, μπορεί να επιλεγεί η γραμμή που βρέθηκε το σφάλμα και να κατευθυνθούμε στο Visual Studio Editor. [40]

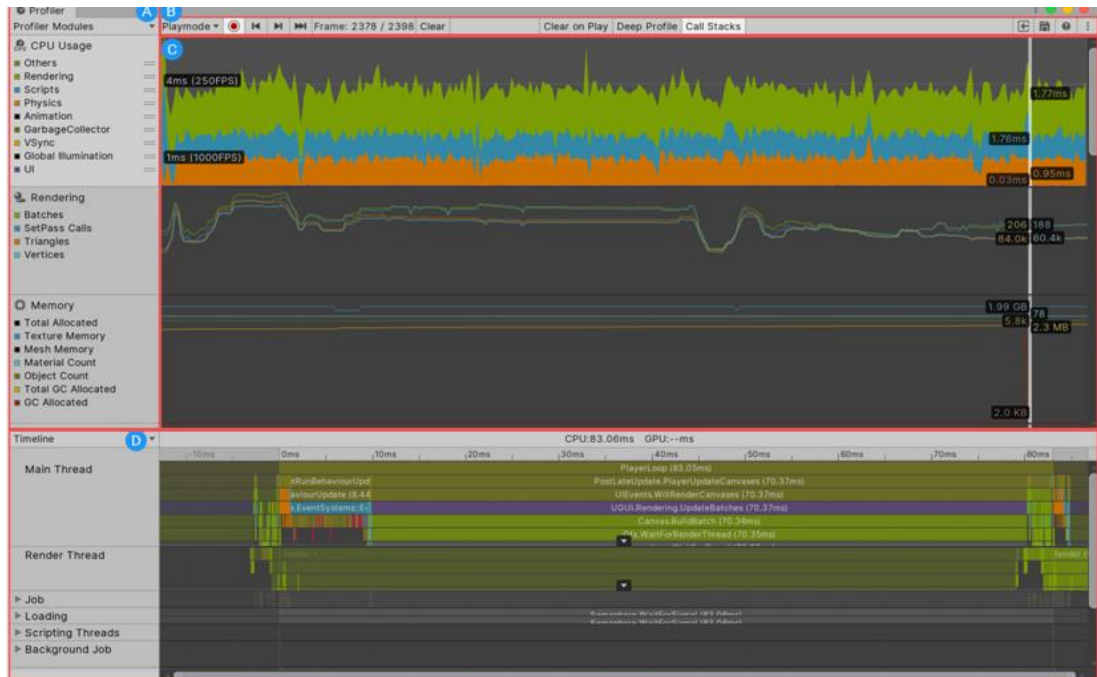


- **Animation view:** Το Παράθυρο Animation στο Unity επιτρέπει τη δημιουργία και την τροποποίηση Animation clips απευθείας μέσα στο Unity. Έχει σχεδιαστεί για να λειτουργεί ως μια ισχυρή και απλή εναλλακτική λύση σε σχέση με εξωτερικά προγράμματα 3D animation. Εκτός από την εμφύχωση της κίνησης, ο επεξεργαστής επιτρέπει επίσης την απόδοση σχεδιοκίνησης (rendering) σε μεταβλητές υλικών και εξαρτημάτων και να ενισχύει τα Animation clips με συμβάντα, συναρτήσεις που καλούνται σε συγκεκριμένα σημεία κατά μήκος της γραμμής χρόνου (Timeline). [41]



- **Profiler window:** Το Unity Profiler είναι ένα εργαλείο που συγκεντρώνει και εμφανίζει δεδομένα σχετικά με την απόδοση της εφαρμογής σε τομείς όπως η CPU, η μνήμη, ο renderer, ο ήχος, κ.α. Είναι ένα χρήσιμο εργαλείο για τον εντοπισμό περιοχών για βελτίωση των επιδόσεων της εφαρμογής. Μπορείτε να επιθεωρήσετε τον κώδικα δέσμης ενεργειών και τον τρόπο με τον οποίο η εφαρμογή χρησιμοποιεί ορισμένα Assets και πόρους που ενδέχεται να την επιβραδύνουν. Εμφανίζει τα αποτελέσματα σχετικά με τις επιδόσεις της εφαρμογής σε μια σειρά διαγραμμάτων όπου εμφανίζονται αιχμές στις επιδόσεις της. Μπορείτε να χρησιμοποιήσετε αυτές

τις πληροφορίες για να λάβετε τεκμηριωμένες αποφάσεις σχετικά με το τι μπορεί να χρειαστεί να βελτιστοποιήσετε στην εφαρμογή σας και για να επιβεβαιώσετε ότι οι βελτιστοποιήσεις σας παράγουν τα αποτελέσματα που περιμένετε. Μπορείτε να το συνδέσετε σε συσκευές στο δίκτυό σας, ή σε συσκευές συνδεδεμένες στο μηχάνημά σας για να ελέγξετε τον τρόπο εκτέλεσης της εφαρμογής στην προβλεπόμενη πλατφόρμα έκδοσης και να συγκρίνετε την απόδοση της εφαρμογής σε αυτές. [42]



## 2.4.2 Unity AR foundation

Το AR Foundation [43] επιτρέπει τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας (AR) πολλαπλών πλατφορμών με το Unity. Σε ένα έργο AR Foundation, επιλέγετε ποια χαρακτηριστικά AR θα ενεργοποιήσετε προσθέτοντας τα αντίστοιχα στοιχεία διαχειριστή στη σκηνή σας. Όταν κατασκευάζετε και εκτελείτε την εφαρμογή σε μια συσκευή, το AR Foundation ενεργοποιεί αυτά τα χαρακτηριστικά χρησιμοποιώντας το εγγενές AR SDK της πλατφόρμας, ώστε να μπορείτε να δημιουργήσετε μία φορά και να την αναπτύξετε στις κορυφαίες πλατφόρμες AR του κόσμου.



Εικόνα 53. Unity AR Foundation

Πηγή: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>

Το πακέτο AR Foundation περιέχει διεπαφές για χαρακτηριστικά AR, αλλά δεν υλοποιεί κανένα χαρακτηριστικό. Για να χρησιμοποιήσετε το AR Foundation σε μια πλατφόρμα-στόχο, χρειάζεστε επίσης ένα ξεχωριστό πακέτο πρόσθετου παρόχου για την εν λόγω πλατφόρμα. Η Unity υποστηρίζει επίσημα τα ακόλουθα πρόσθετα παρόχου:

- Google ARCore XR Plug-in στο Android
- Apple ARKit XR Plug-in στο iOS
- OpenXR Plug-in στο HoloLens 2

Το πακέτο AR Foundation διαθέτει πολλά managers για διάφορες λειτουργίες χρησιμοποιώντας το native AR SDK. Μερικές από αυτές είναι οι εξής:

- **plane detection:** Αναλύει τον χώρο και ανιχνεύει τις επιφάνειες για διάφορους λόγους, όπως για παράδειγμα, την τοποθέτηση ενός αντικειμένου.
- **Track image:** Με την ανίχνευση εικόνας μπορούμε να τοποθετήσουμε ένα εικονικό αντικείμενο αντί για ένα πραγματικό. Οποιαδήποτε εικόνα έχει τραβηχτεί με μία φωτογραφική μηχανή ή ένα κινητό και έχει τοποθετηθεί στην βιβλιοθήκη του AR track image manager μπορεί να ανιχνευθεί και μόλις συμβεί αυτό να ενεργοποιηθεί οποιαδήποτε λειτουργία θελήσει ο προγραμματιστής.
- **Raycast:** Με την λειτουργία του raycast ο προγραμματιστής μπορεί πάλι να κάνει ότι ενέργεια θελήσει, αλλά αυτήν την φορά πατώντας πάνω στο αντικείμενο το οποίο έχει αναφερθεί στο raycast manager.
- **Body and face tracking:** Ανίχνευση προσώπου και σώματος.
- **anchors:** Ανιχνεύει ένα συγκεκριμένο σημείο και το μαρκάρει για οποιαδήποτε χρήση.
- **Meshes:** Δημιουργία διαφόρων χρωμάτων και σχεδίων που βρίσκουμε στο περιβάλλον.

## 3 ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

---

### 3.1 Software Development Life-Cycle

Το Software Development Life-Cycle (SDLC) [44] είναι μία μεθοδολογία, η οποία εξοικονομεί κόστος και χρόνο. Μερικά από τα οφέλη αυτής είναι ο πιο αποδοτικός σχεδιασμός και προγραμματισμός, λιγότερο ρίσκο, το κόστος και η καλύτερη απόδοση των προγραμματιστών. Πιο αναλυτικά η διαδικασία του SDLC είναι:

- **Ανάλυση:** Η συγκεκριμένη διαδικασία περιέχει αναλύσεις σχετικά με το κόστος, τον προγραμματισμό και τους πόρους που θα χρειαστεί η εφαρμογή. Για να γίνει αυτό μαζεύονται πληροφορίες από ειδικούς, από διαχειριστές και από απλούς χρήστες.
- **Σχεδιασμός:** Σε αυτήν την φάση γίνεται η ανάλυση των απαιτήσεων της εφαρμογής. Οι προγραμματιστές ψάχνουν την καλύτερη λύση για την δημιουργία της εφαρμογής όπως τα καλύτερα εργαλεία τα οποία θα χρησιμοποιήσουν και οι τεχνολογικές επιλογές που θα αποδώσουν καλύτερα.
- **Υλοποίηση:** Στην φάση της υλοποίησης οι προγραμματιστές αναλύουν των κώδικα για να μικρύνουν τις διαδικασίες που πρέπει να κάνουν καθημερινά.
- **Δοκιμή:** Μόλις τελειώσει η διαδικασία της υλοποίησης περνάμε στην δοκιμή, οι προγραμματιστές ελέγχουν για τυχών σφάλματα και αν η εφαρμογή πληροί τις προϋποθέσεις των απαιτήσεων του χρήστη.
- **Ανάπτυξη:** Στην φάση της ανάπτυξης χρησιμοποιείται για δοκιμή ένα αντίγραφο της κανονικής εφαρμογής για αναβαθμίσεις ή διορθώσεις. Έτσι ο χρήστης μπορεί να έχει κανονικά πρόσβαση στην εφαρμογή και αν γίνουν τυχών λάθη στο αντίγραφο να μην επηρεάσουν την κανονική εφαρμογή.
- **Συντήρηση:** Τέλος υπάρχει και η φάση της διατήρησης όπου εκεί πραγματοποιείται ο έλεγχος για τα σφάλματα. Ψάχνονται λύσεις για τα προβλήματα διαφόρων χρηστών και ελέγχεται η βελτιστοποίηση και η ασφάλεια της εφαρμογής.

### 3.2 Εργαλεία εφαρμογής

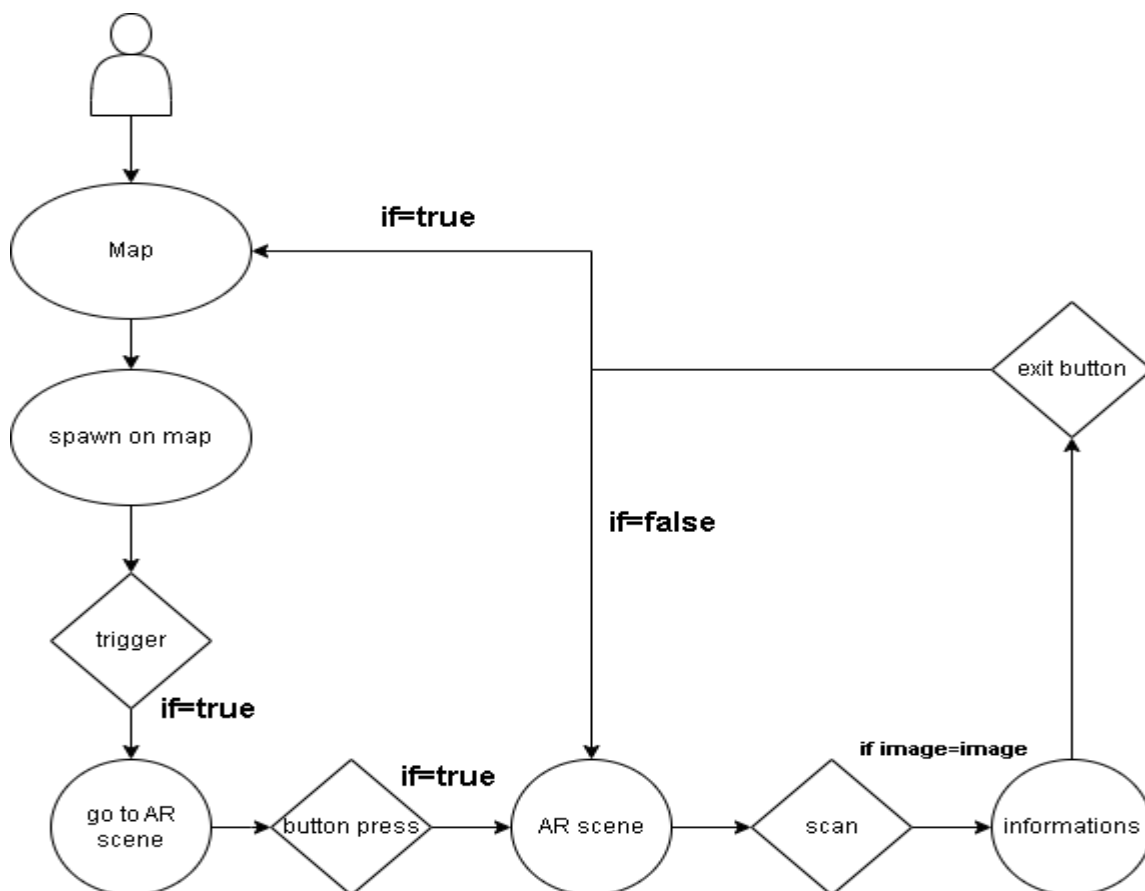
Η εφαρμογή “exploreKastoria” δημιουργήθηκε στην πλατφόρμα Unity. Χρησιμοποιήθηκε το Marbox για την δημιουργία του χάρτη, τα API directions για την πλοήγηση και το spawn on map για την τοποθέτηση των μνημείων. Επίσης, στο Mixed reality χρησιμοποιήθηκε το εργαλείο του Unity, το AR foundation για το άνοιγμα και την λειτουργία της κάμερας και το σκανάρισμα των εικόνων. Τέλος, όλα τα scripts έγιναν με C# στο Microsoft visual studio.

