



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ και ΥΠΟΛΟΓΙΣΤΩΝ

ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ και ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Πλοήγηση σε εσωτερικούς χώρους με χρήση δυνατοτήτων
επαυξημένης πραγματικότητας σε έξυπνες συσκευές**

Φωστίνης Πέτρος

ΑΜ: 711-131104

Επιβλέπων: Χρήστος Τρούσσας, Επ. Καθηγητής

Διπλωματική εργασία υποβληθείσα στο Τμήμα

ΑΙΓΑΛΕΩ, ΟΚΤΩΒΡΙΟΣ 2022

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Ο Δηλών



Πανεπιστήμιο Δυτικής Αττικής,
Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών,
Φωστίνης Πέτρος
© 2022 – Με την επιφύλαξη παντός δικαιώματος

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ και ΥΠΟΛΟΓΙΣΤΩΝ
ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ και ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η παρούσα διπλωματική εργασία παρουσιάστηκε από τον

Φωστίνη Πέτρο

ΑΜ: 711-131104

Επιβλέπων: Χρήστος Τρούσσας, Επ. Καθηγητής

Διπλωματική εργασία υποβληθείσα στο Τμήμα

ΑΙΓΑΛΕΩ, ΟΚΤΩΒΡΙΟΣ 2022

Εισηγητής

Χρήστος Τρούσσας, Επίκουρος Καθηγητής

Εξεταστική Επιτροπή

Χρήστος Τρούσσας

Κλειώ Σγουροπούλου

Γιώργος Μελετίου

Ημερομηνία εξέτασης 20/10/2022

Η έγκριση της διπλωματικής εργασίας δεν υποδηλοί την αποδοχή των γνώμων του συγγραφέα. Κατά τη συγγραφή τηρήθηκαν οι αρχές της ακαδημαϊκής δεοντολογίας.

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

ΠΕΡΙΛΗΨΗ

Η πλοήγηση σε μοντέρνα συμπλέγματα κτιρίων είναι συχνά δύσκολη και περίπλοκη. Με την χρήση εφαρμογών πλοήγησης σε εξωτερικούς χώρους να είναι μέρος της καθημερινότητάς μας, αλλά ανεπαρκείς σε εσωτερικούς χώρους, η παρούσα διπλωματική εργασία στοχεύει να γεμίσει το κενό προσφέροντας δυνατότητες πλοήγησης σε εσωτερικούς χώρους.

Για τον εντοπισμό της τοποθεσίας του χρήστη θα χρησιμοποιηθούν δυνατότητες σάρωσης κωδικών QR, για την αντιστοίχιση ψηφιακού και φυσικού χώρου θα αναπτυχθεί μια πλήρης μακέτα του φυσικού χώρου στο Blender, και για την ιχνηλάτηση της κίνησης και της τοποθεσίας του χρήστη θα αξιοποιηθούν μοντέρνες τεχνολογίες επαυξημένης πραγματικότητας.

Η τελική εφαρμογή είναι ικανή να κατευθύνει τον χρήστη από την εντοπισμένη τοποθεσία του προς τον επιλεγμένο προορισμό του, συγκεκριμένα στους χώρους του τμήματος πληροφορικής του Πανεπιστημίου Δυτικής Αττικής.

Τα αποτελέσματα είναι ικανοποιητικά, με σημαντικό πλεονέκτημα το ελάχιστο κόστος εγκατάστασης, την προσβασιμότητα σε κάθε χρήστη με κινητό, ακόμα και ελλείψει διαδικτυακής σύνδεσης, αλλά με αξιοσημείωτο μειονέκτημα την περιστασιακή ανακρίβεια που χαρακτηρίζει τις τεχνολογίες επαυξημένης πραγματικότητας.

**Πλοήγηση σε εσωτερικούς χώρους με χρήση δυνατοτήτων
επαυξημένης πραγματικότητας έξυπνων συσκευών.**

Φωστίνης Πέτρος

Λέξεις κλειδιά

- Πλοήγηση σε Εσωτερικούς Χώρους
- Unity
- Επαυξημένη Πραγματικότητα
- Εφαρμογή σε κινητές συσκευές

ABSTRACT

Indoor navigation can often be a difficult and complex task. Using navigator applications for the outdoors has become a part of our everyday lives, but is quite lacking when it comes to indoor settings. This paper aims to fill in the gap by providing a fully operational application for indoor navigation.

The user's location will be determined via the scanning of QR Codes, the physical world will be mapped to a digital maquette made in Blender, and the user's movement and active location will be tracked using the capabilities of modern augmented reality technologies.

The final application will be able to guide the user from an understood starting location to a destination of their choice, specifically targeting the Department of Informatics and Computer Engineering building within the Western University of Attica campus.

The results are pleasing, with negligible infrastructure costs, an application available to anyone with a smartphone, and with no need to always be online. Nothing is perfect, however, as augmented-reality-driven location tracking is inherently somewhat unreliable.

Application for indoor navigation based on the augmented reality capabilities of smart devices

Petros Fostinis

Keywords

- Indoor Navigation
- Unity
- Augmented Reality
- Application on Smart Devices

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1	Πρότυπο VR του Ivan Sutherland	σελ 22
Εικόνα 2.2	Πρότυπο AR του Louis Rosenberg	σελ 22
Εικόνα 3.1	Δοκιμή στην ζώνη (3:3)	σελ 29
Εικόνα 5.1	Κάτοψη Τμήματος Πληροφορικής του ΠΑΔΑ	σελ 45
Εικόνα 5.2	Κάτοψη τοποθετημένη στην Unity	σελ 46
Εικόνα 5.3	Κάτοψη με αντικείμενα κύβοι-τοίχοι	σελ 46
Εικόνα 5.4	Δωμάτια εισαγμένα μέσω αρχείου SVG	σελ 47
Εικόνα 5.5	Δωμάτια μορφής Object, χωρίς καθάρισμα	σελ 48
Εικόνα 5.6	Δωμάτια μορφής Object, καθαρισμένα	σελ 48
Εικόνα 5.7	Τοίχοι μορφής Object, με διορθώσεις	σελ 49
Εικόνα 5.8	Υψηλής ποιότητας render του τελικού μοντέλου	σελ 50
Εικόνα 5.9	Τρισδιάστατο μοντέλο τοποθετημένο στην Unity	σελ 50
Εικόνα 5.10	Κάτοψη με υπολογισμένο NavMesh	σελ 51
Εικόνα 5.11	Ψηφιακός χώρος με αντικείμενα προορισμού & QR	σελ 52
Εικόνα 5.12	Αντικείμενα τοποθεσίας χρήστη	σελ 53
Εικόνα 5.13	Βέλος Τοποθεσίας	σελ 54
Εικόνα 5.14	Script Component	σελ 60
Εικόνα 5.15	Λίστα Τοποθεσιών	σελ 60
Εικόνα 5.16	Λίστα Προορισμών	σελ 60
Εικόνα 5.17	Splash Screen & Boot Animation	σελ 65

Εικόνα 5.18	Κυρίως διεπαφή χρήστη	σελ 66
Εικόνα 5.19	Διεπαφή χρήστη με έγκυρες επιλογές	σελ 66
Εικόνα 5.20	Σάρωση Κωδικού Γρήγορης Απόκρισης	σελ 68
Εικόνα 5.21	Φόντο Εικονιδίου	σελ 69
Εικόνα 5.22	Προσκήνιο Εικονιδίου	σελ 69
Εικόνα 5.23	Τελικό Εικονίδιο	σελ 69

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 3.1	Τριγωνισμός με χρήση Beacons	σελ 28
Διάγραμμα 3.2	Διαδικασία αποθήκευσης χάρτη	σελ 32
Διάγραμμα 3.3	Διαδικασία ανάκτησης χάρτη	σελ 32
Διάγραμμα 5.1	Σχέση Android Studio με Unity	σελ 40
Διάγραμμα 5.2	Διάγραμμα ροής της Unity	σελ 42
Διάγραμμα 5.3	Διάγραμμα ροής του Android Studio	σελ 63
Διάγραμμα 6.1	Κατανομή Γκάλοπ Βαθμολογιών	σελ 70
Διάγραμμα 6.2	Αθροισμένες Απόψεις Χρηστών	σελ 71
Διάγραμμα 6.3	Συσχέτιση Βαθμολογιών με Κριτικές	σελ 72

ΚΑΤΑΛΟΓΟΙ ΤΜΗΜΑΤΩΝ ΚΩΔΙΚΑ

Κώδικας Unity 1	Υπολογισμός και εξέφραση διαδρομής	σελ 80
Κώδικας Unity 2	Unload Unity και επιστροφή σε Android Studio	σελ 80
Κώδικας Unity 3	Αρχειοποίηση λιστών τοποθεσιών και προορισμών	σελ 81
Κώδικας Unity 4	Υπολογισμός και χάραξη διαδρομής	σελ 81

ΚΑΤΑΛΟΓΟΣ ΟΡΟΛΟΓΙΩΝ

<u>Ελληνική Ορολογία</u>	<u>Αγγλική Ορολογία</u>	<u>Συντομογραφία</u>
Πλοήγηση σε Εσωτερικούς Χώρους	Indoor Navigation	IDN
Επαυξημένη Πραγματικότητα	Augmented Reality	AR
Ψηφιακή Πραγματικότητα	Virtual Reality	VR
Παγκόσμιο Σύστημα Στιγματοθέτησης	Global Positioning System	GPS
Πλέγμα Πλοήγησης	Navigation Mesh	NavMesh
Κωδικοί Γρήγορης Απόκρισης	Quick Response Codes	QR Codes
Ένδειξη Έντασης Λαμβανομένου Σήματος	Received Signal Strength Indication	RSSI
Διεπαφή Προγραμματισμού Εφαρμογών	Application Programming Interface	API

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	5
ABSTRACT	7
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	9
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ	11
ΚΑΤΑΛΟΓΟΙ ΤΜΗΜΑΤΩΝ ΚΩΔΙΚΑ	12
ΚΑΤΑΛΟΓΟΣ ΟΡΟΛΟΓΙΩΝ	13
ΠΕΡΙΕΧΟΜΕΝΑ	14
ΠΡΟΛΟΓΟΣ	17
1. ΕΙΣΑΓΩΓΗ	18
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	20
2.1. ΕΞΥΠΝΕΣ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ	20
2.2. ΕΦΑΡΜΟΓΕΣ ΕΞΥΠΝΩΝ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ	20
2.3. ΕΦΑΡΜΟΓΕΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ	21
2.4. ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ UNITY	22
2.5. ΠΛΑΙΣΙΟ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ARFOUNDATION	23
2.6. ΔΙΕΠΑΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΦΑΡΜΟΓΩΝ ARCORE	24
2.7. ΕΝΣΩΜΑΤΩΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ANDROID STUDIO	25
2.8. ΕΝΤΟΠΙΣΜΟΣ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗ ΤΟΠΟΘΕΣΙΑΣ	25
3. ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ	27
3.1. A LOW ENERGY IOT APPLICATION USING BEACON FOR INDOOR LOCALIZATION	27
3.2. INDOOR MAPPING BASED ON AUGMENTED REALITY USING UNITY ENGINE	30
4. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ	33

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

4.1. ΠΡΩΙΜΑ ΣΤΑΔΙΑ ΕΡΕΥΝΑΣ	33
4.2. ΜΕΤΑΦΟΡΑ ΣΤΗΝ UNITY	34
4.3. ΕΠΙΛΟΓΕΣ ΕΝΤΟΠΙΣΜΟΥ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗΣ ΤΟΠΟΘΕΣΙΑΣ	36
4.4. ΑΝΤΙΣΤΟΙΧΗΣΗ ΦΥΣΙΚΟΥ ΚΑΙ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ	38
5. ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	40
5.1. ΜΗΧΑΝΙΣΜΟΣ ΠΛΟΗΓΗΣΗΣ ΜΕΣΩ UNITY	40
5.1.1. ΨΗΦΙΑΚΗ ΑΠΕΙΚΟΝΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΧΩΡΟΥ	43
5.1.1.1. ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ ΑΠΟ ΚΑΤΟΨΗ ΣΧΕΔΙΩΝ	45
5.1.1.2. ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ ΑΠΟ ΔΕΔΟΜΕΝΑ	47
5.1.2. ΠΛΕΓΜΑ ΠΛΟΗΓΗΣΗΣ ΚΑΙ ΕΠΙΠΛΕΟΝ ΑΝΤΙΚΕΙΜΕΝΑ	51
5.1.3. ΚΑΜΕΡΕΣ ΣΚΗΝΗΣ UNITY ΚΑΙ ΒΑΣΙΚΟ UI	54
5.1.4. SCRIPTS ΤΗΣ UNITY	58
5.1.4.1. ΕΠΙΚΟΙΝΩΝΙΑ UNITY ΚΑΙ ANDROID STUDIO	58
5.1.4.2. ΥΠΟΛΟΓΙΣΜΟΣ ΔΙΑΔΡΟΜΗΣ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗ ΤΟΠΟΘΕΣΙΑΣ	59
5.1.4.3. ΣΑΡΩΣΗ ΚΩΔΙΚΩΝ QR	61
5.2. ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ ΜΕΣΩ ANDROID STUDIO	62
5.2.1. ΠΑΡΑΓΩΓΟ ΤΗΣ UNITY	64
5.2.2. ΚΙΝΟΥΜΕΝΟ ΣΧΕΔΙΟ ΕΚΚΙΝΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	64
5.2.3. ΚΥΡΙΩΣ ΔΙΕΠΑΦΗ ΤΟΥ ΧΡΗΣΤΗ	65
5.2.4. ΣΑΡΩΣΗ ΚΩΔΙΚΟΥ QR	67
5.2.5. ΕΙΚΟΝΙΔΙΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	68
6. ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ	70
7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	74
ΒΙΒΛΙΟΓΡΑΦΙΑ	77
ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ	80

ΠΡΟΛΟΓΟΣ

Θα ήθελα να ευχαριστήσω τον κύριο Τρούσσα Χρήστο που ανέλαβε την διπλωματική μου εργασία, τον Γιαννακόπουλο Νίκο για την συνεργασία και βοήθεια που προσέφερε στην ανάπτυξη της εφαρμογής, και τέλος την οικογένεια και τους φίλους που με στήριξαν και με υπέμειναν για τόσα χρόνια.

1.ΕΙΣΑΓΩΓΗ

Συχνά, η πλοήγηση σε μεγάλους και περίπλοκους χώρους, όπως σε εκπαιδευτικά τμήματα, μπορεί να είναι εξαιρετικά δύσκολη και πολύπλοκη. Η παρουσία κάποιου ατόμου που γνωρίζει τον χώρο και μπορεί να προσφέρει την βοήθεια του είναι η πιο απλή και αξιόπιστη λύση, αλλά επίσης είναι μη εφαρμόσιμη όταν δεν υπάρχουν τέτοια άτομα διαθέσιμα για να προσφέρουν την γνώση και βοήθεια τους, ή αν υπάρχει δυσκολία επικοινωνίας λόγω έλλειψης κοινής γλώσσας. Με την πρόοδο που έχουν κάνει οι έξυπνες συσκευές την τελευταία δεκαετία, μπορούν να αναπτυχθούν εφαρμογές που ικανοποιούν τον ρόλο αυτό ενός βοηθού-γνώστη.

Στην πλοήγηση σε ανοιχτούς χώρους μέσω έξυπνων συσκευών χρησιμοποιείται η τεχνολογία GPS για εντοπισμό και ιχνηλάτηση τωρινής τοποθεσίας, αλλά με το σήμα της να είναι αναξιόπιστο σε εσωτερικούς χώρους πρέπει να στραφούμε σε άλλες τεχνολογίες. Συχνή εναλλακτική είναι η χρήση WiFi ή BLE Beacons, μα έχει πολλαπλά αρνητικά. Οι μικρές αποστάσεις στις οποίες παραμένουν ακριβείς οι μετρήσεις τους απαιτούν περίπλοκες και ακριβές εγκαταστάσεις υλικού, και ανεξαρτήτως η ακρίβεια τους επηρεάζεται από μεταβλητές τις οποίες δεν ελέγχουμε πλήρως, όπως κινητά εμπόδια και το πλήθος χρηστών ανά πάσα στιγμή.

Για αυτό λοιπόν η εργασία στράφηκε σε ανάπτυξη εφαρμογής που βασίζεται σε χρήση σάρωσης κωδικών γρήγορης απόκρισης για τον εντοπισμό της αρχικής θέσης του χρήστη, και τεχνολογιών επαυξημένης πραγματικότητας για

την ανανέωση της θέσης του χρήστη όσο αυτός κινείται προς τον προορισμό του.

Οι εφαρμογές έξυπνων κινητών συσκευών είναι προγράμματα, σχεδιασμένα να εκτελούνται σε μοντέρνες έξυπνες κινητές συσκευές, οι οποίες μπορούν να αξιοποιήσουν πλήρως την φορητότητα των συσκευών αυτών. Χαρακτηρίζονται από την χρήση των μοντέρνων δυνατοτήτων έξυπνων κινητών συσκευών, όπως διεπαφή μέσω οθονών αφής και χρήση διαφόρων αισθητήρων.

Το αποτέλεσμα της διπλωματικής αυτής είναι μια εφαρμογή που συνδυάζει τις δυνατότητες διεπαφής χρήστη του Android Studio, δυνατότητες δημιουργίας ψηφιακού χώρου μέσω του Blender, και δυνατότητες εύρεσης βέλτιστης διαδρομής και ιχνηλάτησης τοποθεσίας μέσω της Unity.

Στα κεφάλαια που ακολουθούν, θα γίνουν αναφορές στις βασικές τεχνολογίες και πλατφόρμες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής αυτής, σε άλλες λύσεις για το πρόβλημα της πλοήγησης σε εσωτερικούς χώρους και πως συγκρίνονται με την διπλωματική αυτή, θα αναπτυχθούν τα βήματα μελέτης και ανάπτυξης που χρειάστηκαν κατά την ανάπτυξη της εφαρμογής, θα αναλυθούν η διαδικασία και ο τρόπος ανάπτυξης της τελικής εφαρμογής, και τέλος θα παρουσιαστούν αξιολογήσεις, συμπεράσματα και εν δυνάμει μελλοντικές προσθήκες.

2.ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1. ΕΞΥΠΝΕΣ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

Στην σημερινή εποχή, οι έξυπνες κινητές συσκευές αποτελούν αναπόσπαστο κομμάτι της καθημερινότητας μας. Τα χαρακτηριστικά τους συνεχώς εμπλουτίζονται και η δυναμική τους μεγαλώνει, βελτιώνοντας τις δυνατότητες που μας προσφέρουν. Η χρησιμότητά τους είναι καθοριστική, αφού αποτελούν το μέσο που παρέχει την κατάλληλη υποδομή για την λειτουργία εφαρμογών με τις οποίες αλληλεπιδρούμε σε καθημερινή βάση.

Παρόλο που ο αρχικός σκοπός των συσκευών αυτών ήταν περιορισμένος σε ρόλους ψηφιακού προσωπικού βοηθού, όπως π.χ. με την συσκευή *Apple Newton MessagePad*, με τα άλματα τόσο του υλικού όσο και του λογισμικού τις τελευταίες δεκαετίες, οι έξυπνες κινητές συσκευές έχουν αποκτήσει χρησιμότητα ως έξυπνοι προσωπικοί βοηθοί, πολυμεσικές πλατφόρμες προσωπικής διασκέδασης, εργαλεία μάθησης και άλλα.

2.2. ΕΦΑΡΜΟΓΕΣ ΕΞΥΠΝΩΝ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ

Οι εφαρμογές έξυπνων κινητών συσκευών αποτελούν εφαρμογές λογισμικού, κοινώς προγράμματα, σχεδιασμένες να εκτελούνται σε μοντέρνες έξυπνες κινητές συσκευές. Σε αντίθεση με προγράμματα που στοχεύουν σταθερούς προσωπικούς υπολογιστές, εφαρμογές που στοχεύουν πλατφόρμες έξυπνων

κινητών συσκευών μπορούν να αξιοποιήσουν πλήρως την δυνατότητα φορητότητας των αντίστοιχων συσκευών τους.

Χαρακτηρίζονται από την χρήση των μοντέρνων δυνατοτήτων έξυπνων κινητών συσκευών, όπως διεπαφή μέσω οθονών αφής, χρήση αισθητήρων όπως κάμερες για μηχανική όραση ή επιταχυνσιόμετρα, γυροσκόπια, μαγνητόμετρα και αισθητήρες του Παγκόσμιου Συστήματος Στιγματοθέτησης (GPS) για εντοπισμό και ιχνηλάτηση τοποθεσίας.

2.3. ΕΦΑΡΜΟΓΕΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ

Η επαυξημένη πραγματικότητα είναι μια ταχέως αναδυόμενη τεχνολογία που συμπληρώνει τον πραγματικό κόσμο με εικονικά αντικείμενα, φαινομενικά τοποθετημένα στο φυσικό χώρο.^[15]

Με τις ρίζες της τεχνολογίας αυτής σε πρότυπα όπως τα πρώιμα γυαλιά εικονικής πραγματικότητας του Ivan Sutherland το 1968 και τα "Virtual Fixtures" του Louis Rosenberg το 1992, τα άλματα τεχνολογιών τόσο υλικού όσο και λογισμικού, συν η άνοδος των έξυπνων φορητών συσκευών, ιδέες που στο άρθρο "The Cinema of the Future" του Morton Heilig το 1955 έμοιαζαν με ιστορίες επιστημονικής φαντασίας, έχουν γίνει μέρος της καθημερινότητας μας.^[12]

Τα τελευταία χρόνια υπάρχει η τάση να συνδυάζονται τεχνολογίες έξυπνων κινητών συσκευών με τεχνολογίες επαυξημένης πραγματικότητας, για την υλοποίηση εφαρμογών AR που αξιοποιούν την φορητότητα και την άμεση πρόσβαση σε πληροφορίες που προσφέρουν οι έξυπνες κινητές συσκευές.^[16]

Μια εφαρμογή επαυξημένης πραγματικότητας είναι διαδραστική, εκτελείται σε πραγματικό χρόνο και συνδυάζει συν ευθυγραμμίζει πραγματικά με ψηφιακά αντικείμενα στο φυσικό χώρο.^[13] Οι εφαρμογές επαυξημένης πραγματικότητας επιτυγχάνουν τον συνδυασμό αυτό υπερθέτοντας εικόνες, κατασκευασμένες από έξυπνες κινητές συσκευές, στην οπτική του φυσικού κόσμου που βλέπει ο χρήστης.^[14]



Εικόνα 2.1
Πρότυπο VR του Ivan Sutherland



Εικόνα 2.2
Πρότυπο AR του Louis Rosenberg

2.4. ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ UNITY

Για βασική πλατφόρμα της εφαρμογής μας επιλέχθηκε η Unity. Η Unity είναι μια πλατφόρμα ανάπτυξης παιχνιδιών, εφαρμογών ψηφιακής πραγματικότητας και εφαρμογών επαυξημένης πραγματικότητας.

Παρόλο που αρχικά ανακοινώθηκε ως πλατφόρμα ανάπτυξης παιχνιδιών για το λογισμικό Mac OS X, έχει επεκταθεί στο πέρασμα των χρόνων ώστε να υποστηρίζει πολλαπλές πλατφόρμες, όπως και Android, ή οποία είναι η πλατφόρμα για την οποία προορίζεται η εφαρμογή αυτή. Η κυρίως γλώσσα προγραμματισμού που χρησιμοποιείται για ανάπτυξη εφαρμογών σε Unity είναι η C#.

Επιπλέον υποστηρίζει πολλαπλές επεκτάσεις, φτιαγμένες από την Unity Technologies ή από τρίτους. Το ARFoundation της Unity Technologies και το ARCore της Google Inc., παίζουν κομβικό ρόλο στην εφαρμογή μας.

2.5. ΠΛΑΙΣΙΟ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ARFOUNDATION

Το AR Foundation της Unity είναι ένα πλαίσιο ανάπτυξης εμπειριών επαυξημένης πραγματικότητας, με την δυνατότητα να αναπτυχθεί μία φορά και να χτιστεί για πολλαπλές πλατφόρμες χωρίς επιπλέον αλλαγές ή προσαρμογή ανά πλατφόρμα.

Συγκεκριμένα υποστηρίζει το ανοιχτό πρότυπο ανάπτυξης εφαρμογών επαυξημένης πραγματικότητας OpenXR, την διεπαφή προγραμματισμού εφαρμογών επαυξημένης πραγματικότητας ARKit για iOS και την διεπαφή προγραμματισμού εφαρμογών επαυξημένης πραγματικότητας ARCore για AndroidOS.

Επιπλέον το AR Foundation προσπαθεί να προστατεύσει τις εφαρμογές που το χρησιμοποιούν από το να γίνουν γρήγορα παρωχημένες. Προσφέροντας υποστήριξη για τρεις διαφορετικές διεπαφές προγραμματισμού εφαρμογών

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

επαυξημένης πραγματικότητας, υπάρχουν λειτουργίες και δυνατότητες που ίσως να υποστηρίζει μόνο ένα από αυτά τα API. Σε τέτοιες περιπτώσεις, το ARFoundation θα τοποθετήσει τις "ρίζες" των συγκεκριμένων λειτουργιών και δυνατοτήτων ώστε όταν αποκτήσει το ελλιπές API τις δυνατότητες αυτές, η προσθήκη και ενεργοποίηση τους να είναι πιο εύκολη και γρήγορη, αν όχι πλήρως αυτοματοποιημένη.

2.6. ΔΙΕΠΑΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΦΑΡΜΟΓΩΝ ARCORE

Το ARCore είναι η πλατφόρμα της Google Inc. για ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας. Χρησιμοποιώντας διάφορες διεπαφές προγραμματισμού εφαρμογών (*Application Programming Interfaces - APIs*), το ARCore προσφέρει σε έξυπνες συσκευές την δυνατότητα να αισθάνονται το περιβάλλον τους, να κατανοούν τον κόσμο γύρω τους και να αλληλεπιδρούν με πληροφορίες.

Συνοπτικά το ARCore χρησιμοποιεί τέσσερις βασικές δυνατότητες για να ενσωματώσει ψηφιακές πληροφορίες και εικονικά αντικείμενα στον πραγματικό κόσμο όπως τον βλέπει η κάμερα μας:

Ο προσδιορισμός κίνησης (*motion tracking*) επιτρέπει στην συσκευή να καταλαβαίνει και να ιχνηλατεί την τοποθεσία της σε σχέση με τον κόσμο. Η περιβαλλοντική κατανόηση (*environmental understanding*) επιτρέπει στην συσκευή να αναγνωρίζει το μέγεθος και την τοποθεσία κάθε τύπου επιφάνειας: οριζόντιες όπως π.χ. το έδαφος, κάθετες όπως π.χ. τοίχους, και σε κάθε άλλη κλίση, όπως π.χ. έπιπλα. Η κατανόηση βάθους (*depth understanding*) επιτρέπει

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

στην συσκευή να ξεχωρίσει μεταξύ τους αντικείμενα και να εντοπίσει την σειρά και το βάθος που είναι τοποθετημένα. Τέλος η εκτίμηση φωτός (*light estimation*) επιτρέπει στην συσκευή να εκτιμά τις τωρινές συνθήκες φωτισμού του τυχόν χώρου.^[3]

Ο συνδυασμός των δυνατοτήτων αυτών, σε συνεργασία με την συνεχή διεπαφή του χρήστη προσφέρουν την δυνατότητα Ταυτόχρονης Τοπικοποίησης και Χαρτογράφησης (*Simultaneous Localization and Mapping - SLAM*). Το SLAM κατασκευάζει και συνεχώς ενημερώνει έναν χάρτη βασισμένο στο πως κατανοεί τον φυσικό χώρο γύρω του, ενώ ταυτόχρονα ιχνηλατεί την τοποθεσία του χρήστη στον χώρο αυτό.

2.7. ΕΝΣΩΜΑΤΩΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ANDROID STUDIO

Το Android Studio είναι το ενσωματωμένο περιβάλλον ανάπτυξης (*Integrated Development Environment - IDE*) εφαρμογών Android της Google Inc. Βασισμένο στις ισχυρές δυνατότητες του IntelliJ IDE, το Android Studio προσφέρει επιπλέον λειτουργίες στοχευμένες στην ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android όπως προσομοιωτές περιβάλλοντος και λογισμικού συσκευών Android.

2.8. ΕΝΤΟΠΙΣΜΟΣ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗ ΤΟΠΟΘΕΣΙΑΣ

Ο εντοπισμός και η ιχνηλάτηση τοποθεσίας σε εσωτερικούς χώρους μπορούν γενικώς να προσεγγιστούν με δύο διαφορετικούς τρόπους. Η πρώτη προσέγγιση

είναι μέσω Τοπικοποίησης σε Εσωτερικούς Χώρους βασισμένη σε Ζώνες (*Zone-Based Indoor Localization - Z-BIL*) και η δεύτερη είναι μέσω Ακριβής Τοπικοποίησης σε Εσωτερικούς Χώρους (*Precise Indoor Localisation - PIL*).

Η πρώτη προσέγγιση βασίζεται στον διαχωρισμό του χώρου πλοήγησης σε "ζώνες" και μετά στον εντοπισμό της ζώνης στην οποία βρίσκεται ο χρήστης. Η προσέγγιση αυτή είναι ιδανική για τρόπους εντοπισμού και ιχνηλάτησης τοποθεσίας όπως χρήση RSSI μέσω WiFi ή BLE, που δεν έχουν επαρκή ακρίβεια για περισσότερο λεπτομερείς μετρήσεις.

Η δεύτερη προσέγγιση απαιτεί τον ακριβή εντοπισμό της τοποθεσίας του χρήστη, με ακρίβεια περισσότερη του ενός μέτρου. Όπως προαναφέρθηκε, η χρήση μετρήσεων RSSI δεν έχει επαρκή ακρίβεια για την προσέγγιση αυτή, άρα πρέπει να συμπληρωθεί με επιπλέον μετρήσεις, όπως την μέτρηση κίνησης στο πέρας του χρόνου μέσω επιταχυνσιόμετρου και τον εντοπισμό περιστροφής της οπτικής του χρήστη μέσω γυροσκόπιου.^[6]

3. ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ

Στο κεφάλαιο αυτό θα γίνουν αναφορές σε εναλλακτικούς τρόπους επίλυσης του προβλήματος της πλοήγησης σε εσωτερικούς χώρους. Συγκεκριμένα θα αναφερθεί το άρθρο "*A Low Energy IoT Application Using Beacon for Indoor Localization*" των Francesco Pascale, Ennio Andrea Adinolfi, Massimiliano Avagliano, Venanzio Giannella και Andres Salas, που χρησιμοποιεί τεχνολογία BLE Beacons με συσκευές HC-05 και Arduino για τον εντοπισμό τοποθεσίας σε ζώνες δισδιάστατου χώρου. Επίσης θα γίνει αναφορά στο άρθρο "*Indoor Mapping Based on Augmented Reality Using Unity Engine*" των Abhijith Ajith, Albin Mathew Thomas, Jestin Antony, Joel Joseph και Suma R, που χρησιμοποιεί τεχνολογίες επαυξημένης πραγματικότητας και cloud services για χαρτογράφηση χώρου και αποθήκευση διαδρομών αντιστοίχως.

3.1. A LOW ENERGY IOT APPLICATION

USING BEACON FOR INDOOR LOCALIZATION

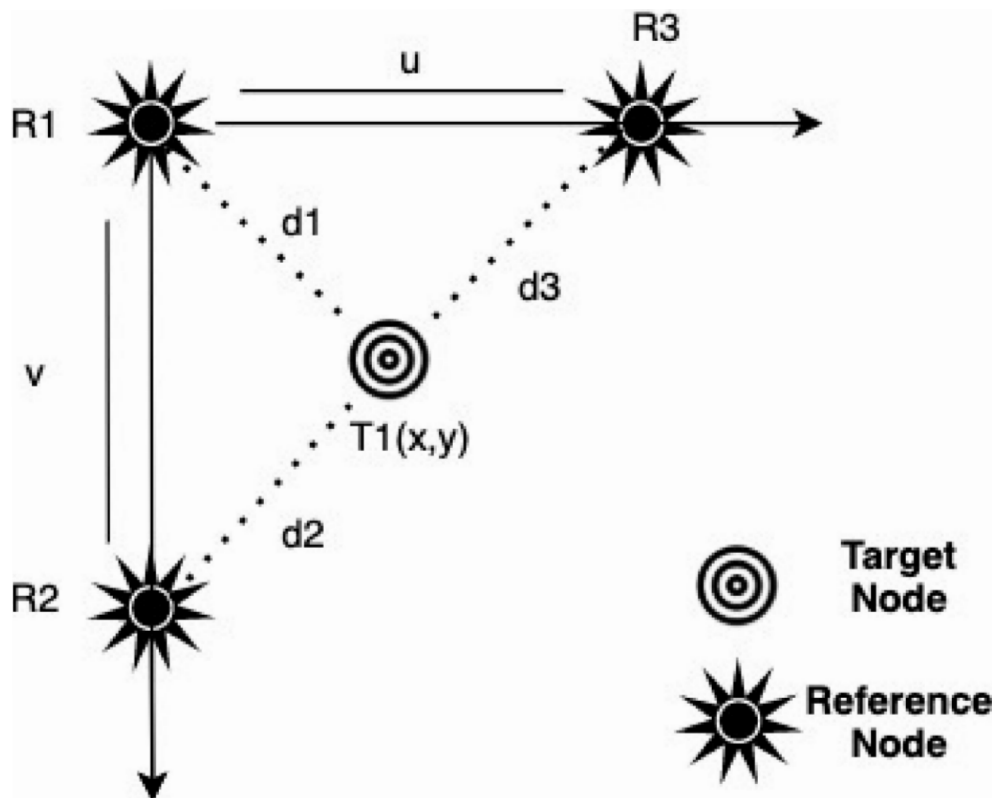
Ο τρόπος εντοπισμού και ιχνηλάτησης τοποθεσίας που επέλεξαν οι Francesco Pascale, Ennio Andrea Adinolfi, Massimiliano Avagliano, Venanzio Giannella και Andres Salas, στο άρθρο τους "*A Low Energy IoT Application Using Beacon for Indoor Localization*"^[6], ήταν βασισμένος σε ζώνες (Z-BL).

Για την συλλογή δεδομένων, χρησιμοποιούν Arduino μικροελεγκτές σε συνεργασία με Bluetooth modules τύπου HC-05 και μετρούν την "ένδειξη έντασης λαμβανομένου σήματος" (*RSSI - Received Signal Strength Indication*) σε dB.

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

Θέτοντας τα HC-05 αυτά σε λειτουργία "Master", μπορούν να προγραμματιστούν μέσω των Arduino με εντολές τύπου "AT", ώστε να αποστέλλονται οι τιμές RSSI.

Τα δεδομένα αυτά αποστέλλονται σε Server, ο οποίος μπορεί να υπολογίσει την τοποθεσία κάθε συσκευής που βρίσκεται σε λειτουργία "Slave", κάνοντας τριγωνισμό βάση τριών διαφορετικών μετρήσεων. Χρησιμοποιώντας μόνο τρεις μετρήσεις λαμβάνουν επαρκείς πληροφορίες για τον υπολογισμό τοποθεσίας σε δισδιάστατο χώρο. Για εφαρμογές που χρειάζονται πληροφορίες και στις τρεις διαστάσεις, π.χ. να συμπεριλαμβάνουν εντοπισμό ορόφου, θα χρειάζεται και μια τέταρτη μέτρηση, άρα και τα επιπλέον αντίστοιχα HC-05 υπό έλεγχο Arduino μικροελεγκτών.



Διάγραμμα 3.1 - Τριγωνισμός με χρήση Beacons.^[6]

Αφότου αρχικοποιηθούν οι συσκευές τύπου "Master", πρέπει να χωριστεί ο φυσικός χώρος σε ζώνες. Με τα HC-05 να έχουν μέγιστο βεληνεκές τα 10m, στην εφαρμογή αυτή επέλεξαν ζώνες διαστάσεων $2 \times 2m^2$. Έπειτα τοποθετούνται συσκευές "Slave", συγκεκριμένα HC-05 modules χωρίς κάποιον μικροελεγκτή, των οποίων την τοποθεσία ιχνηλατούν οι συσκευές "Master".

Τοποθετώντας μια συσκευή τύπου "Slave" στην μέση ενός ελεγχόμενου και άδειου δωματίου, οι συγγραφείς μέτρησαν τιμές διάμεσης ακρίβειας 88%, διάμεσης επανάκλησης 85%, διάμεσης ορθότητας 83% και σκορ F1 87%. Ακολουθεί γράφημα από το συγκεκριμένο paper, που εκφράζει τον φυσικό χώρο, την τοποθεσία των συσκευών τύπου Master ως μπλε κουκίδες, της συσκευής τύπου Client ως κίτρινη κουκίδα και όλες της μετρήσεις τοποθεσίας ως γαλάζιους κύκλους.



Εικόνα 3.1 - Δοκιμή στην ζώνη (3;3).^[6]

Τα αποτελέσματα αυτά δείχνουν θετικά, αλλά σε μη ελεγχόμενους χώρους και με συχνά παράσιτα, όπως οι χώροι του ΠΑΔΑ, τα αποτελέσματα θα είναι πολύ λιγότερο αξιόπιστα.

3.2. INDOOR MAPPING BASED ON AUGMENTED REALITY USING UNITY ENGINE

Ο τρόπος χαρτογράφησης εσωτερικού χώρου που επέλεξαν οι Abhijith Ajith, Albin Mathew Thomas, Jestin Antony, Joel Joseph και Suma R στο άρθρο τους "*Indoor Mapping based on Augmented Reality using Unity Engine*"^[9], είναι βασισμένος στην χρήση δυνατοτήτων επαυξημένης πραγματικότητας για την χαρτογράφηση του χώρου δημιουργώντας μονοπάτια, και την χρήση cloud servers για την αποθήκευση και επανάκτηση των μονοπατιών αυτών.

Για την χαρτογράφηση του εσωτερικού χώρου, έχουν αναπτύξει μια εφαρμογή επαυξημένης πραγματικότητας που τους επιτρέπει να δημιουργήσουν και να αποθηκεύσουν εν δυνάμει διαδρομές σε φυσικούς χώρους. Δημιουργούν μια βάση δεδομένων από "μονοπάτια", την οποία θα γεμίσουν με διαδρομές οι οποίες έχουν "περπατηθεί" χρησιμοποιώντας την εφαρμογή τους.

Συγκεκριμένα η εφαρμογή χρησιμοποιεί δυνατότητες επαυξημένης πραγματικότητας και μηχανικής όρασης για να σαρώσει τον φυσικό χώρο και να εντοπίσει επιφάνειες βάσει των δεδομένων αυτών. Όσο ο χρήστης κινείται σε επιφάνειες που έχουν αναγνωριστεί, η εφαρμογή τοποθετεί "σημεία" στον ψηφιακό χώρο και τα συσχετίζει μεταξύ τους για την δημιουργία μονοπατιών. Συμπληρώνονται με metadata όπως ονομασία, ID και τοποθεσία. Συγκεκριμένα η

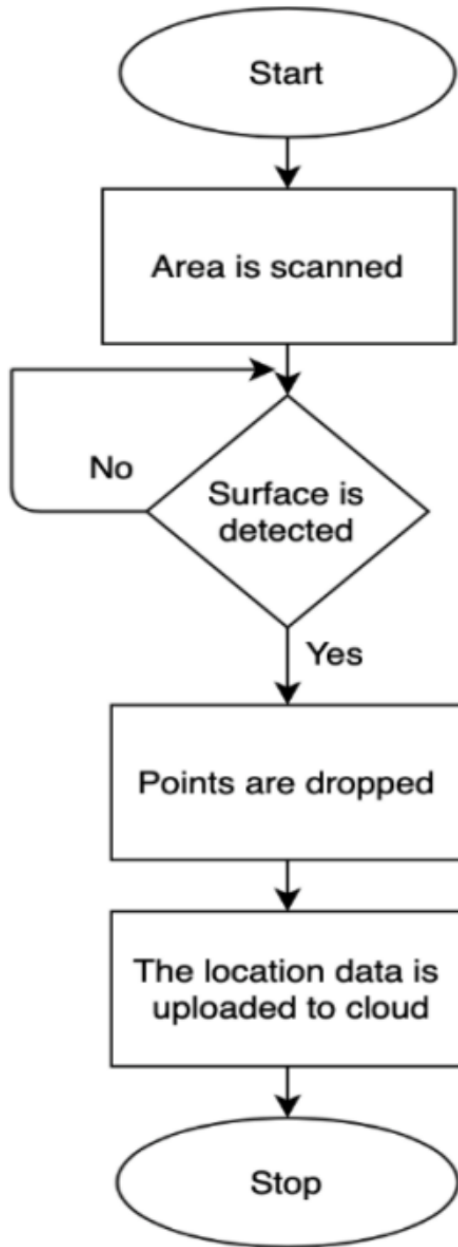
Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

τοποθεσία των μονοπατιών εκφράζεται σε γεωγραφικό μήκος, πλάτος και υψόμετρο. Έχοντας ολοκληρωθεί και συμπληρωθεί με τα προαναφερθέντα metadata, ανεβαίνουν και αποθηκεύονται στο cloud.

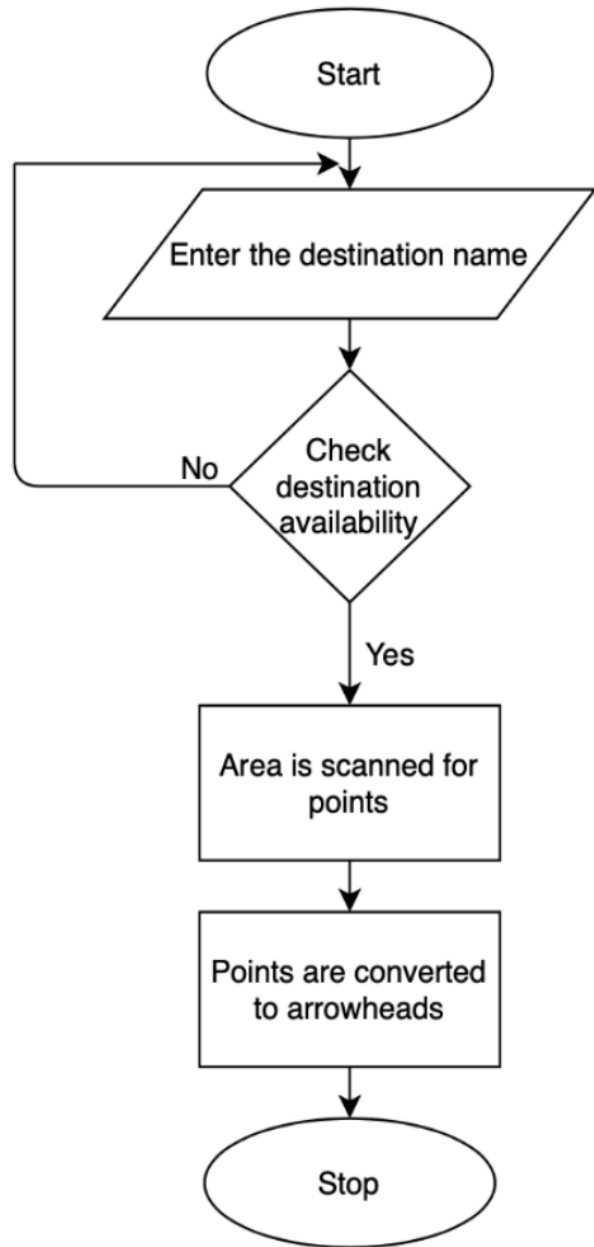
Για την ανάκτηση των μονοπατιών, ο χρήστης επιλέγει τον επιθυμητό του προορισμό, η εφαρμογή σαρώνει το γύρο περιβάλλον ώστε να αναγνωρίσει την τωρινή τοποθεσία του χρήστη, εντοπίζει την τοποθεσία αυτή στην βάση δεδομένων και επιλέγει το προϋπολογισμένο μονοπάτι που αντιστοιχεί στον επιλεγμένο προορισμό και την εντοπισμένη τοποθεσία.

Τα μονοπάτια αυτά μπορούν να ανακτηθούν αργότερα και να χρησιμοποιηθούν για να υπολογιστεί η βέλτιστη διαδρομή προς τον επιλεγμένο προορισμό του χρήστη, βάσει των συντεταγμένων και συσχετίσεων που απέκτησαν τα σημεία αυτά κατά την αρχική τοποθέτησή τους.

Ακολουθούν διαγράμματα από το συγκεκριμένο paper που εκφράζουν τις διαδικασίες αποθήκευσης και ανάκτησης χάρτη.



Διάγραμμα 3.2
 Διαδικασία αποθήκευσης χάρτη^[9]



Διάγραμμα 3.3
 Διαδικασία ανάκτησης χάρτη^[9]

4. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ

Η ανάπτυξη της εφαρμογής ξεκίνησε στο Android Studio, με τα πρώτα σχέδια να είναι μια εφαρμογή αμιγώς ανεπτυγμένη στο Android Studio, με χάρτες τοποθετημένους και αντιστοιχισμένους σε δεδομένα του Google Maps ή του Open Street Map, και χρήση Bluetooth Low Energy (BLE) Beacons για την ιχνηλάτηση της τωρινής τοποθεσίας του χρήστη.

Με την συνέχεια της μελέτης αντίστοιχων εφαρμογών και του πειραματισμού με τα πρωτότυπα της παρούσας εφαρμογής, η ανάπτυξη της εφαρμογής κατέληξε στην χρήση της Unity για την ιχνηλάτηση τοποθεσίας βάσει τεχνολογιών επαυξημένης πραγματικότητας, σε συνεργασία με το Android Studio για την διεπαφή χρήστη, και στη χρήση του προγράμματος δημιουργίας τρισδιάστατων γραφικών Blender για την αντιστοίχιση φυσικού και ψηφιακού κόσμου.

4.1. ΠΡΩΙΜΑ ΣΤΑΔΙΑ ΕΡΕΥΝΑΣ

Το αρχικό development στο Android Studio έδειχνε πολλά υποσχόμενο, συγκεκριμένα όσον αφορά την απόκτηση δικαιωμάτων πρόσβασης σε κάμερα μετά από συγκατάθεση του χρήστη, τον εντοπισμό και την μετάφραση κωδικών γρήγορης απόκρισης (*Quick Response Codes - QR Codes*), και την ανάπτυξη λειτουργικής συν εμφάνισιμης διεπαφής χρήστη (*User Interface - UI*).

Πρώτη προσέγγιση ήταν η χρήση του Google Maps API, ανεπτυγμένο από την Google, προκειμένου να προστεθούν οι κατόψεις του επιθυμητού κτιρίου ως επιπλέον επίπεδα πάνω στον χάρτη του *Google Maps*, με δυνατότητες επιλογής

εμφάνισης συγκεκριμένων ορόφων. Η δημιουργία επιπλέον επιπέδου χρησιμοποιώντας τα σχέδια του ΠΑΔΑ είχε ξεκινήσει στο πρόγραμμα QGIS, το οποίο είναι μία δωρεάν και ανοιχτού κώδικα εφαρμογή επιτραπέζιου συστήματος γεωγραφικών πληροφοριών.

Η χρήση του συγκεκριμένου προγράμματος κρίθηκε απαραίτητη, διότι είναι ο συνδετικός κρίκος που ενώνει τα σχέδια των κατόψεων με τις πραγματικές συντεταγμένες του πανεπιστημίου. Έπειτα πρέπει να δημιουργηθούν τα απαραίτητα shapefile διανυσματικά αρχεία, τα οποία, λόγω του ότι προορίζονται για επιπλέον επίπεδο πάνω στον χάρτη, θα πρέπει να μετατραπούν σε μορφή kml μέσω του QGIS. Δυστυχώς, αν και η συγκεκριμένη ιδέα ήταν αρκετά υποσχόμενη, απορρίφθηκε καθώς αντιμετώπισε το εμπόδιο της αντιστοίχισης του επιπέδου με τον χάρτη.

Περαιτέρω έρευνα οδήγησε στην εύρεση ενός project της Google, στόχος του οποίου ήταν η πλοήγηση σε εσωτερικούς χώρους. Αφότου ληφθούν δικαιώματα συνεισφοράς στο συγκεκριμένο project, μπορούν να αποσταλούν τα σχέδια κάτοψης του επιθυμητού κτιρίου, ώστε να αντιστοιχηθούν σωστά από την Google. Ως αποτέλεσμα είναι διαθέσιμα σε κάθε χρήστη μέσω του Google Maps. Δυστυχώς και αυτή η προσέγγιση απορρίφθηκε, διότι το project αυτό έχει πάψει να υποστηρίζεται.

4.2. ΜΕΤΑΦΟΡΑ ΣΤΗΝ UNITY

Ακολούθησαν πειραματισμοί μεταφοράς της ανάπτυξης εφαρμογής από το Android Studio στην Unity, η οποία είναι πρωταρχικά πλατφόρμα ανάπτυξης Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

βιντεοπαιχνιδιών και πλέον κυριάρχη πλατφόρμα ανάπτυξης εφαρμογών επαυξημένης πραγματικότητας. Τα χαρακτηριστικά αυτά την καθιστούν εξαιρετική στο να δημιουργεί τρισδιάστατους χώρους και να βρίσκει βέλτιστες διαδρομές σε αυτόν, ενώ ιχνηλατεί την τοποθεσία του χρήστη χωρίς να χρειάζεται εγκαταστάσεις υλικού.

Μέσα σε λίγες και σύντομες δοκιμές έγινε προφανές πως η ανάπτυξη τέτοιου τύπου εφαρμογής, με χρήση δυνατοτήτων επαυξημένης πραγματικότητας μέσω της Unity, προσφέρει πολλαπλά θετικά. Πιο γρήγορο development με πιο συχνά και εύκολα iterations, μηδενικό κόστος εξοπλισμού δοκιμής και αργότερα εν δυνάμει εγκατάστασης, δυνατά εργαλεία δημιουργίας εικονικού χώρου και αντιστοίχισης του με τον φυσικό χώρο, εξαιρετικές δυνατότητες τοποθεσίας και ιχνηλάτησης χρήστη στον ψηφιακό χώρο, και υπολογισμού συν χάραγμα της βέλτιστης διαδρομής προς τον προορισμό του. Έτσι επίσης αποφεύγονται απαιτήσεις σύνδεσης δικτύου, servers / cloud, αποθήκευσης μεγάλου βάρους δεδομένων για τις προϋπολογισμένες διαδρομές στις συσκευές των χρηστών, και χρονοβόρες χειροκίνητες συσχετίσεις μεταξύ κόμβων.

Το πρώτο και κυρίως οδόφραγμα που παρουσίασε η Unity ήταν το δύσκολο, ελλιπές και δυσάρεστο περιβάλλον διεπαφής χρήστη (*User Interface - UI*) που μπορεί να δημιουργηθεί χωρίς την προσθήκη κάποιου γραφίστα στην ομάδα ανάπτυξης της εφαρμογής. Επιστρέφοντας όμως στα αποτελέσματα από τους αρχικούς πειραματισμούς για την εφαρμογή αυτή, γνωρίζουμε πως η χρήση του Android Studio για το UI μέρος της εφαρμογής είναι όχι μόνο πολύ πιο γρήγορη και εύχρηστη, αλλά και πως το αποτέλεσμα είναι σημαντικά πιο εμφανίσιμο συν

πιο έξυπνα προσαρμόσιμο στις οθόνες έξυπνων συσκευών κάθε μεγέθους και ανάλυσης.

Το επιπλέον κέρδος της προσέγγισης αυτής είναι ένα παράγωγο της Unity με μοναδικό σκοπό την πλοήγηση, ικανό να αποκοπεί από το συγκεκριμένο περιβάλλον διεπαφής και να μεταφερθεί σε οποιαδήποτε άλλη εφαρμογή που χρειάζεται δυνατότητες πλοήγησης, εφόσον η εφαρμογή αυτή μπορεί να προσφέρει στοιχεία αρχικής τοποθεσίας και προορισμού.

Τέλος, για ευκολία χρήσης, εφαρμόστηκε στο νέο UI το QR Scanner που είχε αναπτυχθεί στα πρώτα στάδια πειραματισμού με την εφαρμογή αυτή, ώστε οι χρήστες να μπορούν να διαλέγουν την τωρινή τους τοποθεσία μέσω QR Codes τοποθετημένα σε κομβικά σημεία του φυσικού χώρου.

4.3. ΕΠΙΛΟΓΕΣ ΕΝΤΟΠΙΣΜΟΥ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗΣ ΤΟΠΟΘΕΣΙΑΣ

Ο πιο συχνός τρόπος εντοπισμού τοποθεσίας κάποιας συσκευής είναι χρησιμοποιώντας το Παγκόσμιο Σύστημα Στιγματοθέτησης (*Global Positioning System - GPS*), ανεπτυγμένο από την κυβέρνηση των Ηνωμένων Πολιτειών της Αμερικής. Παρόλο που το GPS είναι ένα επαρκώς ακριβές σύστημα για τον υπολογισμό τοποθεσίας, καθίσταται άχρηστο σε εσωτερικούς χώρους, εξαιτίας της σπανιότητας ή και έλλειψης δορυφορικού σήματος.^[6] Για τον λόγο αυτό, απορρίφθηκε η χρήση GPS για την συγκεκριμένη εφαρμογή.

Για εφαρμογές πλοήγησης σε εσωτερικούς χώρους, έχουν χρησιμοποιηθεί τεχνικές βασισμένες σε τεχνολογίες ασύρματης επικοινωνίας, όπως το WiFi (IEEE 802.11 b/g) και το Bluetooth Low Energy (BLE).

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

Παρόλο που το WiFi διαθέτει υψηλές ταχύτητες μετάδοσης δεδομένων και δυνατότητα μετάδοσης σε μεγάλες αποστάσεις, το κόστος εγκατάστασης και η κατανάλωση ενέργειας είναι σημαντικά μεγαλύτερα σε σύγκριση με τη χρήση BLE.^[7] Για τον λόγο αυτό, απορρίφθηκε η χρήση WiFi για την συγκεκριμένη εφαρμογή.

Η χρήση BLE Bluetooth Beacons για την ιχνηλάτηση τωρινής τοποθεσίας έδειχνε πολλά υποσχόμενη, αλλά με υψηλά κόστη πειραματισμού. Αυτό προσέθεσε το χαμηλό ή ιδανικά μηδαμινό κόστος εγκατάστασης στους κυρίως στόχους της παρούσας διπλωματικής.

Στην συνέχεια οι πειραματισμοί μεταφέρθηκαν σε χρήση HC-05 BLE modules σε συνεργασία με μικροελεγκτές Raspberry Pi Picos. Το δοκίμιο "*A Low Energy IoT Application Using Beacon for Indoor Localization*"^[6] ήταν εξαιρετικά χρήσιμο για τους πειραματισμούς αυτούς, ανεξαρτήτως της διαφοράς τύπου μικροελεγκτών.

Τα HC-05 είναι προγραμματίσιμα με εντολές AT μέσω των μικροελεγκτών που τα ελέγχουν. Θέτοντας τα HC-05 ως "Masters" μπορεί να συνδεθεί σε αυτά κάθε συσκευή που εκτελεί την εφαρμογή ως "Slave" / "Client", και στέλνοντας τα στοιχεία που συλλέγει κάθε "Master" HC-05 σε έναν server μπορούμε να συνυπολογίσουμε την τοποθεσία κάθε "Client".^[6] Η προσέγγιση αυτή θα απαιτούσε συνεχή σύνδεση της εφαρμογής με το διαδίκτυο, και θα την καθιστούσε άχρηστη σε στιγμές που ο Server θα ήταν μη διαθέσιμος, παράγοντες που οδήγησαν στην απόρριψη της προσέγγισης αυτής.

Τελική προσέγγιση είναι ο εντοπισμός και η ιχνηλάτηση τοποθεσίας μέσω τεχνικών Επαυξημένης Πραγματικότητας. Αρχικά χρησιμοποιείται επιλογή τωρινής

τοποθεσίας μέσω του περιβάλλοντος διεπαφής χρήστη, ή ιδανικότερα εντοπίζοντας την τωρινή τοποθεσία μέσω σάρωσης κωδικών QR. Στην συνέχεια, εφαρμόζονται δυνατότητες μηχανικής όρασης και μετρήσεις ψηφιακών αισθητήρων για την ιχνηλάτηση κίνησης βάσει της αρχικά επιλεγμένης τοποθεσίας. Οι πειραματισμοί με τις τεχνολογίες αυτές προσέφεραν καρπούς εξαιρετικά γρήγορα και με ικανοποιητικά αποτελέσματα.

4.4. ΑΝΤΙΣΤΟΙΧΗΣΗ ΦΥΣΙΚΟΥ ΚΑΙ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ

Για την δημιουργία του ψηφιακού χώρου και της αντιστοίχισης του με τον φυσικό λήφθηκαν υπόψη τρεις προσεγγίσεις. Η πρώτη ήταν η χρήση εφαρμογών επαυξημένης πραγματικότητας είτε για την σάρωση του φυσικού χώρου σε τρισδιάστατο μοντέλο, είτε για την τοποθεσία ψηφιακών "ψίχουλων" στην μορφή Cloud Anchors. Απορρίφθηκαν λόγω του κόστους του χώρου αποθήκευσης και την απαίτηση συνεχής πρόσβασης στο διαδίκτυο, αντιστοίχως.

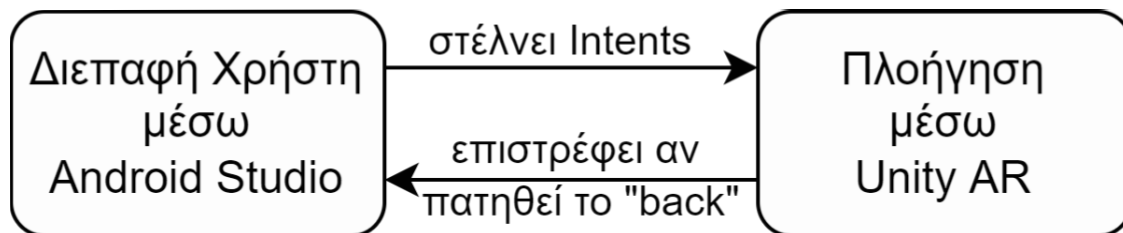
Η δεύτερη προσέγγιση είναι η τοποθέτηση ενός αντικειμένου "πατώματος" στην σκηνή της Unity και μετά η μετακίνηση, κλιμάκωση και περιστροφή πολλαπλών κύβων που πράττουν ως τοίχοι του ψηφιακού χώρου. Το αποτέλεσμα της προσέγγισης αυτής είναι πλήρως ικανοποιητικό, αλλά με δύο σημαντικά αρνητικά μέρη. Το πλήθος των κύβων που χρειάζεται ακόμη και ένας μικρού μεγέθους χώρος, μπορεί να οδηγήσει σε ένα δύσχρηστα υπέρογκο πλήθος κύβων για μεγάλους και περίπλοκους χώρους. Επιπλέον έχουν χαμηλή ακρίβεια απεικόνισης του ψηφιακού χώρου, αν και πλήρως επαρκή για τις δυνατότητες πλοήγησης.

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

Η τρίτη και τελευταία προσέγγιση είναι η δημιουργία του ψηφιακού χώρου σε κάποια εξωτερική εφαρμογή. Για τον σκοπό αυτό στην παρούσα διπλωματική χρησιμοποιήθηκε η εφαρμογή Blender. Το αποτέλεσμα είναι πολύ πιο διαχειρίσιμο από της δεύτερης προσέγγισης, ιδανικό για εγκαταστάσεις με πολλαπλούς ορόφους και πολλαπλά κτίσματα, αλλά πιο δύσκολο και χρονοβόρο όταν απαιτούνται γρήγορες αλλαγές και διορθώσεις.

5. ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή αυτή είναι κυρίως ανεπτυγμένη μέσω Unity και Android Studio. Το μέρος ανεπτυγμένο σε Unity χρησιμοποιεί τεχνολογίες επαυξημένης πραγματικότητας (*Augmented Reality - AR*), λαμβάνει τις τιμές τωρινής τοποθεσίας και επιθυμητού προορισμού από το Android μέρος, υπολογίζει την διαδρομή και την χαράζει τόσο σε έναν χάρτη του χώρου στην πάνω δεξιά γωνία της οθόνης, όσο και στην οπτική της εξωτερικής κάμερας της έξυπνης συσκευής, στοιχεία της οποίας μαζί με μετρήσεις από τους αισθητήρες της έξυπνης συσκευής χρησιμοποιούνται για να ιχνηλατούν την τωρινή τοποθεσία του χρήστη. Στο Android Studio έχει αναπτυχθεί το περιβάλλον διεπαφής του χρήστη (*User Interface - UI*) και θέτει τις τιμές τωρινής τοποθεσίας και επιθυμητού προορισμού, τις οποίες στέλνει στην Unity μέσω intent.



Διάγραμμα 5.1 - Σχέση Android Studio με Unity

5.1. ΜΗΧΑΝΙΣΜΟΣ ΠΛΟΗΓΗΣΗΣ ΜΕΣΩ UNITY

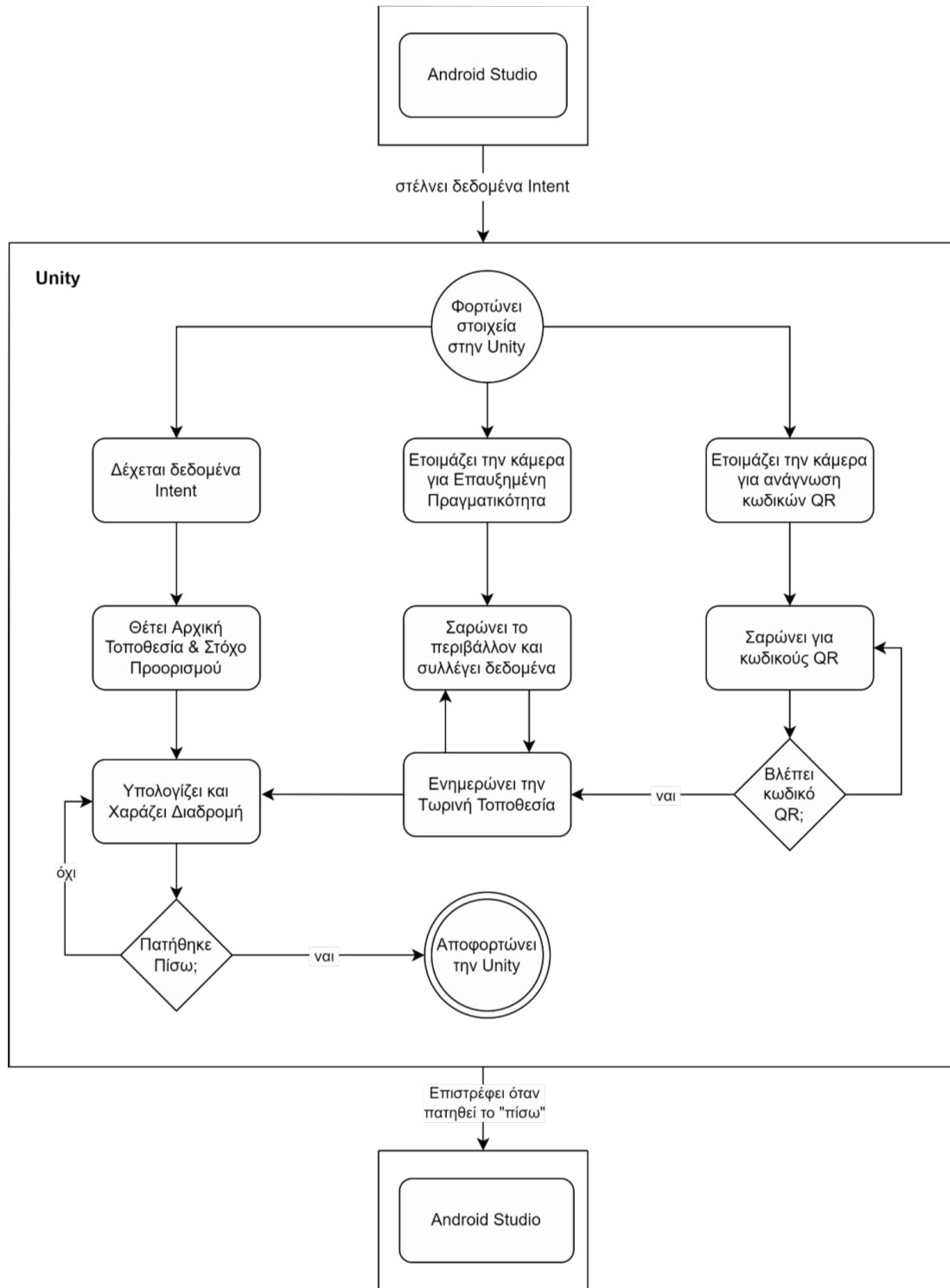
Πρώτο θα αναφερθεί το μέρος ανεπτυγμένο σε Unity, διότι είναι απαραίτητη η κατανόηση του πριν την αναφορά στο μέρος του Android Studio. Αρχικά πρέπει να δημιουργηθεί ένας ψηφιακό χώρο που να αντιστοιχεί στο φυσικό χώρο για

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

τον οποίο προορίζεται η εφαρμογή αυτή, συν να τοποθετηθούν αντικείμενα που να αντιστοιχούν στο πάτωμα και τους τοίχους της φυσικής τοποθεσίας. Βάσει των εικονικών τοίχων αυτών, η Unity μπορεί να υπολογίζει το πλέγμα πλοήγησης (*Navigation Mesh - NavMesh*) που θα χρησιμοποιηθεί για την διαδικασία εύρεσης βέλτιστης διαδρομής μεταξύ τωρινής τοποθεσίας και επιλεγμένου προορισμού.

Για τον υπολογισμό της διαδρομής αυτής, θα τοποθετηθούν αντικείμενα που θα αντιστοιχηθούν στους κόμβους προορισμού και αντικείμενα που θα αντιστοιχηθούν στις εν δυνάμει αρχικές τοποθεσίες / QR points. Οι επιλογές αρχικής τοποθεσίας και προορισμού γίνονται από το μισό του Android το οποίο και τις αποστέλλει στην Unity μέσω Intent.

Στην συνέχεια θα τοποθετηθούν τα αντικείμενα του ARCore και ARFoundation στην σκηνή, δίνοντας συγκεκριμένα προσοχή στην ARCamera που θα χρησιμοποιηθεί για την μηχανική όραση της έξυπνης συσκευής. Επιπλέον θα τοποθετηθεί μια πανοραμική κάμερα εναέριας άποψης που θα χρησιμοποιηθεί για το χάραγμα του χάρτη στην πάνω δεξιά γωνία της οθόνης, και ένα τρισδιάστατο αντικείμενο σε μορφή βέλους που θα υποδεικνύει την τοποθεσία και οπτική γωνία του χρήστη.



Διάγραμμα 5.2 - Διάγραμμα ροής της Unity

5.1.1. ΨΗΦΙΑΚΗ ΑΠΕΙΚΟΝΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΧΩΡΟΥ

Πρώτο και εξαιρετικά κομβικό βήμα είναι σωστή αντιστοίχιση του ψηφιακού χώρου με τον φυσικό χώρο για τον οποίο προορίζεται η εφαρμογή. Υπάρχουν τρεις τρόποι να προσεγγιστεί. Ο πρώτος και στον οποίο θα γίνει μόνο μια σύντομη αναφορά, είναι η δημιουργία μονοπατιών με χρήση δυνατοτήτων επαυξημένης πραγματικότητας, περπατώντας στο φυσικό χώρο από κάποια αρχική τοποθεσία σε κάποιον τελικό προορισμό, και αποθηκεύοντας το μονοπάτι αυτό είτε εσωτερικά στην εφαρμογή είτε, πιο πιθανά, στο cloud.^[9] Τα κυρίως αρνητικά στοιχεία του τρόπου αυτού, και οι λόγοι για τους οποίους απορρίφθηκε, είναι ο όγκος των δεδομένων που απαιτούν αποθήκευση σε κάποιον server, που συνεπάγεται ότι και η τελική εφαρμογή θα χρειαζόταν σύνδεση στο διαδίκτυο, και επίσης η ανάγκη κίνησης στο φυσικό χώρο για διόρθωση προβληματικών μονοπατιών και κόμβων.

Ο δεύτερος τρόπος αντιστοίχισης του φυσικού χώρου με τον ψηφιακό, είναι η χειροκίνητη δημιουργία του τρισδιάστατου ψηφιακού χώρου. Για την προσέγγιση αυτή, απαιτείται μια επαρκούς ποιότητας εικόνα των σχεδίων του χώρου για τον οποίο προορίζεται η εφαρμογή. Σαν πρώτο βήμα πρέπει να τοποθετηθεί ένα δισδιάστατο ορθογώνιο παραλληλόγραμμο που χρησιμοποιεί ως υφή (*Texture - Material*) την εικόνα αυτή των σχεδίων του χώρου. Επιτακτικό είναι οι αναλογίες διαστάσεων της φωτογραφίας και του ορθογώνιου παραλληλόγραμμου να είναι ίσες, αλλιώς υπάρχει το ρίσκο να μην αντιστοιχεί πλήρως ο ψηφιακός μας χώρος με τον φυσικό, προκαλώντας προβλήματα τόσο στην απεικόνιση της διαδρομής του χρήστη όσο και στην ιχνηλάτηση της τωρινής του τοποθεσίας.

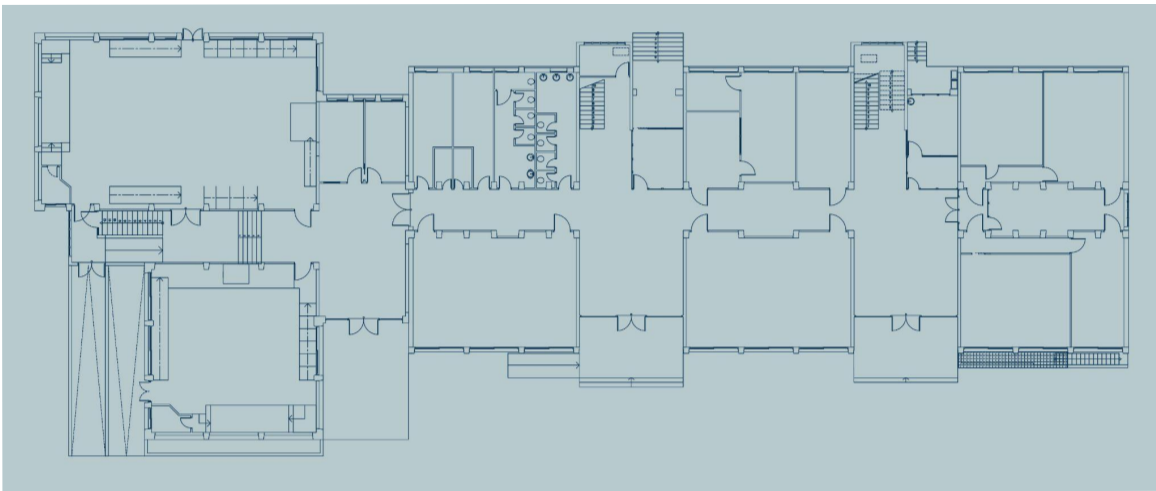
Συνεχίζοντας με την δημιουργία του ψηφιακού χώρου τοποθετούνται αντικείμενα μορφής κύβων, τα οποία μετακινούνται (*translate*), περιστρέφονται (*rotate*) και κλιμακώνονται (*scale*) ώστε να αντιστοιχούν στους τοίχους όπως τους εκφράζουν τα σχέδια του φυσικού χώρου, φροντίζοντας οι βάσεις τους να τέμνουν το ορθογώνιο παραλληλόγραμμο που έχουμε τοποθετήσει ως πάτωμα. Όταν πλέον έχουν τοποθετηθεί και παραμετροποιηθεί όλοι οι κύβοι-τοίχοι, τότε μπορούμε και προαιρετικά να αλλάξουμε την υφή (Texture - Material) του παραλληλογράμμου-πατώματος. Η προσέγγιση αυτή επιτρέπει γρήγορες αλλαγές και διορθώσεις, αλλά είναι εξαιρετικά χρονοβόρα για μεγάλους και περίπλοκους χώρους.

Ο τρίτος τρόπος αντιστοίχισης του φυσικού χώρου με τον ψηφιακό, είναι ο πιο εύκολος και γρήγορος, αλλά απαιτεί την αποτύπωση των σχεδίων σε δεδομένα αντί φωτογραφίας, όπως π.χ. τα σχέδια του Πανεπιστημίου Δυτικής Αττικής, διαθέσιμα σε μορφή XML στην ιστοσελίδα campusplan.uniwa.gr. Τα δεδομένα αυτά μπορούν να εισαχθούν σε προγράμματα δημιουργίας και επεξεργασίας τρισδιάστατων μοντέλων, όπως το Blender και το AutoCad 3D, και να μετατραπούν σε τρισδιάστατα μοντέλα, τα οποία με την σειρά τους θα εισαχθούν στην Unity. Ανάλογα με την μορφή των ψηφιακών σχεδίων, θα χρειαστούν από ελάχιστες έως πολλές διορθώσεις στα τρισδιάστατα μοντέλα αυτά, και αν τα σχέδια είναι ελλιπή θα χρειαστούν και χειροκίνητες προσθήκες. Βάσει των αστάθμητων αυτών παραγόντων, ο τρόπος αυτός μπορεί να είναι ο πιο γρήγορος, απλός και ασφαλής, ή μπορεί ο δεύτερος τρόπος να παραμένει η πιο σοφή προσέγγιση.

5.1.1.1. ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ ΑΠΟ ΚΑΤΟΨΗ ΣΧΕΔΙΩΝ

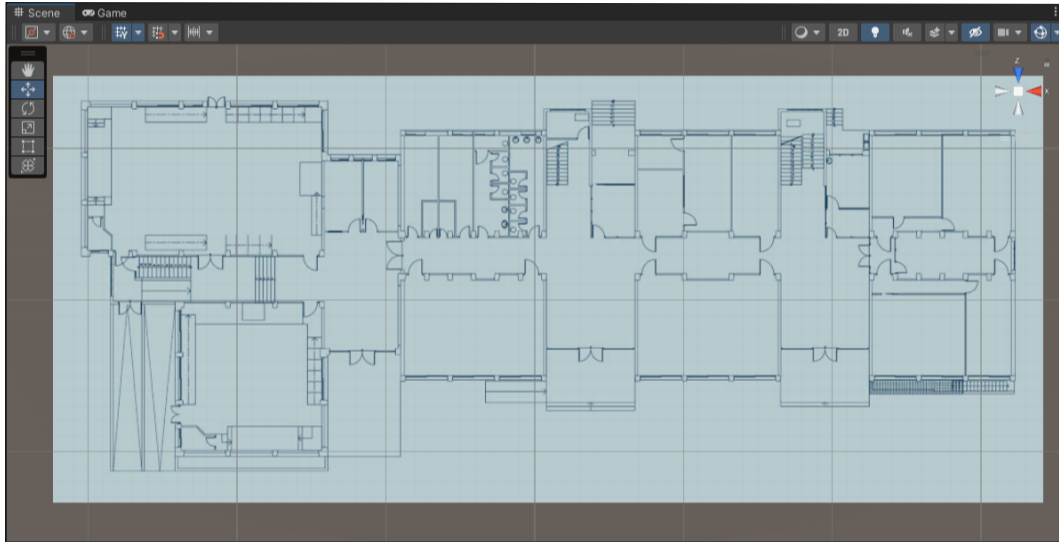
Σαν αρχή χρειάζεται η κάτοψη των σχεδίων του χώρου για τον οποίο προορίζεται η εφαρμογή, η οποία πρέπει και να ψηφιοποιηθεί, βγάζοντας την φωτογραφία, περνώντας την από scanner, ή με οποιοδήποτε δυνατό τρόπο.

Όταν η κάτοψη είναι ψηφιοποιημένη, πρέπει να περαστεί στο πρόγραμμα στο οποίο θα δημιουργηθούν τα τρισδιάστατα αντικείμενα που αντιστοιχούν στους τοίχους. Στο συγκεκριμένο κεφάλαιο θα περιγραφεί το workflow μέσω της Unity.



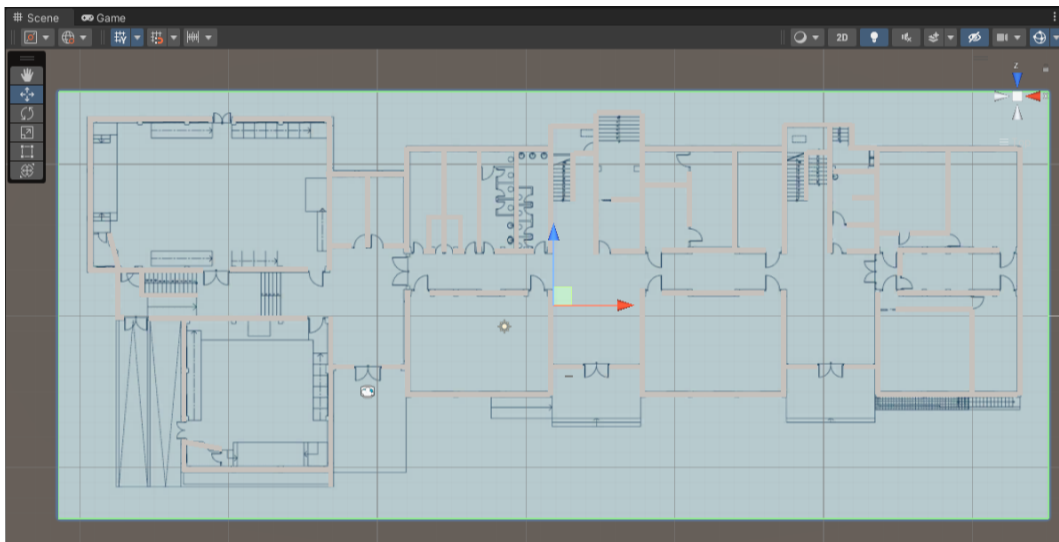
Εικόνα 5.1 - Κάτοψη Τμήματος Πληροφορικής του ΠΑΔΑ

Τοποθετείται ένα δισδιάστατο ορθογώνιο παραλληλόγραμμο και κλιμακώνεται ώστε να έχει ίση αναλογία διαστάσεων με την φωτογραφία, προκειμένου να μην προκύψουν παραμορφώσεις της κάτοψης και να μην χρειάζεται να ασχοληθούμε με UV Unwrapping. Π.χ. με μια φωτογραφία ανάλυσης 1920x1080 θα μπορούσε να εκτελεστεί κλιμάκωση του παραλληλόγραμμου κατά 1.92 και 1.08 αντιστοίχως.



Εικόνα 5.2 - Κάτοψη τοποθετημένη στην Unity

Στην συνέχεια τοποθετούνται μετακινούνται και κλιμακώνονται κύβοι, μέχρι να καλύψουν κάθε τοίχο της κάτοψης, δίνοντας συγκεκριμένη προσοχή στο να μην καλύπτουν πόρτες και άλλα κενά που θα μπορούσαν να χρησιμοποιηθούν ως διαδρομές.



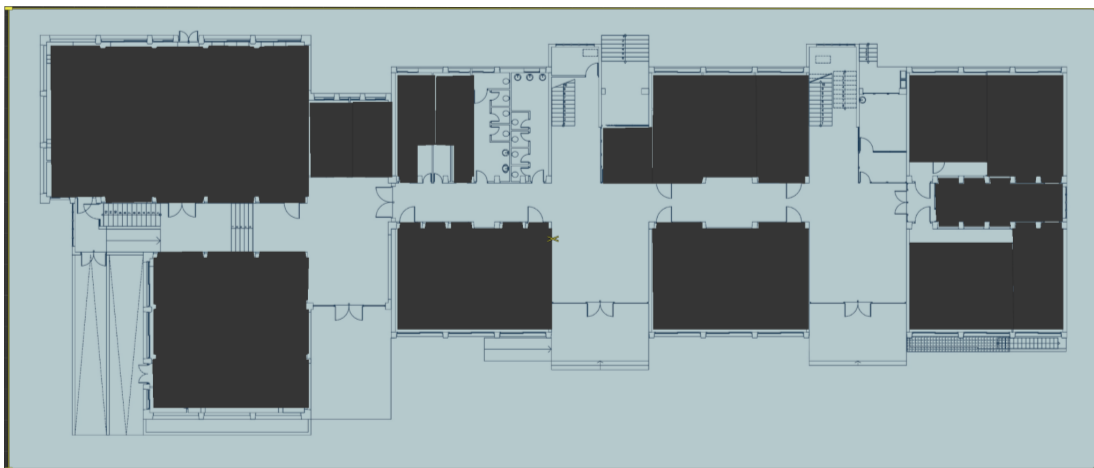
Εικόνα 5.3 - Κάτοψη με αντικείμενα κύβοι-τοιχοί

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

5.1.1.2. ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΧΩΡΟΥ ΑΠΟ ΔΕΔΟΜΕΝΑ

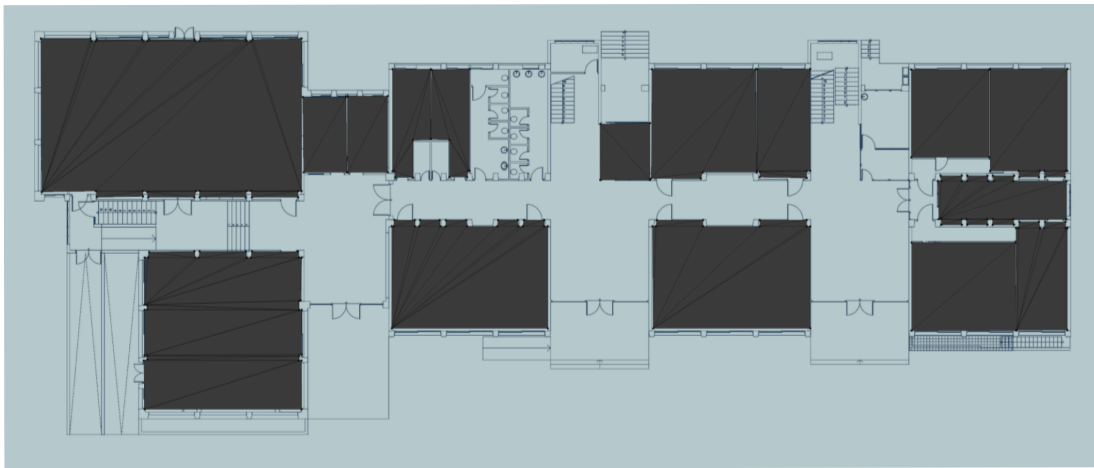
Σαν αρχή χρειάζονται τα σχέδια του χώρου σε κάποια μορφή δεδομένων. Για παράδειγμα, υπάρχουν διαθέσιμα τα σχέδια των αιθουσών του Πανεπιστημίου Δυτικής Αττικής σε μορφή κλιμακούμενων διανυσματικών γραφικών (*Scalable Vector Graphics - SVG*). Στην συνέχεια πρέπει να εισαχθούν τα δεδομένα αυτά στην εφαρμογή δημιουργίας και επεξεργασίας τρισδιάστατων γραφικών της επιλογής μας.

Η εφαρμογή που επιλέχθηκε για αυτό το σκοπό είναι το Blender. Επιλέγοντας *File → Import → Scalable Vector Graphics* και από εκεί το αρχείο SVG στο οποίο είναι χαραγμένα τα σχέδια του χώρου, θα προστεθούν αντικείμενα τύπου *Curve*. Αν υπάρχουν τα σχέδια του χώρου και σε μορφή εικόνας, θα ήταν χρήσιμο να προστεθούν ως αντικείμενο τύπου *Image - Background*, ώστε να χρησιμοποιηθεί σαν σημείο αναφοράς για τις διορθώσεις και αλλαγές που θα ακολουθήσουν.

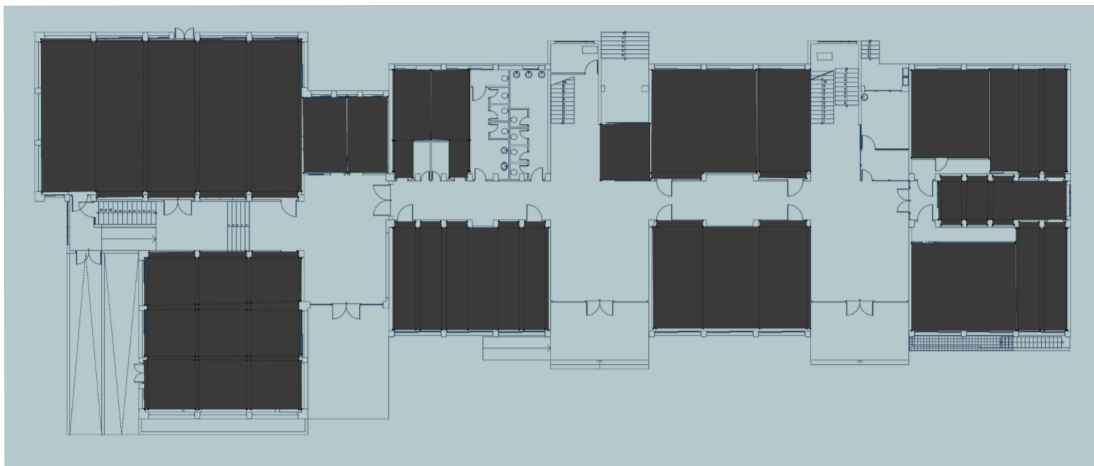


Εικόνα 5.4 - Δωμάτια εισαγμένα μέσω αρχείου SVG

Προκειμένου να γίνουν τα αντικείμενα αυτά πλήρως επεξεργάσιμα σε τρισδιάστατο χώρο, πρέπει να επιλεγθούν όλα εκτός του Background Image και να μετατραπούν από αντικείμενα τύπου Curve σε αντικείμενα τύπου Object. Στην συνέχεια, θα πρέπει να τοποθετηθούν σε πιο ακριβείς θέσεις σύμφωνα με το background image συν να καθαριστούν τα vertices και τα lines που τα αποτελούν.

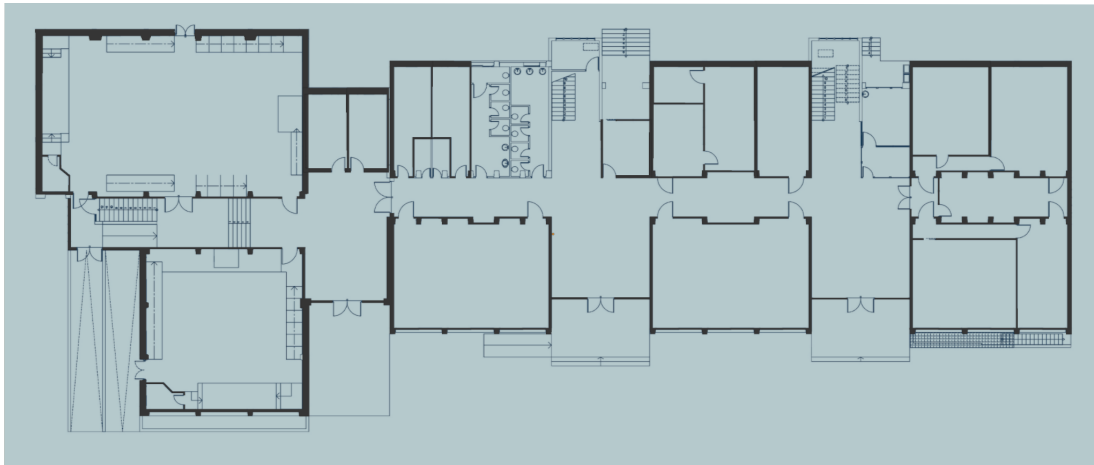


Εικόνα 5.5 - Δωμάτια μορφής Object, χωρίς καθάρισμα



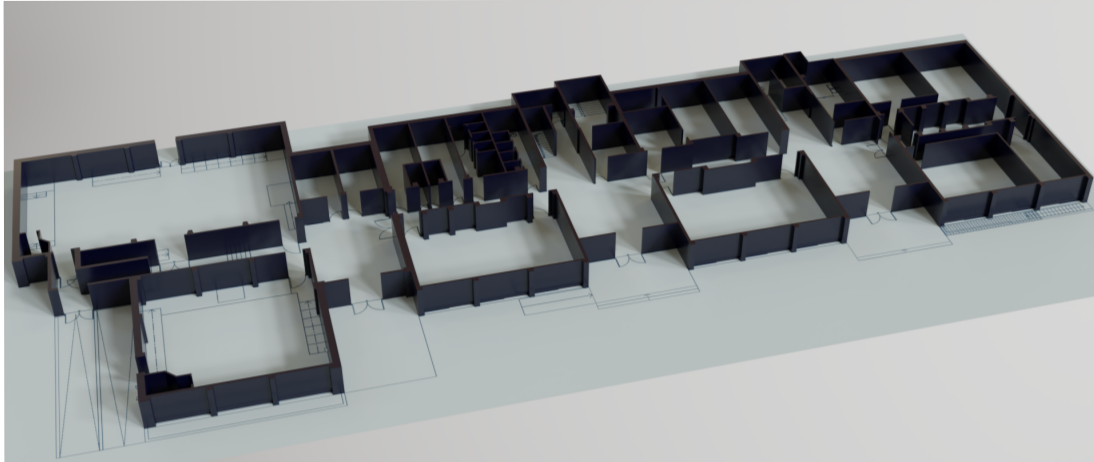
Εικόνα 5.6 - Δωμάτια μορφής Object, καθαρισμένα

Πλέον έχουν δημιουργηθεί αντικείμενα που αντιστοιχούν σε δωμάτια, ενώ η εφαρμογή χρειάζεται αντικείμενα που να αντιστοιχούν σε τοίχους. Εφόσον οι πλευρές των δωματίων που κατασκευάστηκαν ακουμπούν τις έξω πλευρές των τοίχων της κάτοψης, πρέπει να εκτελεστεί ένθεση (*inset*) σε κάθε δωμάτιο και να κλιμακωθεί (*scale*) η ένθεση αυτή προς τα μέσα μέχρι να ακουμπήσει τις μέσα πλευρές των τοίχων. Διαγράφοντας τις εσωτερικές επιφάνειες (*faces*) των αντικειμένων, απομένουν οι τοίχοι που αντιστοιχούν στα δωμάτια αυτά. Τα στοιχεία που μπορούν να αποκτηθούν από την ιστοσελίδα του ΠΑΔΑ είναι ελλιπή, π.χ. λείπουν αίθουσες, αλλά μετά από χειροκίνητο καθάρισμα φτάνουμε σε ένα αρκετά ικανοποιητικό σημείο.



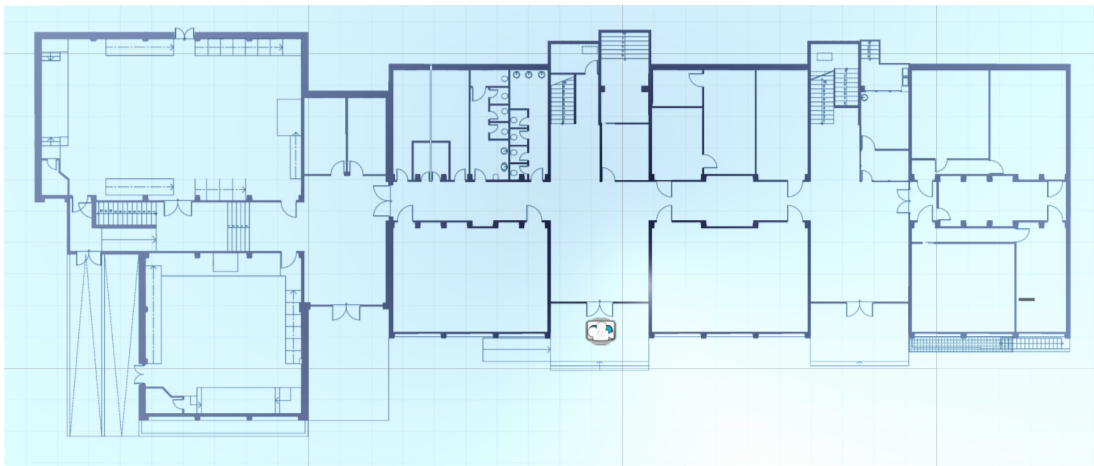
Εικόνα 5.7 - Τοίχοι μορφής *Object*, με διορθώσεις

Στην συνέχεια ενώνονται όλα τα αντικείμενα σε ένα μεγάλο αντικείμενο ώστε να αφαιρεθούν κοινοί τοίχοι και γίνουν ό,τι τελικές αλλαγές και προσθήκες χρειάζονται. Όταν πλέον βρίσκεται σε ικανοποιητικό σημείο, εξωθείται (*extrude*) το μοντέλο στον άξονα Z, ώστε να αποκτήσει ύψος.



Εικόνα 5.8 - Υψηλής ποιότητας render του τελικού μοντέλου

Για να μεταφερθεί το τρισδιάστατο μοντέλο στην Unity, εξάγεται σε μορφή Filmbox (FBX) μέσω του Blender και στην συνέχεια εισάγεται στην Unity εκτελώντας *Assets* → *Import New Assets*. Στα επόμενα κεφάλαια θα χρησιμοποιηθεί το μοντέλο αυτό, αντί των τοίχων που δημιουργήθηκαν μέσω της Unity στο κεφάλαιο 5.1.1.1.

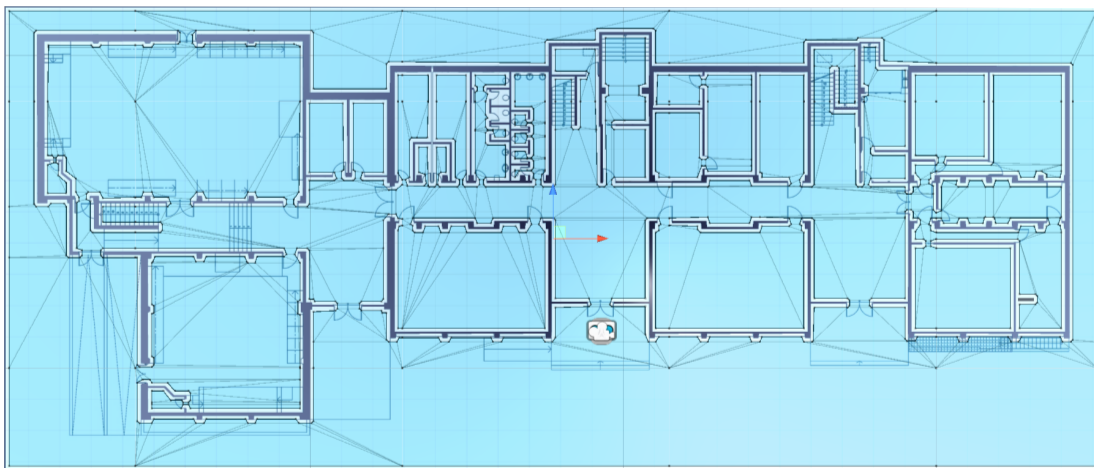


Εικόνα 5.9 - Τρισδιάστατο μοντέλο τοποθετημένο στην Unity

5.1.2. ΠΛΕΓΜΑ ΠΛΟΗΓΗΣΗΣ ΚΑΙ ΕΠΙΠΛΕΟΝ ΑΝΤΙΚΕΙΜΕΝΑ

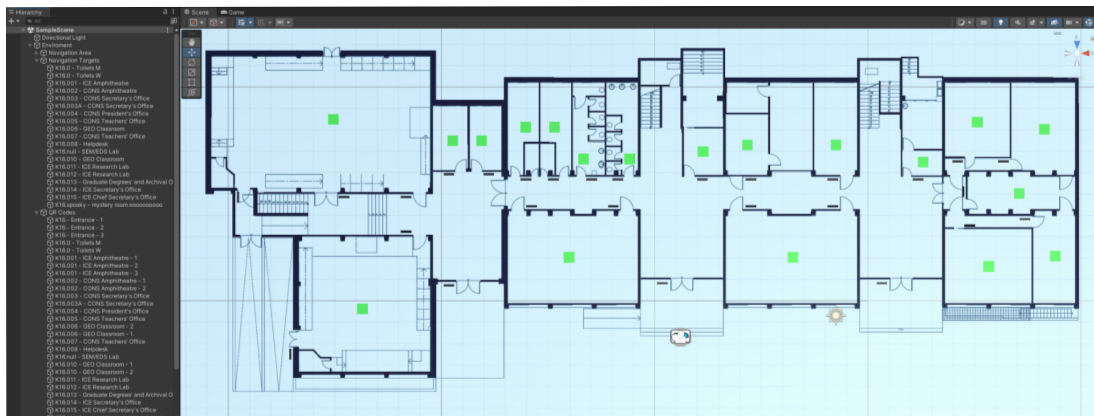
Αρχικά πρέπει να δημιουργηθεί το πλέγμα πλοήγησης (Navigation Mesh - NavMesh), το οποίο εκφράζει που μπορεί να "περπατήσει" η εφαρμογή. Τα δεδομένα αυτά θα χρησιμοποιηθούν για τον υπολογισμό των βέλτιστων διαδρομών από την εκάστοτε τωρινή τοποθεσία προς τον εκάστοτε προορισμό.

Για να γίνει αυτό, πρέπει να τεθούν όλα τα αντικείμενα που αντιστοιχούν στο πάτωμα και τους τοίχους ως "Static", φροντίζοντας οι βάσεις των τοίχων να τέμνουν το πάτωμα. Στο παράθυρο Navigation Mesh της Unity, συγκεκριμένα στο tab Bake, ρυθμίζεται το πάχος του πράκτορα, φροντίζοντας να είναι αρκετά λεπτό ώστε να περνάει από όλες τις επιθυμητές διαδρομές και αρκετά παχύ ώστε να μην περνάει από μη έγκυρα μέρη όπως π.χ. τοίχους, επιλέγουμε Bake και έχουμε έτοιμο τον χώρο στον οποίο θα υπολογιστούν και θα χαραχθούν οι διαδρομές.



Εικόνα 5.10 - Κάτοψη με υπολογισμένο NavMesh

Στην συνέχεια πρέπει να τοποθετηθούν αντικείμενα που να αντιστοιχούν τόσο σε κάθε χώρο, όσο και στις εν δυνάμει αρχικές τοποθεσίες που θα βρίσκονται τοποθετημένα και τα QR Codes. Με πράσινους κύβους θα σηματοδοτηθούν οι προορισμοί, ενώ με μαύρες πλάκες τα σημεία QR. Στην τελική εφαρμογή τα αντικείμενα αυτά είναι κρυμμένα από τον χρήστη και υπάρχουν μόνο για την λειτουργικότητα της εφαρμογής.



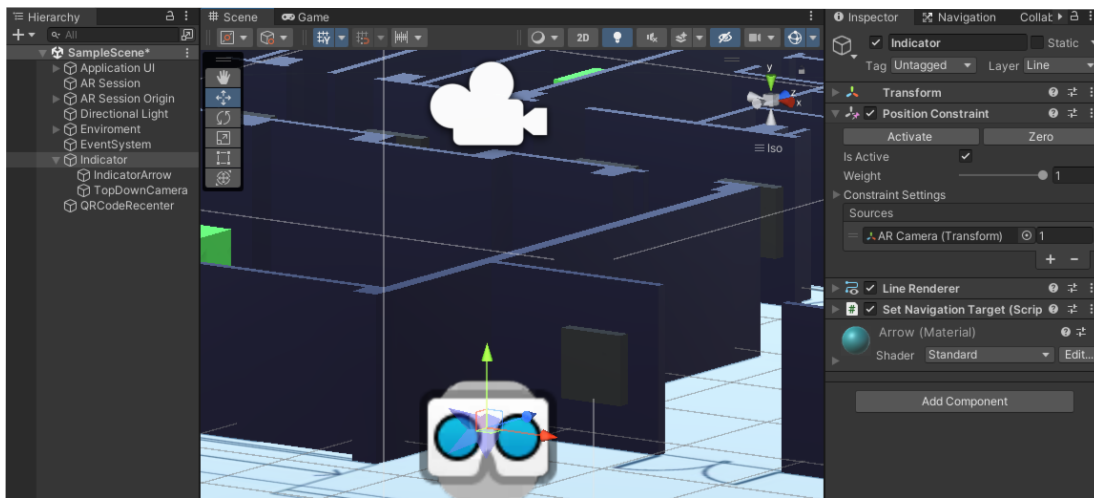
Εικόνα 5.11 - Ψηφιακός χώρος με αντικείμενα προορισμού και QR

Σαν επόμενα βήματα πρέπει να τοποθετηθούν μια σειρά από αντικείμενα του ARFoundation. Στα Build Settings του Project μας, γίνεται αλλαγή πλατφόρμας σε Android, στα Project Settings στην σελίδα "XR Plugin Manager" εγκαθίσταται το Plugin και στην σελίδα του Android επιλέγεται το ARCore. Επιπλέον εγκαθίσταται το ARFoundation μέσω του το Package Manager. Πλέον η εφαρμογή έχει πρόσβαση στις δυνατότητες και τα αντικείμενα των ARCore και ARFoundation, άρα μπορεί να προσθέσει στην σκηνή ένα αντικείμενο "AR Session Origin", το οποίο και ενθυλακώνει το αντικείμενο "AR Camera" που θα χρησιμοποιηθεί ως πηγή πληροφοριών για την εφαρμογή και ως κυρίως οπτική του χρήστη.

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

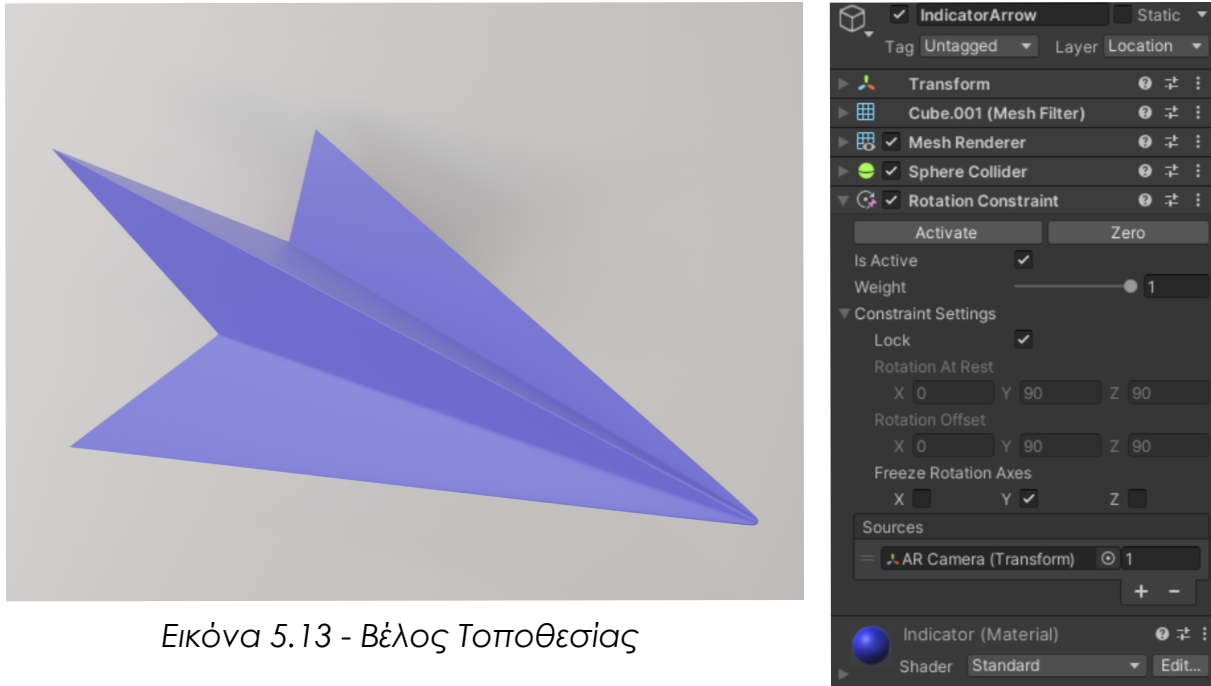
Αργότερα θα εφαρμοστεί κώδικας που θα μετακινεί το αντικείμενο στην αρχική θέση επιλογής του χρήστη, βάσει των QR Points.

Στην συνέχεια τοποθετείται ένα αντικείμενο που θα εκφράζει στον χρήστη την τωρινή του τοποθεσία, και μια επιπλέον κάμερα για την πανοραμική εναέρια άποψη. Η τοποθεσία των δύο αυτών αντικειμένων πρέπει να ανανεώνεται με την θέση του χρήστη στον ψηφιακό χώρο, άρα θα τα τοποθετήσουμε μέσα σε ένα κοινό "Empty" στο οποίο θα προσθέσουμε ένα "Position Constraint Component" με πηγή την ARCamera.



Εικόνα 5.12 - Αντικείμενα τοποθεσίας χρήστη

Για το αντικείμενο που εκφράζει την τωρινή τοποθεσία του χρήστη επιλέχθηκε ένα βέλος, διότι μπορεί να εκφράσει όχι μόνο την τοποθεσία αλλά και την οπτική γωνία του χρήστη. Κατασκευάστηκε στο Blender και εισήχθη στην Unity όπου του προστέθηκε ένα "Rotation Constraint Component" περιορισμένο μόνο στον άξονα Y, ώστε να περιστρέφεται μόνο παράλληλα προς το έδαφος.



Εικόνα 5.13 - Βέλος Τοποθεσίας

Επίσης απαιτείται κάτι που να εκφράζει την διαδρομή που πρέπει να ακολουθήσει ο χρήστης. Για τον σκοπό αυτό θα προστεθεί ένα "*Line Renderer Component*" στο "*Empty*" που περιλαμβάνει το βέλος και την εναέρια κάμερα. Για το πάχος θα οριστεί μια σιγμοειδή καμπύλη τύπου *inverse smoothstep*, ώστε να μειώνεται το πάχος της όσο απομακρύνεται από την κάμερα. Επίσης θα προστεθεί ένα αντίγραφο των τοίχων, με Material τύπου *Occlusion*, το οποίο τους καθιστά αόρατους μα ικανούς να κάνουν *occlude*, στην πράξη να "κρύψουν" την γραμμή διαδρομής όταν στρίβει πίσω τους.

5.1.3. ΚΑΜΕΡΕΣ ΣΚΗΝΗΣ UNITY ΚΑΙ ΒΑΣΙΚΟ UI

Αρχικά θα προστεθεί ένα "*Canvas*" στην σκηνή. Στο "*Canvas*" αυτό θα γίνεται *render* η οπτική της "*AR Camera*", δηλαδή η οπτική της φυσικής εξωτερικής

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

κάμερας της έξυπνης συσκευής του χρήστη, και όποια ψηφιακά αντικείμενα έχει επιτραπεί να κάνουν render μέσω του Culling Mask. Προκειμένου να προσαρμόζεται η οπτική σωστά σε κάθε μέγεθος και ανάλυση οθόνης έξυπνων συσκευών, θα τεθεί στο "Canvas Scaler Component" του "Canvas" η ρύθμιση "UI Scale Mode" ως "Scale with Screen Size".

Επίσης θα προστεθεί μια δεύτερη πανοραμική κάμερα εναέριας άποψης, της οποίας ο σκοπός θα είναι να ενημερώνει το χάρτη στην πάνω δεξιά γωνία της οθόνης. Προκειμένου να ακολουθεί την τοποθεσία του χρήστη, θα δημιουργηθεί ένα αντικείμενο Camera εμφωλευμένο στο "Empty" αντικείμενο με το "Location Constraint Component". Θα μεταφερθεί, θα περιστραφεί και θα κλιμακωθεί η κάμερα αυτή, ώστε να διατηρεί εναέρια άποψη της τωρινής θέσης του χρήστη, και θα τεθεί η προβολή της ως ορθογραφική, προκειμένου ο συγκριτικά μικρός χάρτης τον οποίο θα ενημερώνει να είναι ευανάγνωστος.

Για να μεταφερθεί η οπτική της εναέριας κάμερας στο περιβάλλον διεπαφής του χρήστη (User Interface - UI), θα χρειαστεί να δημιουργήσουμε μια υφή τύπου "Render Texture" και να τεθεί ως το "Target Texture" στις επιλογές της κάμερας αυτής. Ως αποτέλεσμα, σε κάθε καρέ που ανανεώνεται η εναέρια κάμερα, θα ανανεώνεται και το Target Texture με τα δεδομένα της οπτικής της. Προσθέτοντας ένα αντικείμενο Raw Image, εμφωλευμένο στο Canvas Object, και θέτοντας το προαναφερθέν Target Texture ως Texture του Render Image, έχει τοποθετηθεί ένα Minimap UI Element στην οπτική του χρήστη.

Συχνά χαρακτηριστικά των μοντέρνων έξυπνων συσκευών έχουν γίνει οι οθόνες που καλύπτουν ολόκληρο το σώμα της συσκευής (Edge-to-Edge) και οι

εσωτερικές κάμερες που ως αποτέλεσμα βρίσκονται τοποθετημένες μέσα στα πλαίσια της οθόνης, είτε ως εγκοπή (*Notch*) ή ως οπή (*Punch Hole*). Αυτή η σχετικά μοντέρνα αλλά εξαιρετικά διαδεδομένη τάση σχεδιασμού οδήγησε στο να αναθεωρηθεί αρκετές φορές την τοποθεσία του Mini Map. Βάσει προσωπικών μας και δημοσίων δημοσκοπήσεων^[11], αποδείχτηκε πως μεταξύ του κέντρου και των δύο γωνιών, η πάνω δεξιά γωνία είναι η πιο σπάνια και ανεπιθύμητη τοποθεσία εσωτερικής κάμερας τύπου notch ή punch through, άρα και η πιο ασφαλής τοποθεσία για το Minimap UI Element. Για την σωστή κλιμάκωση του Minimap UI Element ανεξαρτήτως ανάλυσης οθόνης, πρέπει τα "Anchor Presets" του "Rect Transform Component" να έχουν τεθεί ως "top right".

Τέλος πρέπει να οριστούν τα "Culling Masks" για τις δύο κάμερες της σκηνής μας. Κάθε κάμερα έχει το δικό της "Culling Mask", μια λίστα επιλογών με όλα τα Layers της σκηνής μας. Κάθε αντικείμενο που ανήκει σε επιλεγμένο Layer θα γίνει Render κανονικά από την κάμερα αυτή, ενώ κάθε αντικείμενο σε μη επιλεγμένο Layer θα εξαιρεθεί (*Cull*) και δεν θα γίνει render από την κάμερα αυτή. Τα αντικείμενα που έχουν τοποθετηθεί στα προηγούμενα κεφάλαια μπορούν να χωριστούν σε τέσσερα γενικευμένα Layer. Το Layer που θα κάνουν Render και οι δύο κάμερες, το Layer που θα κάνει Render η ARCamera και Cull η εναέρια κάμερα, το Layer που θα κάνει Cull η ARCamera και Render η εναέρια, και τέλος το Layer που θα κάνουν Cull και οι δύο κάμερες.

Για την κυρίως κάμερα, το μόνο αντικείμενο που είναι θεμιτό να γίνεται Render είναι η γραμμή που εκφράζει την βέλτιστη διαδρομή μεταξύ της τωρινής θέσης και του επιλεγμένου προορισμού του χρήστη. Επίσης η γραμμή αυτή πρέπει να

χαράζεται και στο Mini Map, άρα θα τοποθετηθεί το "Line Renderer" στο Layer που κάνουν Render και οι δύο κάμερες.

Το αόρατο αντίγραφο των τοίχων είναι χρήσιμο μόνο για την ARCamera, μιας και η εναέρια πανοραμική οπτική της δεύτερης κάμερας καθιστά αχρείαστο το κρύψιμο της γραμμής διαδρομής πίσω από τοίχους. Επίσης με τα δύο αντίγραφα των τοίχων να είναι ίσα και να τέμνονται όλες οι πλευρές τους, μπορεί να προκύψουν σφάλματα απόδοσης. Άρα για λόγους ασφαλούς πρακτικής, τοποθετούνται οι αόρατοι τοίχοι στο Layer που κάνει Render η ARCamera και Cull η εναέρια κάμερα.

Το βέλος που εκφράζει την τοποθεσία και οπτική του χρήστη, συν οι τοποθεσίες των QR Codes, είναι θεμιτό να είναι ορατά στο Mini Map ώστε να προσφέρουν στον χρήστη όσες πληροφορίες χρειάζεται για την αντιστοίχιση της φυσικής τοποθεσίας και της ψηφιακής τοποθεσίας του. Τα αντικείμενα αυτά θα κατηγοριοποιηθούν στο Layer που κάνει Cull η ARCamera και Render η εναέρια κάμερα.

Τέλος τα υπόλοιπα αντικείμενα θα τοποθετηθούν στο Layer που κάνουν Cull και οι δύο κάμερες. Συγκεκριμένα οι κύβοι προορισμού θα τεθούν ως κρυμμένοι και στις δύο οπτικές, διότι δεν προσφέρουν χρήσιμες πληροφορίες στον χρήστη, ενώ προσθέτουν οπτικό θόρυβο. Το τέλος της διαδρομής τους, άρα και ο προορισμός τους, σηματοδοτείται από το τέλος του Line Renderer.

5.1.4. SCRIPTS THΣ UNITY

Το μεγαλύτερο μέρος της λογικής της εφαρμογής περιέχεται στις ενσωματωμένες λειτουργίες των ARSession, AR Camera, Line Renderer και Navigation Mesh που τοποθετήθηκαν και κατασκευάστηκαν στο κεφάλαιο 5.1.2, παρ'όλα αυτά θα χρειαστούν μερικές επιπλέον λειτουργίες οι οποίες θα αναπτυχθούν σε C# και εφαρμοστούν μέσω Scripts.

Τα Scripts αυτά θα είναι παράγωγα της *MonoBehaviour Class* ώστε να εκτελούνται με την εκκίνηση της εφαρμογής. Περιλαμβάνουν μια συνάρτηση *Start()* που εκφράζει ό,τι πρέπει να γίνει με την εκκίνηση του Script, και μια συνάρτηση *Update()* που εκφράζει ό,τι πρέπει να γίνεται σε κάθε κύκλο λογικής της εφαρμογής. Επίσης θα περιλαμβάνουν μεταβλητές χαρακτηρισμένες ως *[SerializedField]* που επιτρέπουν την αντιστοίχιση τους με αντικείμενα τοποθετημένα στην σκηνή της Unity, απλώς κάνοντάς τα drag-and-drop από την σκηνή στο αντίστοιχο κελί του *Script Component*.

5.1.4.1. ΕΠΙΚΟΙΝΩΝΙΑ UNITY ΚΑΙ ANDROID STUDIO

Το πρώτο Script που χρειάζεται, είναι αυτό που διαχειρίζεται την κίνηση και επικοινωνία μεταξύ της Unity και του Android Studio. Πρέπει να μπορεί να λάβει και να χρησιμοποιήσει τα Intent από το Android Studio, και να εκτελεί την κατάλληλη εντολή εξόδου ώστε να μεταφερθεί η εφαρμογή επιτυχώς πίσω στο Android Studio.

Για την λήψη Intent, ελέγχεται σε Compile Time με την χρήση *#if* αν η εφαρμογή γίνεται Compile μέσω Android Studio και όχι μέσω Unity. Αν οι

συνθήκες αυτές ισχύουν, η Unity θα αρχικοποιήσει μια μεταβλητή *AndroidJavaObject* ως *"getIntent"* και θα την χρησιμοποιήσει για να λάβει τα Intent που έστειλε το Android Studio. Τα Intent αυτά θα χρησιμοποιηθούν αργότερα για τον ορισμό τωρινής τοποθεσίας και επιλεγμένου προορισμού. Ο κώδικας αυτός αναφέρεται στο Παράρτημα Κώδικα, στον πίνακα "Κώδικας Unity 1".

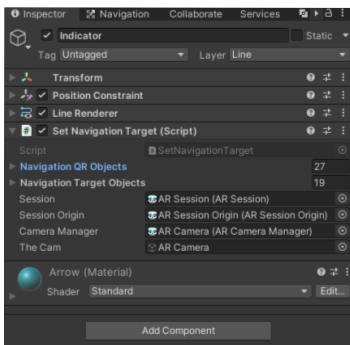
Για την επιστροφή από Unity σε Android Studio, η συνάρτηση *Update()* του συγκεκριμένου Script, θα ελέγχει αν η εφαρμογή εκτελείται σε λογισμικό Android και αν έχει πατηθεί το "πίσω". Αν οι συνθήκες αυτές ισχύουν, η Unity θα εκτελέσει την εντολή *Application.Unload()* και ο κώδικας του Android Studio θα διαχειριστεί τα υπόλοιπα. Ο κώδικας αυτός αναφέρεται στο Παράρτημα Κώδικα, στον πίνακα "Κώδικας Unity 2".

5.1.4.2. ΥΠΟΛΟΓΙΣΜΟΣ ΔΙΑΔΡΟΜΗΣ ΚΑΙ ΙΧΝΗΛΑΤΗΣΗ ΤΟΠΟΘΕΣΙΑΣ

Το δεύτερο Script που χρειάζεται, είναι αυτό που διαχειρίζεται όλη την διαδικασία πλοήγησης. Θέτει τις μεταβλητές τοποθεσίας και προορισμού, υπολογίζει την διαδρομή του χρήστη, και τέλος χαράζει την διαδρομή και ιχνηλατεί την τοποθεσία του χρήστη.

Έχοντας λάβει τις τιμές τοποθεσίας και προορισμού από το Android Studio, πρέπει να εντοπίσει τα αντικείμενα σκηνής στην Unity με τα οποία αντιστοιχούν. Για να επιτευχθεί αυτό, χρειάζεται μία λίστα που περιλαμβάνει τα αντικείμενα προορισμού και μία λίστα που περιλαμβάνει τα αντικείμενα τοποθεσίας. Οι λίστες αυτές πρέπει να τεθούν ως *[SerializeField]*.

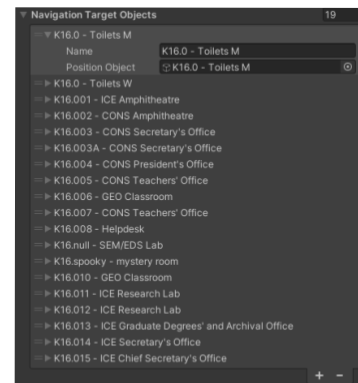
Προκειμένου να δουλέψει σωστά το συγκεκριμένο Script, θα πρέπει να αντιστοιχηθούν όλες οι Serialized μεταβλητές του με αντικείμενα της σκηνής. Προσθέτουμε ένα "Script Component" σε ένα νέο ή υπάρχον "Empty", όπου θα επιλεγθεί το προαναφερθέν Script και τέλος θα αντιστοιχηθούν όλα τα αντικείμενα. Ο κώδικας αυτός αναφέρεται στο Παράρτημα Κώδικα, στον πίνακα "Κώδικας Unity 3".



Εικόνα 5.14
Script Component



Εικόνα 5.15
Λίστα Τοποθεσιών



Εικόνα 5.16
Λίστα Προορισμών

Στην συνέχεια το Script υπολογίζει την βέλτιστη διαδρομή βάσει της τωρινής θέσης και του επιθυμητού προορισμού όπως επιλέχθηκαν, και του πλέγματος πλοήγησης όπως έχει αναφερθεί στο κεφάλαιο 5.1.2. Έχοντας υπολογίσει επιτυχώς την διαδρομή, το Script αρχικοποιεί ένα αντικείμενο *LineRenderer*, το θέτει ως αληθές και αντιστοιχίζει τα σημεία της γραμμής αυτής με τα δεδομένα της

υπολογισμένης βέλτιστης διαδρομής. Ο κώδικας αυτός αναφέρεται στο Παράρτημα Κώδικα, στον πίνακα "Κώδικας *Unity 4*".

Σε κάθε κύκλο λογικής της εφαρμογής, η εφαρμογή θα ανανεώσει την τωρινή τοποθεσία του χρήστη χρησιμοποιώντας τις δυνατότητες του ARCore και θα επαναυπολογίσει συν χαράξει εκ νέου την βέλτιστη διαδρομή βάσει της ανανεωμένης τοποθεσίας χρήστη.

5.1.4.3. ΣΑΡΩΣΗ ΚΩΔΙΚΩΝ QR

Το Script αυτό εκτελεί λειτουργίες εντοπισμού και σάρωσης QR Codes, προσφέροντας στους χρήστες την δυνατότητα να επανατοποθετήσουν τους εαυτούς τους στον ψηφιακό χώρο, χωρίς να χρειάζεται να επιστρέψουν στο Android Studio UI και να επιλέξουν τον επιθυμητό προορισμό τους ξανά. Για να εφαρμοστεί το Barcode Reader που θα χρειαστεί για το σκανάρισμα QR, θα χρησιμοποιηθεί η ανοιχτού κώδικα βιβλιοθήκη Zebra Crossing - ZXing.

Για να προστεθεί η βιβλιοθήκη του Zebra Crossing στην Unity, θα ληφθεί η τελευταία σταθερή κυκλοφορία από το github, θα επιλεχθούν τα αρχεία του ZXing που στοχεύουν την Unity και θα τοποθετηθούν στον φάκελο Assets/Plugins του Unity Project.

Το Script θα ελέγχει σε κάθε κύκλο λογικής της εφαρμογής εάν η κάμερα έχει λάβει νέο καρτέ. Εάν έχει, θα μετατρέψει το καρτέ αυτό σε δισδιάστατη υφή (*Texture*). Εάν το texture αυτό περιλαμβάνει κάποιον κωδικό QR, το ZXing θα το σαρώσει και αν τα περιεχόμενα του κωδικού αντιστοιχούν στην ονομασία

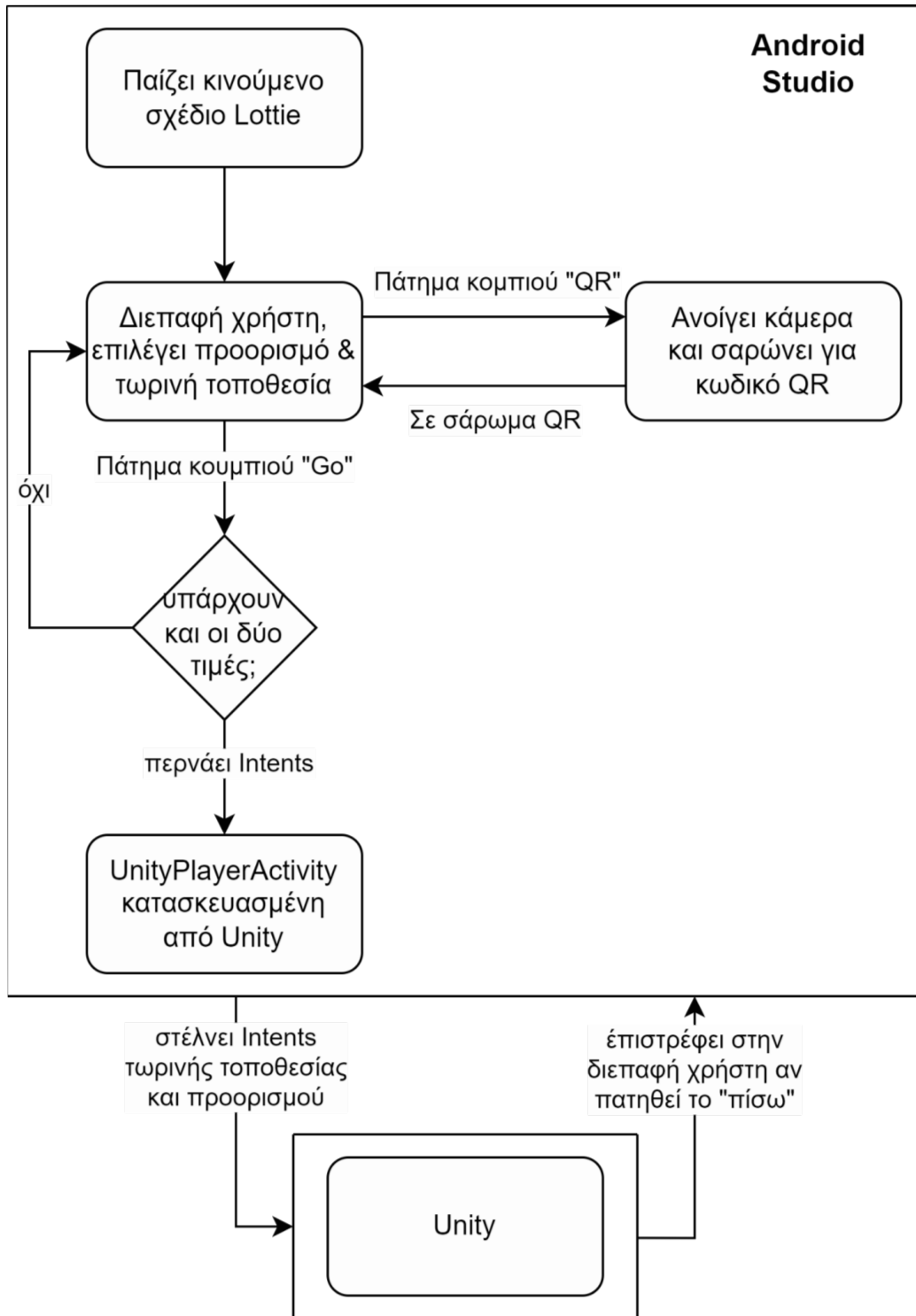
κάποιου αντικειμένου αρχικής τοποθεσίας, θα μεταφερθεί εκεί ο χρήστης στον ψηφιακό χώρο και θα επαναυπολογιστεί η βέλτιστη διαδρομή του.

5.2. ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ ΜΕΣΩ ANDROID STUDIO

Στην συνέχεια θα αναφερθεί το μέρος τη εφαρμογής που έχει αναπτυχθεί σε Android Studio και διαχειρίζεται την διεπαφή του χρήστη με την εφαρμογή. Βάση της ανάπτυξης σε Android Studio είναι το project που έγινε export από την Unity.

Θα αναπτυχθούν τρία ζεύγη δραστηριοτήτων (*Activity*) Java και διατάξεων (*Layout*) XML. Το πρώτο ζεύγος αναπαράγει το κινούμενο σχέδιο εκτέλεσης της εφαρμογής (*boot animation*), το δεύτερο περιλαμβάνει την διεπαφή & διαχειρίζεται τις επιλογές του χρήστη, ενώ, τέλος, το τρίτο θέτει την οπτική της εξωτερικής κάμερας ως οπτική του χρήστη και επιτρέπει την σάρωση κωδικών γρήγορης απόκρισης (*QR Codes*). Τέλος, θα γίνει αναφορά και στον τρόπο δημιουργίας του εικονιδίου της εφαρμογής.

Με το project αυτό να έχει εξαχθεί από την Unity, λείπουν βιβλιοθήκες, ρυθμίσεις και ορισμοί που θα είχαν συμπεριληφθεί εάν ήταν εξ ολοκλήρου υλοποιημένο στο Android Studio. Ως αποτέλεσμα, για την ανάπτυξη των προαναφερθέντων δραστηριοτήτων και διατάξεων θα χρειαστούν αλλαγές και προσθήκες στα αρχεία του *Build Gradle* και του *Android Manifest*.



Διάγραμμα 5.3 - Διάγραμμα ροής του Android Studio

5.2.1. ΠΑΡΑΓΩΓΟ ΤΗΣ UNITY

Όταν η Unity εξάγει projects για επεξεργασία και ανάπτυξη μέσω Android Studio, δημιουργεί ένα αρχείο Java ονόματος *UnityPlayerActivity.java*. Το αρχείο αυτό περιλαμβάνει τις συναρτήσεις που διαχειρίζονται το πέρασμα της σκυτάλης από των κώδικα του Android Studio στις λειτουργίες της Unity και ανάστροφα.

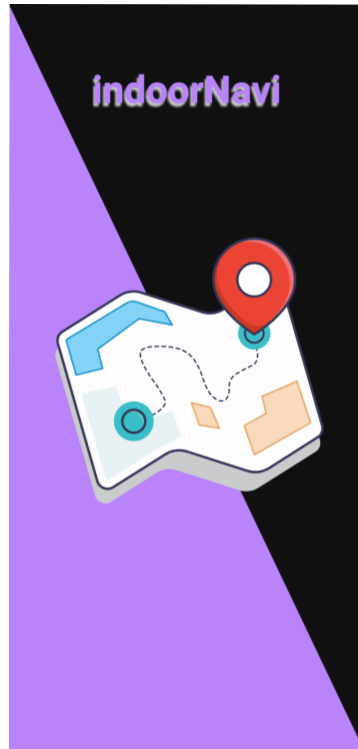
Για την μεταφορά από το Android Studio στην Unity, χρησιμοποιούνται οι συναρτήσεις *onCreate()* και *onNewIntent()*. Η *onNewIntent()* διαχειρίζεται τυχόν Intent που στέλνει το Android Studio στην Unity, όπως στην περίπτωση της παρούσας διπλωματικής την τοποθεσία και τον προορισμό του χρήστη. Η *onCreate()* κατασκευάζει την δραστηριότητα της Unity, φροντίζοντας να περάσει τα τυχόν Intent από την *onNewIntent()*.

Για την επιστροφή από την Unity στο Android Studio, χρησιμοποιείται η συνάρτηση *onUnityPlayerUnloaded()* όταν η Unity εκτελεί εντολή "Unload", ή η συνάρτηση *onUnityPlayerQuitted()* όταν η Unity εκτελεί εντολή "Quit".

5.2.2. ΚΙΝΟΥΜΕΝΟ ΣΧΕΔΙΟ ΕΚΚΙΝΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Με την εκκίνηση της εφαρμογής, αναπαράγεται το κινούμενο σχέδιο εκτέλεσης εφαρμογής (*boot animation*). Για την αναπαραγωγή αυτή θα χρειαστεί η προσθήκη της βιβλιοθήκης *Lottie*, ανεπτυγμένη από την Airbnb Design.

Τα κινούμενα σχέδια αυτά είναι μορφής κλιμακόμενων διανυσματικών γραφικών (*Scalable Vector Graphics - SVG*) και βασισμένα στην γλώσσα σήμανσης XML. Το συγκεκριμένο της παρούσας διπλωματικής, επιλέχθηκε από τις δωρεάν και ανοιχτών δικαιωμάτων επιλογές της πλατφόρμας *LottieFiles*.



Εικόνα 5.17 - *Splash Screen & Boot Animation*

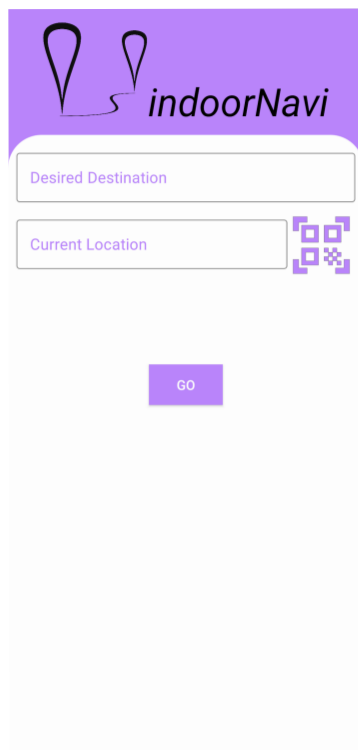
Με την ολοκλήρωση του κινουμένου σχεδίου, ο εφαρμογή μεταβαίνει στην κυρίως διεπαφή χρήστη.

5.2.3. ΚΥΡΙΩΣ ΔΙΕΠΑΦΗ ΤΟΥ ΧΡΗΣΤΗ

Η κυρίως διεπαφή του χρήστη επιτρέπει την επιλογή προορισμού από λίστα επιλογών, την επιλογή τοποθεσίας χρήστη από λίστα επιλογών ή τον εντοπισμό της με σάρωση κωδικού QR, και τέλος την μετάβαση στην Unity. Για την ορθή λειτουργία της θα πρέπει να προστεθούν οι βιβλιοθήκες *AndroidX* και *Material Components*.

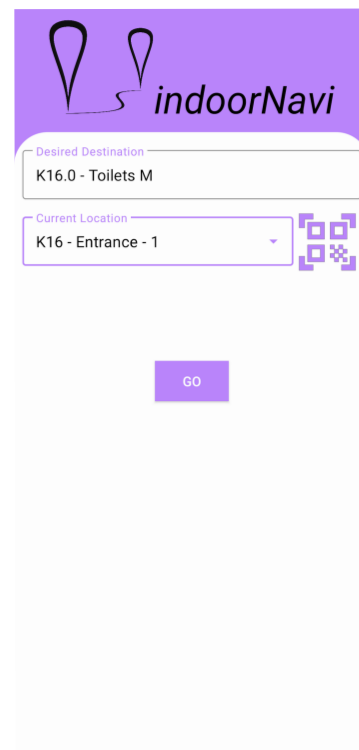
Οι λίστες προορισμών και τοποθεσιών είναι *AutoCompleteTextView*, που περιλαμβάνουν λίστες όλων των πιθανών φυσικών τοποθεσιών του χώρου για τον οποίο προορίζεται η εφαρμογή. Τα ονόματα των τοποθεσιών αυτών είναι αντιστοιχισμένα με τις ονομασίες των κύβων-προορισμών και πλακών-τοποθεσιών της σκηνής στην Unity.

Η επιλογή της λίστας τοποθεσιών μπορεί να γίνει και μέσω σάρωσης κωδικού QR. Το *ImageButton* σχήματος κωδικού QR, ανοίγει την δραστηριότητα σάρωσης κωδικών QR. Η επιλογή τοποθεσίας μέσω σάρωσης βεβαιώνει πως ο χρήστης πράγματι βρίσκεται στην επιλεγμένη τοποθεσία, σε αντίθεση με την επιλογή μέσω λίστας, που δεν έχει τρόπο να επιβεβαιώσει έγκυρη επιλογή.



Εικόνα 5.18

Κυρίως διεπαφή χρήστη



Εικόνα 5.19

Διεπαφή χρήστη με έγκυρες επιλογές

Το κουμπί "Go", σε περίπτωση που τα πεδία προορισμού και τοποθεσίας είναι κενά, ή οι επιλογές τους μη έγκυρες, θα εμφανίσει μήνυμα σφάλματος στον χρήστη. Εναλλακτικά, θα αποστείλει τις τιμές αυτές μέσω Intent στην διεργασία *UnityPlayerActivity*, η οποία με την σειρά της θα ξεκινήσει την Unity, όπου και θα χρησιμοποιηθούν οι μεταβλητές αυτές για τον υπολογισμό και το χάραγμα της διαδρομής του χρήστη.

5.2.4. ΣΑΡΩΣΗ ΚΩΔΙΚΟΥ QR

Για την ορθή λειτουργία της σάρωσης κωδικού QR θα πρέπει να προστεθούν οι βιβλιοθήκες *MLKit.BarcodeScanning*, και *AndroidX.Camera*, ανεπτυγμένες από την Google. Επίσης πρέπει να προστεθεί η βιβλιοθήκη *ZXing*, όπως χρειάστηκε και στις αντίστοιχες λειτουργίες της Unity.

Η δραστηριότητα αυτή, θα εκτελεστεί μόνο αν ο χρήστης πατήσει το *ImageButton* σχήματος QR από την κυρίως διεπαφή, όπως αναφέρεται στο προηγούμενο κεφάλαιο. Κατά την πρώτη εκτέλεση της θα ζητήσει δικαιώματα χρήσης της κάμερας από το λογισμικό.

Όσο η δραστηριότητα αυτή είναι ενεργή, το περιβάλλον διεπαφής του χρήστη θα αντικατασταθεί από την οπτική της εξωτερικής κάμερας συσκευής και η εφαρμογή θα ελέγχει συνεχώς για ύπαρξη κωδικού QR. Αν εντοπιστεί κωδικός, εμφανίζεται το κουμπί "*QR Code Found*", που με την επιλογή του θα σαρώσει τον κωδικό και θα επιστρέψει στην προηγούμενη δραστηριότητα, στέλνοντας τα δεδομένα του κωδικού QR μέσω Intent ως την τωρινή τοποθεσία χρήστη.



Εικόνα 5.20 - Σάρωση Κωδικού Γρήγορης Απόκρισης

5.2.5. ΕΙΚΟΝΙΔΙΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

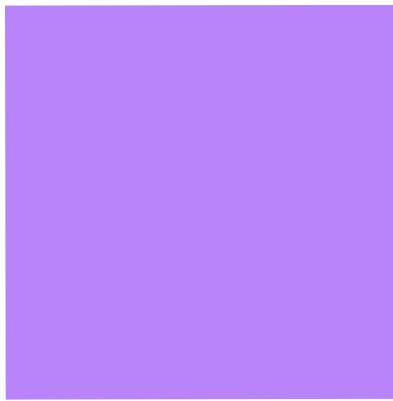
Ένα εικονίδιο μοντέρνας εφαρμογής Android αποτελείται από δύο μέρη, το φόντο (*background*) και το προσκήνιο (*foreground*). Οι δύο αυτές στρώσεις του προσφέρουν την δυνατότητα να προσαρμοστεί έξυπνα στις ρυθμίσεις κάθε παρουσιαστικού λογισμικού Android.

Το εικονίδιο της παρούσας εφαρμογής έχει κατασκευαστεί μέσω του προγράμματος Blender. Μία καμπύλη έχει σμιλευτεί στο σχήμα του αριστερού μισού ενός βέλους τοποθεσίας, έχει μεταμορφωθεί σε αντικείμενο, και τροποποιηθεί με την χρήση ενός "Mirror Modifier" και ενός "Solidify Modifier" για να λάβει την παρούσα μορφή. