



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Συστήματος Επαυξημένης
Πραγματικότητας για την Υποβοήθηση της
Πλοήγησης σε Εσωτερικούς Χώρους

Νίκος Καλορίτης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Γεώργιος Σταμούλης
Καθηγητής ΠΘ

ΣΥΝΕΠΙΒΛΕΠΩΝ

Κωνσταντίνος Κολομβάτος
Επίκουρος Καθηγητής ΠΘ

Λαμία 22 Ιανουαρίου έτος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Συστήματος Επαυξημένης
Πραγματικότητας για την Υποβοήθηση της
Πλοήγησης σε Εσωτερικούς Χώρους

Νίκος Καλορίτης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Γεώργιος Σταμούλης
Καθηγητής ΠΘ

ΣΥΝΕΠΙΒΛΕΠΩΝ

Κωνσταντίνος Κολομβάτσος
Επίκουρος Καθηγητής ΠΘ

Λαμία 21 Ιανουαρίου έτος 2021



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Development of an Augmented Reality System for Indoor Navigation Assistance

Nikos Kaloritis

FINAL THESIS

ADVISOR

Georgios Stamoulis
Professor

CO ADVISOR

Konstantinos Kolomvatsos
Assistant Professor

Lamia 22 January year 2021

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 22/03/2021

Ο Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Στόχος της διπλωματικής εργασίας είναι η δημιουργία μιας εφαρμογής πλοήγησης εσωτερικού χώρου με την χρήση επαυξημένης πραγματικότητας για κινητά τηλέφωνα.

Η εφαρμογή πρόκειται για ένα πρακτικό εργαλείο βοήθειας πλοήγησης που προσφέρει την δυνατότητα προβολής διαδρομής στην οθόνη του χρήστη, επαναυπολογισμό διαδρομής και διόρθωση θέσης με την χρήση QR scanner (Quick Response Scanner). Ακόμα υπάρχει βοήθεια ηχητικού υλικού με την μορφή εντολών. Για την ανάπτυξη της εφαρμογής έχει χρησιμοποιηθεί η μηχανή ανάπτυξης παιχνιδιών Unity. Στην αρχή του κειμένου της εργασίας έχει πραγματοποιηθεί μερική αναφορά ως προς τις τεχνολογίες που έχουν χρησιμοποιηθεί τόσο σε υλικό όσο και αλγόριθμους, έπειτα μία επεξήγηση για την μηχανή ανάπτυξης Unity και τέλος επεξήγηση των βασικών τμημάτων του κώδικα που έχει χρησιμοποιηθεί και σύντομος οδηγός χρήσης της εφαρμογής.

ABSTRACT

The scope of this thesis is to create an offline indoor navigation application for smartphones using augmented reality as a medium for displaying information to the user.

The application is an indoor navigation aid tool with offering the user the ability to view a route using augmented reality, performing route recalculation and position correction using QR(Quick Response Scanner). The application also contains audio assistance instructions. The application development has been achieved using Unity game engine. At the beginning of this thesis there are references about the technologies used in this project, hardware and algorithms, as well as general information about augmented reality. Next there are core information about Unity game engine elements and at last a detailed explanation about the application and its use, as well as a manual for the user.

Table of Contents

ΠΕΡΙΛΗΨΗ	II
ABSTRACT	IV
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	2
1.1 ΟΡΙΣΜΟΣ	2
1.2 ΙΣΤΟΡΙΑ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ	2
1.3 ΤΥΠΟΙ ΣΥΣΚΕΥΩΝ ΓΙΑ ΤΗΝ ΧΡΗΣΗ AR	4
1.4 ΑΙΣΘΗΤΗΡΕΣ ΣΥΣΚΕΥΩΝ	9
1.5 ΤΡΟΠΟΙ ΥΛΟΠΟΙΗΣΗΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ	13
ΚΕΦΑΛΑΙΟ 2 ΠΛΩΗΓΗΣΗ AR	18
2.1 ΠΛΩΗΓΗΣΗ	18
2.1.1 ΠΛΩΗΓΗΣΗ ΣΕ ΕΞΩΤΕΡΙΚΟΥΣ ΧΩΡΟΥΣ	20
2.1.2 ΠΛΩΗΓΗΣΗ ΣΕ ΕΣΩΤΕΡΙΚΟΥΣ ΧΩΡΟΥΣ	21
2.2 ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΟΠΟΙΗΣΗΣ ΚΑΙ ΧΑΡΤΟΓΡΑΦΗΣΗΣ	23
2.2.1 ΔΙΑΔΙΚΑΣΙΑ SLAM ΚΑΙ ΚΑΤΑΝΟΗΣΗ ΣΧΕΣΕΩΝ ΣΕ ΣΗΜΕΙΑ ΤΟΥ ΧΩΡΟΥ	24
2.2.2 ΑΝΑΤΟΜΙΑ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ SLAM	26
2.3 SLAM ΚΑΙ ARCORE	27
ΚΕΦΑΛΑΙΟ 3 UNITY	29
3.1 ΜΗΧΑΝΗ ΑΝΑΠΤΥΞΗΣ ΠΑΙΧΝΙΔΙΩΝ UNITY	29
3.2 Ο ΕΔΙΤΟΡ ΚΑΙ ΤΑ ΠΕΔΙΑ ΤΟΥ	30
3.3 ΣΤΟΙΧΕΙΑ ΤΟΥ UNITY	36
ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ	42
4.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ	42
4.2 ΠΕΡΙΓΡΑΦΗ ΣΚΗΝΗΣ UNITY	43
4.2.1 ΣΤΑΔΙΟ ΑΡΧΙΚΟΠΟΙΗΣΗΣ SESSION ΚΑΙ ARCORE DEVICE	45
4.2.2 ΣΥΣΧΕΤΙΣΜΟΣ ΚΑΜΕΡΑΣ AR ΚΑΙ ΠΛΩΗΓΗΣΗ	52
4.2.3 ΚΑΜΕΡΑ QR ΚΑΙ ΕΠΑΝΑΠΡΟΣΔΙΟΡΙΣΜΟΣ ΘΕΣΗΣ	71
4.3 ΠΛΩΗΓΗΣΗ ΚΑΙ ΧΡΗΣΗ ΕΦΑΡΜΟΓΗΣ	74
ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ	85
ΒΙΒΛΙΟΓΡΑΦΙΑ	86

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Ορισμός

Επαυξημένη πραγματικότητα είναι μία τεχνολογία η οποία επιτρέπει την ενσωμάτωση της ψηφιακής πληροφορίας στο περιβάλλον χρήστη σε πραγματικό χρόνο. Augmented Reality όπως είναι η αγγλική ερμηνεία ή αλλιώς AR σε συντομία είναι ένα είδους διαδραστικού περιβάλλοντος το οποίο χρησιμοποιεί όσους περισσότερους πόρους του παρέχεται από μία συσκευή (εικόνα, ήχος, εφέ, γραφικά) με σκοπό να ενισχύσει την πραγματική εμπειρία χρήστη. Είναι ουσιαστικά ένα εργαλείο που έχει την ικανότητα να χαρακτηρίζει τον κόσμο γύρω μας με ακόμα μεγαλύτερη ακρίβεια και λεπτομέρεια, παρέχοντας επιπλέον γνώση που δεν μας παρέχει το περιβάλλον μας. Σε αντίθεση με την εικονική πραγματικότητα που ουσιαστικά είναι ολοκληρωτικά δομημένη σε ψηφιακή μορφή, η επαυξημένη πραγματικότητα προσθέτει την πληροφορία, απεικονίζει δηλαδή δισδιάστατα (σε σχέση με την οθόνη) ή τρισδιάστατα γραφικά, σε μορφή στρωμάτων πάνω από τον πραγματικό κόσμο, με την βοήθεια κάποιας συσκευής.

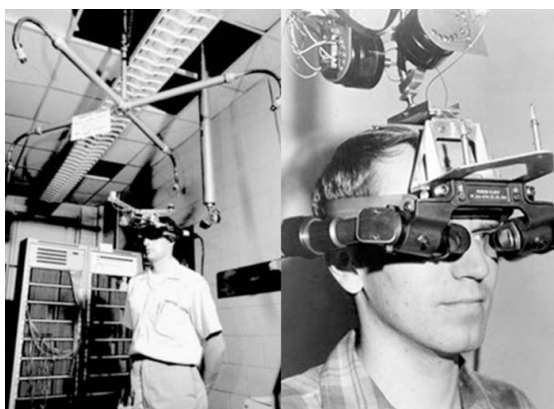
Ο όρος “Επαυξημένη Πραγματικότητα” επινοήθηκε στην εταιρία Boeing πρώτη φορά το 1990 από τον ερευνητή Thomas Caudell του οποίου είχε ζητηθεί, αυτού και του συνάδελφό του David Mizell, να ανακαλύψουν έναν εναλλακτικό τρόπο, από αυτό των ως τότε, διαγραμμάτων και των συσκευών σήμανσης που χρησιμοποιούσαν στα εργοστάσια. Ο Thomas και David πρότειναν να αντικαταστήσουν τις μεγάλες πλάκες ξύλου που απεικόνιζαν τα σχέδια, ξεχωριστά το καθένα, των ορόφων του εργοστασίου με μία είδους συσκευής με ενσωματωμένα γυαλιά που εφαρμόζουν στο κεφάλι και πρόβαλλαν οποιοδήποτε διάγραμμα επιλέξει ο χρήστης σε έναν επαναχρησιμοποιήσιμο πίνακα (εικόνα 1.2). Το αποτέλεσμα ήταν αντί οι εργάτες να έχουν πολλούς και διαφορετικούς πίνακες με διαγράμματα του εργοστασίου να έχουν μόνο μια συσκευή που εξυπηρετεί τον ίδιο σκοπό πολύ πιο οικονομικά και πρακτικά [1].

1.2 Ιστορία Επαυξημένης Πραγματικότητας

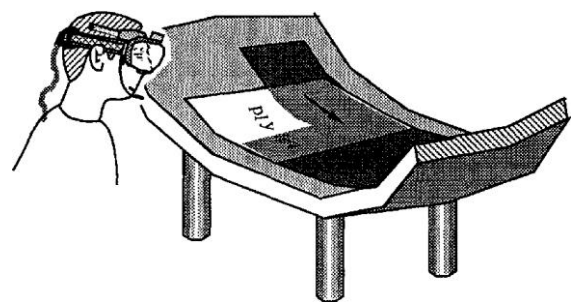
Η επαυξημένη πραγματικότητα συλλήφθηκε πρώτη φορά ως ιδέα αρχικά από τον κινηματογραφιστή Morton Heiling το 1957 του οποίου στόχος ήταν μια καλύτερη εμπειρία για τον θεατή. Το Sensorama, όπως αποκαλείται, είναι μια συσκευή που προσομοιώνει

επιπρόσθετες αισθήσεις (εκτός της όρασης και του ήχου), που δεν έχει πρόσβαση ένας θεατής κατά την διάρκεια μίας ταινίας σε έναν συνηθισμένο χώρο προβολής, όπως η μυρωδιά και η δόνηση. Η συσκευή δεν αποτελούταν από ηλεκτρονικά μέρη αλλά θεωρείται ως η πρώτη απόπειρα αφού εξυπηρετεί τον ίδιο σκοπό μέχρι και σήμερα, την προσθήκη επιπρόσθετων δεδομένων σε μία εμπειρία χρήστη.

Λίγα χρόνια αργότερα το 1968 ο Αμερικανός Ivan Sutherland του πανεπιστήμιο Harvard που θεωρείται και ο “πατέρας των γραφικών στους υπολογιστές” και από τους πρωτοπόρους του διαδικτύου επινόησε μια συσκευή που εφάρμοζε στο κεφάλι ως ένα είδος έμμεσου παραθύρου στον εικονικό κόσμο [2]. Το Sword Of Damocles (εικόνα1.1) όπως αποκαλούνταν ήταν η πρώτη συσκευή εικονικής πραγματικότητας (virtual reality head mounted display VR-HMD) όπως και της επαυξημένης πραγματικότητας [3]. Η συσκευή αρχικά ήταν αρκετά δύσχρηστη και περιορισμένη αφού αποτελούταν από έναν μηχανικό βραχίονα ο οποίος ήταν προσκολλημένος στην οροφή του δωματίου και ήταν υπεύθυνος για την τον εντοπισμό της κεφαλής του χρήστη. Ακόμα, για την συσκευή ήταν απαραίτητη ο προσανατολισμός του βλέμματος. Το σύστημα αναπτύχθηκε περαιτέρω στα επόμενα χρόνια με συνολικά να το αποτελούν έξι υποσυστήματα: έναν διαχωριστή τάσης, έναν πολλαπλασιαστή matrix, μια γεννήτρια φορέων, ένα ακουστικό, έναν αισθητήρα θέσης κεφαλής και έναν υπολογιστή γενικού σκοπού (a clipping divider, matrix multiplier, vector generator, headset, head position sensor, and a general-purpose computer) ενώ η μονάδα ήταν μερικώς διαμπερές και αναπαριστάνονταν απλά γεωμετρικά σχήματα.



Εικόνα 1.1 Συσκευή Sword of Democles



Εικόνα 1.2 Συσκευή τύπου HUD για την προβολή γραφικών προτύπων πάνω σε ειδική επιφάνεια.

Το 1975 ο Myron Krueger ίδρυσε ένα εργαστήριο τεχνητής πραγματικότητας που ονόμασε Videoplace. Ο κύριος στόχος του ήταν να αποδεσμευτεί ο χρήστης από οποιαδήποτε επιπρόσθετη συσκευή. Η συσκευή λειτουργούσε χωρίς ειδικά γάντια ή γυαλιά αλλά με την χρήση προβολέων, κάμερες βιντεοσκόπησης σιλουέτες από τους χρήστες και άλλα υλικά και επέτρεπε μέσω χειρονομιών από τον χρήστη τον έλεγχο σχημάτων (να αλλάξουν μέγεθος σε μια φιγούρα να την μετακινήσουν στον χώρο κ.α) [4]. Οι χρήστες μπορούσαν να είναι σε διαφορετικά δωμάτια και να αλληλεπιδρούν μεταξύ τους μέσω αυτής της τεχνολογίας.

Φυσικά οι παραπάνω προσπάθειες ήταν πρωτόγνωρες και καινούριες στον χώρο της επιστήμης επαυξημένης και εικονικής πραγματικότητας οπότε και ο όρος “επαυξημένη πραγματικότητα” δεν ήταν καθιερωμένος σαν ορισμός μέχρι και το 1989 από τον Jaron Lainer (Virtual Reality) και τον P Caudell (Augmented Reality) το 1990. Τα πρώτα πλήρως ανεπτυγμένα συστήματα επαυξημένης πραγματικότητας αναπτύχθηκαν στο εργαστήριο USAF Armstrong από τον Louis Rosenberg το 1992, όπως και το 1998 χρησιμοποιήθηκε για πρώτη φορά από το διαστημόπλοιο X-38 του κρατικού οργανισμού “NASA” ως σύστημα πλοήγησης [5].

Το 2000 δημιουργήθηκε το πρώτο παιχνίδι επαυξημένης πραγματικότητας. Οι χρήστες έπρεπε να διαθέτουν head-mounted display δηλαδή οθόνη για το κεφάλι τους και ένα χώρο, ένα σακίδιο ώστε να μεταφέρουν έναν υπολογιστή και γυροσκόπια για τον προσανατολισμό. Από το 2010 και μετά, με την άνοδο των έξυπνων κινητών (smartphones) η χρήση της επαυξημένης πραγματικότητας σε εφαρμογές αυξήθηκε όπως και η ζήτηση στην αγορά αφού εξυπηρετούσε όχι μόνο στο τμήμα της ψυχαγωγίας αλλά και της επιστήμης [6].

1.3 Τύποι συσκευών για την χρήση AR

Από το 1957 μέχρι και το 2010 και έπειτα η τεχνολογία όσο αναφορά τα συστήματα επαυξημένης πραγματικότητας έχει αναπτυχθεί αρκετά με αποτέλεσμα την πιο οικονομική και πιο εύκολη κατασκευή υλικού και λογισμικών ως προς την αγορά. Η αρχιτεκτονική και οι πρακτικές έχουν μείνει κατά το μεγαλύτερο μέρος ίδιες και ο στόχος είναι κοινός με τον

αρχικό, η δημιουργία ενός εικονικού κόσμου σαν στρώμα πάνω στον πραγματικό. Οι επικρατέστερες τεχνολογίες που υπάρχουν από το 2000 και μετά είναι οι HUD η Head-Up Display, Head-Mounted Display HMD, τα handheld AR, οι οθόνες ολογράμματος(Holographic displays) και τα πιο ευρέως γνωστά τα “έξυπνα γυαλιά” ή smart glasses. Όλα τα παραπάνω εφαρμόζονται ή χρησιμοποιούνται από εταιρίες παγκοσμίως όμως η αγορά που αφορά το μεγαλύτερο μέρος των καταναλωτών είναι αυτή των handheld AR συσκευών που ανήκουν τα κινητά αφού είναι και τα πιο προσβάσιμα καθώς αποτελούν απαραίτητη συσκευή στην καθημερινότητα [7].

- **Head-Up displays:** Γνωστές και ως HUD, είναι διαφανείς οθόνες προβολής δεδομένων που χαρακτηρίζονται από την δυνατότητα που προσφέρουν στον χρήστη να μην υπάρχει η ανάγκη να αλλάζουν την οπτική γωνία τους κατά την διάρκεια της χρήσης τους. Το όνομα προέρχεται από τις κινήσεις που κάνουν οι πιλότοι σε μαχητικά αεροπλάνα κυρίως (Heads-Up – Κεφάλι προς τα πάνω) αφού είναι αναγκαίο να έχουν ορατότητα προς όλες τις κατευθύνσεις και παράλληλα να παρατηρούν τα όργανα στο πιλοτήριο (κατεύθυνση του οπτικού πεδίου προς τα κάτω). Οι οθόνες HUD (εικόνα 1.3) έχουν το πλεονέκτημα να προβάλλουν ότι απαραίτητη πληροφορία χρειάζεται ένας πιλότος όπως ταχύτητα, κατεύθυνση, ύψος και συντεταγμένες ως ένα επιπρόσθετο στρώμα στο οπτικό πεδίο του. Επίσης ιδιαίτερο πλεονέκτημα είναι ότι δεν χρειάζεται τα μάτια του οδηγού-πιλότου να επανεστιάζουν σε διαφορετικά σημεία για να δουν την πληροφορία εφόσον οι οθόνες είναι σχεδιασμένες ώστε να το αποτρέπουν αυτό και τοποθετημένες κατάλληλα. Για παράδειγμα ένας τρόπος απεικόνισης HUD σε ένα αυτοκίνητο μπορεί να έχει τοποθετημένες οθόνες στον καθρέπτη πόρτας στην γυάλινη προστασία στο οπτικό πεδίο του οδηγού και στον μικρό καθρέπτη [8]. Αρχικά οι οθόνες HUD σχεδιάστηκαν για την χρήση τους στην στρατιωτική αεροπορία αλλά τώρα είναι ιδιαίτερα αισθητή η παρουσία τους στα εμπορικά σκάφη, στο χώρο της αυτοκινητοβιομηχανίας, της ιατρικής και άλλους τομείς [9]. Τα Helmet Mounted Displays (εικόνα 1.4) βρίσκονται στην ίδια κατηγορία με τα Head-Up Displays και χρησιμοποιούνται κυρίως με τον ίδιο σκοπό. Τα HMD είναι φορητά κράνη που αποτελούνται από αντίστοιχες οθόνες για την χρήση AR οι οποίες όμως είναι τοποθετημένες εντός της συσκευής.



Εικόνα 1.3 Προβολή τύπου Head Up Display.



Εικόνα 1.4 Συσκευή τύπου Helmet Mounted Display.

- Handheld Displays:** Οι φορητές οθόνες επαυξημένης πραγματικότητας είναι από τα επικρατέστερα μέσα στα οποία έχει πρόσβαση το μεγαλύτερο ποσοστό του κόσμου. Οι συσκευές αυτές που αποτελούν την κατηγορία των συσκευών χειρός είναι τα Κινητά τηλέφωνα-Smartphones, τα Tablet, και οι συσκευές PDA(personal digital assistant). Η κάθε συσκευή έχει συγκεκριμένα πλεονεκτήματα έναντι των υπολοίπων ενώ χρησιμοποιούνται όλα αναλόγως τις απαιτήσεις του χρήστη. Οι συσκευές χειρός (Handheld Displays) είναι ευρέως γνωστές στο κοινό γιατί είναι συσκευές που χρησιμοποιούνται ήδη στην καθημερινότητα ενώ παράλληλα διαθέτουν τα απαραίτητα υλικά που χρειάζεται μια εφαρμογή AR για να λειτουργήσει με μεγαλύτερη ακρίβεια. Οι HD χρησιμοποιούνται κυρίως για εφαρμογές καθημερινού σκοπού (πχ. Φωτογραφίες, μηνύματα, τηλέφωνο, πλοήγηση), διαθέτουν το απαραίτητο λογισμικό και υλικό με όλους τους κατάλληλους αισθητήρες (MEMS) όπως επιταχυνσιόμετρο, συστήματα GPS, solid state compass (πυξίδα) και κάμερα. Οι κατηγορίες έχουν μερικές διαφορές μεταξύ τους και χρησιμοποιούνται αναλόγως τον σκοπό. Τα smartphone έχουν θετικό το μικρο μέγεθος τους και την εύκολη χρήση τους από τους χρήστες καθώς είναι το πιο κοινό αντικείμενο μεταξύ . Τα tablet έχουν ως κύριο πλεονέκτημα ,σε σχέση με τα υπόλοιπα, την μεγαλύτερη διάμετρο οθόνης δημιουργώντας μία πιο εύκολη και δελεαστική εμπειρία στον χρήστη εφόσον προβάλει τα αντικείμενα AR φυσικά πιο μεγάλα ενώ σαν υλικό είναι παραπλήσιο των smartphone. Τα συστήματα PDA [10] είναι ουσιαστικά φορητοί υπολογιστές με την μορφή smartphone με μεγαλύτερη συνήθως υπολογιστική δύναμη και παρόμοια φορητότητα με αυτή των smartphone αλλά έχουν και μεγαλύτερο κόστος. [11]

- **Spatial AR:** Η τεχνολογία αυτή είναι σχετικά καινούρια αφού δρομολογείται γύρω στο 1990 όπου είχε πιο πολύ τον χαρακτηρισμό “video mapping”. Η τεχνική επαυξημένης πραγματικότητας με την χρήση προβολής έλαβε χώρα πρώτη φορά σε κοινό στο θεματικό πάρκο Disneyland, σε εγκαίνια από το τρενάκι του τρόμου “Haunted Mansion” όπου μέλη των τραγουδιστών της ομώνυμης γυρίστηκαν πρώτα σε ταινία των 16mm και έπειτα προβάλλονταν σε προτομές των προσώπων τους έτσι ώστε να φαντάζουν ότι κινούνται κινούμενα (animated projection) (εικόνα 1.5). [12]



Εικόνα 1.5 Προτομές προσώπων από χαρακτήρες της ταινίας Haunted Mansion.

Το Spatial AR αρχικά χρειάζεται ένα φυσικό τρισδιάστατο υλικό το οποίο να είναι κατάλληλα κατασκευασμένο στο οποίο θα πραγματοποιηθεί έπειτα μία προβολή. Στο επόμενο στάδιο, κατάλληλο λογισμικό θα χαρτογραφήσει τις επιφάνειες που θα χρησιμοποιηθούν για την προβολή ενώ θα δημιουργηθεί το αντίστοιχο περιβάλλον μέσα στο λογισμικό. Έπειτα με την χρήση κάποιου προβολέα τοποθετημένο σε συγκεκριμένη απόσταση και θέση από το αντικείμενο θα αντιστοιχηθούν οι εικόνες στα κατάλληλα σημεία δημιουργώντας την εμπειρία επαυξημένης πραγματικότητας (εικόνα 1.6).



Εικόνα 1.6 Διαφορετική προσέγγιση Spatial AR με επιτραπέζια τρισδιάστατα μοντέλα, για την προσομοίωση βάθους και λεπτομέρειας.

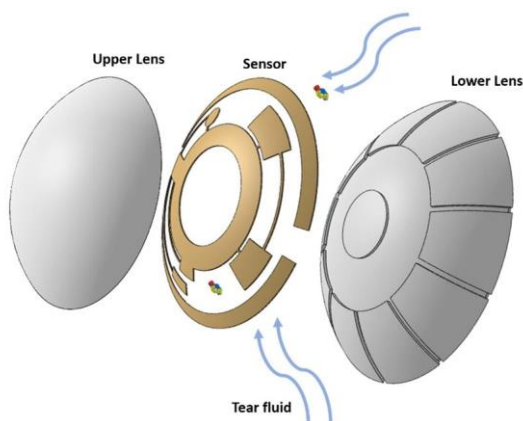
- **Eye-Tap:** Όπως και η τεχνολογία HUD, η προσέγγιση της τεχνολογίας Eye-Tap βασίζεται σε μία συσκευή που τοποθετούνται στην κεφαλή του χρήστη και με την χρήση διαμπερές υλικού την προβολή εικόνων πάνω σε αυτό. Συσκευές τέτοιου τύπου μοιάζουν αρκετά με γυαλιά οράσεως με κάποια επιπλέον χαρακτηριστικά και προσεγγίζουν μία πιο απλοποιημένη εκδοχή των HUD, με την ίδια ακρίβεια και υπολογιστική δύναμη ανεξάρτητα το μικρότερο τους μέγεθος (εικόνα 1.7).



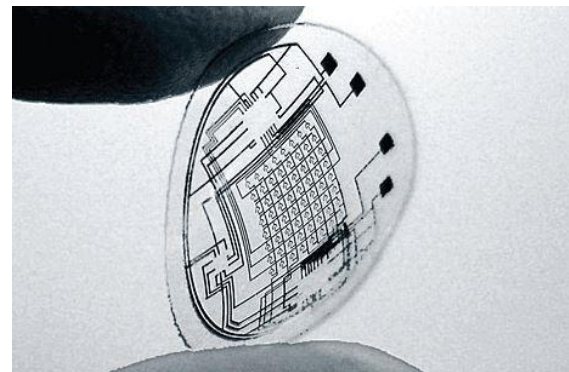
Εικόνα 1.7 Γυαλιά οράσεως με επιπρόσθετη συσκευή που επιτρέπει την λειτουργία AR στο υπάρχον οπτικό πεδίο του χρήστη.

- **Contact lenses:** Η ιδέα για φακούς επαφής επαυξημένης πραγματικότητας πρωτοεμφανίστηκε όταν το 1999 χορηγήθηκε Καναδικό δίπλωμα ευρεσιτεχνίας στον

καθηγητή Steve Mann του πανεπιστημίου Toronto για την ιδέα φακών επαφής που χρησιμοποιούσαν ολογραφικά στοιχεία εστιάζοντας σε κοντινές οθόνες. Ενώ η ιδέα εγκαταλείφθηκε γρήγορα εννέα χρόνια αργότερα η εταιρία Mojo Lens δημιουργήθηκε με σκοπό την δημιουργία AR Lens [13]. Η εταιρία ισχυρίζεται ότι ακόμα και για την αρχή του έργου ήρθε αντιμέτωπη με την εφεύρεση νέων τεχνολογιών όπως η οθόνη Micro-LED που διαθέτει έως και 14.000 PPI(Pixels Per Inches), power data όπως και νέους αλγόριθμους eye-tracking [14]. Τα Mojo Lens και αντίστοιχες εταιρίες ανάπτυξης φακών επαφής σε εφαρμογές επαυξημένης πραγματικότητας είναι ακόμα πρωτότυπα όμως είναι πιθανόν το μέλλον σε τεχνολογίες AR (εικόνα 1.8) (εικόνα 1.9).



Εικόνας 1.8 Φακός επαφής αποτελούμενος από τρία τμήματα, δύο φακών και αισθητήρα ανάμεσα που παρέχει την δυνατότητα AR.



Εικόνα 1.9 Φακός επαφής AR.

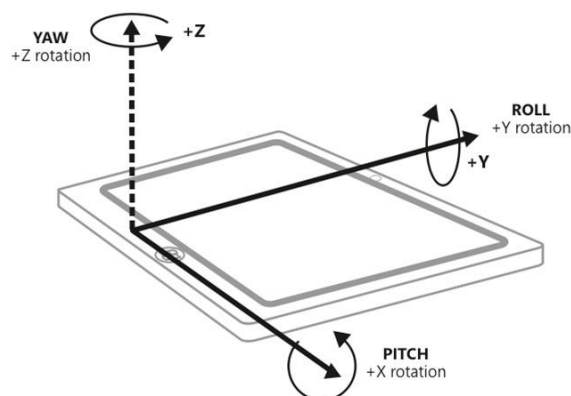
1.4 Αισθητήρες συσκευών

Οι αισθητήρες σε ηλεκτρονικές συσκευές αποτελούν το πιο σημαντικό στοιχείο αλληλεπίδρασης με το περιβάλλον καθώς είναι υπεύθυνα για την ανίχνευση συμβάντων και αλλαγών μέσα σε αυτό όπως και η μεταφορά των δεδομένων σε κάποια βασική δομή της συσκευής για επεξεργασία. Κάθε συσκευή που αναφέρθηκε στο τελευταίο κεφάλαιο της προηγούμενης ενότητας περιέχει πλήθος από τους παρακάτω αισθητήρες. Η διαφορά των παραπάνω συσκευών γενικότερα άλλα και γενικότερα είναι ο τρόπος χρήσης των αισθητήρων με βάση το λογισμικό αλλά και το υλικό που περιέχεται μέσα σε μία συσκευή.

Κάθε αισθητήρας είναι διαφορετικός και υπηρετεί συγκεκριμένο σκοπό προκειμένου να λειτουργήσει σωστά μία εφαρμογή ή υπηρεσία που εξαρτάται ρητά από αυτόν.

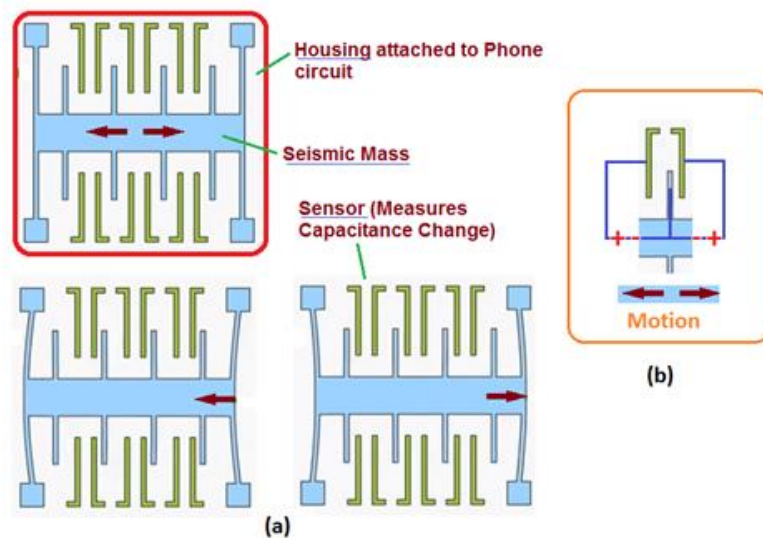
Παρακάτω γίνεται αναφορά ως προς τους αισθητήρες που περιέχονται στις περισσότερες κινητές συσκευές και τον σκοπό που εξυπηρετούν.

- **Magnetometer:** Το μαγνητόμετρο είναι ένας αισθητήρας που μετράει τον μαγνητισμό. Πιο συγκεκριμένα την κατεύθυνση, την δύναμη και την σχετική αλλαγή ενός μαγνητικού πεδίου. Ο αισθητήρας χρησιμοποιεί την σχετικά σύγχρονη τεχνολογία solid state με σκοπό να δημιουργήσει το Hall Effect, το οποίο είναι ένας τύπος μετατροπέα για την μέτρηση μαγνητικού πεδίου. Εξυπηρετεί επίσης και ως πυξίδα σε κινητά (smartphones) [15].
- **Gyroscope:** Το γυροσκόπιο είναι ένας αισθητήρας που δίνει πληροφορίες για την γωνία και ανιχνεύει αν κινείται μία συσκευή (εικόνα 2.0). Ο τρόπος υπολογισμού κίνησης και περιστροφής της συσκευής γίνεται με την χρήση μετρητών καταπόνησης, αντιστάσεων οι οποίοι είναι ευαίσθητοι στην κίνηση και δίνουν διαφορετικά δεδομένα όταν αλλάζει ο προσανατολισμός μιας συσκευής. Επίσης για την σωστή λειτουργία του αισθητήρα χρειάζεται να γίνεται βαθμονόμηση για τον σωστό υπολογισμό δεδομένων [16].



Εικόνα 2.0 Συσκευή που χρησιμοποιεί το γυροσκόπιο μπορεί να καταλάβει την γωνιακή επιτάχυνση.

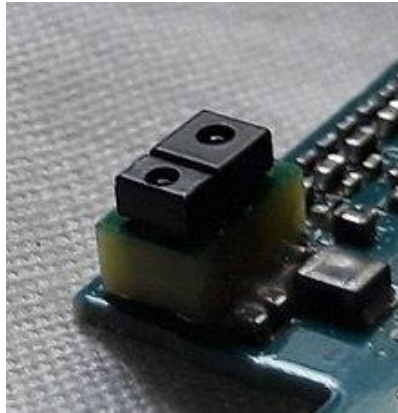
- Accelerometer:** Παρατηρεί σημειώνει αλλαγές στην κίνηση ταχύτητα και περιστροφή συσκευή. Το επιταχυνσιόμετρο είναι ένας αισθητήρας που χρησιμοποιείται για την ανίχνευση κίνησης και περιστροφής σε μία συσκευή. Σε σχέση με το γυροσκόπιο είναι σχεδιασμένο να αντιδράει σε μη βαρυτική επιτάχυνση και ενεργοποιείται με υπερβολικά μικρές κινήσεις της συσκευής και δεν μπορεί να διαχωρίσει την περιστροφή από την επιτάχυνση. Υπάρχουν δύο είδη λειτουργίας. Το πρώτο χρησιμοποιεί μικροσκοπικούς κρυστάλλους ενεργοποιώντας τους με τις δονήσεις από την κίνηση της συσκευής επιτρέποντας να περάσει τάση από μέσα τους δημιουργώντας ένα πέρασμα για την ανάγνωση δεδομένων από τον αισθητήρα. Το δεύτερο χρησιμοποιεί μία δομή από σιλικόνη και πυκνωτές ευαίσθητους στην κίνηση και ανάλογα ασκείται δύναμη στους πυκνωτές οι οποίοι δίνουν τα κατάλληλα δεδομένα στην συσκευή που τα χρησιμοποιεί (εικόνα 2.1) [17].



Εικόνα 2.1 Το επιταχυνσιόμετρο προκειμένου να δείξει την κατεύθυνση της συσκευής μετράει την αλλαγή χωρητικότητας.

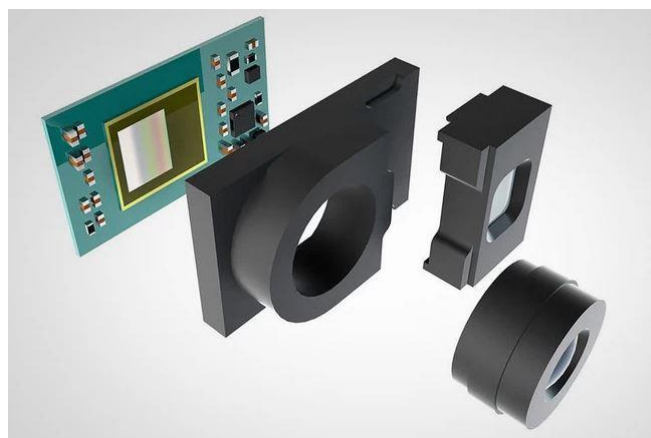
- Proximity Sensor:** Οι αισθητήρες απόστασης/προσέγγισης είναι υπεύθυνοι για την απόσταση, δηλαδή πόσο κοντά ή πόσο μακριά είναι ένα αντικείμενο. Μια συχνή χρήση του είναι η απενεργοποίηση της οθόνης στα smartphone, δηλαδή η λειτουργία ένδειξης πληροφορίας και η λήψη πληροφορίας από την οθόνη αφής, καθώς ο χρήστης την πλησιάζει, προκειμένου να αποτρέψει μη επιθυμητές ενέργειες. Ο αισθητήρας συνδυάζει υπέρυθρες ακτίνες LED με έναν ανιχνευτή εκπομπής φωτός τα

οποία μετρούν την απόσταση από ένα αντικείμενο καθώς η ακτίνα αντανακλά πάνω σε αυτό (εικόνα 2.2) [18].



Εικόνα 2.2 Αισθητήρας εγγύτητας με υπέρυθρες ακτίνες LED.

- **ToF cameras:** Οι κάμερες ToF, ή Time Of Flight Camera χρησιμοποιούν υπέρυθρες για να προσδιορίσουν πληροφορίες για την απόσταση ενός αντικειμένου (εικόνα 2.3). Ο υπολογισμός της απόστασης γίνεται εκπέμποντας παλμούς φωτός (infrared laser) σε μια επιφάνεια μετρώντας τον συνολικό χρόνο που κάνει το φως στον κύκλο του από την στιγμή που στάλθηκε μέχρι να γυρίσει πίσω στον αισθητήρα. Η χρήση αυτών των αισθητήρων μέτρησης βάθους έχει αρκετά πλεονεκτήματα σε σχέση με άλλους λόγω του φθηνού κόστους, ότι χρειάζονται μικρή υπολογιστική ισχύ από την ΚΜΕ (Κεντρική Μονάδα Επεξεργασίας). Επίσης είναι αρκετά χρήσιμοι στο πεδίο AR αφού έχουν πολύ καλή ανταπόκριση σε πραγματικό χρόνο, κάτι απαραίτητο για εφαρμογές AR [19].



Εικόνα 2.3 Τμήματα αισθητήρα υπολογισμού απόστασης TOF camera.

- **GPS**

Ακόμα ένας αισθητήρας απαραίτητος αισθητήρας για τις περισσότερες συσκευές. Το GPS η αλλιώς το Global Positioning System (Παγκόσμιο Σύστημα Θεσιθεσίας) είναι ένας αισθητήρας που βασίζεται στην χρήση δορυφόρων που περιστρέφονται γύρω από την γη. Οι δορυφόροι αυτοί είναι κατάλληλα τοποθετημένοι ώστε κάθε συσκευή, αισθητήρας στον πλανήτη να έχει διαθέσιμους τουλάχιστον τέσσερις (αν και για τον υπολογισμό θέσης είναι απαραίτητοι τουλάχιστον τρεις). Οι δορυφόροι είναι εξοπλισμένοι με ατομικά ρολόγια για τον σωστό υπολογισμό τοποθεσίας. Ενώ το GPS δεν χρειάζεται να έχει πρόσβαση στο διαδίκτυο η λεπτομερής ψηφιακή απεικόνιση στον χρήστη χρειάζεται πρόσβαση, οπότε περιορίζεται αρκετά η χρήση του σε εφαρμογές επαυξημένης πραγματικότητας [20].

1.5 Τρόποι Υλοποίησης Επαυξημένης Πραγματικότητας

Η επαυξημένη πραγματικότητα αν και έχει έναν συγκεκριμένο στόχο την προβολή τρισδιάστατων γραφικών ως ένα επιπλέον στρώμα στην πραγματικότητα ως ένα εργαλείο προς τους ανθρώπους όμως η χρήση της δεν περιορίζεται μόνο εκεί. Εφόσον υπάρχουν διαφορετικές ομάδες ανθρώπων έχουν πρόσβαση σε αυτή την τεχνολογία ανάλογα τις ανάγκες τους επινοούν επιπλέον στόχους και χρήσεις που δεν είχαν τεθεί αρχικά. Αυτό μπορούμε να το δούμε από τα διάφορα πεδία επιστήμης που χρησιμοποιούν AR τεχνολογίες όπως η ιατρική, οι εκπαιδεύσεις στην αεροπορία, στον τουρισμό, σε πλοήγηση, στην βιομηχανία των ηλεκτρονικών παιχνιδιών κ.α. Επομένως είναι λογικό να συμπεράνουμε ότι χρειάζονται και διαφορετικά είδη ανάπτυξης και χρήσης των αλγόριθμων και των υλικών για την ανάπτυξη μιας εφαρμογής/υπηρεσίας AR [21].

Δύο μεγάλες κατηγορίες είναι οι Trigger based και View based [22].

Η κατηγορία Trigger Based αποτελείται από τις εξής υποκατηγορίες:

- **Trigger Based** Είναι κατηγορία που βασίζεται σε κάποιο εξωτερικό ερέθισμα είτε με την μορφή εικόνας ή συγκεκριμένων αντικειμένων των οποίων έχει γίνει ήδη η αναγνώριση πριν την εκτέλεση του κώδικα/προγράμματος (εικόνα 2.4) ή ακόμα και δεδομένα από GPS με στόχο την προβολή των εικονικών αντικειμένων [22].



Εικόνα 2.4 Αναπαράσταση Trigger Based AR. Ο καναπές είναι ένα τρισδιάστατο αντικείμενο εικονιζόμενο σε κινητό "πάνω" από τον πραγματικό χώρο με δυνατότητα τοποθέτησης του ή προσανατολισμού του.

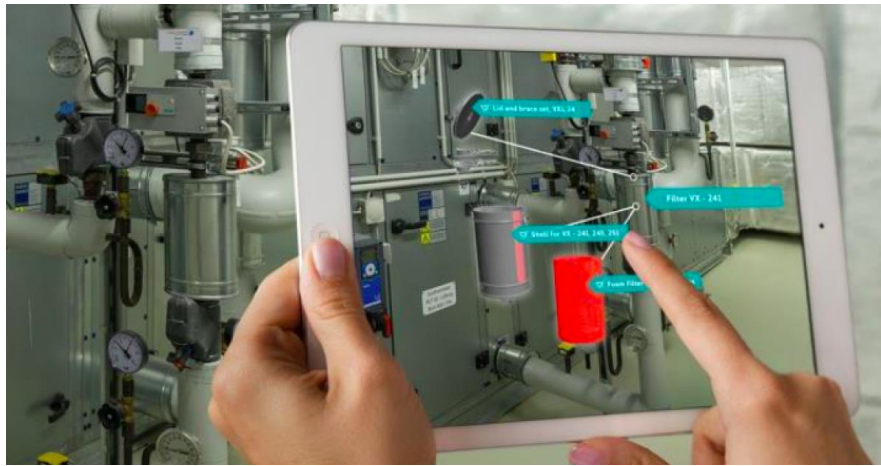
- **Marker-Based:** Η χρήση των marker-based εφαρμογών απαιτεί απαραίτητη τη χρήση μίας εικόνας ή ενός οποιουδήποτε φυσικού σημάδι (marker) στον χώρο είτε αυτό είναι κάποιο απλό χαρτί με εκτυπωμένο το αντίστοιχο σημάδι ή οποιοδήποτε προκαθορισμένο σχήμα ή εικόνα. Όταν γίνεται η αναγνώριση μπορεί να πραγματοποιηθεί μία αντιστοιχία φυσικού-εικονικού περιεχομένου, όπως για παράδειγμα η αναπαράσταση ενός τρισδιάστατου αντικείμενου να αιωρείται στην οθόνη "πάνω" από το marker. Η διαδικασία αναγνώρισης μία εικόνας που έχει οριστεί ως σημάδι από μία εφαρμογή AR είναι η εξής:

Λήψη στιγμιότυπου με την κάμερα, επεξεργασία του στιγμιότυπου σε χαμηλό επίπεδο, διαδικασίες επαναλήψεων αλγόριθμων για απόρριψη αντικειμένων ή εικόνων που υπάρχουν στο στιγμιότυπο και δεν αντιστοιχούν με τα σημάδια, πιθανά στιγμιότυπα που μπορεί να είναι marker, διόρθωση παραμόρφωσης εικόνας και εντοπισμός ορίων, αποκωδικοποίηση των σημαδιών και αντιστοίχιση με έτοιμα σημάδια, υπολογισμός θέσης και προσανατολισμού για την τοποθέτηση προβολής (εικόνα 2.5) [23] [24].



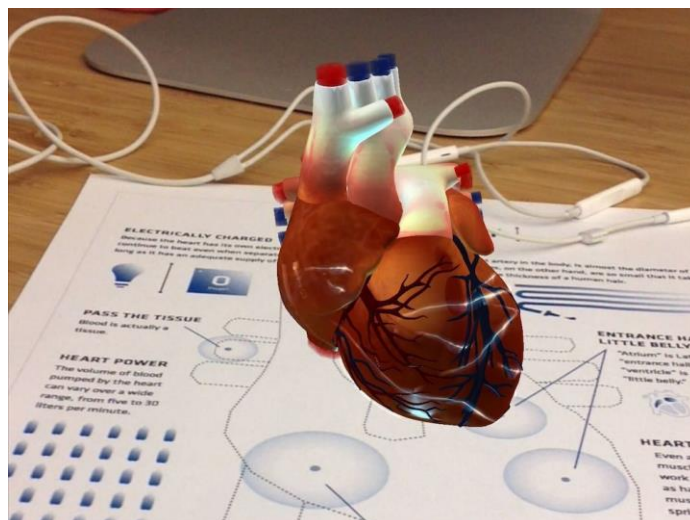
Εικόνα 2.5 Αναπαράσταση Marker Based AR. Το κινητό αφού ανιχνεύσει το σημάδι απεικονίζει ένα σπίτι "πάνω" από αυτό διατηρώντας σταθερή την θέση και τον προσανατολισμό του.

- **Location based:** Χρησιμοποιεί το GPS της συσκευής ώστε να δημιουργήσει σημεία ενδιαφέροντος. Τα σημεία ενδιαφέροντος είναι κυρίως πληροφορίες αποθηκευμένες από την εφαρμογή ως σημάδια πάνω στον χάρτη. Ο χρήστης που έχει πρόσβαση σε αυτή την εφαρμογή έχει πρόσβαση και στα σημεία ενδιαφέροντος ως προβολές/πληροφορίες στην οθόνη του. Οι εφαρμογές της επαυξημένης πραγματικότητας με τοποθεσία είναι ιδιαίτερα δημοφιλής σε διαδικτυακά παιχνίδια αλλά και στην προβολή χρήσιμων πληροφοριών (μουσεία, εστιατόρια, μνημεία, ψυχαγωγικά ,υπηρεσίες κ.α.) [24].
- **Dynamic Augmentation** Η κατηγορία της Δυναμικής Επαυξημένης Πραγματικότητας μεταφράζεται στην προβολή ενός αντικειμένου καθώς αλλάζει. Πιο συγκεκριμένα αρχικά πρέπει να γίνει η αναγνώριση ενός αντικειμένου επιτρεπόμενο από την εφαρμογή, έπειτα είναι απαραίτητο να γίνεται συνεχής αναγνώριση του αντικειμένου σε κάθε καρέ(frame) και η αποτύπωση των πληροφοριών πάνω στο αντικείμενο. Η δυναμική επαυξημένη πραγματικότητα έχει την δυνατότητα να εντοπίζει αλλαγές στην όψη του αντικειμένου που παρακολουθεί και την προσαρμογή ως προς το σχήμα την γωνία ή το μέγεθος των πληροφοριών στο περιβάλλον (εικόνα 2.6) [24].



Εικόνα 2.6 Αναπαράσταση Dynamic Augmentation. Η συσκευή μπορεί να προβάλει στην οθόνη, δυναμικά, χρήσιμες τρισδιάστατες πληροφορίες ως προς την λειτουργία μίας πραγματικής συσκευής ή μηχανήματος.

- **Complex Augmentation:** Αντιστοιχεί στον συνδυασμό των παραπάνω υποκατηγοριών αφού χρησιμοποιεί το διαδίκτυο, την τεχνολογία GPS και Markers. Ένα παράδειγμα που μπορεί να χαρακτηρίσει την κατηγορία αυτή είναι το πρότζεκτ της Google-”Google Glass” τα οποία με την χρήση GPS διάφορων Marker στον χάρτη και οπτικοακουστικά ερεθίσματα που προσφέρουν στον χρήστη και στο πεδίο όρασης του μία πλήρης εμπειρία επαυξημένης πραγματικότητας. Τα αντικείμενα ,με την “πολύπλοκη επαύξηση”, παρουσιάζονται με πολύτιμες πληροφορίες και την δυνατότητα να τροποποιηθούν ώστε να προσεγγίζουν την “πραγματικότητα” του χρήστη (εικόνα 2.7) [24].



Εικόνα 2.7 Αναπαράσταση Complex Augmentation. Με την ανάγνωση ενός σχήματος σε ένα βιβλίο απεικονίζεται η καρδιά σε τρισδιάστατη μορφή μαζί με πληροφορίες. Ο χρήστης μπορεί να αλληλεπιδράσει με το αντικείμενο.

Η κατηγορία View Based αποτελείται από τις εξής υποκατηγορίες:

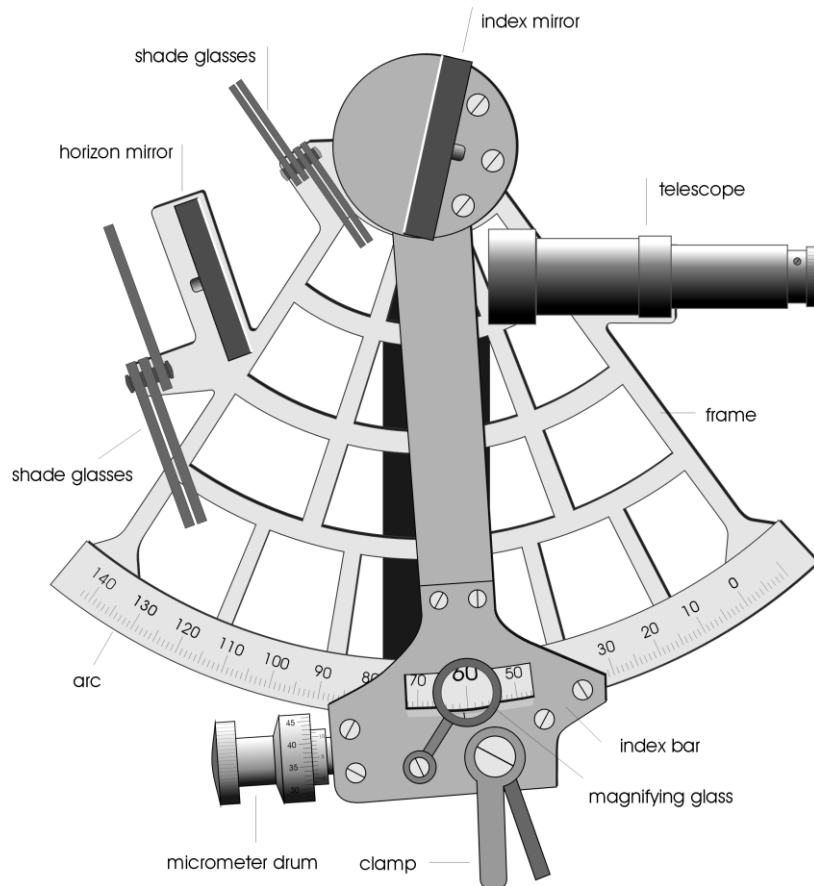
- **Indirect Augmentation:** Έμμεση επαυξημένη πραγματικότητα είναι μία τεχνική που δουλεύει κυρίως ως εργαλείο αφού χρησιμοποιεί στατικές εικόνες του περιβάλλοντος για να επιτύχει τον σκοπό της. Ως παράδειγμα μπορεί να αναφερθεί η φωτογραφία ενός κτηρίου από τον χρήστη και η αλλαγή χρώματος του τοίχου, ή η ακριβής μέτρηση των διαστάσεων του κτηρίου. Είναι αρκετά περιορισμένη υποκατηγορία αφού δεν έχει άμεση αλληλεπίδραση με το περιβάλλον όμως είναι αρκετά ακριβής αφού οι υπολογισμοί πραγματοποιούνται σε ασφαλές μη μεταβαλλόμενο περιβάλλον.
- **Generic Digital Augmentation:** Η τελευταία υποκατηγορία έμμεσος τρόπος προσέγγισης AR που επιτυγχάνει απλά με το να προβάλλει τα ψηφιακά αντικείμενα στην οθόνη σε οποιοδήποτε σημείο του ανατεθεί από το πρόγραμμα χωρίς να υπάρχει κάποια αναφορά στον χώρο. Είναι ιδιαίτερα δημοφιλές σε απλές εφαρμογές κινητών και παιχνίδια [24].

ΚΕΦΑΛΑΙΟ 2 ΠΛΟΗΓΗΣΗ ΑΡ

2.1 Πλοήγηση

Πλοήγηση είναι ένας όρος που αναφέρεται στο επιστημονικό πεδίο που επικεντρώνεται στις ενέργειες που πρέπει να ληφθούν για την παρακολούθηση και τον έλεγχο ενός αντικειμένου όπως ένα όχημα, εφόσον υπάρχει ένας στόχος, ένα σημείο έναρξης και ένα τελικό σημείο προορισμού. Η ετυμολογία της λέξης “Navigation” έχει τις ρίζες από την λατινική λέξη “navigationem” που έχει ως μετάφραση “να πλεύσεις , πλεύσεις στην θάλασσα , να κατευθύνεις ένα πλοίο, ” από την λατινική λέξη navis “ship” , “πλοίο” και agere “ to drive” , “να οδηγήσει” η αλλιώς στην Ελληνική γλώσσα “πλοήγηση” [25].

Η πλοήγηση χρησιμοποιείται ,από τις αρχές του πολιτισμού ,όπου και δημιουργήθηκε από τους ανθρώπους, ως ένα εργαλείο για την μετακίνηση τους. Κατά το μεγαλύτερο μέρος της ιστορίας η κεντρική ιδέα και τεχνικές που χρησιμοποιούνταν ήταν βασισμένες στους χάρτες, τον νυχτερινό ουρανό, κυρίως οι αστερισμοί και συγκεκριμένα άστρα ως σημεία αναφοράς, όπως επίσης το φεγγάρι και ο ήλιος. Αργότερα υπήρχε ανάγκη για την δημιουργία συστημάτων όπως οι συντεταγμένες στους χάρτες ,το γεωγραφικό μήκος και πλάτος. Ο υπολογισμός αποστάσεων και ο προσανατολισμός ήταν, ακόμα, δύσκολος και απαιτούσε αρκετές γνώσεις τόσο ως προς την σωστή ερμηνεία των χαρτών όσο και των αστερισμών και των σημείων αναφοράς που ήταν τότε γνωστά. Οι δυσκολίες όμως αυτές είχαν ως αποτέλεσμα αργότερα την εφεύρεση αντικειμένων όπως ο γνώμονας, το Αραβικό “Kamel”, τον εξάντα (εικόνα 2.8), τον αστρολάβο, την πυξίδα ,κ.α. που διευκόλυνε ιδιαίτερα τον προσανατολισμό του χρήστη και τον έλεγχο την διαδρομής του και των ενεργειών που έπρεπε να λάβει.



Εικόνα 2.8 Απεικόνιση Εξάντα.

Στον σύγχρονο κόσμο ακόμα διδάσκονται τεχνικές προηγούμενων αιώνων και χρήση των εργαλείων τους όμως κατά το μεγαλύτερο τμήμα χρησιμοποιούνται κυρίως ηλεκτρονικά συστήματα και όργανα κατάλληλα για ακριβή υπολογισμό θέσης και προσανατολισμού. Το GPS (σύμπλεγμα δορυφόρων για τον ακριβή υπολογισμό θέσης), συστήματα σόναρ, ραδιοκύματα, ραντάρ, κάμερες και αρκετοί αισθητήρες υπολογισμού απόστασης, βάθους, προσανατολισμού χρησιμοποιούνται συνδυαστικά και μη. Όλοι αυτοί οι αισθητήρες χρησιμοποιούνται στην ναυτιλία, στην αεροπορία και κάποια στην αυτοκινητοβιομηχανία κ.α.

2.1.1 Πλοήγηση Σε Εξωτερικούς Χώρους

Η επαυξημένη πραγματικότητα όπως έχει προαναφερθεί είναι ένα εργαλείο και για αυτό τον λόγο συναντάται σε διάφορα επιστημονικά πεδία και επαγγέλματα. Αν και είναι ακόμα ένα πεδίο της επιστήμης των υπολογιστών που έχει αρκετό περιθώριο βελτίωσης τόσο στο υλικό (hardware) όσο και στην τεχνογνωσία είναι ιδιαίτερα χρήσιμο εργαλείο στο πεδίο της πλοήγησης. Η πλοήγηση σε εξωτερικούς χώρους με την χρήση επαυξημένης πραγματικότητας έχει επιτευχθεί, υλοποιηθεί κατά το μεγαλύτερο μέρος αλλά κυρίως με την αξιοποίηση επιπλέον εργαλείων, συστημάτων ή και αισθητήρων (εικόνα 2.9). Φυσικά αυτό έχει ως συνέπεια οι συσκευές ή τα αντικείμενα, που τα χρησιμοποιούν ως μέσα, να παρουσιάζουν πρακτικούς περιορισμούς λόγω αυτών. Πιο συγκεκριμένα τα περισσότερα από αυτά τα συστήματα καθιστούν απαραίτητη την χρήση διαδικτύου, GPS, από επιταχυνσιόμετρα και την συνεχή ροή δεδομένων από και προς εξωτερικές πηγές τα οποία επίσης δεν είναι αξιόπιστα πολλές φορές, λόγω εξασθένησης σήματος ή γενικότερα απώλεια των δεδομένων αυτών. Ο περιορισμός αυτός καθιστά την τεχνολογία προβληματική σε άλλες εφαρμογές όπως η πλοήγηση σε εσωτερικούς χώρους. Παρόλα αυτά οι απώλειες δεδομένων δεν είναι σε όλες τις περιπτώσεις κρίσιμες και προσφέρουν βάθος ποιότητας [26].



Εικόνα 2.9 Εφαρμογή πλοήγησης AR με την χρήση gps.

2.1.2 Πλοήγηση σε εσωτερικούς χώρους

Η πλοήγηση σε εξωτερικούς χώρους έχει μελετηθεί, έχει εφαρμοστεί και υλοποιηθεί πολλές φορές με διαφορετικούς τρόπους. Σε αντίθεση, για την πλοήγηση σε εσωτερικούς χώρους δεν υπάρχουν πολλές λύσεις που να λειτουργούν εξίσου αποτελεσματικά. Όμως η ανάγκη για έρευνα στον τομέα εσωτερικής πλοήγησης AR για την χρήση του σε χώρους όπως μεγάλα κτήρια, μουσεία, μεγάλα πολυκαταστήματα αυξάνεται όλο και περισσότερο. Η πολυπλοκότητα των κτηρίων αυτών δημιουργεί εμπόδια και προκλήσεις ως προς τον υπολογισμό της διαδρομής η οποία επίσης μπορεί να καταλήγει σε μη επιθυμητή θέση στον πραγματικό χώρο, το οποίο μπορεί να θεωρηθεί σοβαρό ελάττωμα για την εφαρμογή. Ένα παράδειγμα είναι ένα σύστημα εσωτερικής πλοήγησης σε μια εταιρία. Ο χρήστης θέλει να ελέγξει αν κάποιο άλλο άτομο που δουλεύει στην εταιρία βρίσκεται σε μία συγκεκριμένη θέση στον όροφο αυτό. Το σύστημα εσωτερικής πλοήγησης πρέπει να λειτουργεί με απόλυτη ακρίβεια με σκοπό η τελική κατάσταση της διαδρομής, ή αλλιώς ο τερματισμός της, να οδηγεί ακριβώς στο σημείο αυτό και όχι σε κάποιο παραπλήσιο (εικόνα 3.0) [27].



Εικόνα 3.0 Εφαρμογή για πλοήγηση σε εσωτερικούς χώρους με την χρήση beacons.

Ακόμα πρέπει να σημειώσουμε ότι η πλοήγηση σε εσωτερικούς χώρους έχει διαφορές από εξωτερικούς και ως προς τα μέσα που έχουν πρόσβαση στις εφαρμογές της οι χρήστες. Σε πλοία, οχήματα και άλλα, ο χώρος για αισθητήρες, κάμερες, μεγαλύτερα και πιο ακριβή όργανα είναι διαθέσιμος δίχως ανάγκη για μείωση του χώρου αυτού. Αντιθέτως η πλοήγηση AR σε κλειστούς χώρους είναι περιορισμένη σε αυτόν τον τομέα. Η σημασία για εύκολη και οικονομική χρήση μέσων για εφαρμογές AR είναι ύψιστη επομένως τα περισσότερα μέσα που αναφέρθηκαν στο κεφάλαιο ένα αποκλείονται. Το μέσο αυτό που λύνει τα περισσότερα

προβλήματα είναι το ίδιο που χρησιμοποιείται καθημερινά και σε εκτενή χρήση από τον άνθρωπο, τα έξυπνα κινητά, smartphones.

Επιπλέον ως διαφορά πρέπει να αναφερθεί αυτή της διαφοράς χρήσης των αισθητήρων. Αισθητήρες και όργανα όπως GPS έχει αποδειχθεί ότι δημιουργούν αρκετές αποκλίσεις στα δεδομένα που παρέχουν για τον υπολογισμό μίας διαδρομής σε περιβάλλοντα που περικλείονται από τοίχους και αντικείμενα που αποδυναμώνουν ηλεκτρομαγνητικά κύματα. Επίσης αισθητήρες όπως τα επιταχυνσιόμετρα και η χρήση βηματομετρητών δεν είναι αρκετά ώστε να προσφέρουν όλα τα απαραίτητα δεδομένα για την κίνηση του χρήστη, και κυρίως την κατεύθυνση ούτε ακριβής, αφού για την σωστή μέτρηση δεδομένων από τον χρήστη στους αισθητήρες πρέπει να τηρούνται συγκεκριμένοι κανόνες. Για παράδειγμα η μέτρηση βημάτων από τον βηματομετρητή πρέπει να επιτευχθεί με συγκεκριμένη ταχύτητα από τον χρήστη. Αν περπατάει πολύ αργά οι αισθητήρες δεν λαμβάνουν δεδομένα και η έξοδος στο τελικό προϊόν αποκλίνει όσο περισσότερο η εφαρμογή λειτουργεί [28].

Λόγω των παραπάνω αυτή η πτυχιακή εργασία έχουν χρησιμοποιεί μικρός αριθμός αισθητήρων απο αυτούς που έχουν προαναφερθεί. Η προσέγγιση που έχει επιλεγεί βασίζεται κατά το μεγαλύτερο μέρος σε αλγόριθμους που χρησιμοποιεί η Google μέσω του εργαλείου ARCore, το οποίο χρησιμοποιεί αρχές της ρομποτικής ,όπως οι αλγόριθμοι S.L.A.M. η αλλιώς Simultaneous Localization and Mapping. Επίσης ο κύριος αισθητήρας που θα χρησιμοποιεί είναι η κάμερα που διαθέτουν ήδη τα κινητά ενώ οι αλγόριθμοι κάνουν χρήση και του επιταχυνσιόμετρου και βαθύμετρου. Αρχικά θα γίνει μία εισαγωγή για την διαδικασία S.L.A.M. για την καλύτερη κατανόηση των λειτουργιών στο χαμηλό επίπεδο της εφαρμογής και έπειτα θα μελετήσουμε την χρήση των εργαλείων και την υλοποίηση της [28].

Επίσης αναφερόμαστε ως πράκτορα το πιθανόν χρήστη, ρομπότ, κινητό τηλέφωνο που μπορεί να χρησιμοποιηθεί ως πλοηγός.

2.2 Αλγόριθμοι Τοπικοποίησης Και Χαρτογράφησης.

Το ακρόνυμο S.L.A.M η αλλιώς Simultaneous Localization and Mapping ορίζεται κυρίως ως μια διαδικασία, ένα υπολογιστικό πρόβλημα που έχει ως στόχο την κατασκευή και η ενημέρωση (ως προς τις αλλαγές από νέα δεδομένα) ενός χάρτη σε άγνωστο περιβάλλον ενώ παράλληλα διατηρεί γνωστή την τοποθεσία του πράκτορα (συσκευής) σε σχέση με αυτό [28]. Πρωτοεμφανίστηκε το 1987 από έρευνα του Randall Smith και του Peter Cheeseman με την δημοσίευση τους με όνομα “On The Representation and Estimation of Spatial Uncertainty” όπου περιγράφουν μια γενική μέθοδο υπολογισμού κανονικοποιημένων σχέσεων και την συνδιακύμανση μεταξύ σημείων ανά καρέ (frame), τα οποία αναπαριστούν την σχετική θέση των αντικειμένων σε έναν χώρο. Το σύνολο προβλημάτων που καλούνται να αντιμετωπιστούν πραγματοποιούνται από πολλές παραλλαγές της διαδικασίας SLAM και το σύνολο που αποτελεί τις λύσεις στα προβλήματα SLAM έχει αποδειχθεί ότι είναι άπειρο μία ομάδα ερευνητών του Hugh F. Durrant-Whyte στις αρχές του 1990 [29].

Υπάρχουν δύο κύριες προσεγγίσεις στο πρόβλημα:

- Η πρώτη κατηγορία βασίζεται σε αλγόριθμους που χρησιμοποιούν κυρίως στατιστικά μοντέλα ως φίλτρα όπως το φίλτρο Kalman στο οποίο ανήκουν ένα υποσύνολο εναλλακτικών φίλτρων τύπου Kalman ή την μέθοδο Monte Carlo οι οποίοι πραγματοποιούν προβλέψεις και ενημερώνονται αναδρομικά σε κάθε βήμα καθώς διατηρούν τις πληροφορίες που περιέχουν τις καταστάσεις της θέσης και κατεύθυνσης του πράκτορα ως μια συνάρτηση πυκνότητας πιθανότητας [30].
- Η δεύτερη κατηγορία προσεγγίζεται με καθολική βελτιστοποίηση (Global Optimization) και την τεχνική αποθήκευσης συγκεκριμένων σημείων-κλειδιών που υπάρχουν σε ένα καρέ, από μία σειρά καρέ(frames) μιας συνεχής εισόδου(data stream) από δεδομένα, και τον υπολογισμό διαφορών τους για τον προσανατολισμό του πράκτορα. Τα εργαλεία που έχουν χρησιμοποιηθεί στην παρούσα εργασία συμπίπτουν με αυτή την κατηγορία αλγορίθμων [30].

Οι παραπάνω κατηγορίες ομαδοποιούνται περισσότερο ως προς τον τρόπο που υλοποιούνται οι αλγόριθμοι και τον τρόπο που ομαδοποιούν και συσχετίζουν τα δεδομένα για την δημιουργία μιας τοπολογίας.

2.2.1 Διαδικασία SLAM και Κατανόηση σχέσεων σε σημεία του χώρου

Αρχικά για να κατανοήσουμε την λογική που λειτουργούν οι αλγόριθμοι SLAM θα αναπτύξουμε την έννοια των σχέσεων μεταξύ των δεδομένων που αναφέρθηκε στις δύο παραπάνω κατηγορίες. Στην δημοσίευση “A stochastic map for uncertain spatial relationships” των R. Smith Matthew Self και Peter Cheeserman περιγράφονται αυτές οι σχέσεις [31].

Στον πραγματικό κόσμο, περιβάλλον με τα σωστά εργαλεία και όργανα μπορούμε να έχουμε ακριβές μετρήσεις. Η γνώση για την σχετική θέση αντικειμένων είναι δυνατόν να πραγματοποιηθεί με μεγάλη ακρίβεια έτσι ώστε οι αποκλίσεις να μην επηρεάζουν το αποτέλεσμα του συστήματος. Αντιθέτως σε ένα περιβάλλον επαυξημένης πραγματικότητας κάτι τέτοιο δεν είναι δυνατό.

Για καλύτερη επεξήγηση θα χρησιμοποιήσουμε το παράδειγμα από την παραπάνω δημοσίευση εφαρμοσμένη στο παρόν πρόβλημα της πτυχιακής εργασίας.

Αρχικά ας υποθέσουμε ότι ελέγχουμε μία συσκευή όπως ένα κινητό τηλέφωνο σε έναν χώρο. Σε ένα σημείο A υπάρχει ένας φάρος (beacon) ο οποίος μας παρέχει ακριβές πληροφορίες για την θέση του, το διάνυσμα A το οποίο χρησιμοποιούμε και ως ένα μοναδικό αξιόπιστο σημείο-άγκυρα (anchor). Εμείς είμαστε τοποθετημένοι στο σημείο B. Με την προϋπόθεση ότι γνωρίζουμε ακριβής τιμές για την απόσταση από το A στο B, δηλαδή την μεταξύ μας απόσταση με το αξιόπιστο σημείο A, το οποίο γίνεται στον πραγματικό κόσμο, και θέλουμε να μετακινηθούμε σε ένα σημείο Γ, μας επαρκούν όλα τα στοιχεία για να μετακινηθούμε με απόλυτη ακρίβεια.

Αντιθέτως σε ένα σενάριο που λειτουργεί σε περιβάλλον Επαυξημένης πραγματικότητας και χρησιμοποιεί συστήματα SLAM ο υπολογισμός των διανυσμάτων αυτών πρέπει να πραγματοποιηθεί με ανεπαρκής γνώση, το οποίο οδηγεί σε αβεβαιότητες. Οι σχέσεις μεταξύ των σημείων είναι χωρικές. Αυτό έχει ως αποτέλεσμα να μπορούμε να έχουμε μια κατανομή πιθανότητας για κάθε πιθανή θέση. Για ορισμένα σημεία στον χώρο θα έχουμε μεγάλη ακρίβεια ως προς τις τιμές τους ενώ σε άλλα η αβεβαιότητα θα είναι μεγάλη. Για τα σημεία

που έχουν αρκετά σημαντική αβεβαιότητα και είναι πιο πιθανό να δημιουργήσουν αποκλίσεις και λάθη στο σύστημα χρησιμοποιούνται φίλτρα Kalman ή Maximum a Posterior (MAP) εκτιμήσεις. Οι σχέσεις αυτές μεταξύ των σημείων ανανεώνονται περιοδικά ανάλογα με τον αισθητήρα που χρησιμοποιείται για την ανανέωση δεδομένων και επηρεάζουν σε κάθε ανανέωση όλες τις σχέσεις μεταξύ παλιών και καινούριων σημείων ενώ παράλληλα ανανεώνουν και τον χάρτη που δημιουργείται.

Στην δημοσίευση “Globally consistent range scan alignment for environment mapping” από τους Lu και Milios περιγράφεται η σημασία των σχέσεων μεταξύ των σημείων σε έναν χάρτη [32].

Συγκεκριμένα παραθέτει το πρόβλημα των δεδομένων σε σχέση με τις ανακρίβειες από την είσοδο των αισθητήρων. Υποθέτει ότι λαμβάνουμε τα δεδομένα για τον χώρο με την χρήση οδομετρίας (Odometry), δηλαδή την χρήση αισθητήρων κίνησης όπως η κάμερα για τον υπολογισμό της θέσης ενός πράκτορα. Με βάση την οδομετρία με αισθητήρες όπως η κάμερα και μόνο είναι αρκετά δύσκολο να υπολογιστεί η σχετική θέση ενός αντικειμένου. Φυσικά, υπάρχει μια διαδικασία που ακολουθείται για τους υπολογισμούς αυτούς, η σύγκριση κάθε καρτέ από τα δεδομένα που λαμβάνονται. Ευθυγραμμίζοντας δύο τοπικά καρτέ μπορούν να συγκριθούν οι μεταβολές στα αναγνωρισμένα σημεία-κλειδιά από τον αλγόριθμο και να υπολογιστεί μία τοπική αλλαγή στην κατεύθυνση και στην θέση γενικότερα. Βέβαια ακόμη και με αλγόριθμους που χρησιμοποιούν φίλτρα Kalman η παραπάνω διαδικασία δημιουργεί ασυνέπεια σε διαφορετικά σημεία του χάρτη καθώς αυτός ανανεώνεται. Επομένως για την επίλυση των παραπάνω σφαλμάτων πρέπει να διατηρηθούν τοπικά καρτέ των δεδομένων με τις υπολογισμένες θέσεις τους και να συγκρίνονται με γειτονικούς κόμβους για διόρθωση στα μονοπάτια που δημιουργούνται στην τοπολογία.

Όπως έχει γίνει κατανοητό η μεγαλύτερη πρόκληση που καλούνται συνεχώς να λύσουν οι αλγόριθμοι SLAM είναι όχι μόνο η δημιουργία ενός χώρου που να έχει σωστή δομή αλλά και οι ελάχιστες αποκλίσεις για το δίκτυο που θα κινηθεί ένας πράκτορας μέσα σε αυτόν. Συνεπώς υπάρχουν τέσσερις κύριες προκλήσεις: ο άγνωστος χώρος, η μη ελεγχόμενη κάμερα, πραγματικός χρόνος, χωρίς σφάλματα και αποκλίσεις.

2.2.2 Ανατομία ενός συστήματος SLAM

Αισθητήρες δεδομένων: σε συσκευές όπως τα κινητά θα χρησιμοποιηθούν η κάμερα, το επιταχυνσιόμετρο και το γυροσκόπιο. Επίσης είναι δυνατή και η λειτουργία άλλων αισθητήρων όπως προαναφέρθηκαν στο πρώτο κεφάλαιο ως προέκταση τους [33].

Front-End: Σε υψηλό επίπεδο του συστήματος πραγματοποιείται το πρώτο βήμα που είναι η διαδικασία αναγνώρισης συγκεκριμένων σημείων από μία σειρά από εικόνες και η διαδικασία συσχέτισμού τους ως τοπικές εικόνες και συνολικά για την μείωση αποκλίσεων.

Back-End: Σε χαμηλό επίπεδο πραγματοποιείται το επόμενο βήμα που είναι ο συσχέτισμός των σημείων που διακρίνει ο αλγόριθμος καθώς και η διόρθωση των σφαλμάτων μεταξύ τις συσχετίσεις αυτές σε κάθε εικόνα και η σχέση μεταξύ των διαφορετικών καρτέ. Έπειτα το σύστημα αναλαμβάνει να δημιουργήσει την τοπολογία, δηλαδή την γεωμετρία του χώρου και τοποθετείται η κάμερα στο εικονικό περιβάλλον (pose model). Ορισμένοι αλγόριθμοι κατασκευάζουν μία αραιή γεωμετρία του χώρου και από τι αυτός αποτελείται ενώ άλλοι αντιγράφουν με ιδιαίτερη λεπτομέρεια τα διάφορα χαρακτηριστικά του. (εικόνα 3.1)

Αποτέλεσμα SLAM: Το αποτέλεσμα περιλαμβάνει όλα τα χαρακτηριστικά, τις θέσεις και τις σχέσεις καθώς και την τοποθεσία της κάμερας στον κόσμο.

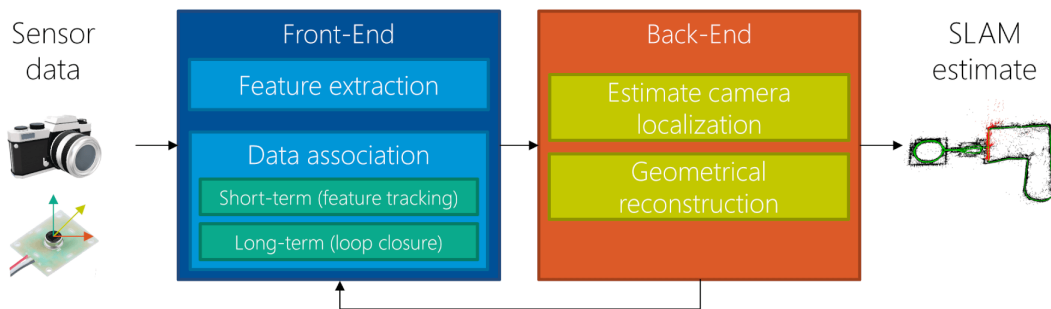


Diagram based on: Cadena, Cesar, et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age." *IEEE Transactions on Robotics* 32.6 (2016): 1309-1332.
 SLAM estimate: R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, Oct. 2015.
 3D Model of motion sensor by gerdulob: <https://www.remix3d.com/details/G009SVNP5RSM>
 3D Model of camera by Microsoft: <https://www.remix3d.com/details/G0095XC931H9>

Εικόνα 3.1 Ανατομία συστημάτων SLAM. Αριστερά απεικονίζεται το μέρος που είναι υπεύθυνο για την μέτρηση δεδομένων από τους αισθητήρες και την ένωσή τους. Δεξιά απεικονίζεται το μέρος που είναι υπεύθυνο για τον υπολογισμό θέσης και γεωμετρίας του χώρου με βάση τα δεδομένα αυτά.

Οι διαδικασίες που περιγράψαμε παραπάνω από τις δύο αυτές δημοσιεύσεις περιγράφουν κυρίως πιο αργούς αλγόριθμους όπως είναι οι SIFT και SURF οι οποίοι χρησιμοποιούνται σαν σημείο αναφοράς αλλά είναι αργοί και δύσχρηστοι για εφαρμογές πραγματικού χρόνου.

Ένας αλγόριθμος που περιγράφει αρκετά παραπλήσια ότι έχει αναφερθεί ως τώρα και τον αλγόριθμο που έχουμε χρησιμοποιήσει στην εργασία είναι ο ORB-SLAM ο οποίος χρησιμοποιεί τον αλγόριθμο ORB ο οποίος είναι παραπλήσιος με τον BRISK ως πιο γρήγορος για χρήση σε πραγματικό χρόνο [34].

2.3 SLAM και ARCore

Το εργαλείο που έχει χρησιμοποιηθεί στην πτυχιακή είναι το ARCore που αναπτύχθηκε από την Google και χρησιμοποιεί παρόμοιους αλγόριθμους με τον παραπάνω με την προσθήκη οδομετρίας και την βοήθεια από αισθητήρες όπως ο αισθητήρας βάθους και το επιταχυνσιόμετρο.

Το ARCore είναι βασισμένο σε τρεις αρχές, το Motion Tracking δηλαδή την ανίχνευση κίνησης, Environmental Understanding δηλαδή την κατανόηση του περιβάλλοντα χώρου και Light Estimation, τον υπολογισμό έντασης φωτός στον χώρο όπως και ανατακλάσεων. Το ARCore επιτρέπει σε οποιαδήποτε σε οποιαδήποτε συσκευή που το υποστηρίζει να ανιχνεύει

τον χώρο σε 6 (έξι) βαθμούς ελευθερίας, κίνησης και προσανατολισμού, σχετική με το περιβάλλον χρησιμοποιώντας μια τεχνική που ονομάζει Concurrent Odometry and Mapping (Ταυτόχρονη Οδομετρία και Χαρτογράφηση). Η τεχνική αυτή έχει την δυνατότητα να ανιχνεύει το μέγεθος και την θέση από επιφάνειες, οριζόντιες, κάθετες άλλα και με κλίση. Ο τρόπος που λειτουργεί η αναγνώριση του περιβάλλοντος γύρω από την συσκευή είναι παρόμοιος με αυτόν που αναφέρθηκε στα συστήματα SLAM. Όταν μια συσκευή που χρησιμοποιεί την τεχνολογία ARCore κινείται στον χώρο, σαρώνει την γεωμετρία του χώρου, δημιουργεί τον πρώτο συσχετισμό για την θέση του μέσα σε αυτόν και συνεχίζει να δημιουργεί συσχετισμούς. Κατά την διάρκεια σάρωσης το ARCore δημιουργεί όπως τα αποκαλεί “feature points” δηλαδή σημεία αναφοράς από τα οποία δημιουργείται ένα αραιό σύμπλεγμα σημείων. Το σύμπλεγμα αυτό εφόσον επεξεργαστεί από τον αισθητήρα κάμερας του κινητού σε συνδυασμό με δεδομένα από το επιταχυνσιόμετρο και το γυροσκόπιο συμβάλει στον υπολογισμό σχετικής θέσης στον χώρο. Το σύμπλεγμα σημείων μπορεί να διαφοροποιηθεί ως προς την κλίση του και το αν είναι οριζόντιο ή κάθετο επίπεδο, καθώς και μεταφράζεται σε τρισδιάστατη επιφάνεια για την χρήση του σε ανάλογο λογισμικό. Η ροή δεδομένων εισόδου και ο συνεχής υπολογισμός θέσης σε κάθε εικόνα που λαμβάνεται πραγματοποιείται επίσης όπως έχει προαναφερθεί. Αξίζει ακόμα να σημειωθεί ότι το ARCore επιτρέπει ανάγνωση δεδομένων από τους αισθητήρες, κάμερας, έως και 60 fps και γυροσκοπίου και επιταχυνσιόμετρου έως και 1000 fps προσφέροντας ελαχιστοποίηση σε απώλεια δεδομένων και αποκλίσεις.

ΚΕΦΑΛΑΙΟ 3 Unity

3.1 Μηχανή Ανάπτυξης Παιχνιδιών Unity

Το Unity είναι μία μηχανή ανάπτυξης παιχνιδιών η οποία υποστηρίζεται από πολλαπλές πλατφόρμες και μπορεί να χρησιμοποιηθεί για την δημιουργία τρισδιάστατων ή και δισδιάστατων εφαρμογών, παιχνιδιών, ή και προσομοιώσεων. Η μηχανή αυτή έχει υιοθετηθεί από εταιρίες πέρα από την βιομηχανία παιχνιδιών, όπως οι βιομηχανίες κινηματογράφου, αυτοκινήτων από κατασκευαστικές εταιρίες, εταιρίες μηχανικής και αρχιτεκτονικής.

Το λογισμικό εμφανίστηκε πρώτη φορά το 2005 αρχικά ως αποκλειστική μηχανή ανάπτυξης παιχνιδιών για το λειτουργικό σύστημα της εταιρίας Apple Inc. το Mac Os X ενώ αργότερα επεκτάθηκε σε πληθώρα από πλατφόρμες όπως σε λειτουργικά συστήματα (Linux, Windows) κονσόλες (Wii, PlayStation, Xbox) smartphones, σε IOS και Android όπως και σε φυλλομετρητές με την βοήθεια εξωτερικών APIs όπως το WebGL. Ο κύριος στόχος με βάση τους δημιουργούς της μηχανής είναι η πρόσβαση του προϊόντος σε περισσότερους προγραμματιστές ως ένα εργαλείο πιο εύκολο στην χρήση του σε σχέση με τα υπόλοιπα της ίδιας κατηγορίας. Το Unity μέσα σε δύο δεκαετίες έχει αποκτήσει αρκετή φήμη στον χώρο των μηχανών ανάπτυξης παιχνιδιών ενώ έχει καταλήξει να είναι μία αρκετά συνήθης επιλογή των προγραμματιστών. Ένα μέρος της ευθύνης για το πόσο δημοφιλής είναι το λογισμικό αυτό έχουν ο μεγάλος αριθμός εκπαιδευτικού υλικού σε συνδυασμό με την πολυπληθή κοινότητα να το υποστηρίζει, το μοντέλο άδειας χρήσης της εταιρίας (δωρεάν χρήση για προγραμματιστές και εταιρίες με έσοδα λιγότερα από 100.000\$), όπως και το Unity Asset Store, ένα είδους διαδικτυακού καταστήματος που προσφέρει δωρεάν και επί πληρωμή σχέδια, τμήματα από κώδικα, και τρισδιάστατα μοντέλα για χρήση. Όλα τα παραπάνω δημιουργούν ένα περιβάλλον που προσεγγίζει μεγαλύτερο εύρος ηλικιών και υπόβαθρου γνώσης.

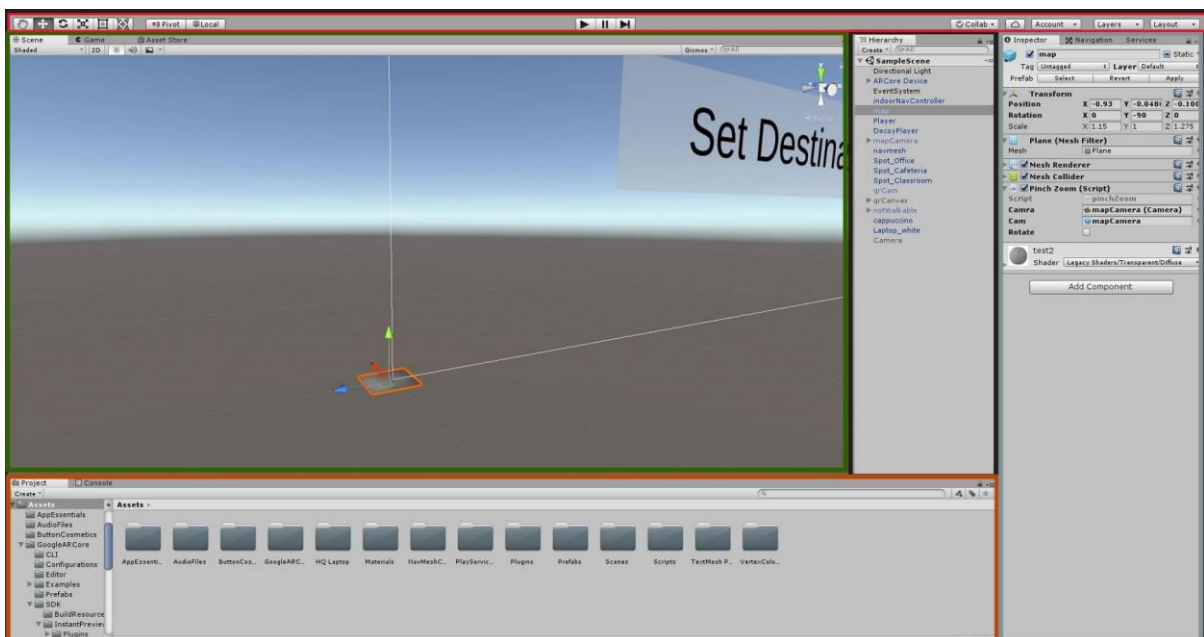
Το Unity έχει δομηθεί με βάση την γλώσσα προγραμματισμού C++ για τον χρόνο εκτέλεσης ενώ στον προγραμματιστή προσφέρει τις γλώσσες C# και Javascript όπου αποκαλεί και UnityScript για την συγγραφή προγραμμάτων και συμπεριφορών που θα

χρησιμοποιήσει η εφαρμογή. Τα τελευταία χρόνια έχει καταργηθεί πλήρως η υποστήριξη της γλώσσας JavaScript από την μηχανή. Η μηχανή επίσης παρέχει μία διεπαφή προγραμματισμού εφαρμογών API που προσφέρει βιβλιοθήκες μεγάλο πλήθος κλάσεων και αντίστοιχα συμπεριφορών διαδικασιών και συναρτήσεων για την εξοικονόμηση χρόνου του προγραμματιστή και την διατήρηση πιο γρήγορου και καθαρού κώδικα. Μία πολύ σημαντική κλάση που περιέχει η βιβλιοθήκη UnityEngine είναι η MonoBehaviour που αποτελεί απαραίτητο συστατικό στοιχείο για την χρήση συναρτήσεων και συμπεριφορών.

Προκειμένου να μπορούμε να περιγράψουμε όλα τα δομικά στοιχεία της εργασίας και τον τρόπο που λειτουργούν πρέπει να γίνουν μερικές αναφορές γενικότερα στα μέρη τα οποία αποτελείται το Unity και την διεπαφή χρήστη GUI που προσφέρει το Unity για την ανάπτυξη ενός προγράμματος ή εφαρμογής. Η ανάλυση θα περιγράψει τα κοινά στοιχεία που χρησιμοποιούνται από όλα τα είδη ανάπτυξης προγραμμάτων και στα απαραίτητα που χρειάζονται κυρίως για ανάπτυξη εφαρμογών που απευθύνονται στην πλατφόρμα Android.

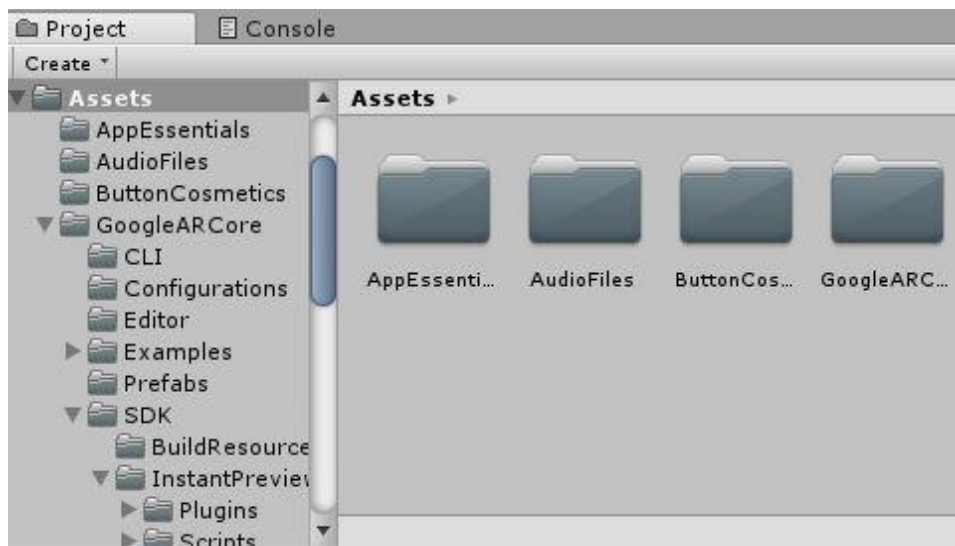
3.2 Ο Editor και τα πεδία του

Παρακάτω παραθέτουμε τον Editor (εικόνα 3.2) ο οποίος είναι η διεπαφή του προγραμματιστή για την ανάπτυξη εφαρμογών και αποτελείται από 5 πρωταρχικά παράθυρα.



Εικόνα 3.2 Editor στο λογισμικό Unity3D.

Ο προγραμματιστής έχει την δυνατότητα να διαμορφώσει τα πεδία εσωτερικά του Unity με βάση τις προτιμήσεις του. Η συγκεκριμένη διάταξη που παρουσιάζεται στις εικόνες είναι μια από αυτές. Επομένως στο κάτω αριστερά (εικόνα 3.2) μέρος της εικόνας υπάρχει

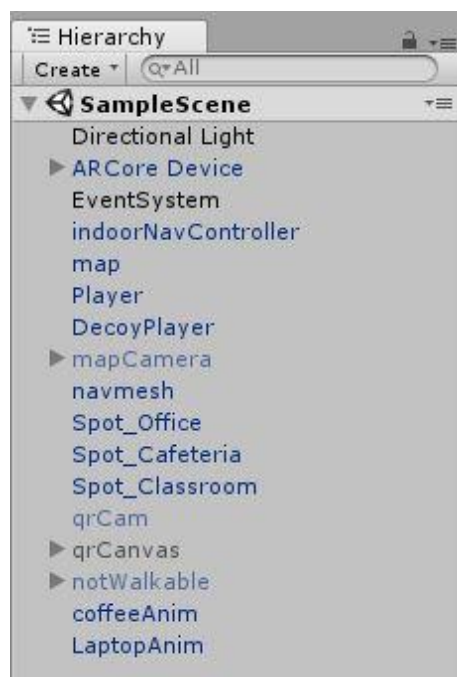


Εικόνα 3.3 Το πεδίο Project και τα στοιχεία του.

το πεδίο “Project” (εικόνα 3.3) το οποίο διαθέτει στον προγραμματιστή μία λίστα από αρχεία. Τα αρχεία αυτά με την δημιουργία ενός καινούργιου project είναι είναι αρχικά τα: Assets και Packages, που αποτελούν απαραίτητα αρχεία για την λειτουργία ενός project στην μηχανή Unity. Τα Assets είναι η ραχοκοκκαλιά ενός πρότζεκτ αφού εκεί πρόκειται ο προγραμματιστής να τοποθετήσει όλα τα απαραίτητα αρχεία που χρειάζεται. Τα πακέτα (Packages) όμως είναι αυτά που θα περιέχουν τα αρχεία που ευθύνονται για τα έτοιμα πακέτα εργαλείων που προσφέρει το Unity όπως το (Analytics Library) καθώς και επιπρόσθετων εξωτερικών.

Το πεδίο Project λειτουργεί ως ένα εργαλείο μεν για τον προγραμματιστή ως προς την κατηγοριοποίηση των αρχείων και επομένως μπορεί να προσφέρει ένα πιο τακτοποιημένο περιβάλλον στην διαδικασία δημιουργίας ενός προγράμματος εφαρμογής. Ωστόσο πρέπει να σημειωθεί ότι το πεδίο Project εξακολουθεί να είναι μία γραφική αναπαράσταση του File Explorer για τον προγραμματιστή, για παράδειγμα στα Windows , πράγμα που σημαίνει, η διαγραφή ή η αλλαγή ενός αντικειμένου ή αρχείου στο πεδίο Project έχει μη ανατρέψιμες αλλαγές στο συνολικό πρότζεκτ.

Hierarchy



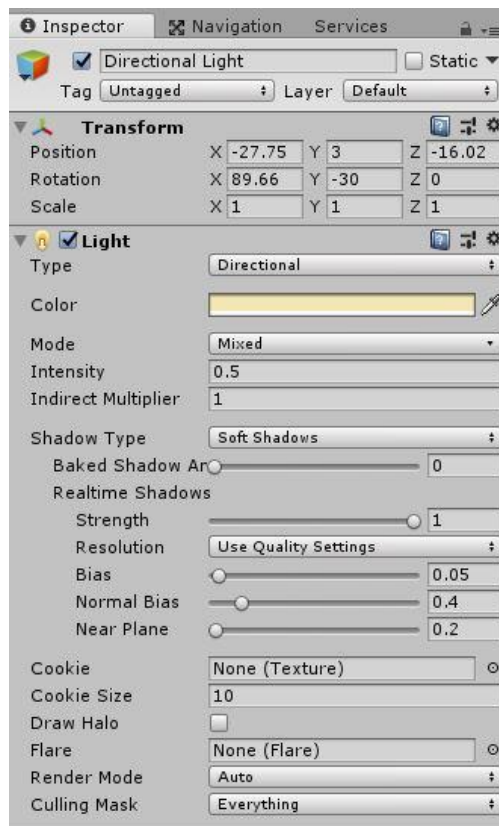
Εικόνα 3.4 Το πεδίο Hierarchy μαζί με τα ενεργά αντικείμενα στη σκηνή με όνομα "Sample Scene".

Το πεδίο “Hierarchy” (εικόνα 3.4) ή “Ιεραρχία” είναι εν συντομία η ένωση μεταξύ των πεδίων Project και Scene. Όταν ένας προγραμματιστής, ή το άτομο που διαχειρίζεται την εφαρμογή, θέλει να προσθέσει αντικείμενα σε μία ενεργή σκηνή στην εφαρμογή θα τα τοποθετήσει στο πεδίο της ιεραρχίας. Ενώ το πεδίο Project περιέχει όλα τα αντικείμενα, scripts, prefabs (συνθέσεις αντικειμένων που έχουν δημιουργηθεί κατά την διάρκεια του ενεργού project), στο πεδίο “Hierarchy” θα τοποθετηθούν μόνο τα απαραίτητα αντικείμενα τα οποία το Unity θα μεταγλωττίσει και θα ενώσει με σκοπό να παραχθεί το εκτελέσιμο αρχείο. Το πεδίο είναι υπεύθυνο και για την ανάθεση script στα αντικείμενα πράγμα το οποίο δεν είναι απαραίτητο στο πεδίο “Project” όπως και την ιεραρχία αντικειμένων. Πιο συγκεκριμένα το “Project” λειτουργεί κυρίως ως αναφορά των αντικειμένων και αρχείων για το “Hierarchy” με την έννοια ότι, ότι αλλαγές πραγματοποιούνται στο πεδίο “Project” πραγματοποιούνται και καθολικά στο project, ενώ οι αλλαγές που πραγματοποιούνται στα αντικείμενα που περιέχονται στην ιεραρχία έχουν αντίκτυπο μόνο στην ενεργή σκηνή και προαιρετικά στο πεδίο Scene (ως προς τι φαίνεται). Όσο αφορά τα αντικείμενα που εμπεριέχονται στην στο πεδίο Hierarchy αυτό που αποτελεί σημαντικό μέρος στην λειτουργία της εφαρμογής είναι η σωστή τοποθέτηση των αντικειμένων ώστε αν υπάρχει

αντιστοιχία “γονέα-παιδιού” (parent objects) να δηλώνετε εκεί. Τέλος, υπάρχει η δυνατότητα μετονομασίας των ενεργών αντικειμένων “μέσα” στο πεδίο χωρίς να υπάρχει κάποια αλλαγή στα αρχεία που βρίσκονται στο πεδίο “Project” δηλαδή στα αυθεντικά αρχεία του ενεργού project.

Ουσιαστικά η ιεραρχία των πεδίων είναι ως εξής: Project : Hierarchy : Scene. Αυτό σημαίνει ότι στο πεδίο Hierarchy έχουμε την κύρια σκηνή Scene όπως ονομάζεται που καταλαμβάνει το μεγαλύτερο μέρος της οθόνης (*όλα τα παράθυρα μπορούν να προσαρμοστούν να μεγενθύνουν όπως και να αφαιρεθούν ανάλογα με τις ανάγκες του προγραμματιστή). Η σκηνή είναι ένα κεντρικό μέσο προβολής των τρισδιάστατων μοντέλων και του GUI της εφαρμογής δηλαδή το περιβάλλον που δημιουργεί ο προγραμματιστής για την εφαρμογή του και τα μενού που εμπεριέχονται σε αυτό. Οι σκηνές είναι αρχεία με κατάληξη .unity και εμπεριέχουν όλα τα ενεργά αντικείμενα και script, κομμάτια κώδικα που έχει ορίσει ο προγραμματιστής, καθώς και τις συνδέσεις μεταξύ τους. Πιο συγκεκριμένα ότι είναι ενεργό στο πεδίο της ιεραρχίας (Hierarchy) και με τυχόν αλλαγές που έχουν γίνει μέσα σε αυτό το πεδίο αποτελεί μέρος της σκηνής. Σε ένα project στο Unity μπορούν να συνυπάρχουν όσες σκηνές ή επίπεδα επιλέξει ο προγραμματιστής, ανάλογα με τις απαιτήσεις του. Τέλος στην ιεραρχία, όταν κάποια αντικείμενα είναι τοποθετημένα εσωτερικά άλλων κληρονομούν κυρίως την ιδιότητα (component) Transform, δηλαδή εξάρτηση στον χώρο του Unity και την κίνηση μέσα σε αυτόν.

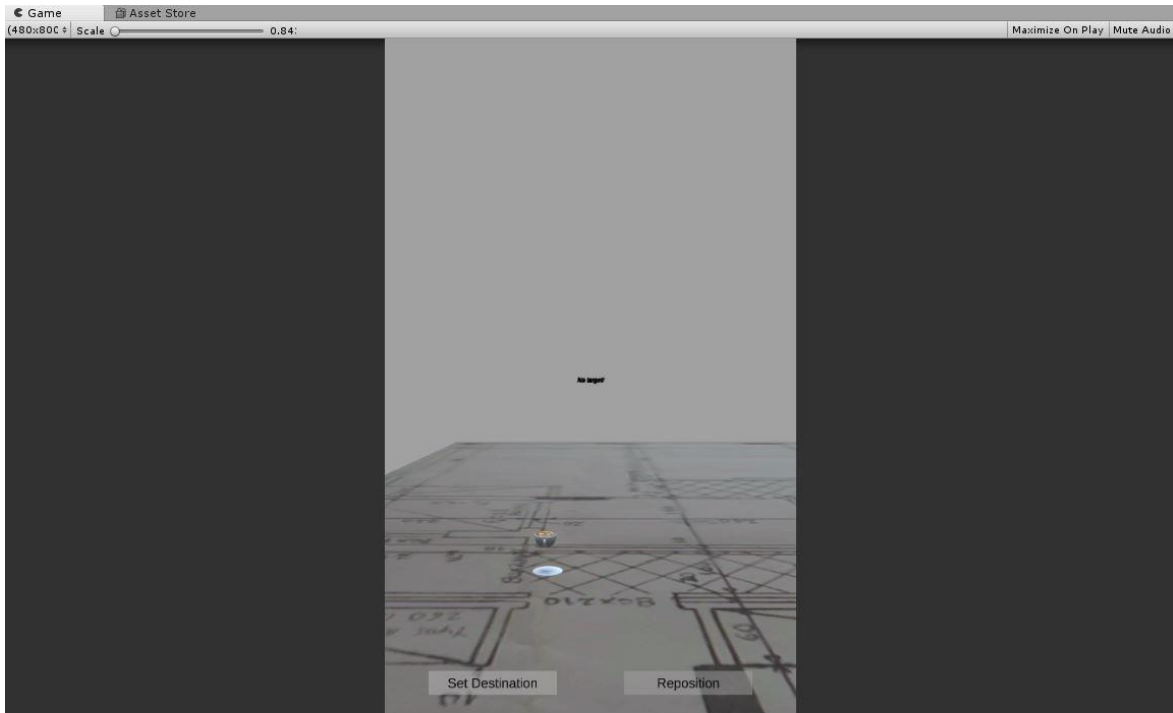
Inspector



Εικόνα 3.5 Το πεδίο Inspector. Στο συγκεκριμένο παράδειγμα απεικονίζει τα στοιχεία του αντικειμένου "Directional Light".

Το πεδίο “Inspector” (εικόνα 3.5) λειτουργεί περισσότερο ως μια προβολή για τις συμπεριφορές των αντικειμένων. Στο πεδίο “Scene” ενώ οι λειτουργίες, τα script και τα components και οι συμπεριφορές υφίστανται, δεν είναι ορατά στον προγραμματιστή επομένως δεν μπορεί να πραγματοποιηθεί κάποια αλλαγή σε τιμές τους. Σε αυτό βοηθάει το πεδίο “Inspector” το οποίο προσφέρει το οπτικό μέσο στον προγραμματιστή για τις συμπεριφορές, τις ιδιότητες και τα χαρακτηριστικά που μπορεί να περιέχει ένα αντικείμενο της ιεραρχίας αλλά και του Project . Είτε αναφερόμαστε σε αρχεία που υπάρχουν εσωτερικά των αρχείων είτε της ενεργής σκηνής ο προγραμματιστής μπορεί έμμεσα μέσω του Inspector να αλλάξει πολλές διαφορετικές τιμές και components ενώ άμεσα είναι ο μόνος τρόπος να αφαιρέσει ή να προσθέσει συμπεριφορές, ιδιότητες ή και scripts στο επιλεγμένο αντικείμενο.

Game



Εικόνα 3.6 Το πεδίο Game. Προσωμοιώνει την τρέχον σκηνή με τα αντικείμενά της ως τελικό εκτελέσιμο.

Το στοιχείο αυτό αν και ανήκει στα πεδία του Editor χρειάζεται απλής αναφοράς. Είναι το πεδίο στο οποίο προβάλλεται το τελικό προϊόν του ενεργού project. Τα αντικείμενα που μεταφράζονται στο πεδίο Game (εικόνα 3.6) είναι και αυτά που είναι τοποθετημένα εντός του εύρους της κάμερας της σκηνής που κάνει render. Επίσης προσφέρει επιλογές προεπισκόπησης της εφαρμογής όπως ανάλυση οθόνης.

Console

Η κονσόλα είναι ένα ακόμα μέρος του Editor και είναι υπεύθυνη για την επίδειξη σφαλμάτων και προειδοποιήσεων και άλλων μηνυμάτων που παράγει το Unity όσο αναφορά τα script. Λειτουργεί όπως οι περισσότερες κονσόλες σε αντίστοιχα και μη λογισμικά, IDE, συντάκτες. Η πρόσβαση σε εξατομικευμένα μηνύματα γίνεται μέσω script με την χρήση της κλάσης Debug ενώ τα σφάλματα και ό τι παράγει το Unity στην κονσόλα επίσης διατίθενται και σε log file για τον προγραμματιστή.

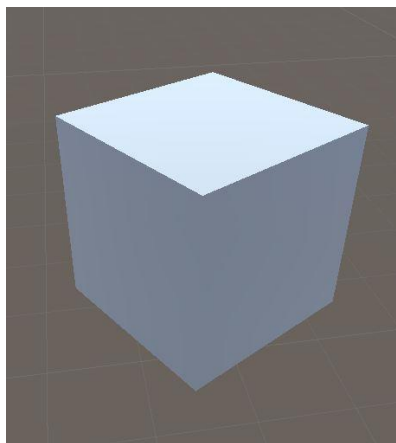
Asset Store

Το asset store αποτελεί ένα δομικό στοιχείο της κοινότητας γύρω από την μηχανή Unity και προσφέρει στους προγραμματιστές πακέτα από υπηρεσίες και προϊόντα επί πληρωμή και μη που μπορούν να χρησιμοποιηθούν εσωτερικά του Unity.

Τα παραπάνω αποτελούν τμήμα της ενδιάμεσης διεπαφής τους προγραμματιστή με την μηχανή και αποτελούν τα βασικά δομικά στοιχεία του λογισμικού. Αξίζει όμως να αναφερθούν και μερικά δομικά στοιχεία του λογισμικού τα οποία έχουν χρησιμοποιηθεί στην εργασία για την καλύτερη κατανόηση του τρόπου λειτουργίας.

3.3 Στοιχεία του Unity

Mesh



Εικόνα 3.7 Mesh με την μορφή κύβου.

Το πλέγμα γραφικών που ονομάζεται Mesh είναι θεμελιώδες στοιχείο γραφικών που διαθέτει το Unity για την αναπαράσταση τρισδιάστατων σχημάτων. Το Unity διαθέτει ένα εργαλείο που λέγεται ProBuilder με το οποίο ο προγραμματιστής μπορεί να διαχειριστεί την γεωμετρία του mesh σε ένα αντικείμενο. Ακόμα προσφέρει βιβλιοθήκες και κλάσεις για δημιουργία γεωμετρίας και γραφικών από τα script για παραγωγή γεωμετρίας την κατά τον χρόνο εκτέλεσης μίας εφαρμογής. Επίσης προκειμένου ένα αντικείμενο να έχει mesh χρειάζεται να διαθέτει στις ιδιότητές του (όπως φαίνεται στο πεδίο Inspector) την ιδιότητα Mesh Filter και Mesh Renderer, των οποίων το πρώτο χρειάζεται ως είσοδο το Mesh του αντικείμενου που διαθέτει αποθηκευμένο το ενεργό project, ενώ το δεύτερο έχει ως είσοδο

την γεωμετρία που δημιουργεί το Mesh Filter απο το Mesh και το μεταφράζει στην σκηνή ,το κάνει ορατό, (render) στην κατάλληλη θέση (εικόνα 3.7).

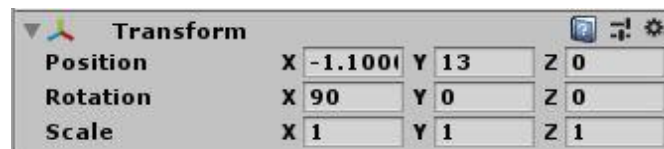
Prefabs



Εικόνα 3.8 Prefabs. Απεικονίζονται ως μπλέ αντικείμενα και αποτελούν αναφορά αντικειμένων. Η διαγραφή τους δεν αποτελεί κίνδυνο.

Prefabs είναι ένα σύστημα του Unity που επιτρέπει στον προγραμματιστή να δημιουργεί να ρυθμίζει και να αποθηκεύει αντικείμενα τύπου GameObjects με την επιθυμητή ιεραρχία αντικειμένων εσωτερικά, συγκεκριμένες ιδιότητες (components) και ρυθμίσεις καθώς και τιμές εσωτερικά των ιδιοτήτων αυτών (εικόνα 3.8). Ο προγραμματιστής έχει την δυνατότητα να μεταβάλλει όπως θέλει ένα αρχείο Prefab. Επίσης η διαγραφή αντικειμένων Prefab απο την ιεραρχία ενός ενεργού Project δεν αποτελεί κίνδυνο αφού οι ρυθμίσεις θέσης και τιμών παραμένουν αποθηκευμένες. Η χρήση των Prefabs είναι προαιρετική με βάση το Unity και την αρχιτεκτονική του, δηλαδή οποιοδήποτε project είναι λειτουργικό και χωρίς Prefabs. Εντούτοις, οι περισσότερες εφαρμογές χρησιμοποιούν Prefabs αφού προσφέρουν επαναχρησιμοποίηση αντικειμένων με αποθηκευμένες ρυθμίσεις ιεραρχίας αλλά και δομικών στοιχείων και ασφάλεια.

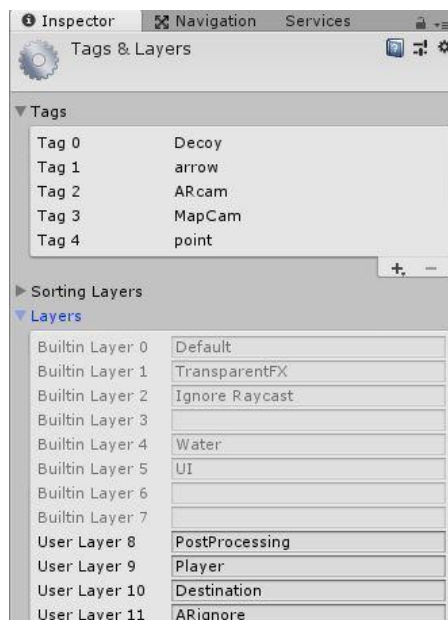
Transform



Εικόνα 3.9 Πεδίο για την απεικόνιση θέσης, κατεύθυνσης και κλίμακας αντικειμένου.

Μια θεμελιώδης ιδιότητα που πρέπει να διαθέτουν όλα τα αντικείμενα εσωτερικά της σκηνής του Unity είναι το Transform. Ενώ δεν είναι ιδιαίτερα πολύπλοκο αποτελεί βασικό στοιχείο αφού καθορίζει την θέση, την θέση περιστροφής και τις διαστάσεις των αντικειμένων (εικόνα 3.9).

Tags και Layers

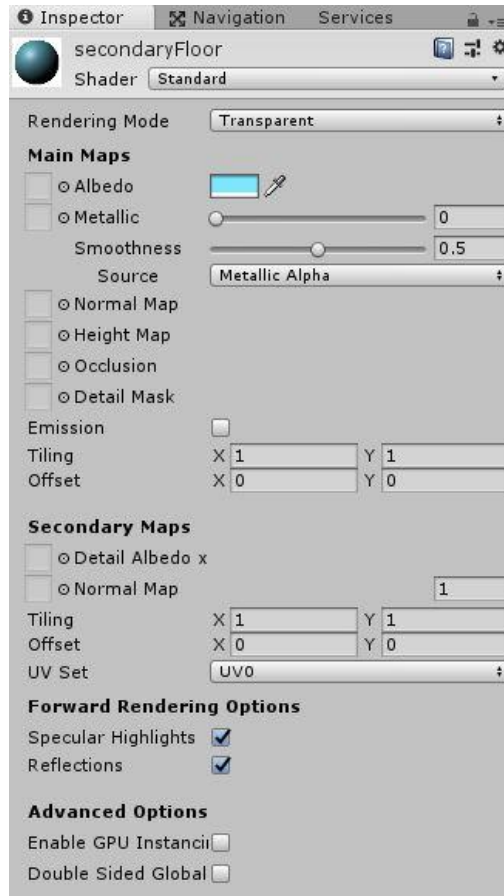


Εικόνα 4.0 Πεδίο Tags και Layers για τον ορισμό καινούριων ή την διαγραφή υπάρχοντων.

Τα δύο αυτά στοιχεία αποτελούν ιδιότητες των αντικειμένων του Unity και ο κύριος σκοπός είναι ο διαχωρισμός των αντικειμένων και των Prefabs γενικότερα εκτός από το όνομά τους και το είδος τους. Είναι ιδιαίτερα χρήσιμα στην διάκριση πανομοιότυπων αντικειμένων όταν ο προγραμματιστής επιθυμεί να εκτελούν διαφορετικές ενέργειες. Αντίστοιχα μπορούν να χρησιμοποιηθούν με σκοπό η μηχανή να λαμβάνει διαφορετικές αποφάσεις για το κάθε layer ή tag. Η πρόσβασή τους πραγματοποιείται κυρίως από τα script. Προσδιορίζονται όμως από μια μεγάλη διαφορά. Τα tags είναι κυρίως ονόματα αναφοράς των αντικειμένων που λειτουργούν ως ένας ακόμα χαρακτηρισμός. Ο διαχωρισμός τους παραμένει σε επίπεδο το πεδίου της σκηνής (εικόνα 4.0). Τα layers είναι στρώματα τα οποία χρησιμοποιούνται για διαχωρισμό στο επίπεδο μετάφρασης και απεικόνισης των αντικειμένων και του mesh που διαθέτουν. Μια πολύ συνήθης χρήση των στρωμάτων είναι η αποφυγή φωτισμού αντικειμένων που έχουν οριστεί να βρίσκονται σε συγκεκριμένα στρώματα στην σκηνή.

Materials

Το “υλικό” Material που διαθέτει το Unity ορίζει τον τρόπο μία επιφάνεια ενός πλέγματος πολύγωνων θα μεταφράζεται. Πιο συγκεκριμένα σε συνδυασμό με την υφή ενός αντικειμένου το υλικό καθορίζει τον τρόπο που θα κάνει render το Unity την επιφάνεια ενός από τα πολύγωνα και συνολικά όλο το mesh του αντικειμένου. Η δημιουργία των Material στο Unity είναι αρκετά απλουστευμένη αφού μπορεί να γίνει μέσω του inspector (εικόνα 4.1), όμως μπορεί να γίνει αρκετά πολύπλοκη αφού η μηχανή προσφέρει την δυνατότητα δημιουργίας σύνθετων Material και μέσω script και με την μορφή κόμβων αποτελούμενων από τιμές και ρυθμίσεις υλικών.



Εικόνα 4.1 Τιμές και ιδιότητες σε ένα "υλικό".

UI

Η διεπαφή χρήστη που προσφέρει το Unity είναι ένα απο τα μέσα που παρέχει ώστε να πραγματοποιήσει επικοινωνία με τον χρήστη. Παρέχει πλαίσια, θέσεις για εικόνες, κουμπιά, πλαίσια κειμένων και πολλά άλλα. Λειτουργεί ως ένα διδιάστατο στατικό στρώμα στην οθόνη

ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

4.1 Γενική Περιγραφή

Το InNav είναι μία εφαρμογή επαυξημένης πραγματικότητας για κινητές συσκευές που έχει ως πρωταρχικό στόχο την σύνθεση από υπάρχον τεχνολογίες και εργαλεία με σκοπό την ακριβή και εύκολη χρήση της για πλοήγηση σε εσωτερικούς χώρους καθώς και την απεικόνιση πληροφοριών για το περιβάλλον που υπάρχει γύρω από τον χρήστη.

Η προσέγγιση εφαρμογών που χρησιμοποιούνται ως εργαλεία είναι συνήθως το απλουστευμένο γραφικό περιβάλλον για τον χρήστη ενώ παράλληλα η ενίσχυση της εφαρμογής με πληροφορίες που θα εξυπηρετήσουν τον σκοπό του “εργαλείου-εφαρμογής”. Μια εφαρμογή επαυξημένης πραγματικότητας είναι εύκολο να πετύχει κάτι τέτοιο λόγω της ανάμειξης γραφικών με το πραγματικό περιβάλλον με την συνεργασία κάμερα-οθόνης.

Όμως κάποια χαρακτηριστικά που υπάρχουν σε όλα τα είδη εφαρμογών παραμένουν κοινά ακόμα και με την χρήση επαυξημένης πραγματικότητας. Αρχικά ένα κοινό που έχουν οι περισσότερες εφαρμογές, εργαλεία και συγκεκριμένα εργαλεία πλοήγησης ακόμα και σε εσωτερικούς χώρους είναι η χρήση τρίτων εργαλείων ως πηγές για τον υπολογισμό θέσης με την χρήση τριγωνισμού, beacons, WIFI GPS. Ενώ αυτά τα εργαλεία είναι αρκετά χρήσιμα για την διόρθωση θέσης της συσκευής σε τυχόν λάθος υπολογισμού στο περιβάλλον, πολλά από αυτά είναι αναξιόπιστα, λόγω κάποιας δυσλειτουργίας ή εξασθένησης σήματος και δεν αποτελούν μέρος μιας συνολικής λύσης για κάθε περιβάλλον, όπως οι διαφορετικές δομές κτηρίων η ακόμα και τα υλικά που είναι κατασκευασμένα και μπορεί να έχουν διαφορετική επίδραση στο σήμα. Επιπρόσθετα πολλά από τα εργαλεία που είναι προσβάσιμα στο κοινό είναι δύσχρηστα ή είναι κατευθυνόμενα προς ένα σχετικά μικρό εύρος ανθρώπων είτε βάση ηλικίας είτε επίπεδο αντίληψης ως προς τεχνολογικά προϊόντα.

Επομένως σύμφωνα με τον γενικό στόχο που προαναφέρθηκε και έπειτα από την ανάλυση μερικών από των γενικών χαρακτηριστικών που προσδιορίζουν μεγάλο πλήθος εφαρμογών στην αγορά πρέπει να διαμορφωθούν κατάλληλα για την δημιουργία ενός εύχρηστου εργαλείου που να πληρεί αυτούς τους στόχους.

Το InNav είναι ένα εργαλείο πλοήγησης σε εσωτερικούς χώρους χωρίς την χρήση διαδικτύου, συστημάτων GPS ή φάρων(beacons) για την σχετική προσέγγιση θέσης και παροχής πληροφοριών του χώρου σε μορφή επαυξημένης πραγματικότητας. Η εφαρμογή χρησιμοποιεί κυρίως την κάμερα και άλλους αισθητήρες όπως το γυροσκόπιο, ως είσοδο δεδομένων για τον υπολογισμό θέσης. Τα δεδομένα αρχικά διαχειρίζεται η βιβλιοθήκη που παρέχει η εταιρία Google μέσω του ARCore. Ο υπολογισμός των διαδρομών, ο προγραμματισμός και η προβολή των αντικειμένων όπως και το περιβάλλον που έχει δομηθεί και υλοποιηθεί η εργασία έχουν πραγματοποιηθεί μέσω του Unity. Ως παράδειγμα για την πτυχιακή εργασία έχει χρησιμοποιηθεί ένα σχέδιο εσωτερικού χώρου που είναι αντιστοιχεί στον πραγματικό χώρο δοκιμής της εφαρμογής.

4.2 Περιγραφή σκηνής Unity

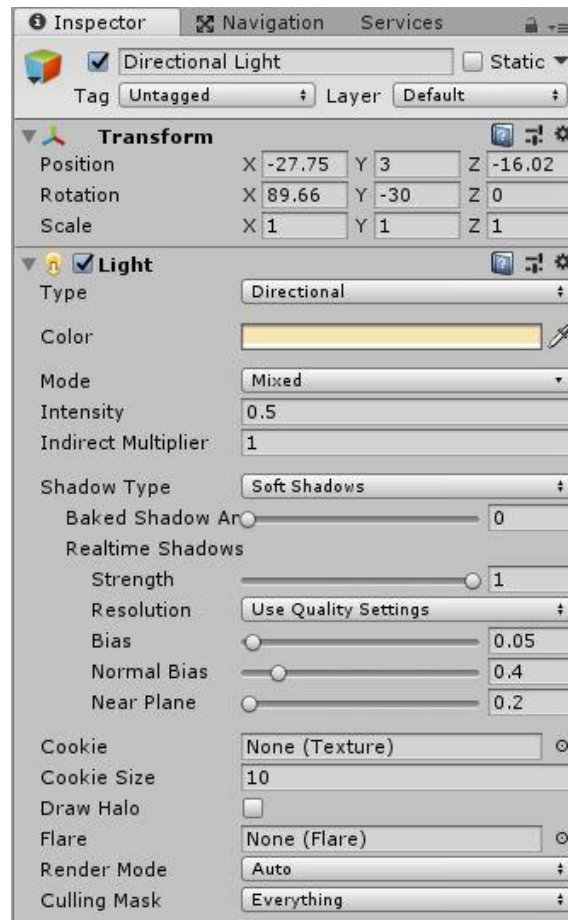
Η περιγραφή των ενεργών αντικειμένων της σκηνής θα προσδιοριστεί κυρίως από τα χαρακτηριστικά τους που εμφανίζονται στο πεδίο Inspector. Αρχικά θα γίνει περιγραφή για την δομή τους στην ιεραρχία και στην σκηνή και την σχέση μεταξύ τους. Η περιγραφή των ρυθμίσεων και των απαιτήσεων της εργασίας για το τελικό εκτελέσιμο αρχείο τύπου “.apk” θα γίνει στο τέλος.

Η εργασία έχει υλοποιηθεί συνολικά σε μία σκηνή Unity με όνομα “SampleScene” στον φάκελο “Scenes”. Η ονομασία της σκηνής σε ένα project στο Unity δεν έχει καμία επίδραση στο ίδιο το project και ειδικά όταν αυτό απαρτίζεται από μία μόνο σκηνή. Όμως η μετανομασία του κατά την διάρκεια ανάπτυξης μπορεί να έχει επιπτώσεις στην αναγνώριση του. Αν το project περιέχει παραπάνω από μία σκηνή τότε υπάρχει σημασία ως προς την σειρά που θα πρέπει να μεταφράσει το Unity τις ενεργές σκηνές κατά την εκτέλεση.

Η περιγραφή των αντικειμένων και των διάφορων prefabs που έχουν χρησιμοποιηθεί θα πραγματοποιηθεί όπως εμφανίζονται στο πεδίο της ιεραρχίας στο project. Η θέση των αντικειμένων στην ιεραρχία δεν έχει σημασία, ούτε δίνει κάποια προτεραιότητα στα αντικείμενα και στην εκτέλεση τους, υπάρχει κυρίως έτσι για την ευκολία του προγραμματιστή. Η ονομασία των αντικειμένων όμως είναι σημαντική αφού εσωτερικά των script μπορεί να γίνει αναφορά ανάλογα με το όνομα του κάθε αντικειμένου.

Επίσης στο πεδίο Inspector, δεν θα γίνει αναλυτική επεξήγηση προς όλες τις παραμέτρους σε κάθε ιδιότητα των αντικειμένων παρά μόνο στις τιμές και τις ιδιότητες που λαμβάνουν σημαντικό ρόλο στην υλοποίηση του project.

Directional Light

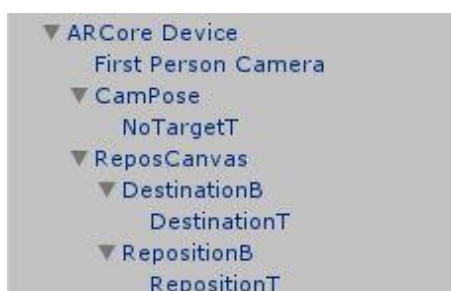


Εικόνα 4.2 Αντικείμενο Directional Light. Υπεύθυνο για τον φωτισμό της σκηνής.

Στην ιεραρχία θα παρατηρήσουμε ως πρώτο αντικείμενο το “Directional Light” το οποίο είναι ένα από τα βασικά αντικείμενα που παρέχει το Unity στην δημιουργία ενός καινούριου project μαζί με μία κάμερα Camera Object και ένα EventSystem που είναι υπεύθυνο για την διαχείριση της γραφικής επιφάνειας χρήστη, δηλαδή αντικείμενα όπως κουμπιά (Buttons), μενού (Menu), προβολών εικόνων και άλλα. Το “κατευθυνόμενο φως” όπως προσδιορίζεται από το Unity είναι ένα από τους τέσσερις τύπους που διαχειρίζεται την προσομοίωση φωτός στην σκηνή. Στην σκηνή η διαχείριση του Directional Light δεν είναι ιδιαίτερα σημαντική γιατί λαμβάνεται υπόψη από την διαδικασία render του Unity μόνο στα αντικείμενα που βρίσκονται στο οπτικό πεδίο μίας απλής κάμερας της σκηνής του Unity. Πιο συγκεκριμένα κάμερες όπως αυτές που έχουν χρησιμοποιηθεί ως AR και ως QR είναι κάμερες που δέχονται δεδομένα από την πραγματική κάμερα της συσκευής (εικόνα 4.2).

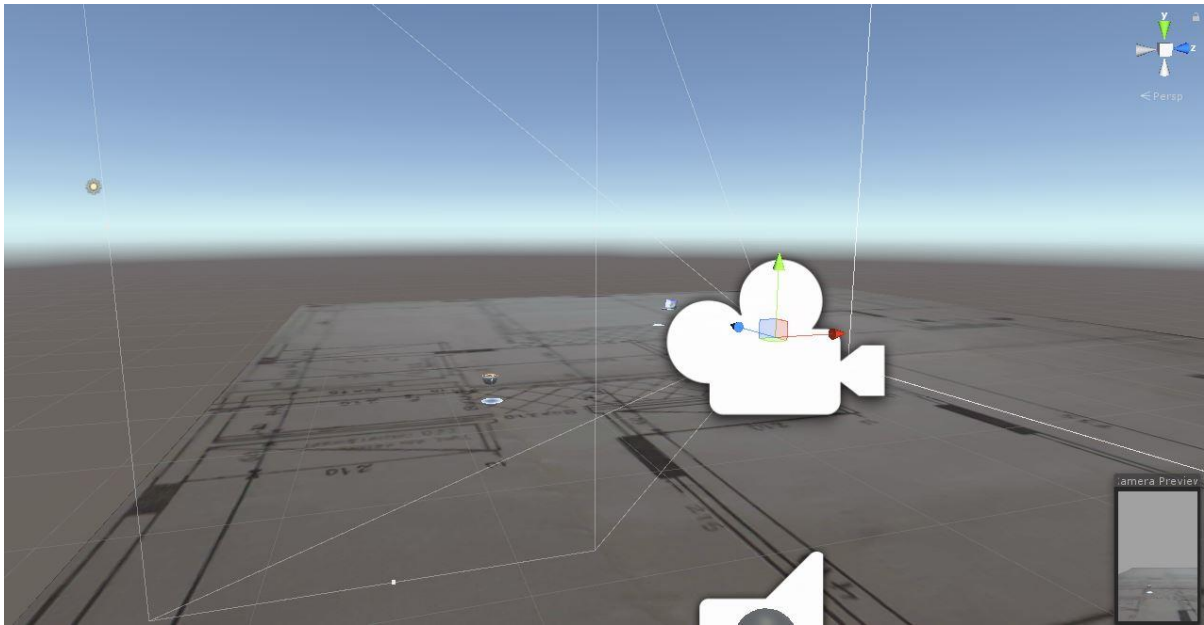
4.2.1 Στάδιο Αρχικοποίησης Session και ARCore Device

Το δεύτερο αντικείμενο που απεικονίζεται στην εικόνα είναι ένα το ARCore Device το οποίο περιέχει την ενεργή κάμερα στην εφαρμογή με το ξεκίνημά της. Αρχικά, προκειμένου να συμπεριλάβουμε το αντικείμενο “ARCore Device” που βρίσκεται στην ιεραρχία πρέπει να εισάγουμε (import), ως asset στο project το ARCore που έχει δημιουργηθεί από την Google , δηλαδή ένα σύνολο από αντικείμενα , αρχεία από κομμάτια κώδικα (scripts), αρχείων διαμόρφωσης υπεύθυνα για την αρχικοποίηση και την ρύθμιση των prefabs που περιέχονται στα αρχεία και τα ίδια τα prefabs. Το prefab αυτό έχει την εξής δομή: ARCore Device → First Person Camera / CamPose / ReposCanvas (εικόνα 4.3).



Εικόνα 4.3 Αντικείμενο AR Core Device με την κάμερα και το UI στην ενεργή σκηνή.

Το ARCore είναι ένα αντικείμενο το οποίο είναι κενό ,όσο αναφορά το mesh του αντικειμένου, (δηλαδή τα γραφικά που δημιουργούν την απεικόνιση του αντικειμένου) και άρα δεν είναι ορατό στην σκηνή. Αποτελεί κυρίως δοχείο, ή γονιό για την κάμερα και το Canvas (τον καμβά) που εμφανίζονται στην σκηνή (εικόνα 4.4). Το αντικείμενο είναι απαραίτητο για την λειτουργία της κάμερας AR αφού λειτουργεί ως αρχείο αρχικοποίησης του αντικειμένου AR και της κάμερας AR. Συγκεκριμένα αποτελείται από ένα script, το “ARCoreSession” που είναι υπεύθυνο για την αρχικοποίηση των μεταβλητών, των τιμών των αντικειμένων και της κάμερας επαυξημένης πραγματικότητας του Google μέσω του API της. Η διαδικασία έναρξης ενός session πραγματοποιείται μέσω της κλάσης LifecycleManager. Προκειμένου να ελέγξουμε κατάλληλα το στιγμιότυπο(instance) της κλάσης χρειάζεται και πιο γενικά το session της AR κάμερας πρέπει να καλέσουμε τις αντίστοιχες μεθόδους.



Εικόνα 4.4 Κάμερα AR και χάρτης.

Μέσω του Unity API έχουν χρησιμοποιηθεί οι μέθοδοι:

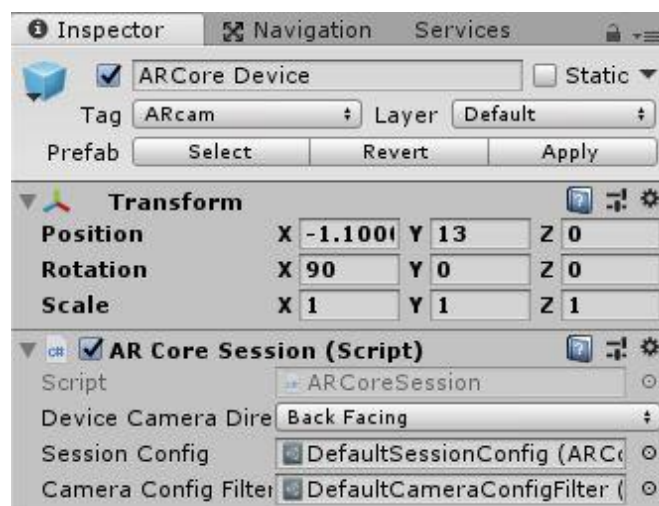
- `void Awake()` στην οποία καλούμε την μέθοδο `CreateSession()` του ARCore API. Ουσιαστικά μέσω της δεύτερης καλούμε την δημιουργία ενός καινούριου `session` όπου θα διατηρηθεί σε όλη την διάρκεια ζωής του αντικειμένου ενώ χρησιμοποιούμε την `Awake()` ως αρχικοποίηση εσωτερικά στο script προκειμένου να φορτωθεί το `session` με το ξεκίνημα της εφαρμογής.
- Η `OnEnable()` που περιέχει την `EnableSession()` λειτουργούν ως εξής. Πρώτα καλείται η μέθοδος `OnEnable()` σε περίπτωση που το αντικείμενο `ARCoreDevice` έχει προηγουμένως απενεργοποιηθεί και ξανα-ενεργοποιεί το `session` που έχει δημιουργηθεί κατά την έναρξη της εφαρμογής στην συσκευή.
- Οι αντίστοιχες μέθοδοι `OnDestroy()` και `OnDisable()` είναι υπεύθυνες για την καταστροφή του `session` και του αντικειμένου και η δεύτερη για την παύση του αντικειμένου και του `session` εφόσον το αντικείμενο δεν είναι ενεργό στην σκηνή.

Ακόμα το script “AR Core Session” περιλαμβάνει τρεις σημαντικές παραμέτρους:

1. Το Device Camera Direction δημιουργεί ένα αντικείμενο, το “DeviceCameraDirection” το οποίο επιστρέφει μία τιμή επιβεβαίωσης τιμής. Αν η τιμή αυτή είναι μηδέν τότε έχουμε επιλέξει την πίσω κάμερα μίας συσκευής ενώ με την τιμή ένα έχουμε επιλέξει την μπροστινή κάμερα.
2. Τα ArcoreSessionConfig και η ARCoreCameraConfigFilter είναι δύο κλάσεις οι οποίες δημιουργούν τα αντικείμενα SessionConfig και CameraConfigFilter.
3. Για το SessionConfig εσωτερικά script “ARCoreSession” ελέγχουμε με συνθήκη αν έχει πραγματοποιηθεί η αντίστοιχη ανάθεση από το αντικείμενο SessionConfig από τα αρχεία που περιέχονται στο πεδίο “Project” όπως επίσης εφόσον έχει γίνει η ανάθεση και τον τρόπο υπολογισμού LightEstimationMode.

Το “ARCoreSession” (εικόνα 4.5) επίσης είναι υπεύθυνο για τον έλεγχο του CameraConfigFilter και αν έχει γίνει ανάθεση σε αυτό.

Και στα δύο αντικείμενα γίνεται έλεγχος για την ανάθεσή τους σε αρκετές από τις μεθόδους εσωτερικά του “ARCoreSession” κάθε φορά που ενεργοποιείται ή ενημερώνεται το script με καινούριες τιμές.



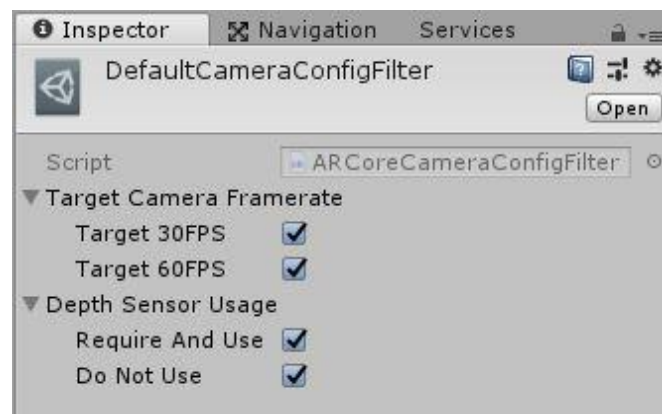
Εικόνα 4.5 Το αρχείο ARCoreSession, υπεύθυνο για τις ρυθμίσεις λειτουργίας κάμερας AR.

Τα δύο αυτά αντικείμενα είναι απαραίτητα για την σωστή λειτουργία δημιουργίας session από το ARCoreSession και γίνεται η ανάθεση τους από τον Editor εσωτερικά του Unity. Το

κάθε ένα από αυτά περιέχουν και ένα script τα οποία είναι υπεύθυνα για την αρχικοποίηση συγκεκριμένων ενεργειών.

AR Configuration Files

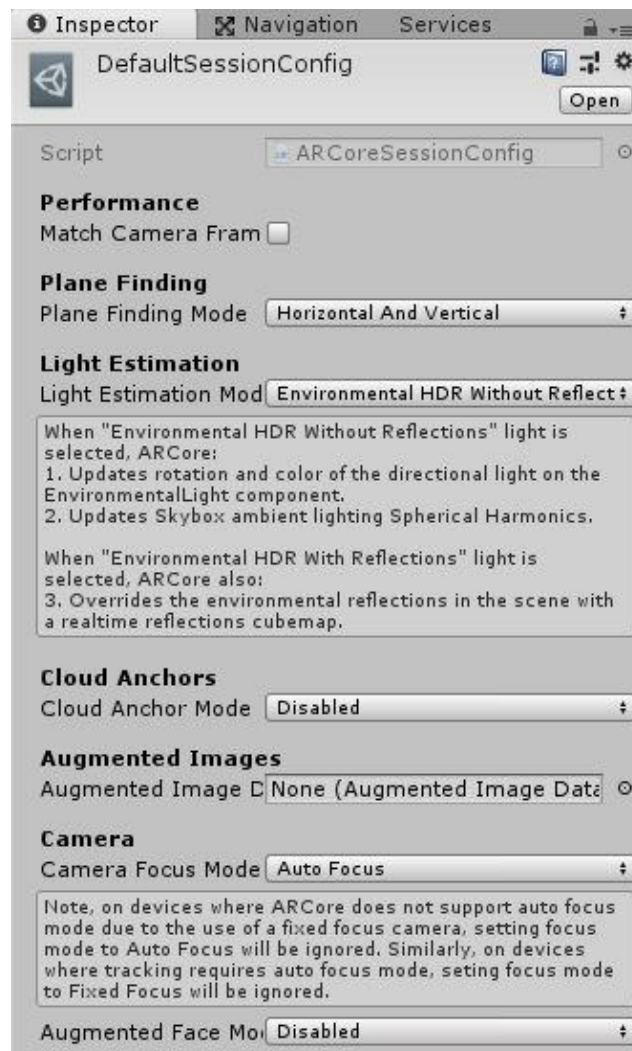
Συγκεκριμένα το ARCoreCameraConfigFilter που είναι τοποθετημένο στο αντικείμενο DefaultCameraConfigFilter περιέχει ρυθμίσεις ως προς τις εικόνες ανά καρέ που θα επεξεργάζονται οι αλγόριθμοι (30 fps ή 60fps) όπως και την επιλογή του αισθητήρα βάθους από την συσκευή. Στο projection έχουν επιλεγθεί και οι δύο επιλογές ως προς τα fps όπως και για την χρήση αισθητήρα βάθους (Depth Sensor) (εικόνα 4.6).



Εικόνα 4.6 Επιλογή χρήσης 30 και 60 καρέ ανά δευτερόλεπτο από την κάμερα όπως και αισθητήρα βάθους για την κάλυψη περισσότερων συσκευών.

Το DefaultSessionConfig (εικόνα 4.7) είναι ένα αντικείμενο το οποίο χρειάζεται το script “ARCoreSession” για να δημιουργήσει καινούριο session και ορίζει σημαντικές ρυθμίσεις όπως η απόδοση της κάμερας όσο αναφορά τα καρέ και την αντιστοιχία της με τα καρέ που μεταφράζει το Unity. Επίσης καθορίζει για το είδος της επιφάνειας που θέλουμε να αναγνωρίσουμε. Εφόσον στην εργασία η συσκευή χρειάζεται να αλλάζει θέση στον χώρο ενώ οι πληροφορίες που έρχονται ως είσοδο στην κάμερα πρέπει να διαχωρίζονται έχει επιλεγθεί να αναγνωρίζονται διαφορετικά και οι οριζόντιες αλλά και οι κάθετες επιφάνειες στον χώρο, σε σχέση με την συσκευή (αν την στιγμή που ο χρήστης ξεκίνησε την εφαρμογή η συσκευή ήταν σε θέση πορτρέτου οι επιφάνειες αναγνωρίζονται αντιστοίχως, οι οριζόντιες επιφάνειες ως οριζόντιες και οι κάθετες ως κάθετες, ενώ αν ανοίξει η συσκευή σε θέση landscape οι αντιστοιχίες θα μεταβληθούν). Ανεξάρτητα όμως με τον τρόπο λειτουργίας και τον προσανατολισμό της συσκευής, η αναγνώριση επιπέδων θα μεταφράζεται το ίδιο. Έπειτα

έχουμε την επιλογή υπολογισμού φωτός όπου έχουμε κάνει την επιλογή Environmental HDR Without Reflection η οποία επιτρέπει στην εφαρμογή και στο ARCore Session να υπολογίζει το φως που έρχεται από τον χώρο προς την κάμερα με ρεαλιστικό υπολογισμό χωρίς να υπολογίζονται οι αντανακλάσεις. Οι ρυθμίσεις για Cloud Anchors και Augmented Images δεν είναι ενεργοποιημένες αφού δεν έχουν χρησιμοποιηθεί για την αναγνώριση σημείων στον χώρο. Επίσης να προσθέσουμε ότι αν και τα cloud Anchors είναι χρήσιμα χρειάζεται πρόσβαση στο διαδίκτυο, ενώ η χρήση τους εκτός διαδικτύου έχει περιορισμένο χρονικό περιθώριο. Τέλος υπάρχει και κατάλληλη ρύθμιση για τον τρόπο που θα λειτουργεί η εστίαση της κάμερας η οποία είναι ρυθμισμένη στην αυτόματη εστίαση, ιδιαίτερα σημαντικό αφού μετακινείται η συσκευή στον χώρο.

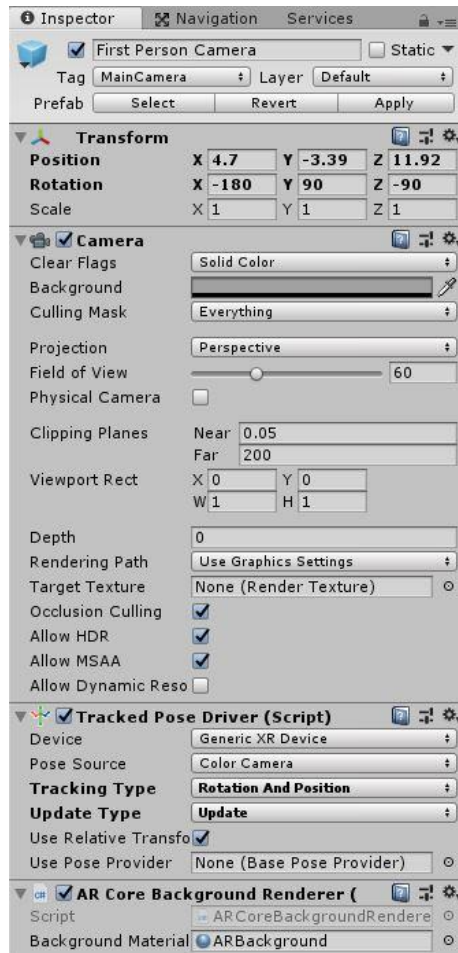


Εικόνα 4.7 Πεδίο επιλογής ποιότητας και αναγνώρισης κάθετων και οριζόντιων επιπέδων από το αρχείο ARCoreSessionConfig.

First Person Camera

Έπειτα απο την αρχικοποίηση των ρυθμίσεων λαμβάνει χώρα το αντικείμενο First Person Camera. Το αντικείμενο στο οποίο έχει δοθεί το δομικό στοιχείο “Camera” το οποίο αποτελεί ένα αντικείμενο rendering απο την σκηνή στην οθόνη. Το αντικείμενο “Camera” χωρίς κάποια τροποποίηση δεν μπορεί να διαβάσει δεδομένα απευθείας απο οποιαδήποτε κάμερα μπορεί να διαθέτει μία συσκευή. Για την λειτουργία της κάμερας ως μετάφρασης δεδομένων απο την φυσική κάμερα του κινητού έχει χρησιμοποιηθεί το script AR Core Background Renderer. Το script χρησιμοποιεί τα δεδομένα του αισθητήρα κάμερας της συσκευής για να τα αντιστοιχήσει με την “Camera” που προσφέρει το Unity.

Συγκεκριμένα στην μέθοδο OnEnable() του Unity, ελέγχει αν υπάρχει και αν είναι ενεργό ένα session, που επιβεβαιώνει μέσω της κλάσης Lifecycle, όπως επίσης και ενεργοποιεί την διαδικασία μετατροπής απορρόφησης δεδομένων απο την κάμερα σε μία δισδιάστατη εικόνα προκειμένου η κάμερα του Unity να την κατανοήσει. Η διαδικασία ξεκινά με το EnableARBackgroundRendering(); ενώ δημιουργούμε ένα texture για την απεικόνιση σε 2D εικόνα. Στο script επίσης υπάρχει η διαδικασία μετάφρασης των δεδομένων με “invert culling” το οποίο θα ορίσει αν ένα πολύγωνο, δηλαδή αν ένα σχήμα ή μία επιφάνεια θα μεταφραστεί απο το πρόγραμμα. Στην μία περίπτωση οπου ο οπίσθιος αισθητήρας κάμερας είναι ενεργός τότε δεν χρειάζεται να ανατραπεί η διαδικασία ενώ με τον εμπρόσθιο αισθητήρα κάμερας θα χρειαστεί να γίνει inverted culling. Στο script AR Core Background Renderer και εσωτερικά στην μέθοδο _UpdateShaderVariables() χρησιμοποιείται ο τύπος Material απο το Unity με σκοπό την εμφάνιση της εικόνας που λαμβάνουμε απο την κάμερα της συσκευής (εικόνα 4.8).



Εικόνα 4.8 Ρυθμίσεις κάμερας AR με επιλογή ανίχνευσης θέσης και προσανατολισμού.

Επιπρόσθετο δομικό στοιχείο που περιέχει το First Person Camera είναι το tracked Pose Driver το οποίο είναι script που ρυθμίζει τις προτιμήσεις της κάμερας. Συγκεκριμένα στην εφαρμογή έχουμε επιλέξει την συσκευή Generic XR Device ώστε να δηλώνει ότι πρόκειται για μία συσκευή επομένως να λαμβάνει κυρίως δεδομένα. Μια σημαντική ρύθμισή που επηρεάζει κατά πολύ την ακρίβεια της εφαρμογής είναι η επιλογή Tracking Type. Ενώ έχει γίνει η επιλογή Rotation And Position δηλαδή να υπολογίζει τις σχέσεις και στον προσανατολισμό αλλά και στην θέση της συσκευής η επιλογή Only Position επιφέρει πιο ακριβείς πληροφορίες και δημιουργεί μία πιο drift free εμπειρία. Ο λόγος που έχει επιλεγεί η πρώτη και όχι η δεύτερη είναι κυρίως γιατί αντικείμενα και prefabs που χρειαζόμαστε για να δημιουργήσουμε ένα πιο κατανοητό περιβάλλον AR ως εμπειρία του χρήστη, όπως θα δούμε παρακάτω, χρειάζονται και υπολογισμό Rotation.

Η διαδικασία δημιουργίας session, δηλαδή η αρχικοποίηση της συσκευής AR όσο αναφορά τον τρόπο που θα λαμβάνονται τα δεδομένα όπως και τον τρόπο που θα λειτουργεί η κάμερα ενεργοποιείται πρώτη στην σχέση αντικειμένων ARCoreDevice - First Person Camera. Πιο συγκεκριμένα δεν υπάρχει κάποια σειρά που θα φορτωθούν τα ενεργά script από την εφαρμογή, όμως προκειμένου να λειτουργήσει σωστά η κάμερα AR από το αντικείμενο First Person Camera χρειάζεται να έχουν αρχικοποιηθεί όλες οι απαραίτητες ρυθμίσεις από το script του ARCoreDevice(ARCoreSession.cs) και αντίστοιχα από τα DefaultSessionConfig και DefaultSessionConfigFilter

Με βάση τα παραπάνω μπορούμε να περιγράψουμε το πρώτο στάδιο της εφαρμογής. Ουσιαστικά η συσκευή με το άνοιγμα του προγράμματος ενώ φορτώνει όλα τα αρχεία, περιμένει πρώτα να αρχικοποιηθούν οι ρυθμίσεις ενώ δημιουργεί ένα καινούριο AR Session. Έπειτα ο αισθητήρας της κάμερας με το που αναγνωρίσει τα πρώτα δεδομένα, αναγνωρίζει συνεχώς καινούρια και σχεδιάζει μια εικόνα και την προβάλλει συνεχώς, σε κάθε frame στην οθόνη της συσκευής. Από το σημείο αυτό η συσκευή έχει κάνει localize και ο χρήστης έχει την επιλογή είτε να κάνει σάρωση QR είτε να καθορίσει έναν προορισμό στον χάρτη.

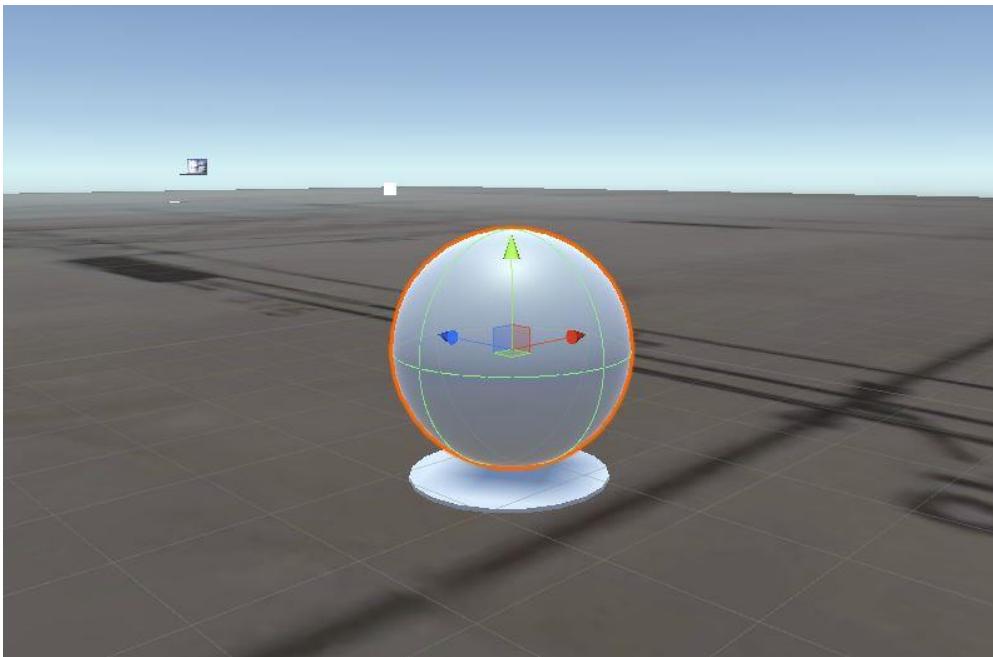
4.2.2 Συσχετισμός Κάμερας AR και Πλοήγηση

Navigation και NavMesh

Η αιτία ως χρήσης της μηχανής ανάπτυξης παιχνιδιών Unity στην εργασία δεν ήταν μόνο για την δημιουργία ενός αντικειμένου AR μόνο. Το πλεονέκτημα που μας παρέχει είναι η ευκολία δημιουργίας αντικειμένων και η οπτικοποίηση τους. Σαν επόμενο βήμα έχει η δημιουργία μίας σχέσης των δεδομένων AR και η αναπαράστασή τους εσωτερικά της μηχανής. Στο δεύτερο τμήμα έχει αρχικοποιηθεί ένα prefab, αντικείμενο υπεύθυνο για τον συσχετισμό μεταξύ κάμερας AR και μηχανής. Ακόμα έχουμε δημιουργήσει έναν agent(πράκτορα) σε συνδυασμό με το Navigation Mesh του Unity για τον υπολογισμό του πιο γρήγορου μονοπατιού και έναν χαρακτήρα που αντιστοιχεί στην συσκευή όπως επίσης έχουμε τοποθετήσει και έναν χάρτη σε 1:1 κλίμακα σε αντιστοιχία με τον πραγματικό χώρο

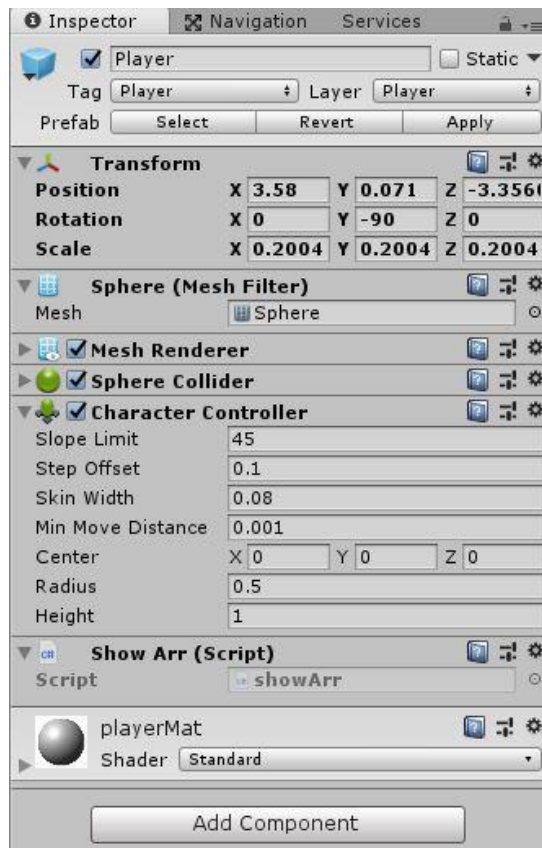
που αντιπροσωπεύει η εφαρμογή. Προκειμένου όμως να είναι κατανοητές οι σχέσεις μεταξύ των prefabs αυτών πρέπει να εξηγήσουμε τον τρόπο λειτουργίας εύρεσης του πιο σύντομου μονοπατιού.

Αντικείμενο Player



Εικόνα 4.9 Σφαιρικό σχήμα που αντιπροσωπεύει το αντικείμενο Player.

Το στάδιο σχεδιασμού των script για τις ρυθμίσεις της κάμερας AR μέσω του ARCore API είναι ολοκληρωμένο μέχρι αυτό το σημείο, όμως προκειμένου να υπάρχει αντιστοιχία κινήσεων του πραγματικού κόσμου με την μηχανή του Unity και η τοπικοποίηση να λάβει χώρα εσωτερικά της μηχανής Unity πρέπει να πραγματοποιηθούν οι αντίστοιχοι συσχετισμοί. Στην ιεραρχία έχει δημιουργηθεί ένα αντικείμενο που το έχουμε ονομάσει “Player” (εικόνα 4.9).



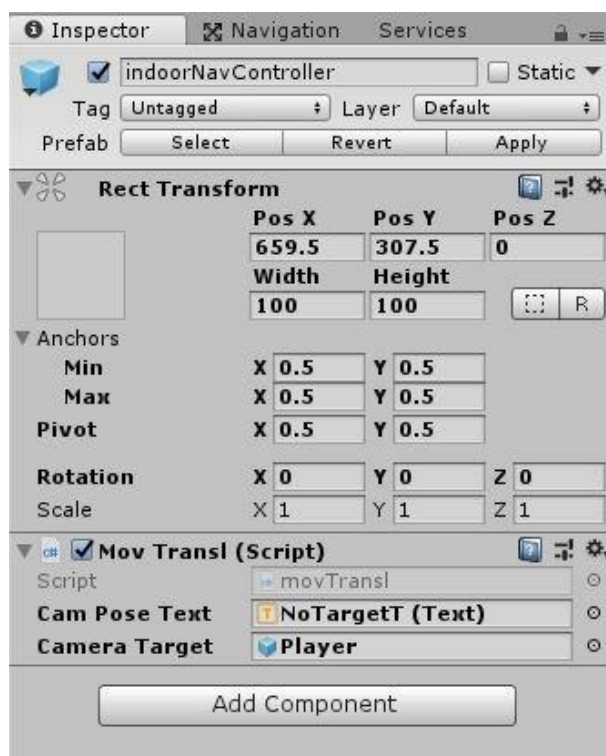
Εικόνα 5.0 Ρυθμίσεις αντικειμένου Player.

Το αντικείμενο αυτό είναι ένα τρισδιάστατο σφαιρικό αντικείμενο και παίζει τον ρόλο του χρήστη/συσκευής. Το Player είναι ένα prefab το οποίο είναι αρχικοποιημένο ως θέση σε ένα σημείο του χάρτη και περιέχει τις ιδιότητες, όπως μπορεί να παρατηρηθεί από τον Inspector, Transform που δηλώνει την θέση και τον προσανατολισμό όπως και σε όλα τα αντικείμενα εσωτερικά του Unity, ακόμα και αν μην διαθέτουν mesh, Sphere mesh filter και Renderer υπεύθυνα για το σχήμα και την επιλογή της μηχανής να μεταφράζει το σχήμα ως υπαρκτό και έπειτα έχουμε το Sphere Collider και Character Controller καθώς και το script Show Arr.

Τα τρία αυτά components είναι εξίσου σημαντικά αλλά προς το παρόν θα αναφερθεί μόνο το Character Controller το οποίο προσφέρει την ιδιότητα σε ένα αντικείμενο να μετακινείται στον χώρο όπως ένας χαρακτήρας σε ένα παιχνίδι. Το δομικό αυτό στοιχείο αν και δεν προσφέρει μεγάλη ελευθερία στο αντικείμενο είναι κατάλληλο για την εργασία αφού χρειάζεται μόνο η κατεύθυνση και η ικανότητα του αντικειμένου να προχωράει στον χώρο. Το Character Controller επίσης είναι λειτουργικό μόνο όταν υπάρχει κάποια είσοδος από δεδομένα (εικόνα 5.0).

Player Ένωση IndoorNav Controller

Ο στόχος είναι η ένωση των δεδομένων απο την κάμερα που μας δίνονται απο τα script των αντικειμένων ARCoreDevice και First Person Camera με το αντικείμενο Player. Το μέσο για την ένωση αυτών των δύο είναι η δημιουργία ενός “κενού” αντικειμένου στην ιεραρχία το οποίο έχει ονομαστεί τυχαία “indoorNavController”. Το αντικείμενο αυτό περιέχει το script “MovTransl” και έχει ως δημόσιες μεταβλητές ένα πεδίο για κείμενο Text το οποίο παρέχουμε εσωτερικά του ARCoreDevice στο UI “CamPose” που περιέχει το NotargetT και ένα πεδίο τύπου GameObject που έχει τοποθετηθεί το αντικείμενο Player (εικόνα 5.1).



Εικόνα 5.1 Ρυθμίσεις και ιδιότητες αντικειμένου IndoorNavController.

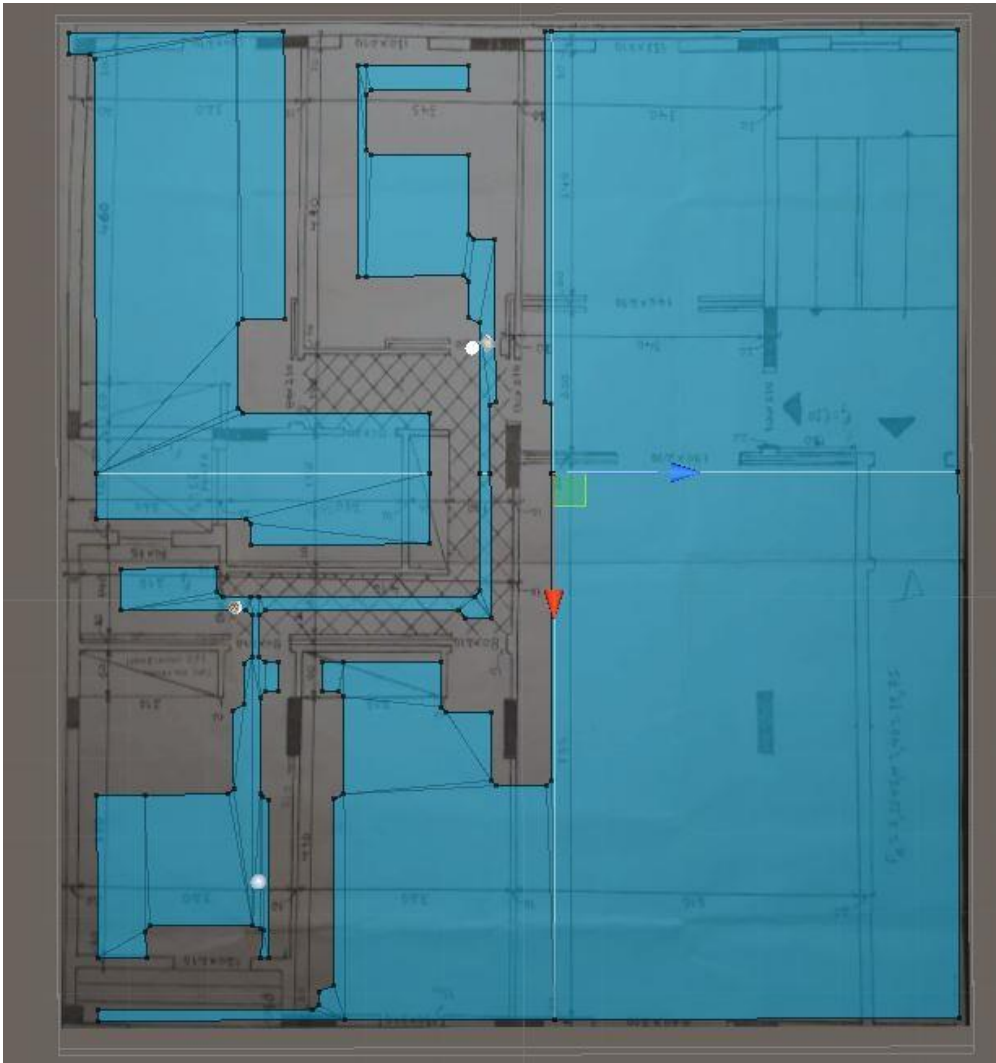
Πρώτη μέθοδος που καλείται και στα περισσότερα script είναι η Start() που εκτελεί τα πρώτα κομμάτια κώδικα και χρησιμεύει για αρχικοποίηση μεταβλητών. Μέσα στην Start() αρχικοποιούμε ένα αντικείμενο τύπου Vector3 το οποίο θα χρησιμοποιήσουμε για να λαμβάνουμε τα δεδομένα απο την κάμερα AR. Στην συνέχεια η μέθοδος Update() είναι υπεύθυνη για κάθε frame αρχικά να λαμβάνει μέσω του Session.Status και Session.Tracking αν το Session που είναι ενεργό έχει είσοδο δεδομένων απο τον αισθητήρα κάμερας και

παρουσιάζει το αντίστοιχο μήνυμα. Το μήνυμα ως αρχικοποίηση περιέχει το κείμενο του της μεταβλητής `NotargetT`. Αν το `Session.Tracking` επιστρέψει τιμή η οποία δηλώνει ότι δεν συμβαίνει κάποιου είδους αναγνώριση απο τον αισθητήρα, είτε λόγω έλλειψης φωτισμού στον χώρο είτε εμποδίων που είναι τοποθετημένα υπερβολικά κοντά στον αισθητήρα, τότε για την διάρκεια που συμβαίνει αυτό, το UI του Unity ειδοποιεί τον χρήστη ότι η αναγνώριση έχει χαθεί με το κατάλληλο μήνυμα.

Προκειμένου να λάβουμε τα δεδομένα της AR κάμερας χρησιμοποιούμε την κλάση `Frame` η οποία περιέχει ένα στιγμιότυπα απο το ενεργό αντικείμενο `First Person Camera` για κάθε frame που είναι ενεργό και εντοπίζει κάποια εικόνα απο το περιβάλλον. Εσωτερικά της κλάσης `frame` υπάρχει η μέθοδος `Pose` η οποία παρέχει την πόζα της συσκευής AR και το μεταφράζει με την βοήθεια του `Vector3 position` σε συντεταγμένες του Unity, `x y z`. Καθώς διατηρούμε σε κάθε frame τις τιμές του κινητού στον χώρο και στο Unity στο `m_prevARPosePosition` με την μεταβλητή τύπου `GameObject cameraTraget` που αποτελεί αναφορά στο αντικείμενο `Player` κάνουμε αντιστοιχία των συντεταγμένων. Με αυτό τον τρόπο σε κάθε καρέ του Unity όσο ενημερώνεται ο αισθητήρας αντιστοίχως θα ενημερώνεται και ο `Player` στην σκηνή.

NavMesh

Το `Navigation Mesh` η αλλιώς όπως αποκαλείται `NavMesh` που αναφέρθηκε προηγουμένως είναι μια διαδικασία του Unity να δημιουργεί ένα στρώμα πάνω στις επιφάνειες των τρισδιάστατων σχημάτων που περιέχει στην σκηνή το οποίο χρησιμοποιείται για τον υπολογισμό διαδρομών και “έξυπνης” κίνησης των `agent` στην σκηνή. Το Unity API προσφέρει διαδικασίες κλάσεις και μεθόδους για την δημιουργία `navigation mesh` και έλεγχο των `agent` σε αυτό. Στο `NavMesh` μπορούν να συμπεριλαμβάνονται ή όχι αντικείμενα με δύο τρόπους. Απενεργοποιώντας αντικείμενα που δεν επιθυμούμε να έχουν `NavMesh` ή ορίζοντας συγκεκριμένα στρώματα. Με την επιλογή `Bake` πραγματοποιείται η δημιουργία κατασκευής `NavMesh` ενώ με την `Clear` διαγράφεται (εικόνα 5.2).

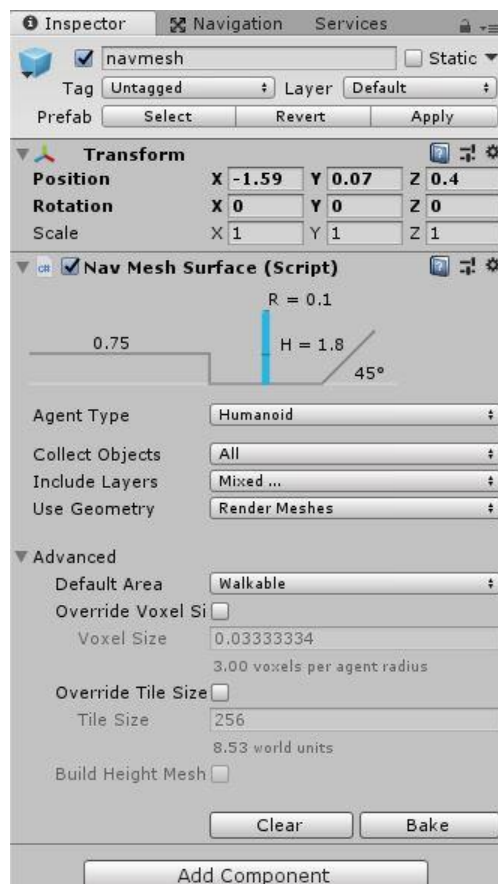


Εικόνα 5.2 Ο χάρτης μετά την χρήση Nav Mesh.

Στην εργασία έχουν χρησιμοποιηθεί τα script `NavMeshSurface` και `NavMeshModifier` των οποίων το πρώτο είναι υπεύθυνο για την δημιουργία μιας συνέχειας απο πολύγωνα με την μορφή mesh το οποίο προορίζεται για την αναγνώριση κόμβων μέσα σε αυτό. Για να δημιουργηθεί ένα NavMesh χρειάζεται να υπάρχουν στην σκηνή σχήματα(τρισδιάστατα αντικείμενα) τα οποία τα ίδια εφόσον έχουν mesh θα θεωρούνται υπολογίσιμα απο τον αλγόριθμο εσωτερικά του `NavMeshSurface`.

Εσωτερικά του `NavMeshSurface` υπάρχουν οι μέθοδοι `BuildNavMesh()` και `UpdateNavMesh()` οι οποίοι αποτελούν μέρος της δημιουργίας του πλέγματος μάζας στην

σκηνή. Ο τρόπος που υπολογίζονται τα πολύγωνα και οι σχέσεις μεταξύ τους αφορούν τα ενεργά τρισδιάστατα αντικείμενα στην σκηνή ενώ αντικείμενα στην ιεραρχία όπως οι εικόνες και τα κουμπιά που περιέχονται στο UI αγνοούνται. Για παράδειγμα αν υπάρχουν δύο τετράγωνα αντικείμενα στην σκηνή τα οποία είναι απομακρυσμένα και δεν έχουν κανέναν συσχετισμό όσο αναφορά τα πολύγωνα που τα σχηματίζουν ή η μεταξύ τους απόσταση είναι μεγάλη αλλά εξακολουθούν και τα δύο να είναι ενεργά τότε ο αλγόριθμος NavMeshSurface θα δημιουργήσει όλα τα κατάλληλα πολύγωνα που θα καλύψουν την περιοχή πάνω από τα τετράγωνα καθώς και την συνολική περιοχή σε ευθεία που έχει η μεταξύ τους απόσταση. Κατά συνέπεια, είναι κατανοητό ότι αν η τοποθέτηση των αντικειμένων στην ενεργή σκηνή δεν είναι προσεκτική, το NavMesh θα κατασκευαστεί με πολλά παραπάνω πολύγωνα τα οποία δεν θα ακολουθούν την ροή του προγράμματος. Το script NavMeshSurface ακόμα, περιέχει κατάλληλες ρυθμίσεις για τον τύπο του πράκτορα που θα χρησιμοποιηθεί, το ποια στρώματα (layers) θα είναι αυτά που θα υπολογιστούν κατά την διάρκεια δημιουργίας του NavMesh καθώς και αν η περιοχή από τα πολύγωνα, δηλαδή το mesh, θα είναι “walkable”, άρα και περιοχή που μπορεί να αναγνωρίσει ο πράκτορας (εικόνα 5.3).



Εικόνα 5.3 Ρυθμίσεις Αρχείου Nav Mesh Surface.

Στο πρώτο τμήμα ρυθμίσεων αναγνώρισης πράκτορα στο πεδίο Inspector υπάρχει η επιλογή ύψους πλάτους κλίσης, που δηλώνει την κλίση την οποία ένας πράκτορας έχει πρόσβαση (αν η επιλογή είναι 45, δηλαδή 45 μοίρες, ο πράκτορας έχει πρόσβαση σε αντικείμενα που βρίσκονται σε διαφορετικό ύψος στον άξονα Y και σχηματίζουν γωνία ίση ή μικρότερη των 45 μοιρών) και το ύψος (σκαλοπάτι) που μπορεί να “ανέβει”. Οι επιλογές αυτές έχουν σημασία καθώς όταν ο πράκτορας υπολογίζει το μονοπάτι με σκοπό έναν προορισμό θα βρει την βέλτιστη διαδρομή. Αυτό σημαίνει ότι δεν προσωμειώνει μία ρεαλιστική διαδρομή ενός ατόμου, καθώς μετακινείται πολύ κοντά σε τοιχώματα και σε γωνίες και επομένως αν η διάμετρος του πράκτορα έχει για παράδειγμα αρχικοποιηθεί ως μεγαλύτερη από κάποιο πέρασμα, ο πράκτορας δεν θα μπορεί να ολοκληρώσει την διαδρομή.

Το δεύτερο script κατηγορίας NavMesh που έχει χρησιμοποιηθεί στην εργασία είναι το NavMeshModifier υπεύθυνο για ενέργειες που θα κάνει το NavMesh ως προς την κατασκευή του. Τα δύο αυτά script συνεργάζονται καθώς τα αντικείμενα που έχουν το script NavMeshModifier ως ιδιότητα στο inspector δίνουν την επιλογή να μην συμπεριληφθούν στην επίστρωση απο πολύγωνα του NavMesh. Αυτό σημαίνει ότι το script NavMeshSurface θα σχηματίσει πολύγωνα γύρω απο τα αντικείμενα με εφαρμοσμένο το script NavMeshModifier και ενεργοποιημένη την επιλογή Override Area με Area Type “Not Walkable”. Καθώς το mesh που κατασκευάζεται έχει την μορφή “στρώματος” πάνω απο τα αντικείμενα της σκηνής αν ένα αντικείμενο που θέλουμε να γίνει override δεν το έχουμε επιλέξει κατάλληλο για υπολογισμό στο mesh, το script θα αγνοήσει και το σχήμα και την περιοχή που βρίσκεται κάτω απο αυτό.

Οι αλγόριθμοι αναζήτησης κόμβων που χρησιμοποιεί το Unity είναι τροποποιημένοι A* αλγόριθμοι και συμπεριφέρονται αντιστοίχως.

Η διαδικασία δημιουργίας NavMesh στην σκηνή έχει ως εξής. Στο πεδίο Hierarchy - Scene έχουν τοποθετηθεί :

mapCamera

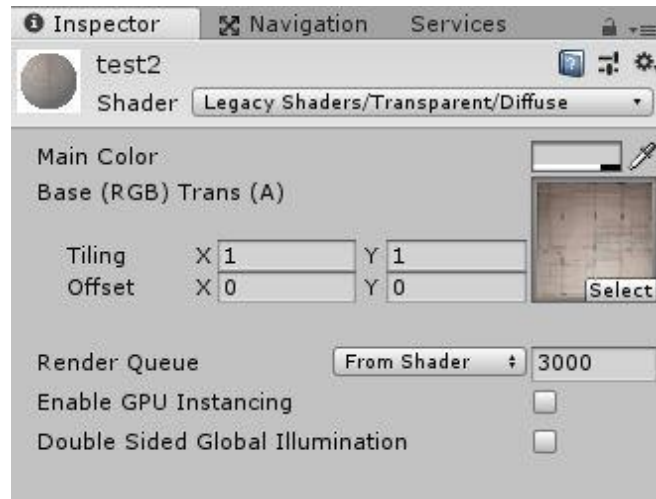
Το αντικείμενο mapCamera είναι ένα από τα τρία αντικείμενα τύπου “Camera” της ιεραρχίας και αντίστοιχα της σκηνής το οποίο είναι υπεύθυνο για την προβολή του χάρτη στον χρήστη προκειμένου να ορίσει μία διαδρομή. Ως αντικείμενο κάμερας αξίζει κυρίως να αναφερθεί ότι ο τρόπος προβολής που έχει επιλεγεί είναι ο Perspective, δηλαδή προοπτικός (εικόνα 5.4).



Εικόνα 5.4 Αντικείμενο mapCamera, η κάμερα υπεύθυνη για την προβολή του χάρτη στην επιλογή διαδρομών.

Ο χάρτης map

Στο πεδίο ιεραρχίας υπάρχει ένα αντικείμενο με το όνομα “map”, το οποίο αποτελεί τον χάρτη που έχει χρησιμοποιηθεί στην εφαρμογή (εικόνα 5.5). Ο χάρτης αποτελείται από ένα plane του Unity, δηλαδή ένα αντικείμενο με mesh στο σχήμα του επιπέδου το οποίο έχει διαστάσεις 1:1 με τον πραγματικό κόσμο που αναπαριστά. Το στοιχείο Mesh Renderer του αντικειμένου map έχει αρχικοποιηθεί ως μη ενεργό, δηλαδή δεν είναι ορατό στην σκηνή, λόγω του ότι χρησιμοποιείται ως σημείο προβολής και για το αντικείμενο mapCamera και για την κάμερα επαυξημένης πραγματικότητας First Person Camera. Όταν η κάμερα AR απενεργοποιείται το mesh του χάρτη ενεργοποιείται και αντίθετα.



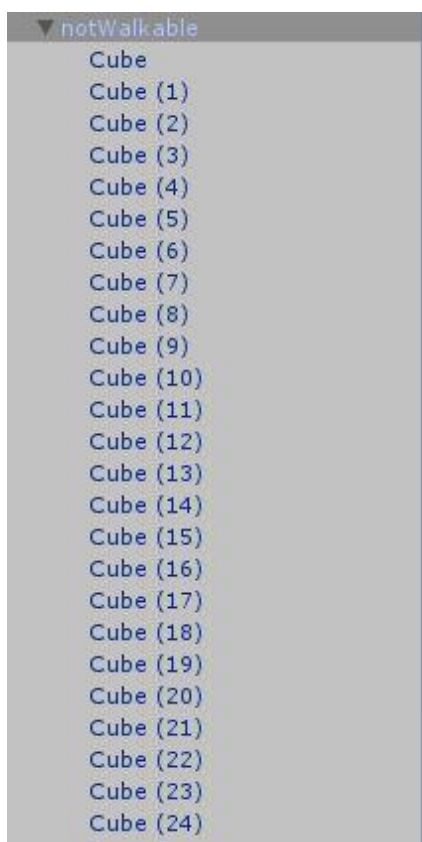
Εικόνα 5.5 Ρυθμίσεις αντικειμένου χάρτη.

Ακόμα ο χάρτης περιέχει ένα script, το pinchZoom το οποίο είναι υπεύθυνο για τις ενέργειες, zoom in και out, δηλαδή η απομάκρυνση της κάμερας Unity mapCamera που είναι προσανατολισμένη προς τον χάρτη. Αρχικά, με την χρήση εισόδου απο την οθόνη με την Input.GetTouch λαμβάνει το πλήθος εισόδου απο την οθόνη, δηλαδή πόσα δάκτυλα έχει ο χρήστης στην οθόνη του και την θέση τους σε αυτήν. Επίσης περιέχει την μεταβλητή Plane τύπου Plane με την οποία θα διασταυρωθεί η πληροφορία θέσης εισόδου και θα πραγματοποιηθεί η αντίστοιχη ενέργεια. Τέλος για την εφαρμογή αυτής της ενέργειας έχει χρησιμοποιηθεί το σύστημα RayCasting του Unity με το οποίο η μηχανή ανάλογα την είσοδο που έχει ορίσει ο χρήστης, δημιουργεί “ακτίνες” οι οποίες “ταξιδεύουν” εσωτερικά της σκηνής και επιστρέφουν τιμές απο τα αντικείμενα που έχουν έρθει σε επαφή. Πιο απλοϊκά, αν ο χρήστης ακουμπήσει την οθόνη με δύο δάκτυλα το Unity θα καταγράψει την θέση τους και θα στείλει την πληροφορία απο το αντικείμενο της κάμερας προς το αντικείμενο του χάρτη. Αν ο χρήστης απομακρύνει τα δάκτυλα μεταξύ τους, η κάμερα που είναι ορισμένη στο αντικείμενο map θα απομακρυνθεί.

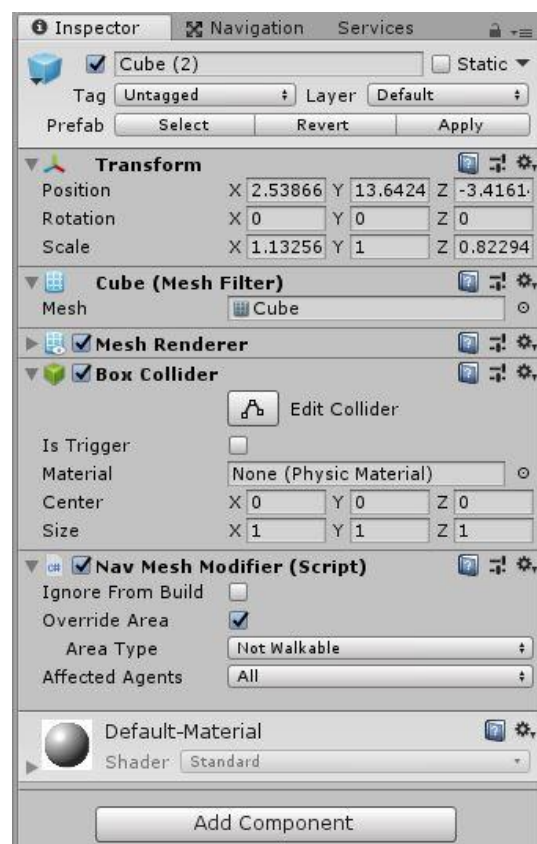
Επιπρόσθετα πρέπει να αναφερθεί ότι στον χάρτη είναι τοποθετημένα τα αντικείμενα Spot_Office Spot_Cafeteria και Spot_Classroom, τα οποία είναι απλοί δίσκοι του Unity και δηλώνουν τις εισόδους των δωματίων στον χώρο. Η χρήση τους όμως λαμβάνει μέρος στο αντικείμενο DecoyPlayer και qrCam.

NotWalkable

Το αντικείμενο NotWalkable είναι ένα κενό αντικείμενο που περιέχει εικοσιτέσσερα (24) παιδιά από κύβους προσαρμοσμένους κατάλληλα στον χάρτη (εικόνα 5.6). Ουσιαστικά αντιπροσωπεύουν τους τοίχους του χώρου ενώ το κάθε ένα από αυτά περιέχει το script NavMeshModifier και έχει ενεργοποιημένη την επιλογή Override Area σε συνδυασμό με τον τύπο περιοχής "Not Walkable" (εικόνα 5.7). Επίσης η κλίμακα των εικονικών τοίχων είναι μεγαλύτερη από αυτή της πραγματικότητας, για πιο ρεαλιστικό αποτέλεσμα στην λειτουργία του πράκτορα (Nav Mesh Agent).

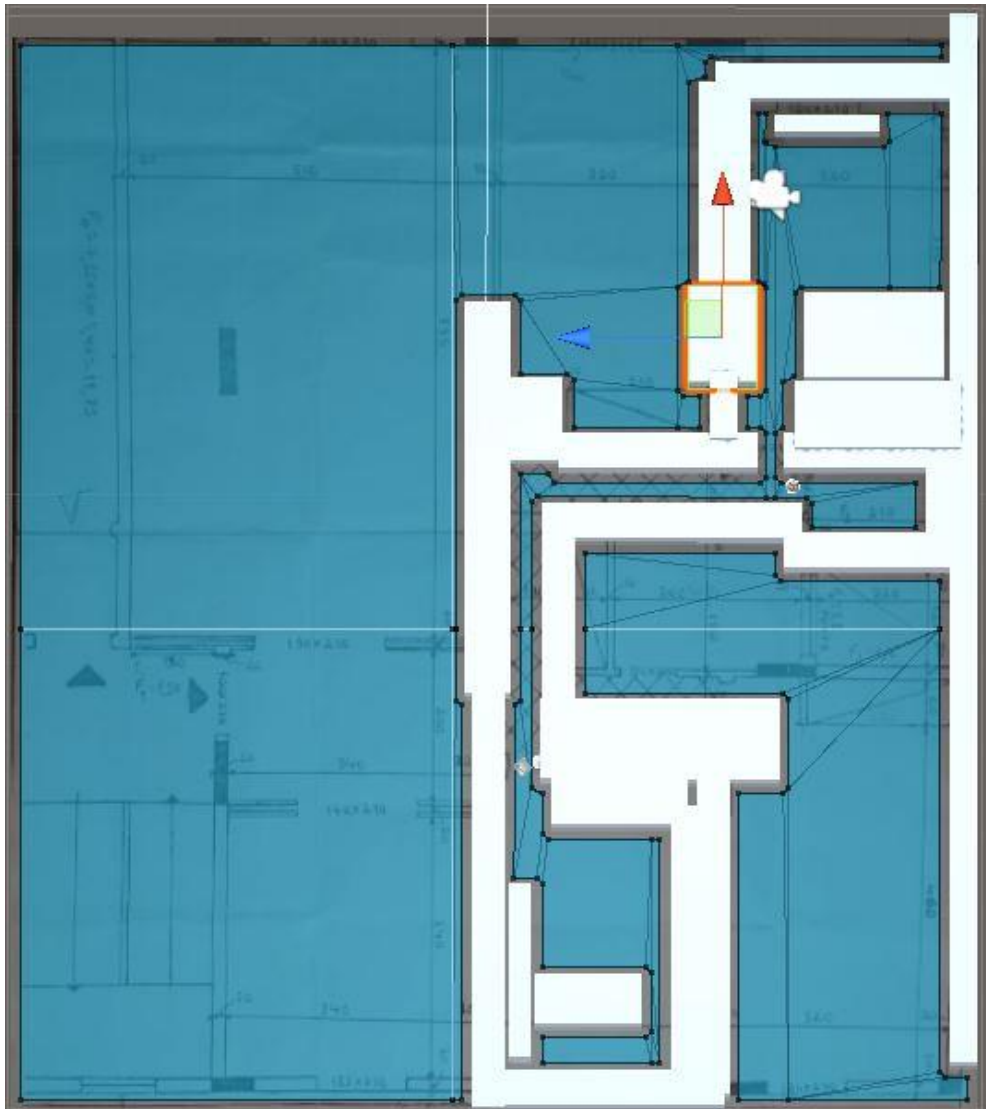


Εικόνα 5.6 Αντικείμενο notWalkable αποτελούμενο από αριθμό αντικειμένων τύπου "κύβος" διαμορφωμένα ειδικά στον χάρτη.



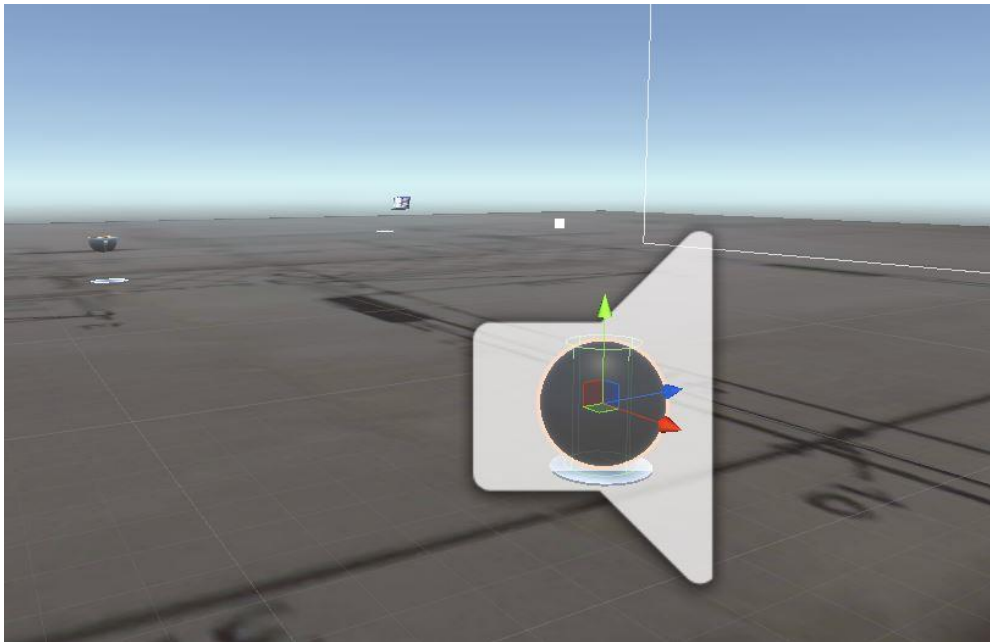
Εικόνα 5.7 Ρυθμίσεις σε ένα από τα πανομοιότυπα αντικείμενα Cube.

Με όλα τα παραπάνω η διαδικασία NavMesh είναι αντιληπτή συνολικά και μπορεί να εξηγηθεί πιο απλά. Αρχικά δημιουργούμε τον χάρτη και τους τοίχους ή ότι αντικείμενο δεν θέλουμε να συμπεριλαμβάνει το NavMesh στην σκηνή του Unity. Έπειτα με την ενέργεια Bake δημιουργούμε το NavMesh του χάρτη και η εφαρμογή περιμένει είσοδο. Ο πράκτορας απο δω και στο εξής έχει την δυνατότητα να κινηθεί πάνω στο αντικείμενο “map” με τους αντίστοιχους περιορισμούς (εικόνα 5.8).



Εικόνα 5.8 Ο χάρτης της εργασίας με εμφανή τον χώρο που απαγορεύεται να προχωρήσει ο πράκτορας.

Το αντικείμενο DecoyPlayer



Εικόνα 5.9 Το σφαιρικό αντικείμενο DecoyPlayer.

DecoyPlayer (εικόνα 5.9) έχει ονομαστεί το αντικείμενο το οποίο έχουμε χρησιμοποιήσει για τις εξής τρεις ενέργειες:

- κίνηση του πράκτορα στον χάρτη
- δημιουργία δεικτών, βέλη που δείχνουν την διαδρομή στον χρήστη
- ηχητικές εντολές διαδρομής

Αρχικά το αντικείμενο είναι πανομοιότυπο του Player με τις εξής προσθήκες. Περιέχει NavMeshAgent ένα script, το DecoyAgent και AudioSource, ένα component που είναι απαραίτητο για την αναπαραγωγή αρχείων ήχου στην σκηνή του Unity (εικόνα 6.0).

1. Στην μέθοδο Start αρχικοποιείται η μεταβλητή onceStill η οποία δηλώνει ότι το αντικείμενο αρχικά είναι ακίνητο. Εσωτερικά της LateUpdate έχουν ανανεώνουμε σε κάθε frame την θέση του DecoyPlayer και την συγκρίνουμε με την τιμή του προηγούμενου frame (στιγμιότυπου). Η κίνηση του αντικειμένου πραγματοποιείται απο την ιδιότητα (component) NavMeshAgent σε συνδυασμό με την κλάση Ray. Αν ο agent δεν κινείται και η “ακτίνα” έχει είσοδο από την οθόνη της συσκευής τότε με

την χρήση της μεθόδου `SetDestination` ο πράκτορας θέτει προορισμό το σημείο αυτό. Το βήμα, η ταχύτητα η ακτίνα και η επιτάχυνση του πράκτορα ελέγχονται απο το component `NavMeshAgent`.

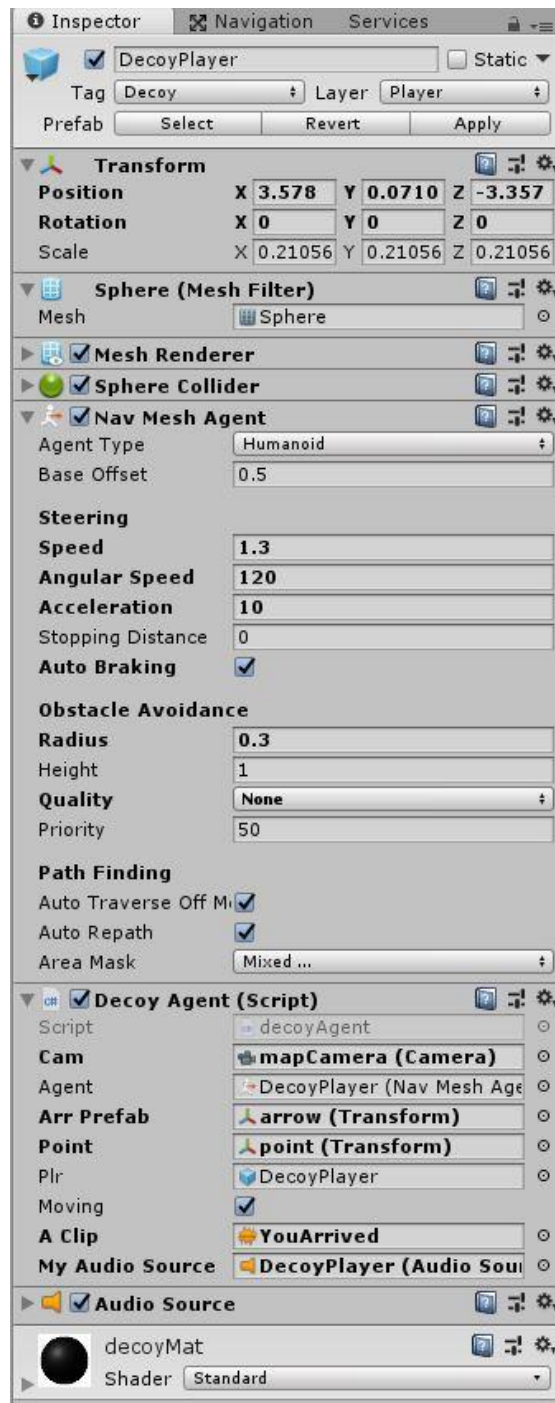
2. Η λειτουργία δημιουργίας απο βέλη στον χάρτη πραγματοποιείται απο την μέθοδο `spawnArrows()` η οποία δημιουργεί βέλη με βάση την θέση του πράκτορα στον χάρτη κάθε ένα δευτερόλεπτο και εφόσον κινείται, χρόνο που έχουμε ορίσει με την μεταβλητή `time` και χρησιμοποιούμε εσωτερικά της `LateUpdate()`. Η δημιουργία απο βέλη καθώς και απο σημεία “point” πραγματοποιείται με την μέθοδο `Instantiate()` και η καταστροφή τους στην `DestroyObjects()` με την χρήση `Destroy()`.

Τα βέλη που έχουν δημιουργηθεί στον χάρτη με την `Instantiate()` δεν έχουν αρχικοποιηθεί, επομένως δεν είναι χρηστικά αφού δεν έχουν προσανατολισμό και δεν μπορούν να χρησιμοποιηθούν για να υποδείξουμε την διαδρομή σωστά στον χρήστη. Τα αντικείμενα “arrow” είναι αντικείμενα των οποίων το mesh μεταφράζεται απο το Unity δηλαδή είναι ορατά ακόμα και όταν η κάμερα `mapCamera` είναι ανενεργή στην σκηνή, σε σχέση με το αντικείμενο `map` του οποίου το mesh είναι ενεργό όταν το `mapCamera` είναι ενεργό.

Το παραπάνω πρόβλημα αντιμετωπίζεται εσωτερικά της μεθόδου `pointAnchor()` σε συνδυασμό με το script “ShowArr” στο αντικείμενο `Player`. Αρχικά, στην μέθοδο `pointAnchor()` γίνεται εντοπισμός των αντικειμένων `arrow` και `point` με βάση το όνομα που τους δίνει το Unity στην σκηνή ως κλώνους. Έπειτα για την καλύτερη διαχείρισή τους γίνεται μετανομασία με αύξουσα αριθμητική σειρά που δημιουργήθηκαν. Στο σημείο αυτό με την μέθοδο `LookAt()` μέσα σε επαναληπτική δομή και ως είσοδο την θέση κάθε βέλος απο την σκηνή πραγματοποιείται ο συσχετισμός μεταξύ τους. Στην τελευταία δομή επανάληψης απενεργοποιούμε όλα τα mesh απο τα αντικείμενα `arrow` εκτός απο τα δύο πρώτα. Παράλληλα με την μέθοδο `pointAnchor()` στο αντικείμενο `Player` εσωτερικά του script `showArr()` υπάρχει η μέθοδος `OnTriggerEnter()` η οποία αποτελεί μέθοδο του Unity και ενεργοποιείται όταν δύο αντικείμενα περιέχουν το component “collider” και δηλώνει την σύγκρουση δύο αντικειμένων. Πιο ουσιαστικά δηλώνει ότι αν δύο αντικείμενα είναι τοποθετημένα σε μία θέση και το ένα βρεθεί το πολύ σε μία συγκεκριμένη απόσταση απο το δεύτερο θα πραγματοποιηθεί μία ενέργεια. Επίσης να σημειωθεί πως τα

αντικείμενα τέτοιου είδους σχέσης δεν μπορούν να έχουν φυσικές δυνάμεις αλληλεπίδρασης. Επομένως με την χρήση αυτής της μεθόδου, καθώς το αντικείμενο Player κινείται στον χώρο, αν έρθει σε επαφή με το κατάλληλο point που βρίσκεται κάτω από το αντίστοιχο απενεργοποιημένο βέλος στην σκηνή, ενεργοποιεί το βέλος. Η αιτία πίσω από την διαδικασία είναι μία δημιουργία ρεαλιστικής εμπειρίας για τον χρήστη αφού τα βέλη δεν θα φαίνονται “μέσα” από τους τοίχους του πραγματικού κόσμου στην κάμερα επαυξημένης πραγματικότητας.

3. Ως τρίτη διαδικασία είναι αυτή της δημιουργίας ηχητικών αντικειμένων με σκοπό την καλύτερη πλοήγηση στον χώρο. Στην εργασία υπάρχει ο φάκελος AudioFiles που περιέχει αρχεία με ξεχωριστές εντολές ανάλογα τον προσανατολισμό που βρίσκεται ο χρήστης την δεδομένη στιγμή. Η διαδικασία αυτή δημιουργείται εσωτερικά στο αντικείμενο arrow στο script audioTrigger ενώ στο αντικείμενο DecoyPlayer πραγματοποιείται μόνο η τελική εντολή αν η συσκευή έχει φτάσει στον προορισμό της. Ο υπολογισμός πραγματοποιείται εσωτερικά της LateUpdate() σε κάθε frame με την μέτρηση απόστασης μεταξύ των αντικειμένων DecoyPlayer και Player στην σκηνή.

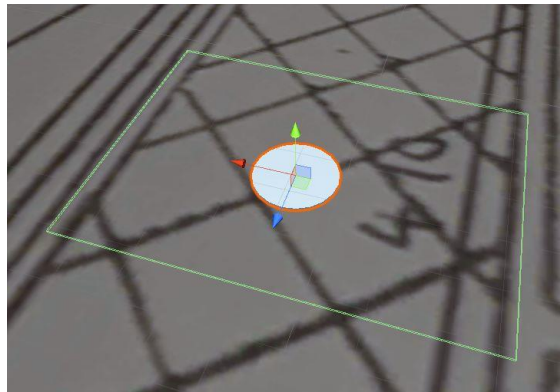


Εικόνα 6.0 Ρυθμίσεις του αντικειμένου DecoyPlayer.

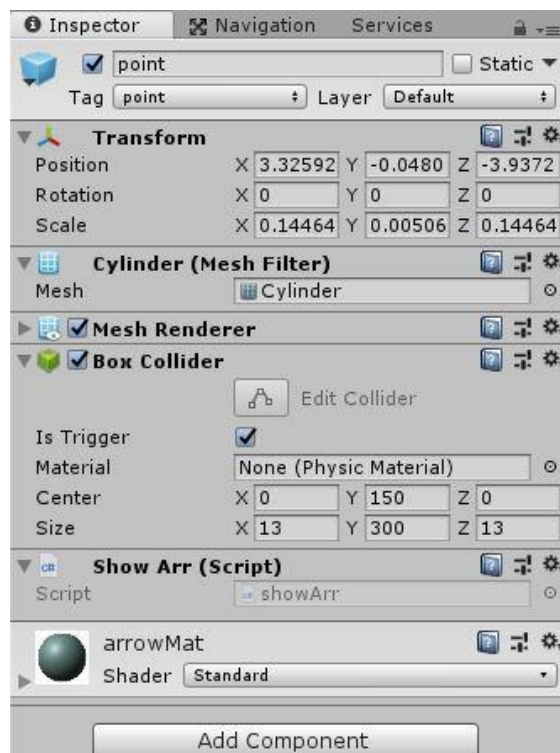
Όταν ο χρήστης φτάσει στο προορισμό του τα βελάκια καταστρέφονται και μπορεί να επιλέξει κάποια άλλη ενέργεια.

Αντικείμενο point

Είναι ένα αντικείμενο στο σχήμα δίσκου (εικόνα 6.1) το οποίο έχει χρησιμοποιηθεί για δύο σκοπούς. Ο πρώτος είναι αποκλειστικά και μόνο ως σημάδια για τον κάθε προορισμό στον χάρτη ενώ ο δεύτερος λειτουργεί ως αναφορά σε κάποια αρχεία script για την ενεργοποίηση αντικειμένων arrow στην σκηνή (εικόνα 6.2).

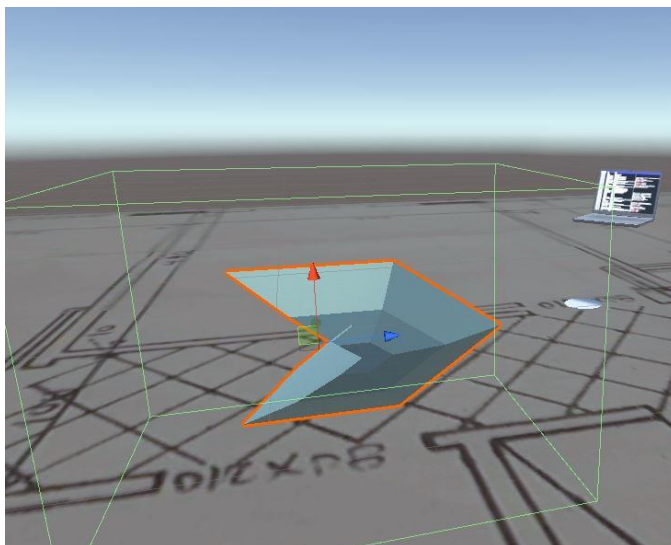


Εικόνα 6.1 Αντικείμενο point. Το λευκό είναι το mesh-μάζα του αντικειμένου ενώ το πράσινο είναι το πεδίο σύγκρουσης.



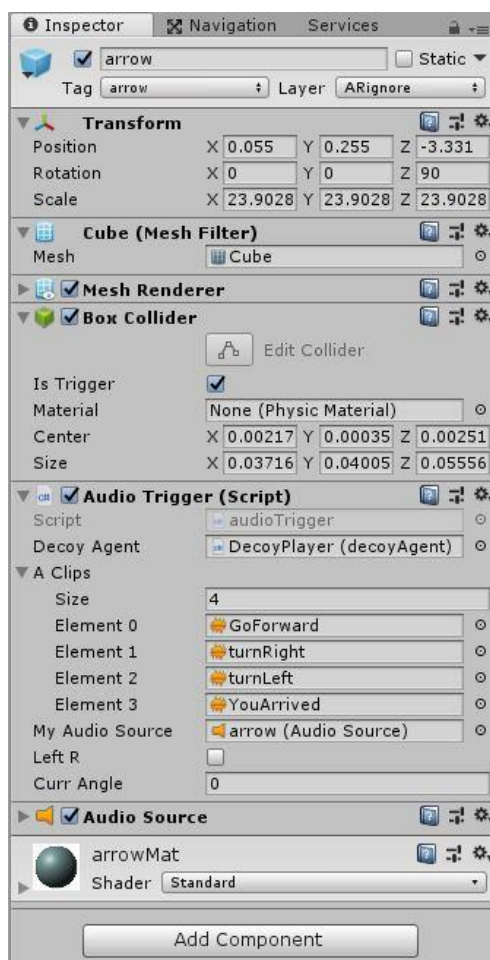
Εικόνα 6.2 Ρυθμίσεις αντικειμένου point.

Αντικείμενο Arrow



Εικόνα 6.3 Αντικείμενο arrow και το πεδίο σύγκρουσής του.

Το αντικείμενο arrow είναι ένα αντικείμενο στο σχήμα βέλους και η γεωμετρία του έχει κατασκευαστεί σε εξωτερικό πρόγραμμα (εικόνα 6.3). Η θέση του στον χώρο δεν έχει σημασία διότι ως prefab δεν υφίσταται στην σκηνή αν δεν δημιουργηθεί από κάποιο άλλο αντικείμενο. Περιέχει component AudioSource το οποίο είναι υπεύθυνο για την αναπαραγωγή ήχου σε τρισδιάστατο χώρο. Το script audioTrigger στο αντικείμενο Arrow είναι υπεύθυνο για την ανάθεση αρχείων ήχου και της σωστής επιλογής της κατάλληλης εντολής ανάλογα με τον προσανατολισμό του βέλους που θα είναι ο χρήστης πιο κοντά (εικόνα 6.4). Λειτουργεί ως εξής. Το script decoyAgent στο αντικείμενο DecoyPlayer δημιουργεί αντικείμενα arrow. Την στιγμή της δημιουργίας τους τα βέλη δεν είναι προσανατολισμένα και για αυτό έχει χρησιμοποιηθεί ο υπολογισμός γωνίας των arrow σε μέθοδο Update() στο αρχείο audioTrigger εσωτερικά των arrow και όχι σε Start(). Εφόσον τα βέλη προσανατολιστούν το script audioTrigger αναμένει το αντικείμενο player που αντιπροσωπεύει τον χρήστη ή την συσκευή μέχρι να εισέλθει στην αντίστοιχη περιοχή που θα ενεργοποιήσει τον κώδικα στην OnTriggerEnter.



Εικόνα 6.4 Ρυθμίσεις αντικειμένου arrow.

Η συσκευή καθώς πλοηγείται στην διαδρομή με σκοπό να φτάσει σε κάποιον προορισμό έρχεται σε επαφή με κάποιο “Collider” ενός από τα βέλη που έχουν δημιουργηθεί στην σκηνή. Σε εκείνο το σημείο το βέλος-κλώνος με το αρχείο audioTrigger υπολογίζει την διαφορά σε μοίρες σε σχέση με τον Player-χρήστη. Αν η γωνία μεταξύ του arrow και του χρήστη είναι μικρότερη από -30 μοίρες, τότε ο χρήστης πρέπει να στρίψει αριστερά, από -30 έως 30 ευθεία και 30 και πάνω να κατευθυνθεί προς τα δεξιά. Οι υπόλοιποι υπολογισμοί στο αρχείο audioTrigger που αφορούν τον υπολογισμό γωνίας σχετίζονται κυρίως με την μηχανή Unity. Πιο συγκεκριμένα, τα αντικείμενα, prefabs, κατά την δημιουργία τους έχουν αρχικοποιημένο προσανατολισμό και καθώς το πρόγραμμα είναι ενεργό, ο μετέπειτα προσανατολισμός τους εξαρτάται από αυτόν. Επομένως χρειάζεται να αρχικοποιηθούν όλοι οι προσανατολισμοί το ίδιο για τον σωστό υπολογισμό γωνίας άρα και αρχείο ήχου.

4.2.3 Κάμερα QR και Επαναπροσδιορισμός Θέσης

QR κάμερα



Εικόνα 6.5 Αντικείμενο qrCanvas, μαζί με τα αρχεία του και παιδί του αποτελεί την κάμερα QR.

Εικόνα Αντικείμενο qrCanvas, μαζί με τα αρχεία του και παιδί του αποτελεί την κάμερα QR (εικόνα 6.5). Το αντικείμενο qrCam είναι ουσιαστικά η τρίτη και η τελευταία κάμερα που έχει χρησιμοποιηθεί στην σκηνή και στην εφαρμογή. Η χρήση της είναι η ανίχνευση και η ανάγνωση συγκεκριμένων κωδικών QR που έχουν τοποθετηθεί απο πριν στον χώρο προκειμένου να διορθωθεί η θέση της συσκευής στον χώρο. Προκειμένου η κάμερα του Unity να μας παρέχει την επιθυμητή λειτουργία ανάγνωσης έχουν δημιουργηθεί δύο script τα οποία είναι το QrCamEnable και το Place Spawn (εικόνα 6.6).



Εικόνα 6.6 Ρυθμίσεις κάμερας QR.

Το QrCamEnable είναι το αρχείο κώδικα το οποίο έχει χρησιμοποιηθεί για την ανάγνωση των κωδικών τύπου QR. Αρχικά για να μπορέσει η κάμερα του Unity να χρησιμοποιήσει τον

αισθητήρα κάμερας της συσκευής το script πρέπει να περιέχει τις μεταβλητές τύπου, RawImage το οποίο είναι ένα “texture” δύο διαστάσεων όπου με την βοήθεια της μεταβλητής AspectRatioFitter δημιουργεί ένα ορθογώνιο παραλληλόγραμμο αναλογικά με την οθόνη της συσκευής. Το ορθογώνιο αυτό μεταφράζει τα δεδομένα που λαμβάνει από τον αισθητήρα που έχει επιλεγθεί. Με την μεταβλητή “cupi” η οποία έχει τιμή ανάλογα με το τι επιστρέφει η μεταβλητή “devices” τύπου WebCamDevice, αποφασίζει για το αν η συσκευή θα χρησιμοποιεί τον εμπρόσθιο αισθητήρα κάμερας ή τον οπίσθιο αισθητήρα.

Ακόμα προκειμένου να αξιοποιηθούν τα δεδομένα που έχουν ληφθεί από την κάμερα έχει χρησιμοποιηθεί η βιβλιοθήκη “ZXing” η οποία είναι ένα project ανοικτού κώδικα για ανάγνωση και δημιουργία κωδικών QR. Μέσω του script αφού έχουμε επιβεβαιώσει ότι υπάρχει διαθέσιμη κάμερα από την συσκευή για λήψη δεδομένων καλείται η μέθοδος scanCode(). Εσωτερικά αυτής έχει χρησιμοποιηθεί η μεταβλητή barcodeReader της κλάσης IbarcodeReader η οποία δημιουργεί ένα στιγμιότυπο. Μέσω του barcodeReader καλείται η μέθοδος Decode() όπου με ορίσματα το texture που έχουμε δημιουργήσει και τις διαστάσεις του πραγματοποιείται η μετάφραση του QR κωδικού. Επίσης να αναφερθεί ότι όσο το αντικείμενο qrCam είναι ενεργό γίνεται scan κάθε ένα δευτερόλεπτο.

Εφόσον η μέθοδος Decode() επιστρέφει κάποιο αποτέλεσμα σταματάμε την λειτουργία της κάμερας και καλούμε την μέθοδο spawnP1r() με όρισμα τον κωδικό που λήφθηκε από αυτήν, με την μορφή αλφαριθμητικού, η οποία είναι τοποθετημένη εσωτερικά του script placeSpawn. Εκεί πραγματοποιείται η σύγκριση της συμβολοσειράς που διαβάστηκε από τον κωδικό QR με τις προκαθορισμένες συμβολοσειρές – αντικείμενα point στον χάρτη και ανάλογα αν υπάρχει η συμβολοσειρά που μόλις σαρώθηκε, τα αντικείμενα player και decoyAgent θα μεταφερθούν αντιστοίχως.

UI

Τα περισσότερα αντικείμενα αλληλεπιδρούν μεταξύ τους με διάφορες σχέσεις, είτε μέσω των script είτε με την ιεραρχία εκτός από τις κάμερες οι οποίες μπορούν να είναι ενεργές ατομικά. Εκτός από τις κάμερες, ένα ακόμα στοιχείο που δεν έχει περιγραφεί και είναι διαφορετικό ανάλογα με την ενεργή κάμερα στην σκηνή είναι η διεπαφή χρήστη που περιέχει κυρίως κουμπιά για την επιλογή συγκεκριμένων ενεργειών.

Το UI για το αντικείμενο ARCoreDevice αποτελείται από δύο κουμπιά και ένα αντικείμενο τύπου text. Το text χρησιμοποιείται για ένδειξη εύρεσης επιφάνειας, ή αν χαθεί κάποια επιφάνεια και δεν μπορεί να αναγνωριστεί ο χώρος αλλάζει η ένδειξη κατάλληλα

μέσω script. Το DestinationB ενεργοποιεί την κάμερα του χάρτη mapCamera καθώς και το mesh renderer του χάρτη με σκοπό την εμφάνισή του mesh του χάρτη στην οθόνη του χρήστη, ενώ το RepositionB απλά ενεργοποιεί την κάμερα QR, qrCam.

Το αντικείμενο mapCamera περιέχει το αντικείμενο switchCamB το οποίο επιστρέφει στην κάμερα επαυξημένης πραγματικότητας ενώ με το κουμπί pathCorrection καταστρέφει όλα τα αντικείμενα arrow στην σκηνή προκειμένου ο χρήστης να διορθώσει το μονοπάτι που θέλει να προσδιορίσει.

Τέλος υπάρχει προαιρετικό κουμπί στην διεπαφή χρήστη της κάμερας QR για επιστροφή στην κάμερα AR σε περίπτωση λάθους του χρήστη.

4.3 Πλοήγηση και Χρήση Εφαρμογής

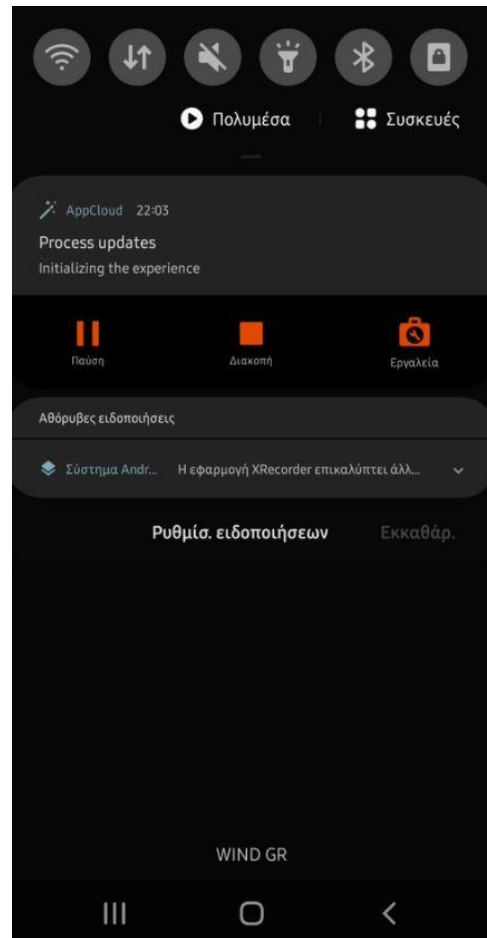
Όταν ο χρήστης ανοίγει το πρόγραμμα κατευθύνεται αμέσως στην κάμερα επαυξημένης πραγματικότητας η οποία είναι υπεύθυνη να αναγνωρίζει τα χαρακτηριστικά του χώρου όπως αναφέραμε με τους αλγόριθμους τύπου SLAM. Η εφαρμογή έχει αρχικοποιημένη μία θέση στον χάρτη, η οποία λόγω της μη χρήσης σύνδεσης στο διαδίκτυο είναι πολύ πιθανό να είναι λάθος. Ανεξαρτήτως αν η θέση αντιπροσωπεύει την πραγματικότητα ή όχι ο χρήστης έχει δύο επιλογές. Η επιλογή του χάρτη του προσφέρει την δυνατότητα να επιλέξει προορισμό, ή να διορθώσει κάποιο μονοπάτι αν πραγματοποιήθηκε κάποια λάθος ενέργεια από τον ίδιο. Η επιλογή της σάρωσης, απενεργοποιεί την κάμερα AR και ενεργοποιεί την κάμερα QR για την ανάγνωση κωδικών QR. Με την επιλογή αυτή ο χρήστης μπορεί μέσω επιλεγμένων σημείων στον χώρο να σαρώσει τον κατάλληλο κωδικό και να γίνει υπολογισμός της θέσης του στον χάρτη. Σε συγκεκριμένα σημεία στον χάρτη επίσης έχουν τοποθετηθεί αντικείμενα που δίνουν κάποια πληροφορία για το δωμάτιο ή τον χώρο που έχει αποφασίσει να κατευθυνθεί ο χρήστης. Τα αντικείμενα αυτά είναι ορατά από την κάμερα AR και είναι ορατά μόνο όταν ο χρήστης πλησιάσει τον αντίστοιχο χώρο. Η πλοήγηση στον χώρο έχει πραγματοποιηθεί με την βοήθεια απεικόνισης τρισδιάστατων αντικειμένων με το σχήμα από βέλη τα οποία κατευθύνουν τον χρήστη στον προορισμό του και δείχνουν πάντα το επόμενο βέλος που ακολουθεί. Επίσης υπάρχει επιπλέον βοήθεια από ακουστικές εντολές για καλύτερη διευκρίνιση της διαδρομής για το τι ενέργεια πρέπει να πραγματοποιήσει ο χρήστης.

Ας πραγματοποιήσουμε αρχικά μία υποθετική σειρά ενεργειών χρήσης της εφαρμογής. Ο χρήστης ανοίγει την εφαρμογή και η κάμερα υπολογίζει και αρχικοποιεί την θέση. Οι εικόνες παρακάτω δείχνουν ότι καθώς ανοίγει η εφαρμογή φαίνεται το υδατογράφημα Unity (εικόνα 6.7) και το διαδίκτυο στο οποίο δεν έχουμε πρόσβαση (εικόνα 6.8). Ακόμα οι δύο επόμενες

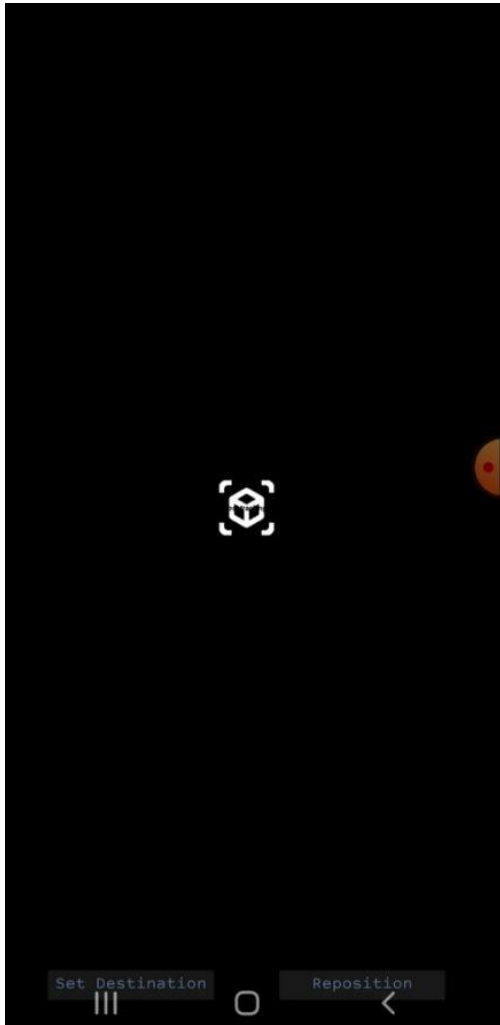
απο αυτές δείχνουν την αρχικοποίηση της εφαρμογής η οποία προσπαθεί να αναγνωρίσει το πρώτο επίπεδο του session (εικόνα 6.9) (εικόνα 7.0).



Εικόνα 6.7 Λογότυπο κατά την έναρξη της εφαρμογής.



Εικόνα 6.8 Η υπηρεσία WIFI είναι απενεργοποιημένη. Δεν υπάρχει βοήθεια WIFI στον υπολογισμό θέσης.

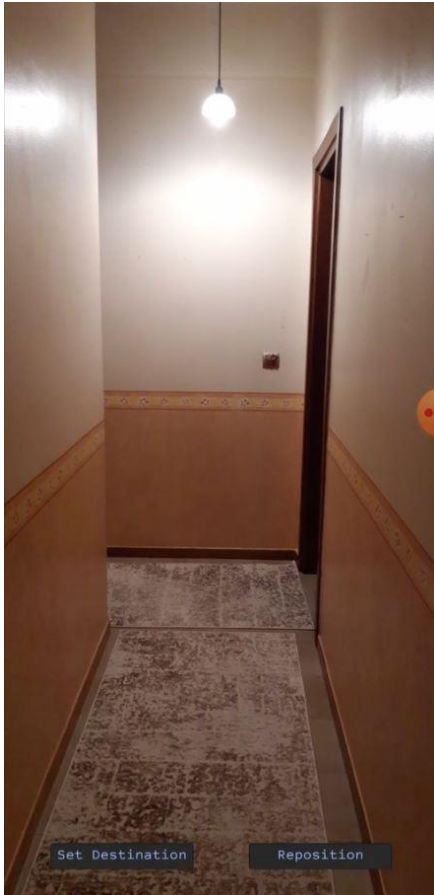


Εικόνα 6.9 Οθόνη αρχικοποίησης.

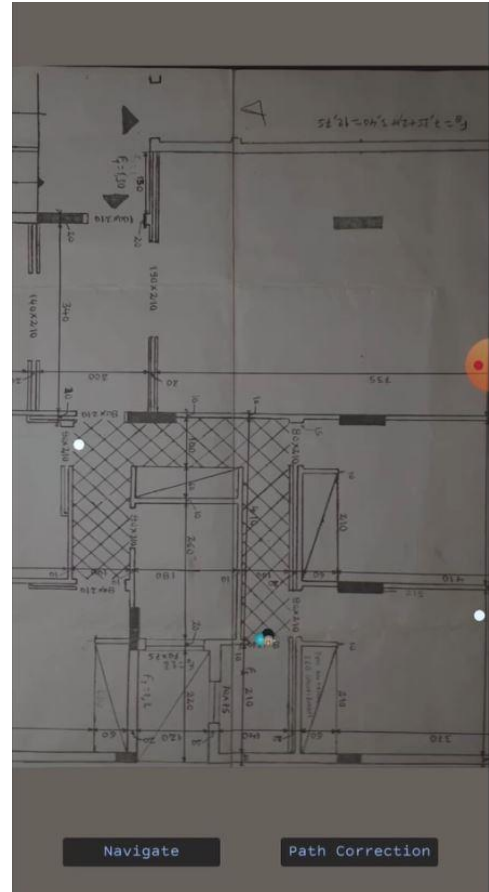


Εικόνα 7.0 Οθόνη αναγνώρισης επιπέδου και αρχικοποίησης.

Πραγματοποιείται αναγνώριση επιπέδου (εικόνα 7.1) και έχει γίνει τοπικοποίηση στον χώρο για την συσκευή.

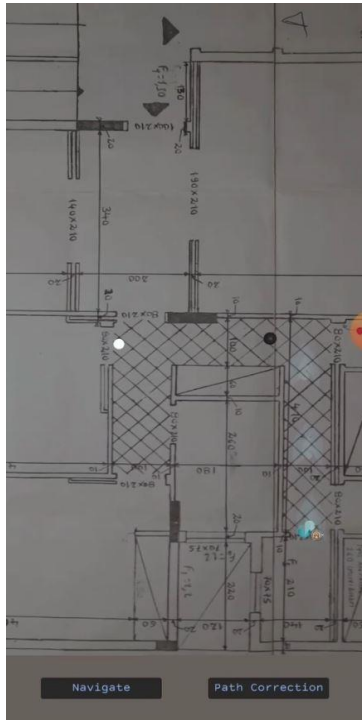


Εικόνα 7.1 Αναγνώριση επιπέδου.

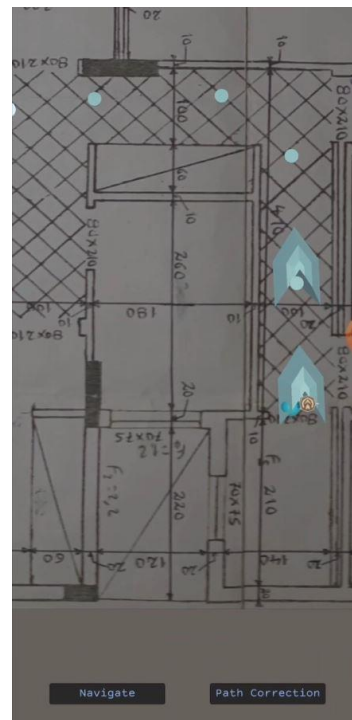


Εικόνα 7.2 Οθόνη κάμερας χάρτη και επιλογή προορισμού.

Ανοίγοντας τον χάρτη με το κουμπί “Set Destination” βλέπουμε μία κάτοψη του χάρτη (εικόνα 7.2) στην οποία μπορεί ο χρήστης να κινηθεί με ευκολία, με απλές κινήσεις των δακτύλων όπως και να τον μεγεθύνει (εικόνα 7.3) (εικόνα 7.4). (Ο χάρτης μπορεί να αντικατασταθεί με οποιοδήποτε αρχείο τύπου εικόνας άμεσα ώστε να είναι ευανάγνωστος από τον χρήστη, ή μπορούν να δημιουργηθούν αντίστοιχα τρισδιάστατα μοντέλα του κτηρίου που λειτουργεί η εφαρμογή).



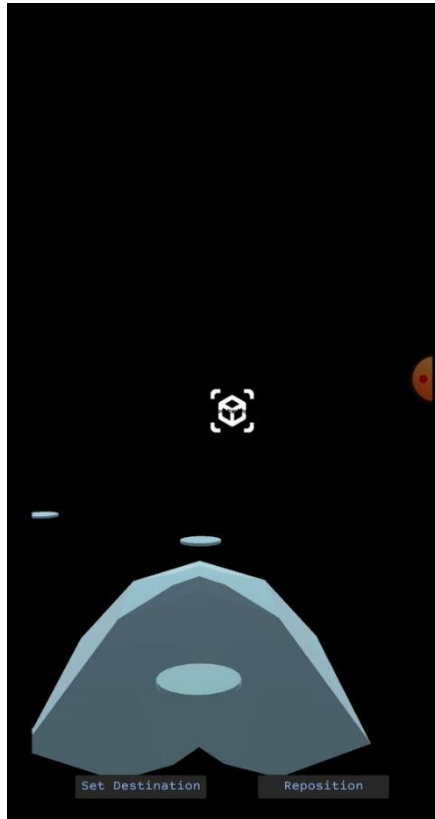
Εικόνα 7.3 Μεγέθυνση χάρτη και διαδικασία υπολογισμού πιο σύντομου μονοπατιού από τον πράκτορα.



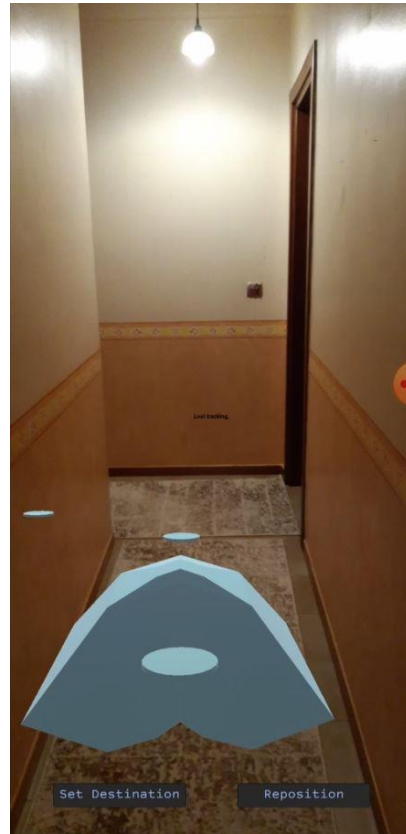
Εικόνα 7.4 Μονοπάτι και εικονιζόμενα βέλη στον χάρτη.

Με το κουμπί “Path Correction” ο χρήστης μπορεί να σβήσει όλα τα ενεργά βέλη και σημεία που βρίσκονται στον χώρο σε περίπτωση που έχει κάνει λάθος και θέλει να θέσει ξανά κάποιον άλλο προορισμό ή θέλει να προσαρμόσει την τοποθεσία του σε κάποια άλλη μέσα στο κτήριο.

Ο χρήστης τώρα θέλοντας να κατευθυνθεί στον προορισμό του επιλέγει το κουμπί “Navigate” το οποίο θα τον επαναφέρει στη κάμερα AR και θα προβάλλει τα βέλη στον χώρο. Η διαδικασία αναγνώρισης επιπέδου στον χώρο ενεργοποιείται ξανά όμως δεν αποτελεί αρχικοποίηση της εφαρμογής όπως πριν (εικόνα 7.5). Απλά χρειάζεται να επαναπροσδιορίσει την συσκευή ανάλογα με τις προηγούμενες εισόδους. Στις παρακάτω εικόνες βλέπουμε αρχικά, καθώς έχουμε κινήσει το κινητό τα βέλη μοιάζουν να υπάρχουν μέσα στον τοίχο με λάθος κατεύθυνση (εικόνα 7.6). Όμως με το που αναγνωριστεί το επίπεδο τα βέλη προσανατολίζονται και τοποθετούνται ακριβώς εκεί που πρέπει (εικόνα 7.7).



Εικόνα 7.5 Αλλαγή κάμερας και επαναφορά προηγούμενης κατάστασης δεδομένων.

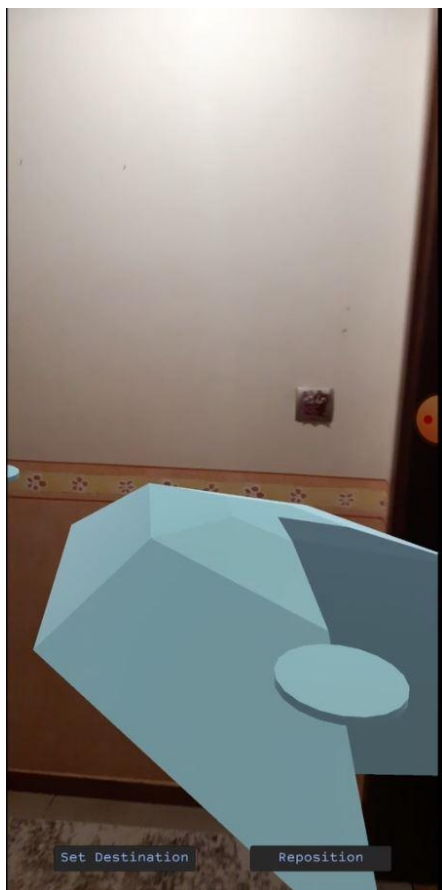


Εικόνα 7.6 Αναγνώριση και Αντιστοίχιση δεδομένων και προσανατολισμός.

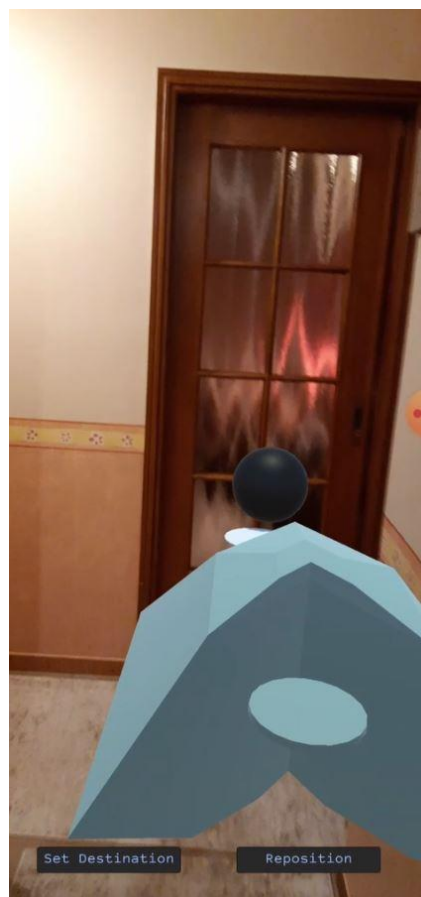


Εικόνα 7.7 Προσανατολισμός βελών.

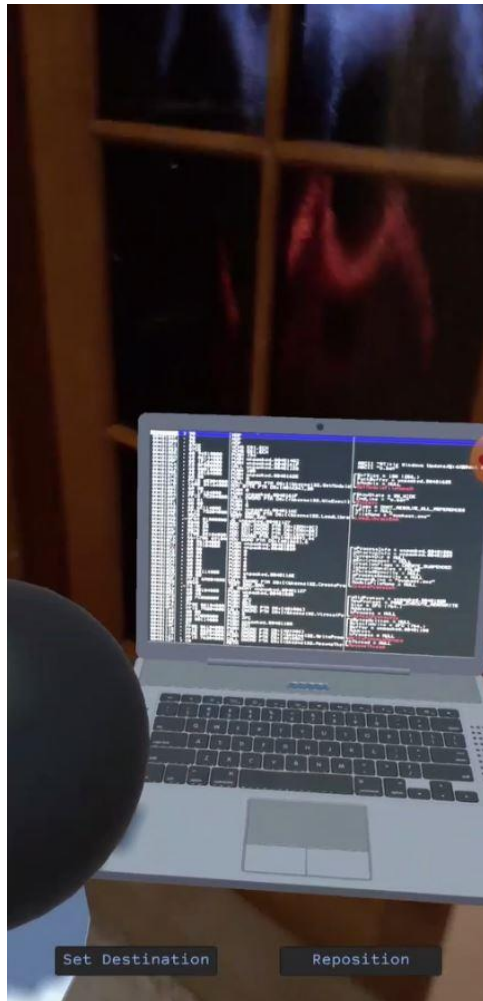
Στις παρακάτω εικόνες ο χρήστης έχει ξεκινήσει να περπατάει προς τον προορισμό του και καθώς περπατάει τα βέλη του εμφανίζονται και υπολογίζοντας την κατεύθυνση του και την κατεύθυνση των βελών ενεργοποιείται μία φωνητική εντολή. Στο συγκεκριμένο παράδειγμα ο εκφωνητής λέει “Go forward” δηλαδή “Πηγαίνατε ευθεία” και “Go left” δηλαδή “Στρίψτε αριστερά” (εικόνα 7.8) (εικόνα 7.9). Καθώς φτάνει σε μία συγκεκριμένη απόσταση από τον agent ο εκφωνητής ανακοινώνει “You have arrived your destination.” δηλαδή “φτάσατε στον προορισμό σας.” και εμφανίζεται το αντίστοιχο εικονίδιο που αντιστοιχεί στον χώρο (εικόνα 8.0).



Εικόνα 7.8 Αριστερή στροφή.



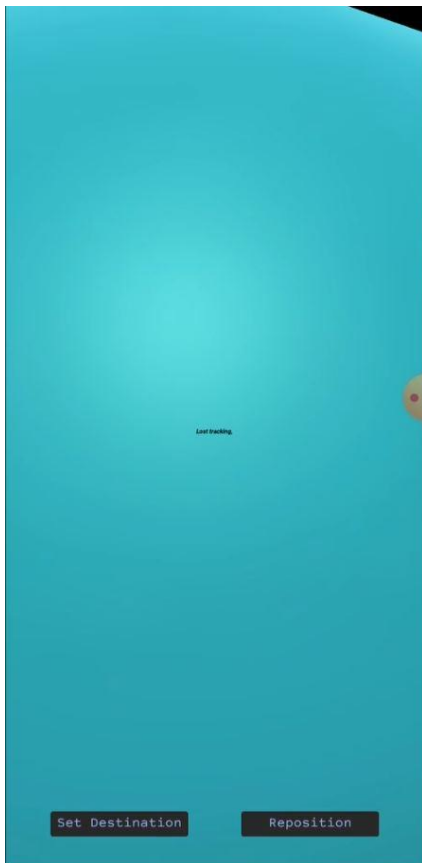
Εικόνα 7.9 Συνέχεια διαδρομής και ορατότητα προορισμού.



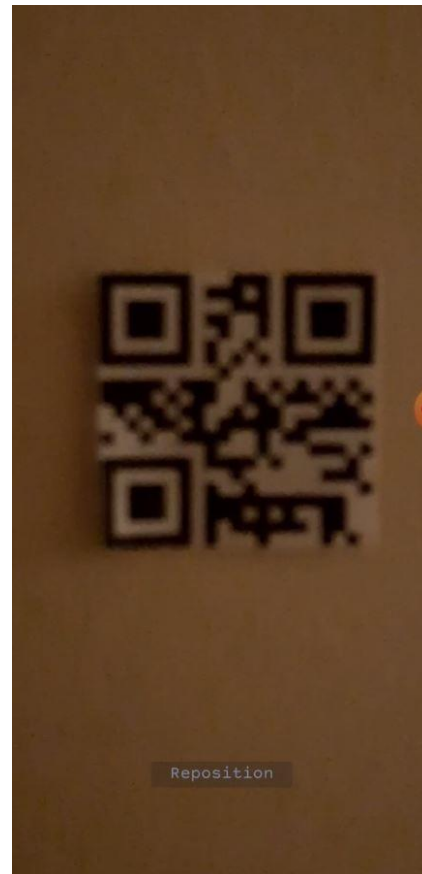
Εικόνα 8.0 Επιτυχία διαδρομής και απεικόνιση ενός φορητού υπολογιστή ως ένδειξη περιεχομένου του προορισμού.

Εφόσον η διαδρομή έχει πραγματοποιηθεί με επιτυχία, τα βέλη έχουν εξαφανιστεί από την σκηνή και δεν φαίνονται στον χρήστη, ο οποίος μπορεί να αποφασίσει να κλείσει την εφαρμογή ή να δημιουργήσει με παρόμοιες ενέργειες μια καινούρια διαδρομή.

Ακόμα ας δούμε τι γίνεται στην περίπτωση που δεν μπορεί η εφαρμογή να αναγνωρίσει κάποιο επίπεδο. Όταν γίνεται αυτό ενεργοποιείται η ένδειξη “Lost Tracking” και τα τρισδιάστατα μοντέλα λειτουργούν με παράξενες συμπεριφορές (εικόνα 8.1). Με το που βρει ξανά η συσκευή κάποιο επίπεδο στον χώρο που έχει αναγνωρίσει, τα μοντέλα επαναπροσανατολίζονται και τοποθετούνται στις θέσεις που ήταν πριν συμβεί αυτό το γεγονός. Ακόμα ο χρήστης μπορεί να επιλέξει για επαναπροσδιορισμό θέσης την σάρωση μέσω QR κωδικό, με το κουμπί “Reposition”, σε περίπτωση που δεν μπορεί να ανακτηθεί κάποια πληροφορία για κάποιο αντικείμενο ή γιατί δεν ξέρει ο ίδιος που βρίσκεται στο κτήριο. Οι κωδικοί QR είναι εκτυπωμένοι και τοποθετημένοι σε αντίστοιχα σημεία στον χώρο οι οποίοι αντιπροσωπεύουν, όπως για παράδειγμα αίθουσα 1, εστιατόριο, εργαστήριο (εικόνα 8.2). Αφού η κάμερα QR κάνει σάρωση του κωδικού ενεργοποιείται η κάμερα AR και ξαναγίνεται προσδιορισμός των αντικειμένων (εικόνα 8.3) (εικόνα 8.4).



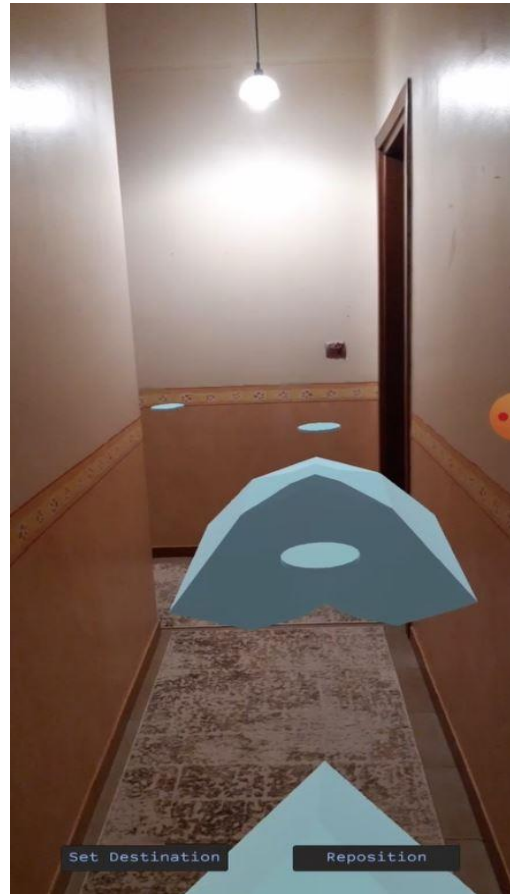
Εικόνα 8.1 Απώλεια δεδομένων ή λάθος κίνηση και αστοχία υπολογισμού θέσης και προσανατολισμού από τον χρήστη.



Εικόνα 8.2 Μεταφορά στην κάμερα QR και αναγνώριση κωδικού QR.



Εικόνα 8.3 Αλλαγή σε κάμερα AR και επαναπροσδιορισμός θέσης.



Εικόνα 8.4 Επιτυχής επαναπροσδιορισμός θέσης.

ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα

Το πρόγραμμα που υλοποιήθηκε αποτελεί ολοκληρωμένη εφαρμογή όσο αναφορά την λειτουργία της. Ο στόχος έχει πραγματοποιηθεί κατά το μεγαλύτερο μέρος. Ο χρήστης μπορεί να περιηγηθεί στον χώρο ακολουθώντας τα κατάλληλα τοποθετημένα βέλη και στο τέλος να φτάσει στον προορισμό που έχει ορίσει ο ίδιος με τις αντίστοιχες ενδείξεις και πληροφορίες.

Το παραπάνω είναι αρκετό ώστε να θεωρηθεί επιτυχημένη η εργασία ως προς τους στόχους που έχουμε θέσει αρχικά. Τα συμπεράσματα που προέκυψαν από τα αποτελέσματα της εργασίας σχετίζονται περισσότερο με βελτιώσεις απόδοσης και αισθητικής.

Η απόδοση της εφαρμογής εξαρτάται σε μεγάλο βαθμό στο SDK της Google που είναι υπεύθυνο για την χρήση αλγορίθμων τύπου S.L.A.M για την αναγνώριση συνεχών εικόνων και την διατήρηση μιας διαδρομής. Επομένως για μια εφαρμογή που στοχεύει σε μικρότερες αποκλίσεις δεδομένων και λιγότερα λάθη χρειάζεται το συγκεκριμένο SDK να ενημερωθεί μελλοντικά με βελτιστοποιήσεις. Εναλλακτικά ο στόχος μπορεί να επιτευχθεί με την χρήση κάποιου άλλου πιο προηγμένου SDK ή ακόμα και την δημιουργία ενός. Επίσης μπορούν να πραγματοποιηθούν και διαφορετικοί τρόποι διαχείρισης εργαλείων του Unity που καλύπτουν τις ατέλειες του SDK. Όπως για παράδειγμα τον συνεχή έλεγχο του χρήστη στον χάρτη και τον επαναπροσανατολισμό του σε συγκεκριμένες γραμμές στον χάρτη.

Το δεύτερο μέρος της εργασίας είναι εξίσου σημαντικό και αφορά τον τρόπο παρουσίασης της εργασίας. Πραγματοποιήθηκαν όλες οι αναγκαίες προσπάθειες για την δημιουργία ενός περιβάλλοντος κατανοητού από τον καθημερινό χρήστη με μερικά ελαττώματα κυρίως στο αισθητικό μέρος τα οποία μπορούν να χρησιμοποιηθούν σε τελικό προϊόν άλλα αποτελούν περισσότερο υποκατάστατα υλικά.

BIBΛΙΟΓΡΑΦΙΑ

1. Caudell, T. P., & Mizell, D. W. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, https://www.researchgate.net/publication/3510119_Augmented_reality_An_application_of_heads-up_display_technology_to_manual_manufacturing_processes.
<https://doi.org/10.1109/hicss.1992.183317>
2. Sutherland, I. E. (1968). A head-mounted three dimensional display. Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I on - AFIPS 68 (Fall, Part I). doi:10.1145/1476589.1476686
3. The Sword of Damocles (virtual reality). (2020, July 02). Retrieved from [https://en.wikipedia.org/wiki/The_Sword_of_Damocles_\(virtual_reality\)](https://en.wikipedia.org/wiki/The_Sword_of_Damocles_(virtual_reality))
4. Krueger, M. W. (1985). "VIDEOPLACE": A Report from the ARTIFICIAL REALITY Laboratory. Leonardo, 18(3), 145. doi:10.2307/1578043 ,http://dada.compart-bremen.de/docUploads/videoplace_paper.pdf
5. Augmented Reality – The Past, The Present and The Future. (n.d.). Retrieved from <https://www.interaction-design.org/literature/article/augmented-reality-the-past-the-present-and-the-future>
6. Grover, D. (2020, February). Augmented reality history, background and philosophy. Retrieved July, 2020, from [https://wiki.mq.edu.au/display/ar/Augmented reality history, background and philosophy](https://wiki.mq.edu.au/display/ar/Augmented+reality+history,+background+and+philosophy)
7. Craig, A. B. (2013). *Understanding augmented reality: Concepts and applications*. Amsterdam: Morgan Kaufmann/Elsevier.
8. Enrique Cano, Pavel Gonzalez, Marcos Maroto and Diego Villegas ,Head-up Displays (HUD) in driving Retrieved from https://www.researchgate.net/publication/301776934_Head-up_displays_in_driving
9. Park, H. S., Park, M. W., Won, K. H., Kim, K., & Jung, S. K. (2013, December 04). In-Vehicle AR-HUD System to Provide Driving-Safety Information. Retrieved from <https://onlinelibrary.wiley.com/doi/full/10.4218/etrij.13.2013.0041>
10. Beal, V. (n.d.). PDA - personal digital assistant. Retrieved January 9, 2020, from <https://www.webopedia.com/TERM/P/PDA.html>.
11. Wagner, D., & Schmalstieg, D. (n.d.). Handheld Augmented Reality Displays. IEEE Virtual Reality Conference (VR 2006). doi: 10.1109/vr.2006.67
12. Bimber, O., & Raskar, R. (2005). Spatial Augmented Reality. doi: 10.1201/b10624

13. Dorrier, J., Dorrier, J., DorrierJason, J., Jason, Singularity Hub, & Singularity Hub. (2020, January 20). Mojo Vision's Augmented Reality Contact Lenses Kick off a Race to AR on Your Eye. Retrieved January 22, 2020, from <https://singularityhub.com/2020/01/17/mojo-visions-augmented-reality-contact-lenses-kick-off-a-race-to-ar-in-your-eye/>
14. (n.d.). Retrieved January 22, 2020, from <https://www.eenewseurope.com/news/14000ppi-microled-display-worlds-densest-only-048mm-across>
15. Baird, W. H. (2008). Magnetometer construction and applications for introductory physics. *American Journal of Physics*, 76(9), 807-811. doi:10.1119/1.2919741, <https://aapt.scitation.org/doi/10.1119/1.2919741>
16. Passaro, V. M., Cuccovillo, A., Vaiani, L., Carlo, M. D., & Campanella, C. E. (2017). Gyroscope Technology and Applications: A Review in the Industrial Perspective. *Sensors*, 17(10), 2284. doi:10.3390/s17102284
17. Faisal, I. A., Purboyo, T. W., & Ansori, A. S. (2019). A Review of Accelerometer Sensor and Gyroscope Sensor in IMU Sensors on Motion Capture. *Journal of Engineering and Applied Sciences*, 15(3), 826-829. doi:10.36478/jeasci.2020.826.829
18. Proximity Sensor. (n.d.). Retrieved from <https://www.sciencedirect.com/topics/engineering/proximity-sensor>
19. Foix, S., Alenya, G., & Torras, C. (2011). Lock-in Time-of-Flight (ToF) Cameras: A Survey. *IEEE Sensors Journal*, 11(9), 1917-1926. doi:10.1109/jsen.2010.2101060
20. Kumar, S., & Moore, K. B. (2002). *Journal of Science Education and Technology*, 11(1), 59-80. doi:10.1023/a:1013999415003
21. Wagner, D., & Schmalstieg, D. (n.d.). Handheld Augmented Reality Displays. IEEE Virtual Reality Conference (VR 2006). doi:10.1109/vr.2006.67, <https://ieeexplore.ieee.org/document/1667684>
22. Amanda Edwards Stewart, Tim hoyt, & Greg M. Reger (2016) Classifying Different Types of Augmented Reality Technology, https://www.researchgate.net/publication/315701832_Classifying_different_types_of_augmented_reality_technology
23. Siltanen, S. (2012). Theory and applications of marker-based augmented reality. Retrieved from <https://www.vtresearch.com/sites/default/files/pdf/science/2012/S3.pdf>.
24. Ahir, K. (2019, September 03). What is the difference between Marker based and Markerless AR? Retrieved from <https://medium.com/@kumar.ahir/what-is-the-difference-between-marker-based-and-markerless-ar-192fb9fa09c5>

25. Navigation. (2020, May 30). Retrieved from <https://en.wikipedia.org/wiki/Navigation#Etymology>
26. Narzt, W., Pomberger, G., Ferscha, A., Kolb, D., Muller, R., Wieghardt, J., . . . Lindinger, C. (2006, March). Augmented reality navigation systems. DOI: 10.1007/s10209-005-0017-5 ,https://www.researchgate.net/publication/220606626_Augmented_reality_navigation_systems
27. Rehman, U., & Cao, S. (2016). Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE Transactions on Human-Machine Systems*, 1-12. doi:10.1109/thms.2016.2620106 ,https://www.researchgate.net/publication/310393165_Augmented-Reality-Based_Indoor_Navigation_A_Comparative_Analysis_of_Handheld_Devices_Versus_Google_Glass
28. Simultaneous localization and mapping. (2020, May 18). Retrieved from https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
29. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., . . . Leonard, J. J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309-1332. doi:10.1109/tro.2016.2624754 ,http://rpg.ifi.uzh.ch/docs/TRO16_cadena.pdf
30. THRUN, S., BURGARD, W., & FOX, D. (1998). A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. doi:doi:10.1023/a:1008806205438 ,<https://link.springer.com/article/10.1023/A:1008806205438>
31. Smith, R., Self, M., & Cheeseman, P. (1987). A Stochastic Map For Uncertain Spatial Relationships. Retrieved from https://www.researchgate.net/publication/234808346_A_stochastic_map_for_uncertain_spatial_relationships.
32. LU, F., & MILIOS, E. (1997). Globally Consistent Range Scan Alignment for Environment Mapping. Retrieved from <https://link.springer.com/article/10.1023/A:1008854305733>.
33. Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1052-1067. doi:10.1109/tpami.2007.1049 ,https://www.doc.ic.ac.uk/~ajd/Publications/davison_etal_pami2007.pdf

34. Burgard, W., Stachniss, C., Grisetti, G., Steder, B., Kummerle, R., Dornhege, C., . . . Tardos, J. D. (2009). A comparison of SLAM algorithms based on a graph of relations. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. doi:10.1109/iros.2009.5354691
https://www.researchgate.net/publication/221065312_A_Comparison_of_SLAM_Algorithms_Based_on_a_Graph_of_Relations