### 3.3 Περιγραφή της εφαρμογής

Η συγκεκριμένη εφαρμογή δημιουργήθηκε με σκοπό ενίσχυση της εμπειρίας ξενάγησης του χρήστη στα αξιοθέατα και μνημεία της πόλης της Καστοριάς, μέσω χρήσης επαυξημένης πραγματικότητας σε έξυπνες κινητές συσκευές Android. Ο χρήστης επιλέγει το σημείο ενδιαφέροντος (μνημείο-αξιοθέατο) στο οποίο επιθυμεί να περιηγηθεί. Με τη βοήθεια των χαρτών πλοήγησης και των οδηγιών που διαθέτει η εφαρμογή, ο χρήστης καθοδηγείται από το σημείο που βρίσκεται μέχρι το σημείο ενδιαφέροντος. Όταν ο χρήστης πλησιάσει κοντά στο σημείο ενδιαφέροντος, η εφαρμογή επαυξημένης πραγματικότητας ενεργοποιείται, μέσω του ανοίγματος της κάμερας της έξυπνης κινητής συσκευής Android. Σκανάροντας το QR code που βρίσκεται στην περιοχή του σημείου ενδιαφέροντος (μνημείο-αξιοθέατο), εμφανίζονται στην οθόνη της έξυπνης κινητής συσκευής Android, συνοπτικές πληροφορίες για το συγκεκριμένο αξιοθέατο, μέσω τεχνικών μικτής κι επαυξημένης πραγματικότητας. Τέλος, πατώντας το κουμπί «Έξοδος» μεταφέρεται στην αρχική οθόνη του χάρτη πλοήγησης της εφαρμογής, προκειμένου να συνεχιστεί η πλοήγηση.

### 3.4 Δομή της εφαρμογής

Η ολοκληρωμένη δομή και το διάγραμμα ροής αποτυπώνεται στην παρακάτω εικόνα:

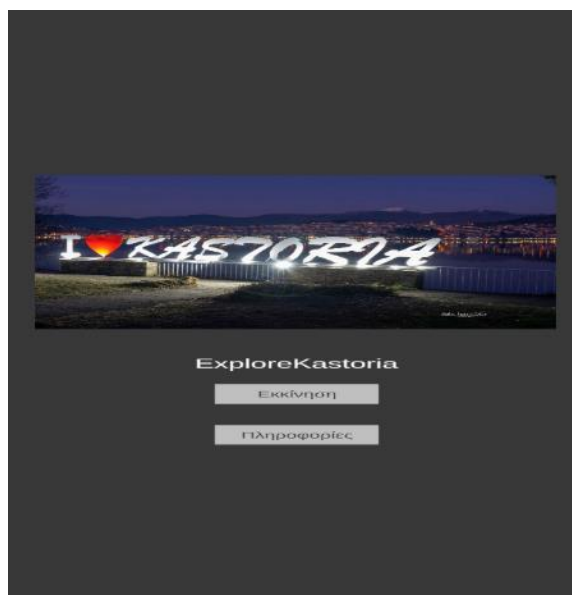


Εικόνα 54. Διάγραμμα ροής εφαρμογής “exploreKastoria”

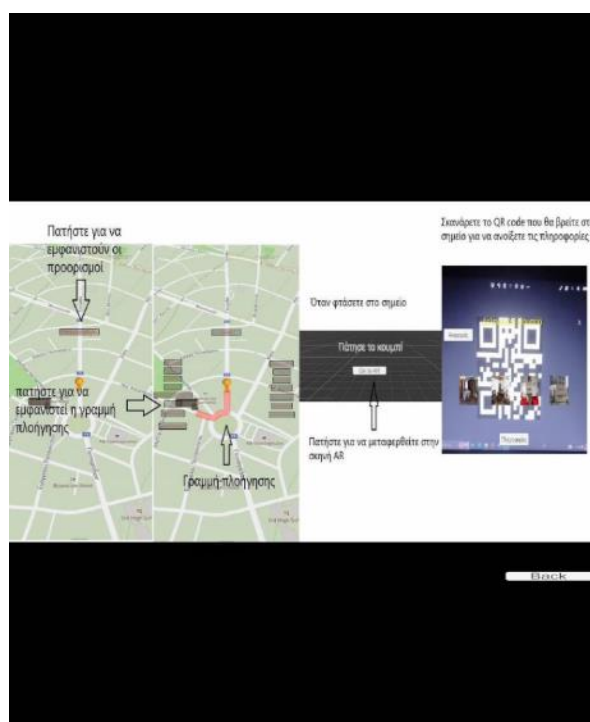


## Αρχική σκηνή και Χάρτης

Ανοίγοντας την εφαρμογή έχουμε την αρχική σκηνή στην οποία περιέχονται δύο κουμπιά, ένα της εκκίνησης και ένα των πληροφοριών. Πατώντας της εκκίνησης ξεκινά η εφαρμογή και πατώντας τις πληροφορίες μας ανοίγει ένας νέος καμβάς δείνοντας μας πληροφορίες σχετικά με την εφαρμογή αλλά και ένα κουμπί back που με το πάτημά του μας επιστρέφει στην αρχική σκηνή.

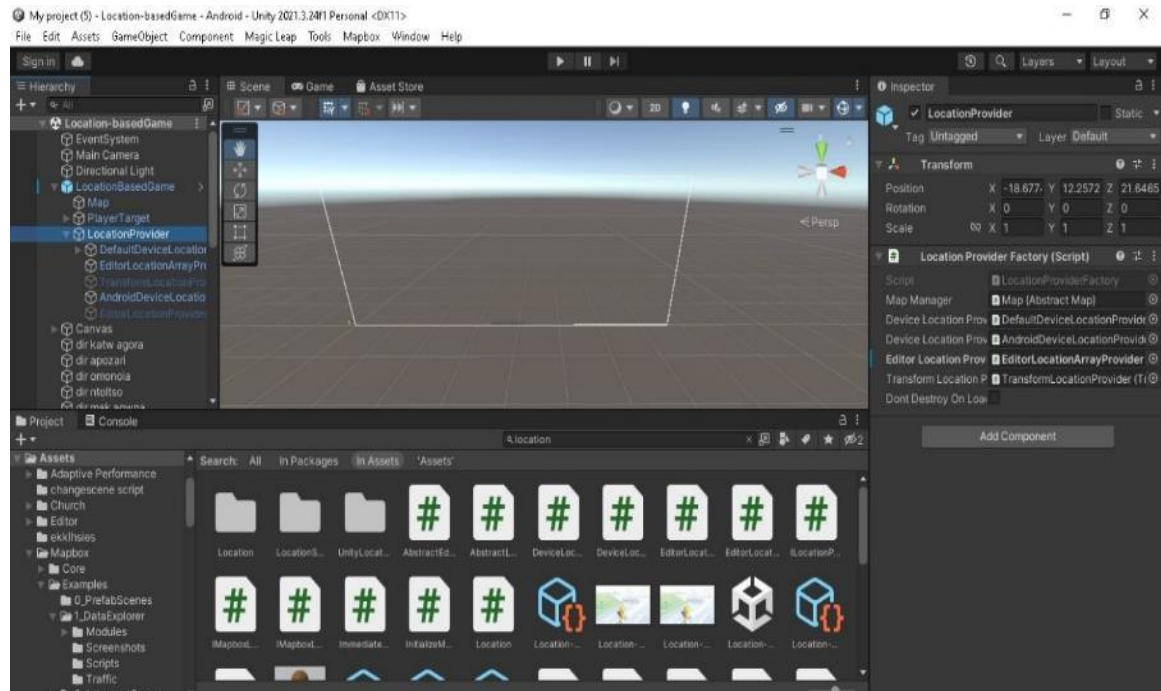


Εικόνα 55. Αρχική σκηνή



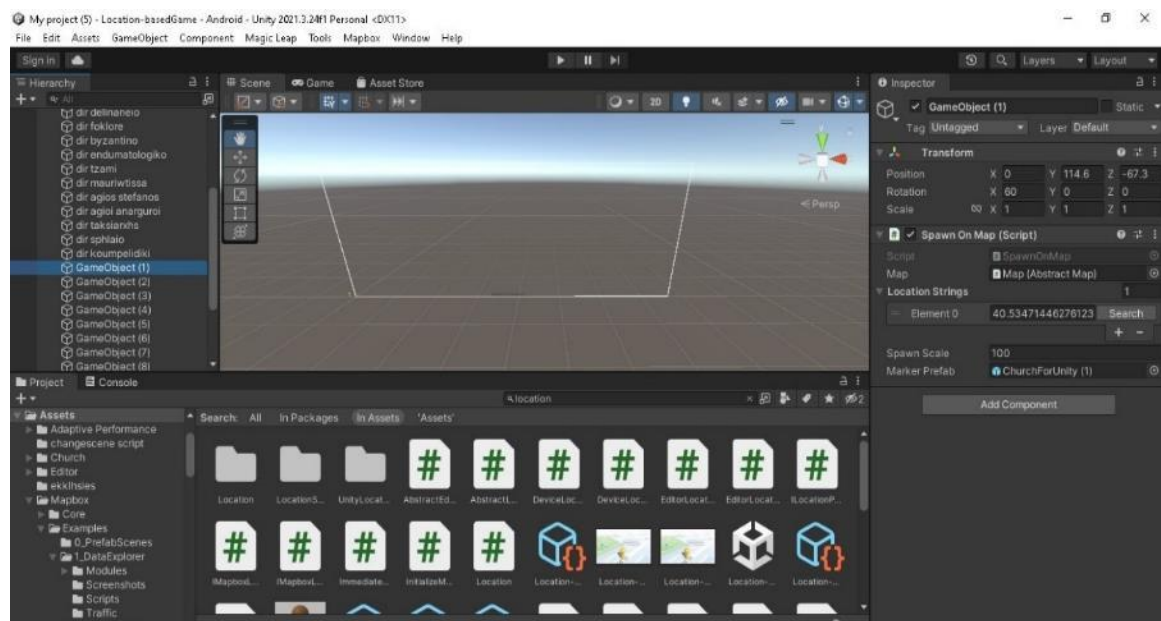
Εικόνα 56. Πληροφορίες

Πατώντας την εκκίνηση εμφανίζεται ο χάρτης στην τοποθεσία όπου βρισκόμαστε μέσω του GPS, που μας δίνεται από το εργαλείο του Mapbox, [Location provider](#).



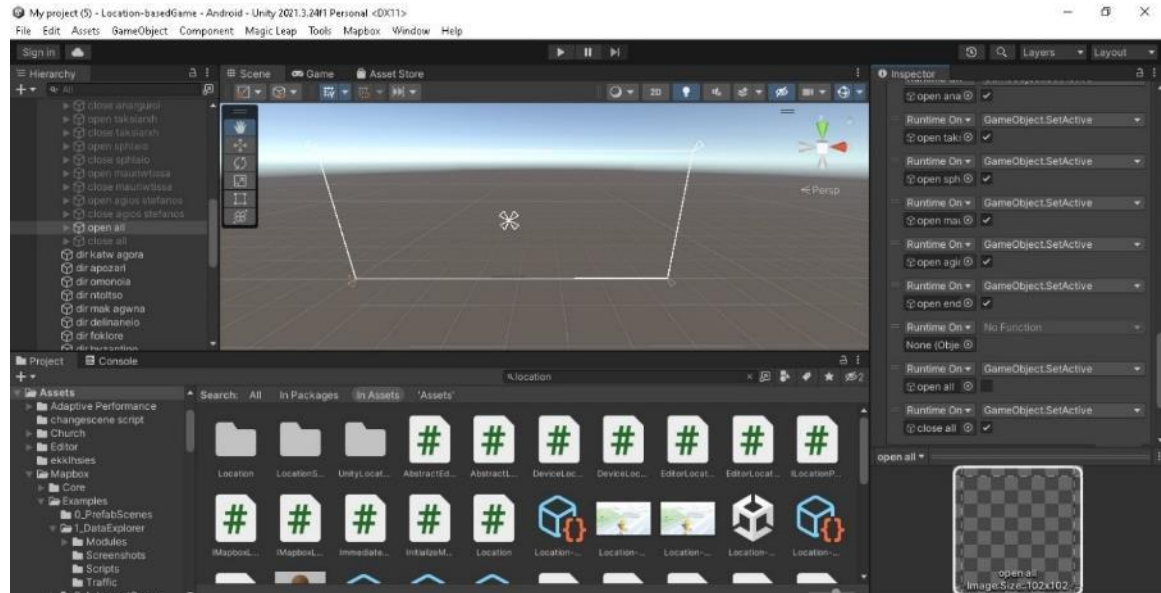
Εικόνα 57. Location provider

Επίσης εμφανίζονται και objects τα οποία έχω τοποθετήσει στα σημεία που βρίσκονται τα μνημεία και τα αξιοθέατα, μέσω του script [spawn](#) του Mapbox, τέλος εμφανίζεται και ο παίκτης στο σημείο που βρισκόμαστε μέσω του script [ImmediatePosition](#) του Mapbox.



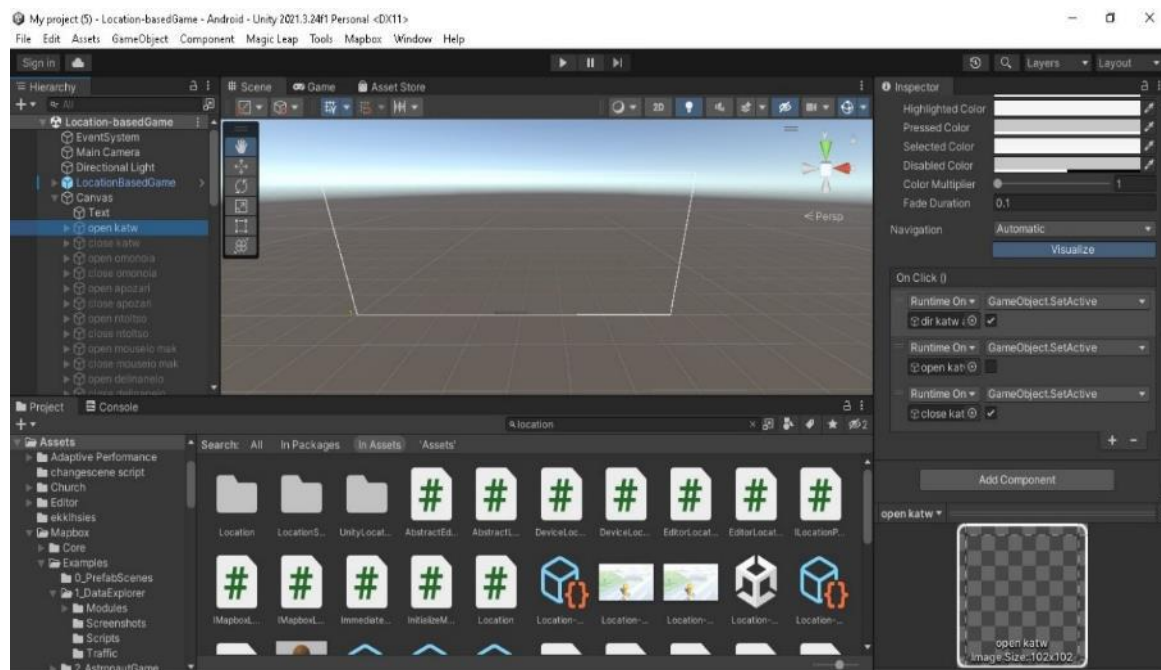
Εικόνα 58. Spawn on map

Στην οθόνη εμφανίζεται και ένα button για το άνοιγμα των οδηγιών. Πατώντας το απενεργοποιείται και ενεργοποιείται το κουμπί έξοδος. Εν ολίγοις στην θέση του εμφανίζεται το κείμενο έξοδος.Επίσης, εμφανίζονται και τα κουμπιά τα οποία αναφέρουν τα μνημεία και αξιοθέατα που έχουν τοποθετηθεί.

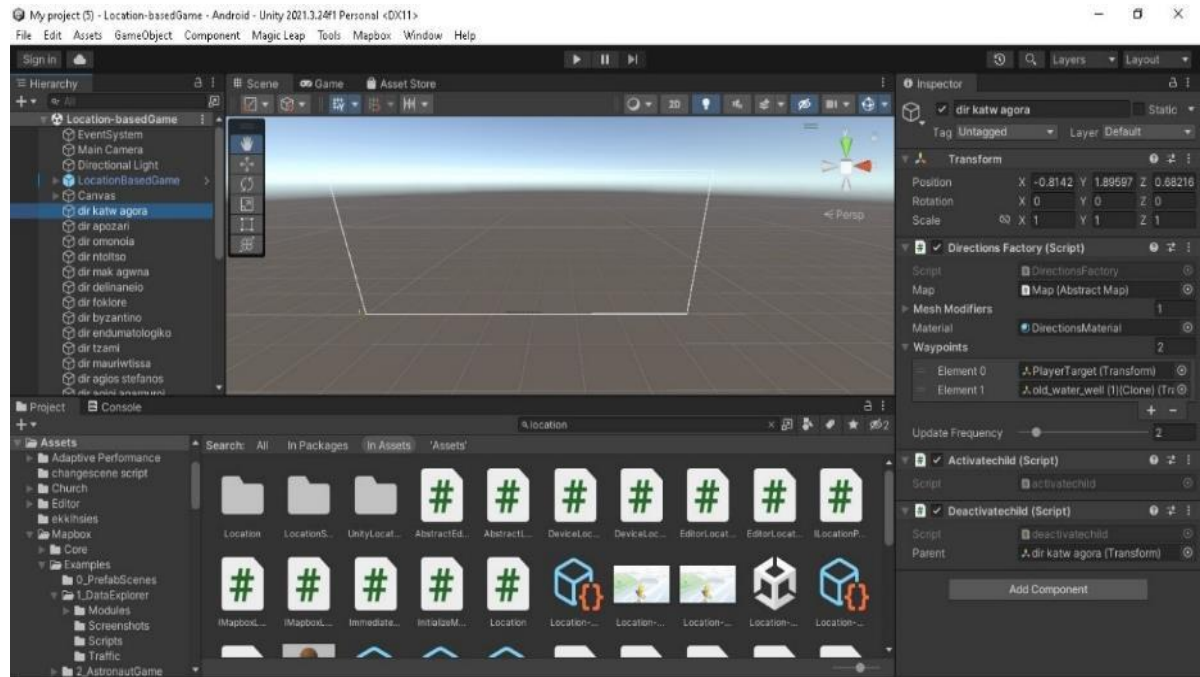


Εικόνα 59. Location button

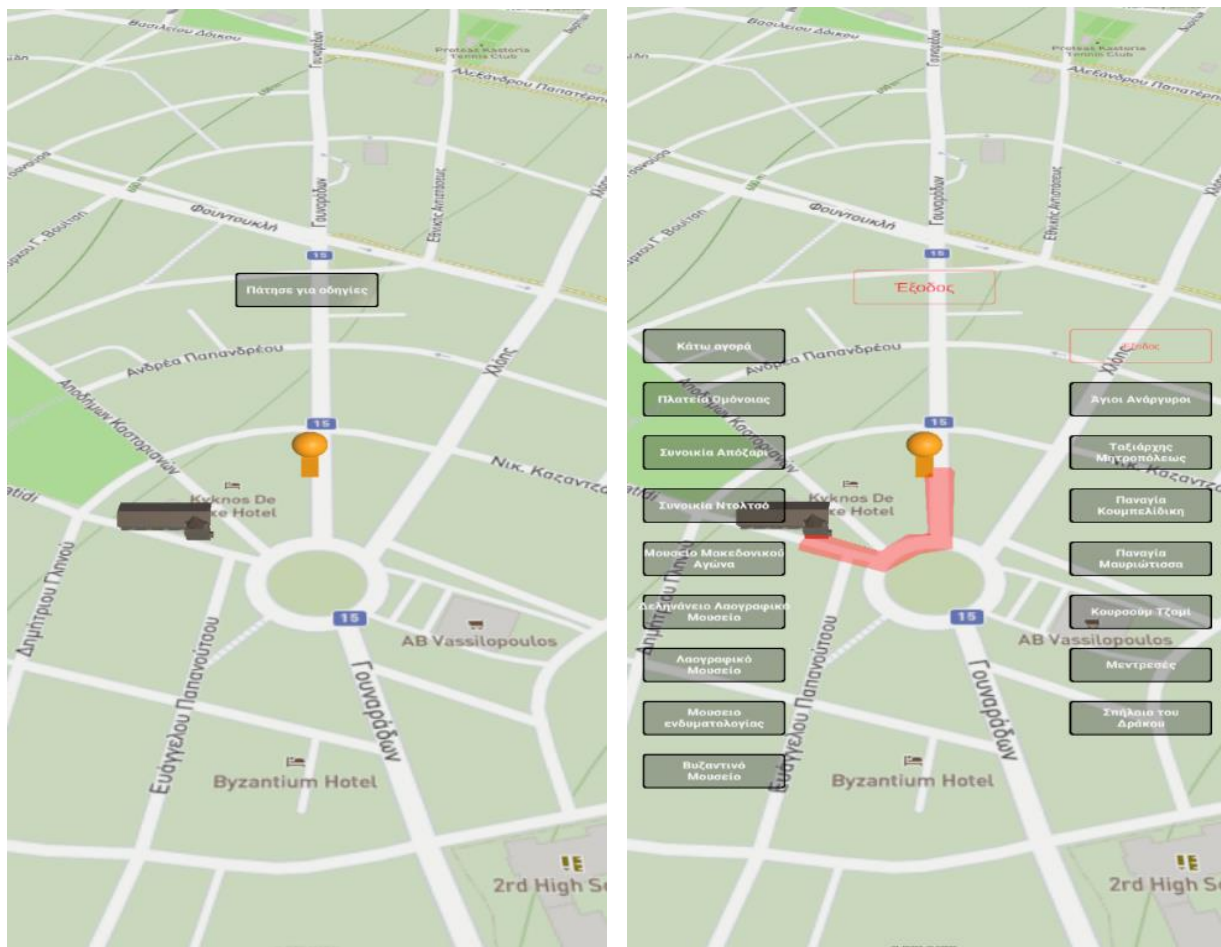
Πατώντας ένα από αυτά γίνεται το ίδιο στην περίπτωση του κουμπιού έξοδος και ενεργοποιείται το script [directionsfactory](#) του mapbox. Στην συνέχεια εμφανίζεται μία γραμμή πλοήγησης στο σημείο ενδιαφέροντος η οποία αναφέρεται μόνο στους κεντρικούς δρόμους,



Εικόνα 60. Directions button

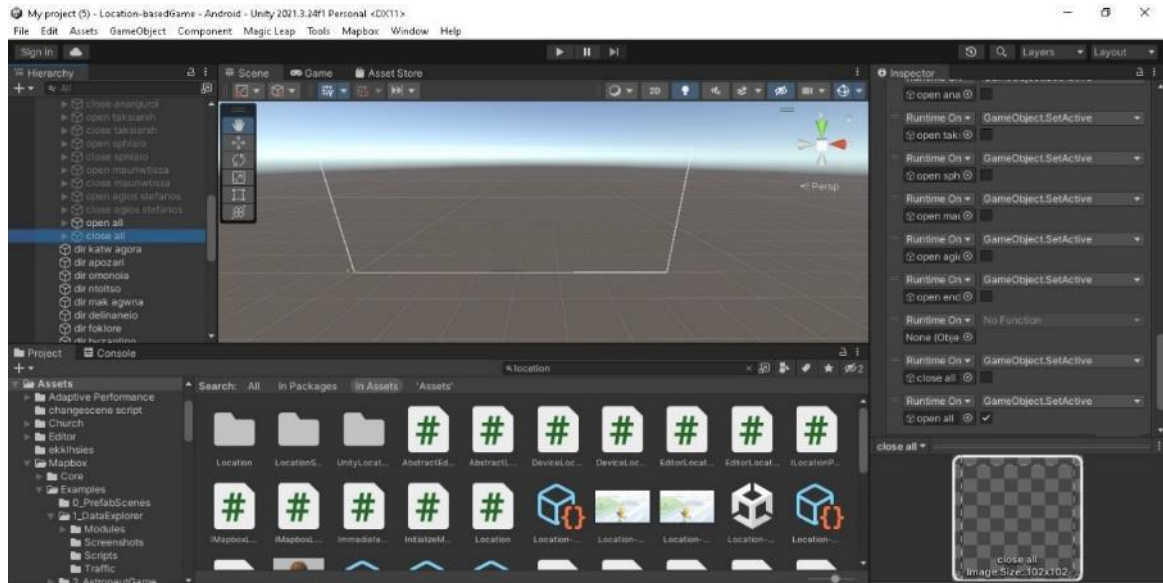


Εικόνα 61. Directions Factory



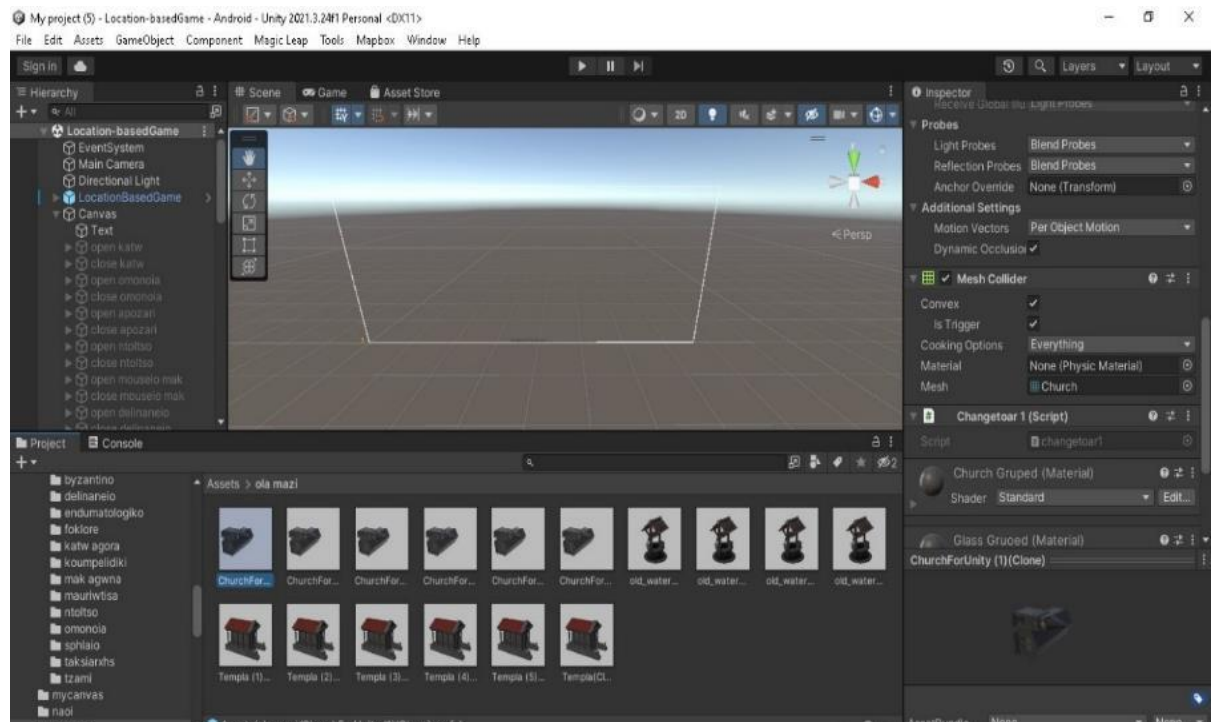
Εικόνα 62. Directions screenshots

Το κουμπί έξοδος απενεργοποιεί το script και το κουμπί έξοδος των οδηγιών απενεργοποιεί όλα τα κουμπιά εκτός από τα κουμπιά εξόδου αν υπάρχουν ακόμα ενεργές οδηγίες.



Εικόνα 63. Close button

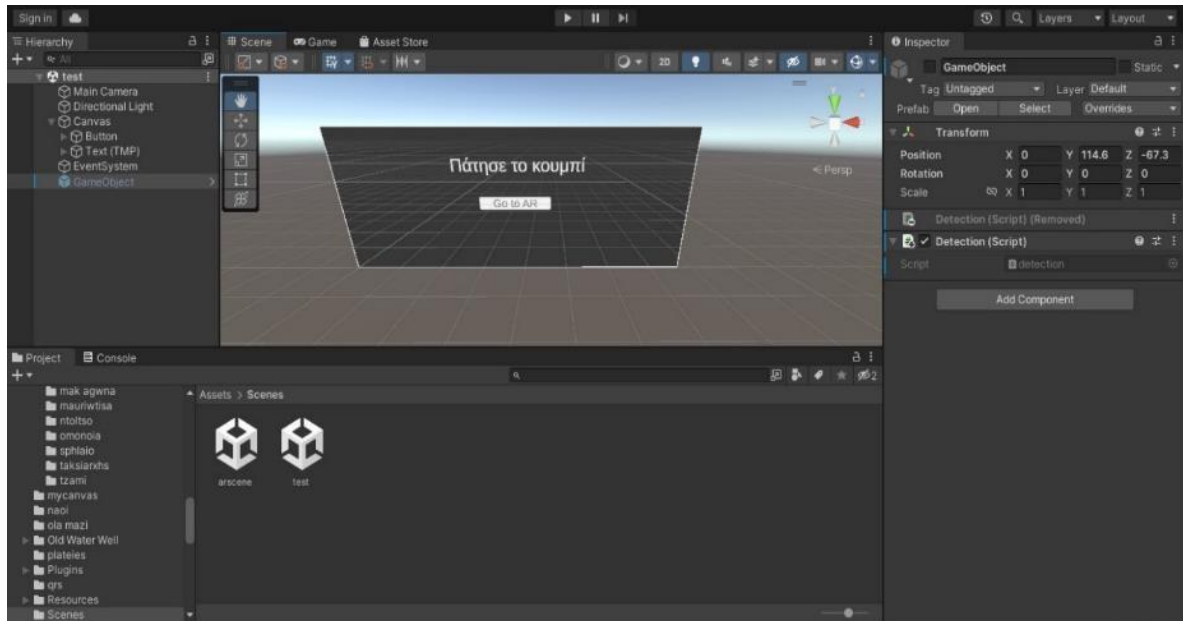
Φτάνοντας κοντά στο αντικείμενο ενεργοποιείται το trigger αλλαγής σκηνής μέσω των colliders και του script που δημιουργήσα [changetoar1](#).



Εικόνα 64. Changetoar1 trigger script

## 2<sup>η</sup> σκηνή: Ομαλή μετάβαση σε AR

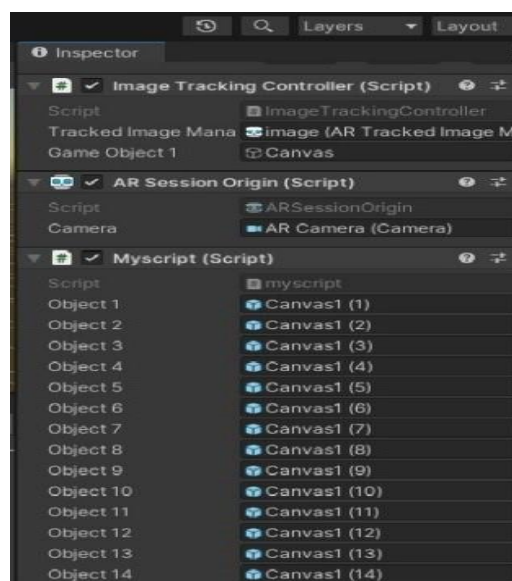
Αυτή η σκηνή δημιουργήθηκε για μία ομαλή μετάβαση από τον χάρτη σε μία σκηνή AR μέσω κάμερας. Σε αυτήν θα εμφανιστεί ένα κείμενο που μας ζητείτε να πατήσουμε το κουμπί που γράφει go to AR. Όταν το πατήσουμε ενεργοποιείται το script [detection](#) και αλλάζουμε σκηνή.



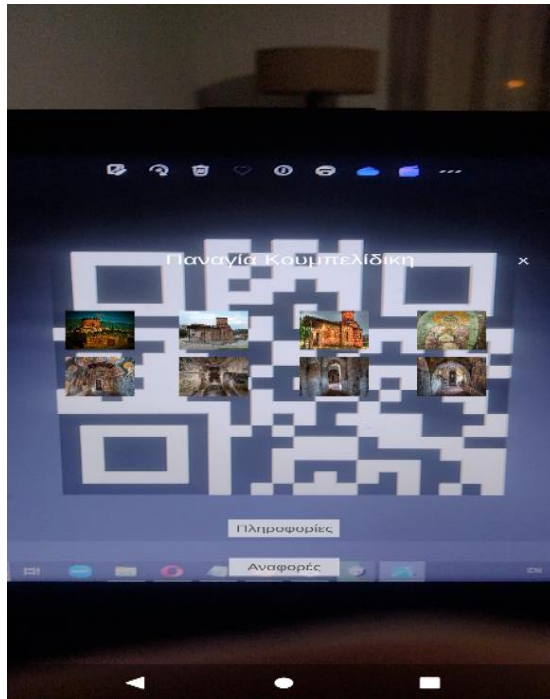
Εικόνα 65. Second Scene detection script for change scene

## 3<sup>η</sup> σκηνή: AR και MR

Στην τρίτη και τελευταία σκηνή ενεργοποιείται η κάμερα και εμφανίζεται ένα κείμενο «βρες και σκάνανε το qr code» μόλις βρεθεί και σκαναριστεί μέσω του track image manager του AR foundation της Unity, του [image tracking controller](#) και του [myscript](#) εμφανίζονται εικόνες και κουμπιά πληροφοριών κι αναφορών για το συγκεκριμένο σημείο.

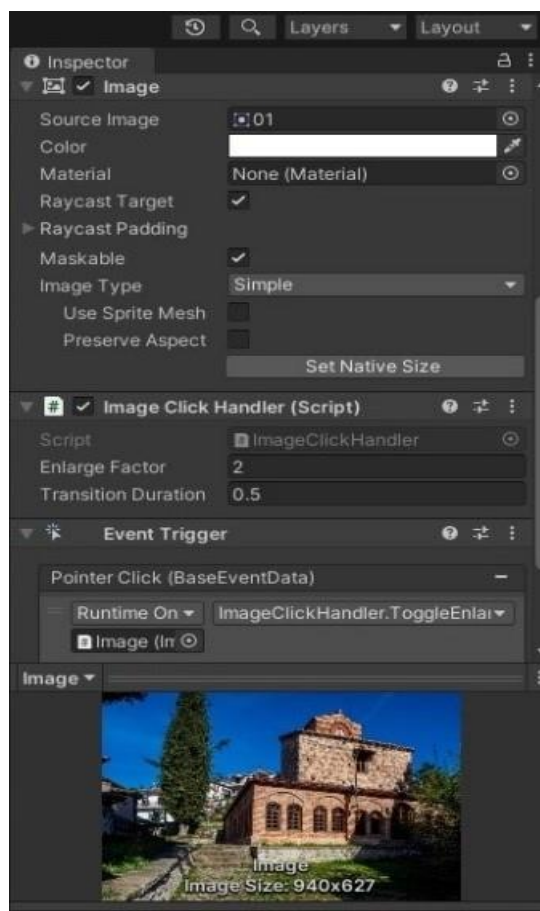


Εικόνα 66. Image track and myscript

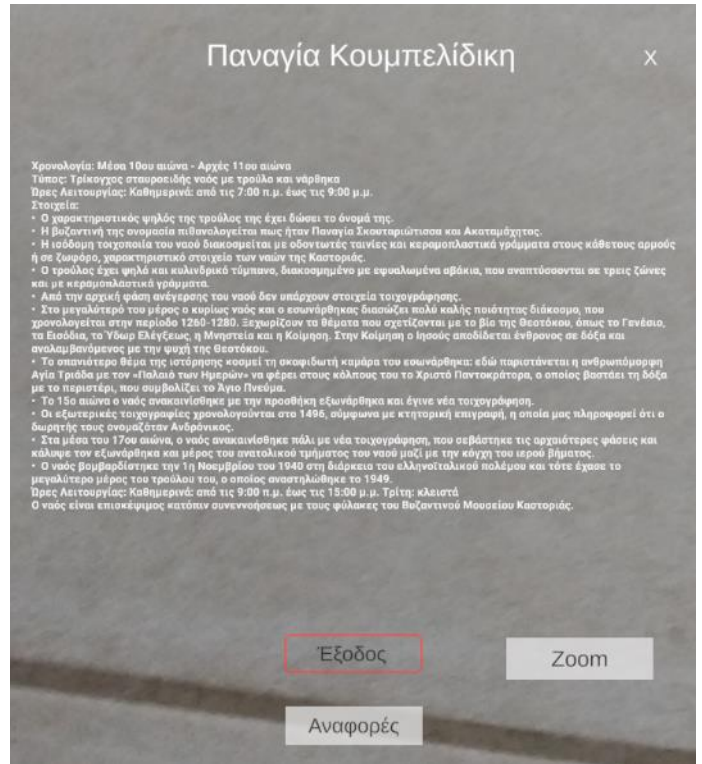
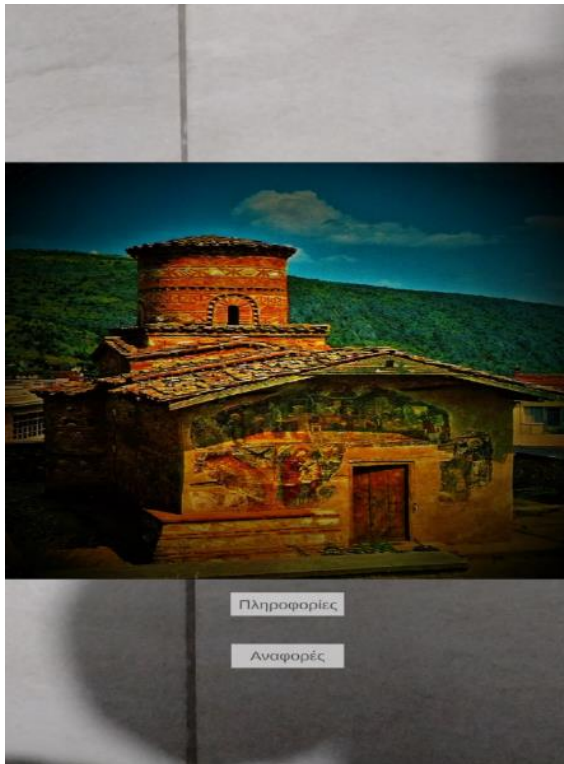


Εικόνα 67. QR scan from app screenshot

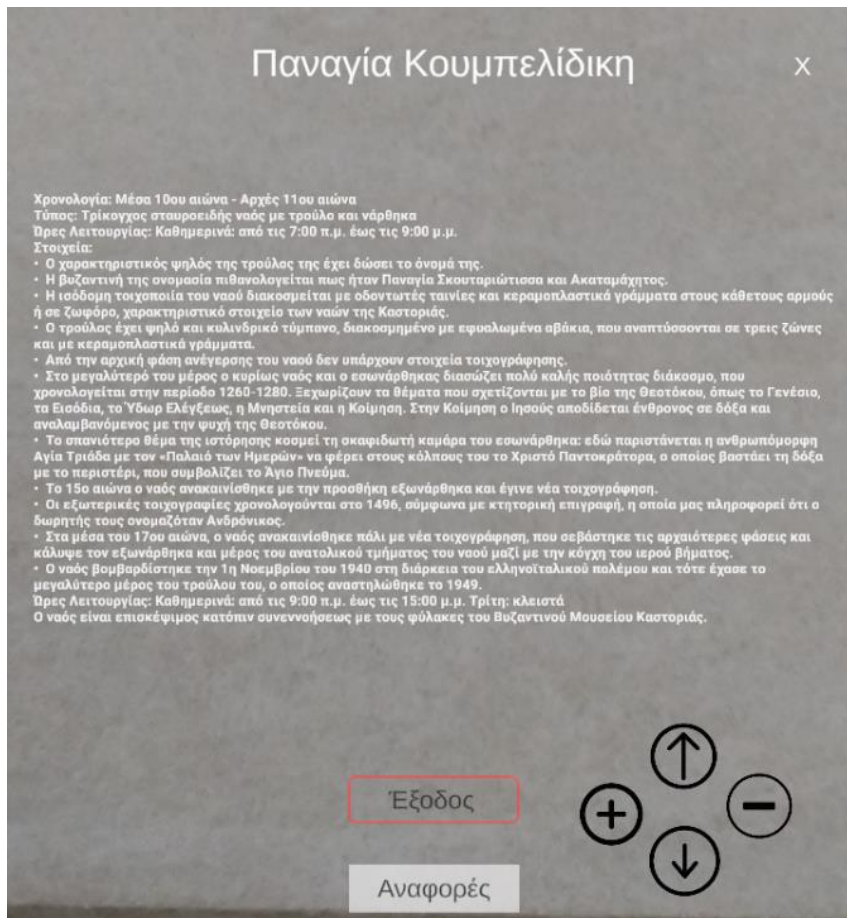
Πατώντας μία οποιαδήποτε εικόνα, αυτή μεγεθύνεται και πατώντας την ξανά επιστρέφει στο αρχικό μέγεθος μέσω του script [Image click handler](#) και του event trigger onclick.



Εικόνα 68. ImageClickHandler script for images



Εικόνα 69. Images from app screenshot



Εικόνα 70. Info and zoom from app screenshots



Πατώντας το κουμπί πληροφορίες απενεργοποιείται και ενεργοποιείται το κουμπί έξοδος. Επίσης, ενεργοποιείται ένας νέος καμβάς με ένα κείμενο για τις πληροφορίες του συγκεκριμένου σημείου που με το κουμπί έξοδος απενεργοποιείται και ενεργοποιείται και πάλι το κουμπί των πληροφοριών. Το ίδιο συμβαίνει και με το κουμπί των αναφορών. Επίσης ενεργοποιείται και το κουμπί ζουμ το οποίο αν το πατήσουμε ενεργοποιεί με την σειρά του 4 νέα κουμπιά που μεγενθύνουν ή σμικρύνουν το κείμενο και το κατευθύνουν πάνω ή κάτω, αυτό γίνεται με την βοήθεια των script [ZoomText](#) και [MoveText](#) .

Πατώντας το κουμπί X (εξόδου), ενώ βρισκόμαστε στις πληροφορίες, μας δίνεται η δυνατότητα να σκανάρουμε ξανά το QR code μέσω του [exitback](#) script.

Τέλος, πατώντας το κουμπί X (εξόδου) στον αρχικό καμβά έξω από τις πληροφορίες ενεργοποιείται το [exit](#) script και μεταφερόμαστε και πάλι στην σκηνή 1 που είναι ο χάρτης.

## Συμπεράσματα

---

Είναι γεγονός πως οι σύγχρονες τεχνολογίες της επαυξημένης και της μικτής πραγματικότητας ενισχύουν σημαντικά την εμπειρία του χρήστη σε τέτοιου είδους εφαρμογές. Για να λειτουργήσουν οι συγκεκριμένες τεχνολογίες απαιτούνται έξυπνες κινητές συσκευές (ή/και εξειδικευμένος εξοπλισμός όπως AR Headsets), που στην εποχή μας χρησιμοποιούνται καθημερινά από το ευρύ κοινό χρηστών, ανεξαρτήτως ηλικίας. Πλέον όσο η τεχνολογία εξελίσσεται θα υπάρξουν πολλές ακόμα καινοτόμες εφαρμογές επαυξημένης και μικτής πραγματικότητας ακόμα και για τις καθημερινές δραστηριότητες του κάθε χρήστη και η εξοικείωσή του με την τεχνολογία θα γίνει πολύ πιο εύκολη.

Μετά την απασχόληση με διάφορα λογισμικά για τη δημιουργία της εφαρμογής συμπεραίνουμε ότι η Unity Game Engine είναι μία πλατφόρμα ανάπτυξης παιχνιδιών και όχι μόνο, που αφορούν ένα ευρύ φάσμα εφαρμογών σε διάφορους τομείς της καθημερινής ζωής, για ποικίλες πλατφόρμες. Επίσης, είναι σχετικά εύκολη στη χρήση και το σημαντικότερο είναι πως διαθέτει πολύ καλή τεκμηρίωση και μεγάλη κοινότητα χρηστών/προγραμματιστών, κάτι που βοηθά πάρα πολύ στην υποστήριξη των νέων προγραμματιστών.

Χρησιμοποιώντας τα διάφορα εργαλεία τα οποία διαθέτει, όπως το AR foundation, η διαδικασία γίνεται απλούστερη. Το πακέτο AR foundation που είναι ενσωματωμένο στη Unity, μας βοηθά στην υποστήριξη των τεχνολογιών κι εφαρμογών της επαυξημένης πραγματικότητας, με τα διάφορα modules που διαθέτει για διάφορες λειτουργίες, όπως το image track, το οποίο και αξιοποίησα για τη λειτουργία σάρωση των QR codes και τη χρήση των σε Marker-based AR. Το εργαλείο Marbox, μας δίνει την δυνατότητα της δημιουργίας ενός δυναμικού χάρτη, η οποία υλοποιείται πολύ εύκολα μέσα στο περιβάλλον της Unity, όπως και η λειτουργία της πλοήγησης σε αυτόν τον δυναμικό χάρτη μέσω του Directions API που διαθέτει.

Η πειραματική εφαρμογή “exploreKastoria” που αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας είναι μία εφαρμογή επαυξημένης και μικτής πραγματικότητας με σκοπό την ενίσχυση της εμπειρίας τουριστικής περιήγησης στα αξιοθέατα της πόλης της Καστοριάς. Είναι πλήρως λειτουργική και καλύπτει ένα πολύ μεγάλο μέρος των αρχικών απαιτήσεων. Με την πλοήγηση μέσα από δυναμικούς χάρτες και με τη βοήθεια της τεχνολογίας GPS, ο χρήστης μπορεί να εξερευνήσει, πιο εύκολα και γρήγορα, μερικά από τα πιο σημαντικά αξιοθέατα και μνημεία της Καστοριάς κι έτσι να έχει στη διάθεσή του πληθώρα ψηφιακών πληροφοριών σε ποικίλες μορφές (κείμενο, εικόνα, ήχος, animation, video) για τα σημεία που βρίσκεται, χρησιμοποιώντας τεχνολογίες επαυξημένης και μικτής πραγματικότητας, κάτι που αποδεδειγμένα ενισχύει τη εμπειρία ξενάγησης και περιήγησης.

## Προτάσεις για μελλοντικές επεκτάσεις

Μερικές από τις βελτιώσεις που μπορούν να γίνουν στην συγκεκριμένη εφαρμογή είναι η επέκταση αυτής και σε άλλα μέρη, έξω από την πόλη της Καστοριάς και η τοποθέτηση περισσότερων μνημείων και αξιοθέατων σε αυτήν.

Επιπροσθέτως, μπορεί να αναβαθμιστεί η λειτουργία της, ώστε να μπορεί να αξιοποιηθεί και σε συσκευές IOS, αλλά και στο web. Μπορεί να τοποθετηθεί ένα search bar, αντί για buttons, αν προστεθούν και άλλα αντικείμενα για την πλοήγηση στον χάρτη και η μεταφορά από τη μία σκηνή στην άλλη να γίνεται μέσω των συντεταγμένων (Location-based AR) και όχι μόνο μέσω δεικτών κι αντικειμένων (Marker-based AR) για να αποφευχθούν τυχόν λάθη.

Επίσης, άλλο ένα σημείο είναι η βελτίωση των οδηγιών προσανατολισμού και κατευθύνσεων (directions) προς τα επιλεγμένα αξιοθέατα. Στο σημείο αυτό, μπορούν να προστεθούν και άλλες δυνατότητες οδηγιών πλοήγησης με διάφορα μεταφορικά μέσα, όπως μια διαδρομή με τα πόδια, όχι μόνο από κεντρικούς δρόμους, αλλά και να προστεθεί ο χρόνος που θα χρειαστεί ο χρήστης για να φτάσει σε έναν συγκεκριμένο προορισμό.

Τέλος να αναφέρουμε ότι με την αξιολόγηση των διαφόρων χρηστών που θα χρησιμοποιήσουν την συγκεκριμένη εφαρμογή, όταν θα διατεθεί προς δοκιμή από το κοινό, είναι βέβαιο πως θα υπάρξουν περαιτέρω προτάσεις βελτιώσεων των λειτουργιών της εφαρμογής που μπορούν να γίνουν σε επόμενη έκδοσή της.

## Βιβλιογραφία

---

- [1] Z. Wu, J. Guo, S. Zhang, C. Zhao and X. Ma, ""An AR Benchmark System for Indoor Planar Object Tracking"," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019.
- [2] W. Slany, ""A mobile visual programming system for Android smartphones and tablets"," in *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2012.
- [3] R. Van Krevelen, "Augmented Reality: Technologies, Applications, and Limitations.," 10.13140/RG.2.1.1874.7929., Amsterdam, The Netherlands, 2007.
- [4] T. Caudell and D. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes.," in *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences. 2. vol.2.*, 1992.
- [5] J. M. Loomis, R. G. Golledge and R. L. Klatzky, "Personal guidance system for the visually impaired using GPS, GIS, and VR technologies," in *Proceedings of the First Annual International Conference Virtual Reality and Persons with Disabilities*, 1993.
- [6] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, 1999.
- [7] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid and D. Hallaway, "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system," *Computers & Graphics*, vol. 23, no. 6, pp. pp. 779-785, 1999.
- [8] G. Reitmayr and D. Schmalstieg, "Mobile collaborative augmented reality," , 2001, pp. 114-123, doi: 10.1109/ISAR.2001.970521., in *Proceedings IEEE and ACM International Symposium on Augmented Reality*, New York, NY, USA, 2001.
- [9] V. Vlahakis, J. Karigiannis, M. Tsotros, M. Gounaris, L. Almeida, D. Stricker, T. Gleue, I. T. Christou, R. Carlucci and N. Ioannidis, "Archeoguide: First results of an augmented reality, mobile computing system in cultural heritage sites," in *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage (VAST '01)*, New York, NY, USA, 2001.
- [10] G. Reitmayr and T. W. Drummond, "Going out: Robust model-based tracking for outdoor augmented reality," in *Fifth IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006)*, Santa Barbara, CA, USA, 2006.
- [11] A. Morrison, A. Oulasvirta, P. Peltonen, S. Lemmelä, G. Jacucci, G. Reitmayr, J. Näsänen and A. Juustila, "Like bees around the hive: A comparative study of a mobile augmented reality map," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, New York, NY, USA, 2009.

- [12] O. Bergig, N. Hagbi, J. El-Sana and M. Billingham, "In-place 3dsketching for authoring and augmenting mechanical systems," in *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, Orlando, Florida, USA, 2009.
- [13] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *9th IEEE International Conference on Computer Vision (ICCV 2003)*, Nice, France, 2003.
- [14] G. Klein and D. Murray, "Parallel Tracking and Mapping on a Camera Phone," in *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, Orlando, Florida, USA, 2009.
- [15] A. K. Noor, "The HoloLens Revolution," The American Society of Mechanical Engineers, 4 October 2016. [Online]. Available: <https://www.asme.org/topics-resources/content/the-hololens-revolution>. [Accessed 31 August 2023].
- [16] D. Smith, "How Does An Accelerometer Work In A Smartphone? Bill Hammack, The Engineer Guy, Explains," International Business Times, 23 May 2012. [Online]. Available: <https://www.ibtimes.com/how-does-accelerometer-work-smartphone-bill-hammack-engineer-guy-explains-full-text-699762>. [Accessed 31 August 2023].
- [17] "How Does a Gyroscope Sensor Work in Your Smartphone?," TechAhead, 22 August 2019. [Online]. Available: <https://www.techaheadcorp.com/knowledge-center/how-gyroscope-sensor-work-in-smartphone/>. [Accessed 31 August 2023].
- [18] Y. Cai, Y. Zhao, X. Ding and J. M. Fennelly, "Magnetometer basics for mobile phone applications," 2012.
- [19] H. Costa, E. Sprout, S. Teng, M. McDaniel, J. Hunt, D. Boudreau, T. Ramroop, K. Rutledge and H. Hall, "Global Positioning System (GPS) | Education," National Geographic Society, 19 December 2022. [Online]. Available: <https://education.nationalgeographic.org/resource/gps/>. [Accessed 31 August 2023].
- [20] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Transactions on Information and Systems*, Vols. E77-D, no. 12, pp. 1321-1329, 1994.
- [21] J. Tony, "Augmented Reality in Healthcare: Use Cases, Examples and Trends," Fingent, 22 June 2021. [Online]. Available: <https://www.fingent.com/blog/augmented-reality-in-healthcare-use-cases-examples-and-trends/>. [Accessed 31 August 2023].
- [22] S. I. Studio, "Augmented Reality (AR) in Healthcare," Medium.com, 19 March 2019. [Online]. Available: <https://medium.com/swevens/augmented-reality-ar-in-healthcare-3c12bdf86a8e>. [Accessed 31 August 2023].
- [23] L. Bocchibianchi, "How is augmented reality used in construction?," GAMMA AR Technologies S.à r.l., 5 May 2023. [Online]. Available: <https://gamma-ar.com/how-is-augmented-reality-used-in-construction/>. [Accessed 31 August 2023].
- [24] V. Fedorychak, "7 Applications for Augmented Reality (AR) in Construction," SmartTek Solutions LLC, 03 April 2023. [Online]. Available:

<https://smarttek.solutions/blog/augmented-reality-in-construction/>. [Accessed 31 August 2023].

- [25] T. Constable, "How is the military uses Mixed Reality (MR) for Training and Operations," WARFIGHTER DIGITAL, 17 January 2023. [Online]. Available: <https://www.warfighterpodcast.com/blog/how-is-the-military-uses-mixed-reality-mr-for-training-and-operations/>. [Accessed 31 August 2023].
- [26] G. A. & C. website, "About Google Arts & Culture," Google, 15 September 2022. [Online]. Available: <https://support.google.com/culturalinstitute/partners/answer/4395223>. [Accessed 31 August 2023].
- [27] L. Natural History Museum, "Virtual Museum: 13 ways to explore from home | Natural History Museum," The Trustees of The Natural History Museum, London, 26 July 2023. [Online]. Available: <https://www.nhm.ac.uk/visit/virtual-museum.html>. [Accessed 31 August 2023].
- [28] C. Ferrandez, "5 amazing augmented reality museum experiences," Poplar Studio, 8 October 2021. [Online]. Available: <https://poplar.studio/blog/5-amazing-augmented-reality-museum-experiences/>. [Accessed 31 August 2023].
- [29] E. Jagger, "What's the Difference Between an SDK and an API?," Abstract. The API Company, 25 July 2023. [Online]. Available: <https://www.abstractapi.com/guides/sdk-vs-api>. [Accessed 31 August 2023].
- [30] A. Accelerator, "Visual Studio: Most Up-to-Date Encyclopedia, News & Reviews," Academic Accelerator, [Online]. Available: <https://academic-accelerator.com/encyclopedia/visual-studio>. [Accessed 31 August 2023].
- [31] B. Howe, "MapBox Explained," Medium, 2- October 2022. [Online]. Available: <https://blakebhowe.medium.com/introduction-2b278b438adb>. [Accessed 31 August 2023].
- [32] J. Petty, "What is Unity 3D and what is it used for?," CONCEPT ART EMPIRE, 14 March 2019. [Online]. Available: <https://conceptartempire.com/what-is-unity/>. [Accessed 31 August 2023].
- [33] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | The Project window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/ProjectView.html>. [Accessed 31 August 2023].
- [34] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | The Scene view," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/UsingTheSceneView.html>. [Accessed 31 August 2023].
- [35] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | The Hierarchy window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/Hierarchy.html>. [Accessed 31 August 2023].

- [36] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | The Inspector window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/UsingTheInspector.html>. [Accessed 31 August 2023].
- [37] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | The Game view," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/GameView.html>. [Accessed 31 August 2023].
- [38] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Create Gameplay | GameObjects," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/GameObjects.html>. [Accessed 31 August 2023].
- [39] Unity\_Technologies, "Unity User Manual 2022.3 (LTS)GraphicsLightingThe Lighting window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/lighting-window.html>. [Accessed 31 August 2023].
- [40] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Unity's interface | Console Window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/Console.html>. [Accessed 31 August 2023].
- [41] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Animation | Animation Clips | Animation Window Guide | Using the Animation view," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/animator-UsingAnimationEditor.html>. [Accessed 31 August 2023].
- [42] Unity\_Technologies, "Unity User Manual 2022.3 (LTS) | Working in Unity | Analysis | Profiler overview | The Profiler window," Unity Technologies, 15 September 2022. [Online]. Available: <https://docs.unity3d.com/Manual/ProfilerWindow.html>. [Accessed 31 August 2023].
- [43] Unity\_Technologies, "AR Foundation," Unity Technologies, 26 July 2023. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>. [Accessed 31 August 2023].
- [44] AWS, "What Is SDLC (Software Development Lifecycle)?," AWS, 8 October 2021. [Online]. Available: <https://aws.amazon.com/what-is/sdlc/>. [Accessed 31 August 2023].
- [45] T. Höllerer, S. Feiner and J. Pavlik, "Situated documentaries: Embedding multimedia presentations in the real world," in *IEEE International Symposium on Wearable Computers*, 1999.

## Παράρτημα Α. Κώδικας εφαρμογής

---

### CameraPermissionHandler.cs

Κώδικας για αποδοχή πρόσβασης στην κάμερα.

```
using UnityEngine;
using UnityEngine.Android;

public class CameraPermissionHandler : MonoBehaviour
{
    void Start()
    {
        // ζητά άδεια για την κάμερα
        if (!Permission.HasUserAuthorizedPermission(Permission.Camera))
        {
            Permission.RequestUserPermission(Permission.Camera);
        }

        // αποδοχή πρόσβασης κάμερας
        EnableCamera();
    }

    void EnableCamera()
    {
        // έλεγχος αν έγινε αποδοχή
        if (Permission.HasUserAuthorizedPermission(Permission.Camera))
        {
            // πρόσβαση στην κάμερα
            WebCamTexture webcamTexture = new WebCamTexture();
            Renderer renderer = GetComponent<Renderer>();
            renderer.material.mainTexture = webcamTexture;
            webcamTexture.Play();
        }
    }
}
```

### Detection.cs

Αυτός ο κώδικας χρησιμοποιείται για μία αλλαγή σκηνής.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class detection : MonoBehaviour
{
    private void Start()
    {

        SceneManager.LoadScene(3); // Φόρτωση της σκηνής 3
    }
}
```



## changetoar1.cs

Ο κώδικας αυτός ελέγχει αν το trigger έχει ενεργοποιηθεί ήδη (hasTriggered). Εάν δεν έχει ενεργοποιηθεί, φορτώνει αμέσως την σκηνή 1. Εάν έχει ενεργοποιηθεί, υπάρχει μια καθυστέρηση των 120 δευτερολέπτων πριν φορτωθεί ξανά η σκηνή 1.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class changetoar1 : MonoBehaviour
{
    private static bool hasTriggered = false; // Καθορίζει αν το trigger έχει
    ενεργοποιηθεί ήδη

    private void OnTriggerEnter(Collider other) // Εάν το trigger δεν έχει
    ενεργοποιηθεί ακόμα
    {
        if (!hasTriggered)
        {
            SceneManager.LoadScene(2); // Φόρτωση της σκηνής 1
            hasTriggered = true; // Ενημέρωση ότι το trigger έχει ενεργοποιηθεί
        }
        else
        {
            StartCoroutine(TriggerAfterDelay(120f)); //εκκίνηση της καθυστέρησης
        }
    }

    private IEnumerator TriggerAfterDelay(float delay)
    {
        yield return new WaitForSeconds(delay);

        // Εκτέλεση του δεύτερου γεγονότος μετά την καθυστέρηση
        SceneManager.LoadScene(2); // Φόρτωση της σκηνής 2
    }
}
```

## Myscript.cs

Αυτός ο κώδικας είναι ένα λεξικό το οποίο συνδέεται με τον κώδικα ImageTrackingController και χρησιμοποιείται για την ενεργοποίηση και απενεργοποίηση αντιστοιχιζόμενων αντικειμένων με συγκεκριμένες εικόνες.

```
using UnityEngine;

public class myscript : MonoBehaviour
{
    // Δηλώσεις των GameObject που θα αντιστοιχηθούν με τις εικόνες
    public GameObject Object1;
    public GameObject Object2;
    public GameObject Object3;
    public GameObject Object4;
    public GameObject Object5;
    public GameObject Object6;
    public GameObject Object7;
}
```

```

public GameObject Object8;
public GameObject Object9;
public GameObject Object10;
public GameObject Object11;
public GameObject Object12;
public GameObject Object13;
public GameObject Object14;
public GameObject Object15;
public GameObject Object16;
public GameObject Object17;

void Start()
{
// Λήψη του ImageTrackingController που είναι συνδεδεμένο με το ίδιο GameObject
ImageTrackingController trackingController =
GetComponent<ImageTrackingController>();
// Προσθήκη των GameObjects στο Image Tracking Controller με τα αντίστοιχα ονόματα
εικόνων
    trackingController.AddGameObjectToTrack("agios stefanos", Object1);
    trackingController.AddGameObjectToTrack("ag anarguroi", Object2);
    trackingController.AddGameObjectToTrack("taksiarxhs", Object3);
    trackingController.AddGameObjectToTrack("sphlaio", Object4);
    trackingController.AddGameObjectToTrack("mauriwtissa", Object5);
    trackingController.AddGameObjectToTrack("tzami", Object6);
    trackingController.AddGameObjectToTrack("koumpelidiki", Object7);
    trackingController.AddGameObjectToTrack("katw agora", Object8);
    trackingController.AddGameObjectToTrack("apozari", Object9);
    trackingController.AddGameObjectToTrack("ntoltso", Object10);
    trackingController.AddGameObjectToTrack("omonoia", Object11);
    trackingController.AddGameObjectToTrack("mak agwna", Object12);
    trackingController.AddGameObjectToTrack("delinaneio", Object13);
    trackingController.AddGameObjectToTrack("foklore", Object14);
    trackingController.AddGameObjectToTrack("byzantino", Object15);
    trackingController.AddGameObjectToTrack("endumatologiko", Object16);
    trackingController.AddGameObjectToTrack("mentreses", Object17);

}
}

```

## ImageTrackingController.cs

Αυτός ο κώδικας υλοποιεί έναν ελεγκτή για την ανίχνευση εικόνων (image tracking) σε μια εφαρμογή που χρησιμοποιεί την τεχνολογία επαυξημένης πραγματικότητας (AR) με τη χρήση ARFoundation στο Unity.

```

using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;
using System.Collections.Generic;

public class ImageTrackingController : MonoBehaviour
{
    public ARTrackedImageManager trackedImageManager; // O ARTrackedImageManager
    υπεύθυνος για την διαχείριση των tracked images
    public Dictionary<string, GameObject> gameObjectsToActivate = new
    Dictionary<string, GameObject>();
}

```

```

// Dictionary που αντιστοιχεί τα ονόματα των εικόνων με τα αντίστοιχα GameObjects
// που θα ενεργοποιούνται
    public GameObject gameObject1;
// GameObject που θα ενεργοποιείται για την πρώτη εικόνα που αναγνωρίζεται

    private Dictionary<string, ARTrackedImage> trackedImages = new
Dictionary<string, ARTrackedImage>();
// Αποθηκεύει τις ARTrackedImages που αναγνωρίζονται ανάλογα με το όνομα τους

    private void Awake()
    {
// Απενεργοποίηση όλων των ARTrackedImageManagers εκτός από τον δικό μας
        ARTrackedImageManager[] imageManagers =
FindObjectsOfType<ARTrackedImageManager>();
        foreach (ARTrackedImageManager manager in imageManagers)
        {
            if (manager != trackedImageManager)
            {
                Destroy(manager);
            }
        }
    }

    private void Start()
    {
// Εγγραφή στο event για τις αλλαγές στις tracked images
        trackedImageManager.trackedImagesChanged += OnTrackedImagesChanged;
// Φόρτωση των tracked images
        LoadTrackedImages();
    }

    private void OnDisable()
    {
// Απεγγραφή από το event για τις αλλαγές στις tracked images
        trackedImageManager.trackedImagesChanged -= OnTrackedImagesChanged;
    }

    private void OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs eventArgs)
    {
// Εκτελείται όταν αλλάζει η κατάσταση μιας tracked image
        foreach (var trackedImage in eventArgs.added)
        {
// Προσθήκη της tracked image στο dictionary
            trackedImages[trackedImage.referenceImage.name] = trackedImage;
// Ενεργοποίηση του αντίστοιχου GameObject
            ActivateGameObject(trackedImage.referenceImage.name);
        }

        foreach (var trackedImage in eventArgs.updated)
        {
// Εκτελούνται ενημερώσεις για το αντικείμενο που αναγνωρίζεται
            UpdateGameObject(trackedImage.referenceImage.name);
        }

        foreach (var trackedImage in eventArgs.removed)
        {
// Αφαίρεση της tracked image από το dictionary
            trackedImages.Remove(trackedImage.referenceImage.name);
// Απενεργοποίηση του αντίστοιχου GameObject
            DeactivateGameObject(trackedImage.referenceImage.name);
        }
    }
}

```

```

        private void ActivateGameObject(string imageName)
        {
            // Ενεργοποίηση του αντίστοιχου GameObject
            if (gameObjectsToActivate.TryGetValue(imageName, out GameObject
gameObject))
            {
                gameObject.SetActive(true);
            }
            // Απενεργοποίηση ενός συγκεκριμένου GameObject
            gameObject1.SetActive(false);
        }

        private void UpdateGameObject(string imageName)
        {
            // Εκτελείτε αν χρειαστούν ενημερώσεις στο αντικείμενο που αναγνωρίζεται
        }

        private void DeactivateGameObject(string imageName)
        {
            // Απενεργοποίηση του αντίστοιχου GameObject
            if (gameObjectsToActivate.TryGetValue(imageName, out GameObject
gameObject))
            {
                gameObject.SetActive(false);
            }
        }

        public void AddGameObjectToTrack(string imageName, GameObject gameObject)
        {
            // Προσθήκη αντικειμένου για αναγνώριση στο dictionary
            gameObjectsToActivate[imageName] = gameObject;
        }

        private void LoadTrackedImages()
        {
            // Φόρτωση όλων των tracked images που βρίσκονται ήδη στον χώρο
            foreach (ARTrackedImage trackedImage in trackedImageManager.trackables)
            {
                trackedImages[trackedImage.referenceImage.name] = trackedImage;
                ActivateGameObject(trackedImage.referenceImage.name);
            }
        }

        private void OnEnable()
        {
            // Εγγραφή στο event όταν γίνει ανανέωση του σκηνικού
            trackedImageManager.trackedImagesChanged += OnReloadedScene;
        }

        private void OnReloadedScene(ARTrackedImagesChangedEventArgs eventArgs)
        {
            // Καθαρισμός του dictionary πριν φορτώσουμε ξανά τις tracked images
            trackedImages.Clear();

            // Φόρτωση των tracked images
            LoadTrackedImages();

            // Απεγγραφή από το event για να αποφευχθούν πολλαπλές εγγραφές
            trackedImageManager.trackedImagesChanged -= OnReloadedScene;
        }
    }
}

```

## ImageClickHandler.cs

Αυτός ο κώδικας επιτρέπει τη δυναμική επέκταση και συρρίκνωση μιας εικόνας κάθε φορά που γίνεται κλικ πάνω της. Η εικόνα μεγαλώνει προς το μέγεθος του defaultSize enlargeFactor και κεντράρεται στον καμβά, ενώ στη συρρίκνωση επιστρέφει στο αρχικό μέγεθος και τοποθετείται στην αρχική θέση.

```
using UnityEngine;
using UnityEngine.UI;

public class ImageClickHandler : MonoBehaviour
{
    private bool isEnlarged = false; // Κατάσταση επέκτασης
    private Vector2 defaultSize; // Προεπιλεγμένο μέγεθος εικόνας
    private int defaultSiblingIndex; // Προεπιλεγμένος αριθμός αδελφού
    private Vector2 defaultPosition; // Προεπιλεγμένη θέση εικόνας

    [SerializeField]
    private float enlargeFactor = 2.0f; // Παράγοντας επέκτασης
    [SerializeField]
    private float transitionDuration = 0.5f; // Διάρκεια μετάβασης

    private RectTransform rectTransform; // Το Component RectTransform της εικόνας
    private Canvas canvas; // Το Component Canvas στο οποίο ανήκει η εικόνα

    private void Start()
    {
        rectTransform = GetComponent<RectTransform>();
        defaultSize = rectTransform.sizeDelta;
        defaultSiblingIndex = transform.GetSiblingIndex();
        defaultPosition = rectTransform.anchoredPosition;

        canvas = GetComponentInParent<Canvas>();
    }

    public void ToggleEnlarge()
    {
        if (isEnlarged)
        {
            StartCoroutine(TransitionToDefaultSize());
        }
        else
        {
            StartCoroutine(TransitionToEnlargedSize());
        }
    }

    private System.Collections.IEnumerator TransitionToDefaultSize()
    {
        float time = 0.0f;
        Vector2 initialSize = rectTransform.sizeDelta;
        Vector2 targetSize = defaultSize;

        // Αποθήκευση του τρέχοντος αριθμού αδελφού
        int currentSiblingIndex = transform.GetSiblingIndex();

        while (time < transitionDuration)
        {
            time += Time.deltaTime;
            float t = Mathf.Clamp01(time / transitionDuration);
            rectTransform.sizeDelta = Vector2.Lerp(initialSize, targetSize, t);
        }
    }
}
```

```

        yield return null;
    }

    rectTransform.sizeDelta = defaultSize;
    rectTransform.anchoredPosition = defaultPosition;

    // Επαναφορά του αριθμού αδελφού
    transform.SetSiblingIndex(currentSiblingIndex);

    isEnlarged = false;
}

private System.Collections.IEnumerator TransitionToEnlargedSize()
{
    float time = 0.0f;
    Vector2 initialSize = rectTransform.sizeDelta;
    Vector2 targetSize = defaultSize * enlargeFactor;

    // Ορισμός του αριθμού αδελφού στη μεγαλύτερη τιμή
    transform.SetAsLastSibling();

    while (time < transitionDuration)
    {
        time += Time.deltaTime;
        float t = Mathf.Clamp01(time / transitionDuration);
        rectTransform.sizeDelta = Vector2.Lerp(initialSize, targetSize, t);

        // Κέντραρισμα της εικόνας στο Canvas
        Vector2 canvasCenter =
canvas.GetComponent<RectTransform>().rect.center;
        Vector2 targetPosition = canvasCenter;
        rectTransform.anchoredPosition = Vector2.Lerp(defaultPosition,
targetPosition, t);

        yield return null;
    }

    rectTransform.sizeDelta = targetSize;
    isEnlarged = true;
}
}

```

## Exitback.cs

Κώδικας αλλαγής σκηνής.

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class exitback : MonoBehaviour
{
    public void ConfirmExit()
    {
        SceneManager.LoadScene(3); // Φόρτωση της σκηνής 3
    }
}

```

## Exit.cs

Κώδικας αλλαγής σκηνής.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class exit : MonoBehaviour
{
    public void ConfirmExit()
    {
        SceneManager.LoadScene(1); // Φόρτωση της σκηνής 1
    }
}
```

## SpawnOnMap.cs

Κώδικας του Mapbox με μία τροποποίηση βάζοντας μία καθυστέρηση ώστε να μην υπάρξει κάποιο κόλλημα και ενεργοποιηθεί αμέσως το Trigger.

Αυτός ο κώδικας δημιουργεί αντικείμενα (markers) στον χάρτη σύμφωνα με τις γεωγραφικές τοποθεσίες που έχουμε δηλώσει, χρησιμοποιώντας το Mapbox. Το σενάριο παρουσιάζει μια καθυστέρηση (startWait) πριν από την κλήση της μεθόδου Initialize() που δημιουργεί τα αντικείμενα.

Επίσης, η μέθοδος OnUpdate() ενημερώνει συνεχώς τις θέσεις και τις κλίμακες των αντικειμένων βάσει των γεωγραφικών τους τοποθεσιών.

Αυτή η κλάση χρησιμοποιείται για την τοποθέτηση markers στον χάρτη με βάση τις παρεχόμενες γεωγραφικές συντεταγμένες.

```
namespace Mapbox.Examples
{
    using UnityEngine;
    using Mapbox.Utils;
    using Mapbox.Unity.Map;
    using Mapbox.Unity.MeshGeneration.Factories;
    using Mapbox.Unity.Utilities;
    using System.Collections.Generic;
    using UnityEngine.SceneManagement;

    public class SpawnOnMap : MonoBehaviour
    {
        // Ο χάρτης στον οποίο θα προστεθούν τα αντικείμενα
        [SerializeField]
        AbstractMap _map;

        // Οι συντεταγμένες των τοποθεσιών στον χάρτη
        [SerializeField]
        [Geocode]
        string[] _locationStrings;
        Vector2d[] _locations;

        // Η κλίμακα των αντικειμένων που θα τοποθετηθούν
        [SerializeField]
        float _spawnScale = 100f;
    }
}
```

```

// Το prefab του marker που θα δημιουργηθεί
[SerializeField]
GameObject _markerPrefab;

List<GameObject> _spawnedObjects;
private float startWait = 8f;

void Start()
{
    // Καθυστέρηση της κλήσης της μεθόδου Initialize() μετά από έναν χρόνο
    Invoke("Initialize", startWait);
}

void Initialize()
{
    // Μετατροπή των δεδομένων συντεταγμένων από τα strings σε Vector2d
    _locations = new Vector2d[_locationStrings.Length];
    _spawnedObjects = new List<GameObject>();
    for (int i = 0; i < _locationStrings.Length; i++)
    {
        var locationString = _locationStrings[i];
        _locations[i] = Conversions.StringToLatLon(locationString);

        // Δημιουργία αντικειμένου marker
        var instance = Instantiate(_markerPrefab);

        // Τοποθέτηση του marker στον χάρτη με βάση την γεωγραφική τοποθεσία
        instance.transform.localPosition =
        _map.GeoToWorldPosition(_locations[i], true);

        // Ορισμός της κλίμακας του marker
        instance.transform.localScale = new Vector3(_spawnScale,
        _spawnScale, _spawnScale);

        // Προσθήκη του marker στη λίστα _spawnedObjects
        _spawnedObjects.Add(instance);
    }
}

private void OnUpdate()
{
    int count = _spawnedObjects.Count;
    for (int i = 0; i < count; i++)
    {
        var spawnedObject = _spawnedObjects[i];
        var location = _locations[i];

        // Ενημέρωση της θέσης και της κλίμακας του marker βάσει της
        γεωγραφικής τοποθεσίας
        spawnedObject.transform.localPosition =
        _map.GeoToWorldPosition(location, true);
        spawnedObject.transform.localScale = new Vector3(_spawnScale,
        _spawnScale, _spawnScale);
    }
}
}
}

```



## DirectionsFactory.cs

Κώδικας του Mapbox: Είναι υπεύθυνος για τη δημιουργία και την ανανέωση μιας διαδρομής πάνω στον χάρτη. Χρησιμοποιεί τις παρεχόμενες γεωγραφικές τοποθεσίες ως waypoints για να κατασκευάσει και να ανανεώσει τη διαδρομή μεταξύ τους. Επίσης, επιτηρεί τις αλλαγές στις θέσεις των waypoints και ανανεώνει τη διαδρομή κατάλληλα.

```
namespace Mapbox.Unity.MeshGeneration.Factories
{
    using UnityEngine;
    using Mapbox.Directions;
    using System.Collections.Generic;
    using System.Linq;
    using Mapbox.Unity.Map;
    using Mapbox.Unity.MeshGeneration.Modifiers;
    using Mapbox.Unity.MeshGeneration.Data;
    using Mapbox.Utills;
    using Mapbox.Unity.Utilities;
    using System.Collections;

    public class DirectionsFactory : MonoBehaviour
    {
        [SerializeField]
        AbstractMap _map;

        [SerializeField]
        MeshModifier[] MeshModifiers;

        [SerializeField]
        Material _material;

        [SerializeField]
        Transform _playerTransform; // Αναθέστε το transform του παίκτη στον
Inspector.

        [SerializeField]
        Vector2d _secondWaypoint; // Αναθέστε τις συντεταγμένες του δεύτερου
σημείου στο Inspector.

        private List<Vector3> _cachedWaypoints;

        [SerializeField]
        [Range(1, 10)]
        private float UpdateFrequency = 2;

        private Directions _directions;
        private int _counter;

        GameObject _directionsGO;
        private bool _recalculateNext;

        private void Awake()
        {
            if (_map == null)
            {
                _map = FindObjectOfType<AbstractMap>();
            }
            _directions = MapboxAccess.Instance.Directions;
            _map.OnInitialized += Query;
            _map.OnUpdated += Query;
            _cachedWaypoints = new List<Vector3>();
        }
    }
}
```

```

        _cachedWaypoints.Add(_playerTransform.position); // Θέστε τη θέση του
παίκτη ως το πρώτο σημείο
        _recalculateNext = false;

        foreach (var modifier in MeshModifiers)
        {
            modifier.Initialize();
        }
    }

    private IEnumerator StartAfterInitialization()
    {
        // Περιμένετε ένα καρέ για να εξασφαλίσετε ότι άλλα αντικείμενα έχουν
πρωταρχικοποιηθεί.
        yield return null;
        // Τώρα, μπορείτε να καλέσετε το Query μετά από ένα καρέ για να
διασφαλίσετε ότι όλα είναι έτοιμα.
        Query();
        StartCoroutine(QueryTimer());
    }

    private void OnDisable()
    {
        _map.OnInitialized -= Query;
        _map.OnUpdated -= Query;
    }

    private void Query()
    {
        // Τροποποιήστε τη μέθοδο Query για να χρησιμοποιεί τη θέση του παίκτη
ως το πρώτο σημείο και τις παρεχόμενες συντεταγμένες ως το δεύτερο σημείο.
        var wp = new Vector2d[] {
            _playerTransform.GetGeoPosition(_map.CenterMercator, _map.WorldRelativeScale),
            _secondWaypoint };
        var _directionResource = new DirectionResource(wp,
RoutingProfile.Driving);
        _directionResource.Steps = true;
        _directions.Query(_directionResource, HandleDirectionsResponse);
    }

    private IEnumerator QueryTimer()
    {
        while (true)
        {
            yield return new WaitForSeconds(UpdateFrequency);
            for (int i = 0; i < _cachedWaypoints.Count; i++)
            {
                if (_cachedWaypoints[i] != _playerTransform.position) //
Αλλάξτε _waypoints[i].position σε _cachedWaypoints[i]
                {
                    _recalculateNext = true;
                    _cachedWaypoints[i] = _playerTransform.position; //
Αλλάξτε _waypoints[i].position σε _cachedWaypoints[i]
                }
            }
            if (_recalculateNext)
            {
                Query();
                _recalculateNext = false;
            }
        }
    }
}

```

```

        private void HandleDirectionsResponse(DirectionsResponse response)
        {
            if (response == null || response.Routes == null ||
response.Routes.Count < 1)
            {
                return;
            }
            var meshData = new MeshData();
            var dat = new List<Vector3>();
            foreach (var point in response.Routes[0].Geometry)
            {
                dat.Add(Conversions.GeoToWorldPosition(point.x, point.y,
_map.CenterMercator, _map.WorldRelativeScale).ToVector3xz());
            }
            var feat = new VectorFeatureUnity();
            feat.Points.Add(dat);

            foreach (var mod in MeshModifiers)
            {
                if (mod != null and mod.Active)
                {
                    mod.Run(feat, meshData, _map.WorldRelativeScale);
                }
            }
            CreateGameObject(meshData);
        }

        private GameObject CreateGameObject(MeshData data)
        {
            if (_directionsGO != null)
            {
                Destroy(_directionsGO);
            }
            _directionsGO = new GameObject("direction_waypoint_entity");
            _directionsGO.transform.parent = transform;
            var mesh = _directionsGO.AddComponent<MeshFilter>().mesh;
            mesh.subMeshCount = data.Triangles.Count;
            mesh.SetVertices(data.Vertices);
            _counter = data.Triangles.Count;
            for (int i = 0; i < _counter; i++)
            {
                var triangle = data.Triangles[i];
                mesh.SetTriangles(triangle, i);
            }
            _counter = data.UV.Count;
            for (int i = 0; i < _counter; i++)
            {
                var uv = data.UV[i];
                mesh.SetUVs(i, uv);
            }
            mesh.RecalculateNormals();
            _directionsGO.AddComponent<MeshRenderer>().material = _material;
            return _directionsGO;
        }

        public void Start()
        {
            StartCoroutine(StartAfterInitialization());
        }
    }
}

```

## ZoomText.cs

Κώδικας μεγένθυσης και σμίκρυνσης κειμένου.

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

public class ZoomText : MonoBehaviour
{
    public TextMeshProUGUI[] textsToZoom; // Ένας πίνακας από κείμενα που θα
    υποστούν μεγένθυση/σμίκρυνση.

    public float zoomAmount = 1.1f; // Ο συντελεστής μεγένθυσης.

    // Μέθοδος που εκτελεί τη μεγένθυση των κειμένων.
    public void ZoomIn()
    {
        foreach (TextMeshProUGUI textToZoom in textsToZoom)
        {
            textToZoom.fontSize *= zoomAmount; // Αυξάνει το μέγεθος της
            γραμματοσειράς του κειμένου.
        }
    }

    // Μέθοδος που εκτελεί τη σμίκρυνση των κειμένων.
    public void ZoomOut()
    {
        foreach (TextMeshProUGUI textToZoom in textsToZoom)
        {
            textToZoom.fontSize /= zoomAmount; // Μειώνει το μέγεθος της
            γραμματοσειράς του κειμένου.
        }
    }
}
```

## MoveText.cs

Κώδικας μετακίνησης κειμένου προς τα πάνω και προς τα κάτω.

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

public class MoveText : MonoBehaviour
{
    public List<TextMeshProUGUI> textObjects; // Δημιουργία λίστας για
    αποθήκευση αντικειμένων TextMeshProUGUI

    public float moveDistance = 10f; // Ρυθμίστε την απόσταση μετακίνησης
    όπως χρειάζεται.

    // Μέθοδος που μετακινεί το κείμενο προς τα πάνω.
    public void MoveUp()
    {
        foreach (TextMeshProUGUI textToMove in textObjects)
        {
            Vector3 position = textToMove.rectTransform.localPosition;
            position.y += moveDistance; // Αυξάνει τον υ-άξονα της θέσης του
κειμένου.
            textToMove.rectTransform.localPosition = position; // Ορίζει τη νέα
θέση του κειμένου.
        }
    }

    // Μέθοδος που μετακινεί το κείμενο προς τα κάτω.
    public void MoveDown()
    {
        foreach (TextMeshProUGUI textToMove in textObjects)
        {
            Vector3 position = textToMove.rectTransform.localPosition;
            position.y -= moveDistance; // Μειώνει τον υ-άξονα της θέσης του
κειμένου.
            textToMove.rectTransform.localPosition = position; // Ορίζει τη νέα
θέση του κειμένου.
        }
    }
}
```

## DeviceLocationProvideAndroidNative.cs

Κώδικας για την εμφάνιση του χάρτη σύμφωνα με το σημείο στο οποίο βρισκόμαστε.

```
namespace Mapbox.Unity.Location
{
    using UnityEngine;
    using System.Collections;
    using System.Globalization;
    using System;
    using System.IO;
    using System.Text;
    using Mapbox.Utils;

    public class DeviceLocationProviderAndroidNative : AbstractLocationProvider,
    IDisposable
    {
        // Ελάχιστη απόσταση (μετρημένη σε μέτρα) που πρέπει να μετακινηθεί
        // συσκευή προτού ενημερωθεί η τοποθεσία.
        [SerializeField]
        [Tooltip("Η ελάχιστη απόσταση (μετρημένη σε μέτρα) που πρέπει να
        μετακινηθεί συσκευή προτού ενημερωθεί η τοποθεσία. Υψηλές τιμές όπως 500
        υποδηλώνουν λιγότερη επιβάρυνση.")]
        float _updateDistanceInMeters = 0.0f;

        // Ελάχιστο χρονικό διάστημα μεταξύ ενημερώσεων τοποθεσίας, σε χιλιοστά
        // του δευτερολέπτου.
        [SerializeField]
        [Tooltip("Το ελάχιστο χρονικό διάστημα μεταξύ ενημερώσεων τοποθεσίας, σε
        χιλιοστά του δευτερολέπτου. Δεν είναι λογικό να πέσετε κάτω από τα 500ms.")]
        long _updateTimeInMilliseconds = 1000;

        private WaitForSeconds _wait1sec;
        private WaitForSeconds _wait5sec;
        private WaitForSeconds _wait60sec;
        /// <summary>Ελέγχει τον παροχέα τοποθεσίας μόνο στο αιτούμενο διάστημα
        // ενημέρωσης για μείωση του φορτίου.</summary>
        private WaitForSeconds _waitUpdateTime;
        private bool _disposed;
        private static object _lock = new object();
        private Coroutine _pollLocation;

        private AndroidJavaObject _activityContext = null;
        private AndroidJavaObject _gpsInstance;
        private AndroidJavaObject _sensorInstance;

        ~DeviceLocationProviderAndroidNative() { Dispose(false); }
        public void Dispose()
        {
            Dispose(true);
            GC.SuppressFinalize(this);
        }

        protected virtual void Dispose(bool disposeManagedResources)
        {
            if (!_disposed)
            {
                if (disposeManagedResources)
                {
                    shutdown();
                }
                _disposed = true;
            }
        }
    }
}
```

```

    }
}

private void shutdown()
{
    try
    {
        lock (_lock)
        {
            if (null != _gpsInstance)
            {
                _gpsInstance.Call("stopLocationListeners");
                _gpsInstance.Dispose();
                _gpsInstance = null;
            }
            if (null != _sensorInstance)
            {
                _sensorInstance.Call("stopSensorListeners");
                _sensorInstance.Dispose();
                _sensorInstance = null;
            }
        }
    }
    catch (Exception ex)
    {
        Debug.LogError(ex);
    }
}

protected virtual void OnDestroy() { shutdown(); }

protected virtual void OnDisable() { shutdown(); }

protected virtual void Awake()
{
    // Ασφαλή μέτρα για να μην εκτελείται όταν είναι απενεργοποιημένο ή
    // δεν έχει επιλεγεί ως παροχέας τοποθεσίας
    if (!enabled) { return; }
    if (!transform.gameObject.activeInHierarchy) { return; }

    _wait1sec = new WaitForSeconds(1);
    _wait5sec = new WaitForSeconds(5);
    _wait60sec = new WaitForSeconds(60);
    // Ρυθμίζει την ενημέρωση αν το αιτούμενο χρονικό διάστημα ενημέρωσης
    // είναι απαράδεκτα χαμηλό
    _waitUpdateTime = _updateTimeInMilliseconds < 500 ? new
WaitForSeconds(0.5f) : new WaitForSeconds((float)_updateTimeInMilliseconds /
1000.0f);

    _currentLocation.IsLocationServiceEnabled = false;
    _currentLocation.IsLocationServiceInitializing = true;

    if (Application.platform == RuntimePlatform.Android)
    {
        getActivityContext();
        getGpsInstance(true);
        getSensorInstance();

        if (_pollLocation == null)
        {
            _pollLocation = StartCoroutine(locationRoutine());
        }
    }
}

```

```

    }
}

private void getActivityContext()
{
    using (AndroidJavaClass activityClass = new
AndroidJavaClass("com.unity3d.player.UnityPlayer"))
    {
        _activityContext =
activityClass.GetStatic<AndroidJavaObject>("currentActivity");
    }

    if (null == _activityContext)
    {
        Debug.LogError("Δεν ήταν δυνατή η λήψη της δραστηριότητας
UnityPlayer");
        return;
    }
}

private void getGpsInstance(bool showToastMessages = false)
{
    if (null == _activityContext) { return; }

    using (AndroidJavaClass androidGps = new
AndroidJavaClass("com.mapbox.android.unity.AndroidGps"))
    {
        if (null == androidGps)
        {
            Debug.LogError("Δεν ήταν δυνατή η λήψη της κλάσης
'AndroidGps'");
            return;
        }

        _gpsInstance =
androidGps.CallStatic<AndroidJavaObject>("instance", _activityContext);
        if (null == _gpsInstance)
        {
            Debug.LogError("Δεν ήταν δυνατή η λήψη της στιγμιοτύπου
'AndroidGps'");
            return;
        }

        _activityContext.Call("runOnUiThread", new AndroidJavaRunnable(()
=> { _gpsInstance.Call("showMessage", "Έναρξη της παρακολούθησης τοποθεσίας");
})));

        _gpsInstance.Call("startLocationListeners",
_updateDistanceInMeters, _updateTimeInMilliseconds);
    }
}

private void getSensorInstance()
{
    if (null == _activityContext) { return; }

    using (AndroidJavaClass androidSensors = new
AndroidJavaClass("com.mapbox.android.unity.AndroidSensors"))
    {
        if (null == androidSensors)
        {

```



```

        Debug.LogError("Δεν ήταν δυνατή η λήψη της κλάσης
'AndroidSensors'");
        return;
    }

    _sensorInstance =
androidSensors.CallStatic<AndroidJavaObject>("instance", _activityContext);
    if (null == _sensorInstance)
    {
        Debug.LogError("Δεν ήταν δυνατή η λήψη της στιγμιοτύπου
'AndroidSensors'");
        return;
    }

    _sensorInstance.Call("startSensorListeners");
}

private IEnumerator locationRoutine()
{
    while (true)
    {
        // Δεν ήταν δυνατή η λήψη της δραστηριότητας του παίκτη, περίμενε
και ξαναδοκίμασε
        if (null == _activityContext)
        {
            SendLocation(_currentLocation);
            yield return _wait60sec;
            getActivityContext();
            continue;
        }
        // Δεν ήταν δυνατή η λήψη της στιγμιοτύπου του προσθετικού GPS,
περίμενε και ξαναδοκίμασε
        if (null == _gpsInstance)
        {
            SendLocation(_currentLocation);
            yield return _wait60sec;
            getGpsInstance();
            continue;
        }

        // Ενημέρωση του προσανατολισμού της συσκευής
        if (null != _sensorInstance)
        {
            _currentLocation.DeviceOrientation =
_sensorInstance.Call<float>("getOrientation");
        }

        bool locationServiceAvailable =
_gpsInstance.Call<bool>("getIsLocationServiceAvailable");
        _currentLocation.IsLocationServiceEnabled =
locationServiceAvailable;

        // Η εφαρμογή ενδέχεται να έχει ξεκινήσει χωρίς την τοποθεσία,
αλλά να την ενεργοποιεί αργότερα. Ελέγξτε κατά καιρούς
        if (!locationServiceAvailable)
        {
            _currentLocation.IsLocationServiceInitializing = true;
            _currentLocation.Accuracy = 0;
            _currentLocation.HasGpsFix = false;
            _currentLocation.SatellitesInView = 0;
        }
    }
}

```

```

        _currentLocation.SatellitesUsed = 0;

        SendLocation(_currentLocation);
        _gpsInstance.Call("stopLocationListeners");
        yield return _wait5sec;
        _gpsInstance.Call("startLocationListeners",
_updateDistanceInMeters, _updateTimeInMilliseconds);
        yield return _wait1sec;
        continue;
    }

// Αν φτάσαμε μέχρι εδώ, σημαίνει ότι οι υπηρεσίες τοποθεσίας λειτουργούν
    _currentLocation.IsLocationServiceInitializing = false;

    try
    {
        AndroidJavaObject locNetwork =
        _gpsInstance.Get<AndroidJavaObject>("lastKnownLocationNetwork");
        AndroidJavaObject locGps =
        _gpsInstance.Get<AndroidJavaObject>("lastKnownLocationGps");

        // Εύκολες περιπτώσεις: είτε καμία είτε και οι δύο τοποθεσίες
        // GPS ή δικτύου δεν είναι διαθέσιμες
        if (null == locGps & null == locNetwork) {
        populateCurrentLocation(null); }
        if (null != locGps && null == locNetwork) {
        populateCurrentLocation(locGps); }
        if (null == locGps && null != locNetwork) {
        populateCurrentLocation(locNetwork); }

        // Διαθέσιμη η τοποθεσία GPS και δικτύου: καθορίστε ποια θα
        // χρησιμοποιηθεί
        if (null != locGps && null != locNetwork) {
        populateWithBetterLocation(locGps, locNetwork); }

        _currentLocation.TimestampDevice =
        UnixTimestampUtils.To(DateTime.UtcNow);
        SendLocation(_currentLocation);
    }
    catch (Exception ex)
    {
        Debug.LogErrorFormat("Σφάλμα πρόσθετου GPS: " +
ex.ToString());
    }
    yield return _waitUpdateTime;
}

private void populateCurrentLocation(AndroidJavaObject location)
{
    if (null == location)
    {
        _currentLocation.IsLocationUpdated = false;
        _currentLocation.IsUserHeadingUpdated = false;
        return;
    }

    double lat = location.Call<double>("getLatitude");
    double lng = location.Call<double>("getLongitude");
    Utils.Vector2d newLatLng = new Utils.Vector2d(lat, lng);
    bool coordinatesUpdated =
!newLatLng.Equals(_currentLocation.LatitudeLongitude);

```

```

        _currentLocation.LatitudeLongitude = newLatLng;

        float newAccuracy = location.Call<float>("getAccuracy");
        bool accuracyUpdated = newAccuracy != _currentLocation.Accuracy;
        _currentLocation.Accuracy = newAccuracy;

        // Χωρίστε το με το 1000. Το Android επιστρέφει χιλιοστά
        δευτερολέπτου, εργαζόμαστε με δευτερόλεπτα
        long newTimestamp = location.Call<long>("getTime") / 1000;
        bool timestampUpdated = newTimestamp != _currentLocation.Timestamp;
        _currentLocation.Timestamp = newTimestamp;

        string newProvider = location.Call<string>("getProvider");
        bool providerUpdated = newProvider != _currentLocation.Provider;
        _currentLocation.Provider = newProvider;

        bool hasBearing = location.Call<bool>("hasBearing");
        // Εκτιμήστε το bearing μόνο όταν το αντικείμενο τοποθεσίας έχει πραγματικά
        bearing. Το Android γεμίζει το bearing (που δεν είναι ίσο με τον προσανατολισμό
        της συσκευής) μόνο όταν η συσκευή κινείται. Διαφορετικά ορίζεται σε '0.0'
        //https://developer.android.com/reference/android/location/Location.html#getBearing()
        // Δεν θέλουμε αυτό όταν περιστρέφουμε ένα χάρτη ανάλογα με την κατεύθυνση που
        κινείται ο χρήστης, επομένως δεν ενημερώνουμε το 'heading' με το '0.0'
        if (!hasBearing)
        {
            _currentLocation.IsUserHeadingUpdated = false;
        }
        else
        {
            float newHeading = location.Call<float>("getBearing");
            _currentLocation.IsUserHeadingUpdated = newHeading !=
            _currentLocation.UserHeading;
            _currentLocation.UserHeading = newHeading;
        }

        float? newSpeed = location.Call<float>("getSpeed");
        bool speedUpdated = newSpeed != _currentLocation.SpeedMetersPerSecond;
        _currentLocation.SpeedMetersPerSecond = newSpeed;

        // Σημαίνετε την τοποθεσία ως ενημερωμένη εάν οποιαδήποτε από τις
        παρακάτω συνθήκες αληθεύει
        // Debug.LogFormat("coords:{0} acc:{1} time:{2} speed:{3}",
        coordinatesUpdated, accuracyUpdated, timestampUpdated, speedUpdated);
        _currentLocation.IsLocationUpdated =
            providerUpdated
            || coordinatesUpdated
            || accuracyUpdated
            || timestampUpdated
            || speedUpdated;

        // Αν χρειάζεται, ξανασχοληθείτε. Προειδοποιεί αυτήν τη στιγμή.
        //bool networkEnabled =
        _gpsInstance.Call<bool>("getIsNetworkEnabled");
        bool gpsEnabled = _gpsInstance.Call<bool>("getIsGpsEnabled");
        if (!gpsEnabled)
        {
            _currentLocation.HasGpsFix = null;
            _currentLocation.SatellitesInView = null;
            _currentLocation.SatellitesUsed = null;
        }
        else
        {

```

```

        _currentLocation.HasGpsFix = _gpsInstance.Get<bool>("hasGpsFix");
        _currentLocation.SatellitesInView =
_gpsInstance.Get<int>("satellitesInView");
        _currentLocation.SatellitesUsed =
_gpsInstance.Get<int>("satellitesUsedInFix");
    }
}

/// <summary>
/// Εάν είναι διαθέσιμες τόσο η τοποθεσία GPS όσο και η τοποθεσία δικτύου,
χρησιμοποιήστε την πιο πρόσφατη/καλύτερη
/// </summary>
/// <param name="locGps"></param>
/// <param name="locNetwork"></param>
private void populateWithBetterLocation(AndroidJavaObject locGps,
AndroidJavaObject locNetwork)
{
    // Ελέγξτε ποια τοποθεσία είναι πιο πρόσφατη
    long timestampGps = locGps.Call<long>("getTime");
    long timestampNet = locNetwork.Call<long>("getTime");
    if (timestampGps > timestampNet)
    {
        populateCurrentLocation(locGps);
        return;
    }

    // Ελέγξτε ποια τοποθεσία έχει καλύτερη ακρίβεια
    float accuracyGps = locGps.Call<float>("getAccuracy");
    float accuracyNet = locNetwork.Call<float>("getAccuracy");
    if (accuracyGps < accuracyNet)
    {
        populateCurrentLocation(locGps);
        return;
    }

    // Προεπιλογή στην τοποθεσία του δικτύου
    populateCurrentLocation(locNetwork);
}

#if UNITY_ANDROID

private string time2str(AndroidJavaObject loc)
{
    long time = loc.Call<long>("getTime");
    DateTime dtPlugin = new DateTime(1970, 1, 1, 0, 0, 0,
DateTimeKind.Utc).Add(TimeSpan.FromMilliseconds(time));
    return dtPlugin.ToString("yyyyMMdd HHmmss");
}

private string loc2str(AndroidJavaObject loc)
{
    if (null == loc) { return "loc: NULL"; }

    try
    {
        double lat = loc.Call<double>("getLatitude");
        double lng = loc.Call<double>("getLongitude");

        return string.Format(CultureInfo.InvariantCulture, "loc: {0}, {1}
acc:{2} b:{3} {4} {5}",
lat, lng,

```

```

        loc.Call<float>("getAccuracy"),
        loc.Call<float>("getBearing"),
        time2str(loc),
        loc.Call<string>("getProvider")
    );
    }
    catch (Exception e)
    {
        return "ERROR: " + e.Message;
    }
}
#endif

private void SendLocation(Location location)
{
    if (null != _gpsInstance)
    {
        bool gpsEnabled = _gpsInstance.Call<bool>("getIsGpsEnabled");
        if (!gpsEnabled)
        {
            _gpsInstance.Call("stopLocationListeners");
        }
    }

    if (location.IsLocationUpdated)
    {
        _currentLocation = location;
        OnLocationUpdated(location);
    }
}

private void onLocationUpdated(Location location)
{
    // Η τοποθεσία της συσκευής ενημερώθηκε
    Debug.Log("Ενημερώθηκε η τοποθεσία: " + location.LatitudeLongitude);

    // Εδώ μπορείτε να εκτελέσετε άλλον κώδικα ή να ενημερώσετε το UI με
    τη νέα τοποθεσία.
}

protected virtual void Update()
{
    // Εδώ μπορείτε να προσθέσετε πρόσθετον κώδικα για το Update του
    παιχνιδιού σας, αν χρειαστεί.
}

private void OnDrawGizmos()
{
    // Αυτή είναι μια μέθοδος επαφής για εμφάνιση της τρέχουσας τοποθεσίας
    στον Unity Editor. Μπορείτε να τη χρησιμοποιήσετε για να ελέγξετε αν η τοποθεσία
    ανανεώνεται όπως πρέπει.

    Gizmos.color = Color.green;
    Gizmos.DrawWireSphere(new
    Vector3((float)_currentLocation.LatitudeLongitude.x, 0,
    (float)_currentLocation.LatitudeLongitude.y), 1);
}
}
}

```

## ImmediatePositionWithLocationProvider.cs

Κώδικας για την τοποθέτηση του παίκτη στο σημείο που βρισκόμαστε σύμφωνα με τον κώδικα του locationProvider.

```
namespace Mapbox.Examples
{
    using Mapbox.Unity.Location;
    using Mapbox.Unity.Map;
    using UnityEngine;

    public class ImmediatePositionWithLocationProvider : MonoBehaviour
    {
        bool _isInitialized; // Μεταβλητή για να ελέγξουμε αν ο χάρτης έχει
        αρχικοποιηθεί.

        ILocationProvider _locationProvider; // Τοποθεσία πάροχος που θα
        χρησιμοποιηθεί.
        ILocationProvider LocationProvider
        {
            get
            {
                if (_locationProvider == null)
                {
                    // Αν δεν έχουμε ορίσει κάποιον τοποθεσία πάροχο,
                    χρησιμοποιούμε τον προεπιλεγμένο πάροχο.
                    _locationProvider =
                    LocationProviderFactory.Instance.DefaultLocationProvider;
                }

                return _locationProvider;
            }
        }

        Vector3 _targetPosition; // Η θέση προορισμού στον 3D χώρο.

        void Start()
        {
            // Ορίζουμε έναν ακροατή για το γεγονός ότι ο χάρτης έχει
            αρχικοποιηθεί.
            LocationProviderFactory.Instance.mapManager.OnInitialized += () =>
            _isInitialized = true;
        }

        void LateUpdate()
        {
            if (_isInitialized)
            {
                var map = LocationProviderFactory.Instance.mapManager;
                // Αναφορά στον χάρτη. Ορίζουμε τη θέση του αντικειμένου (το οποίο έχει αυτόν τον
                κώδικα) στη θέση του χρήστη βάσει των γεωγραφικών συντεταγμένων.
                transform.localPosition =
                map.GeoToWorldPosition(LocationProvider.CurrentLocation.LatitudeLongitude);
            }
        }
    }
}
```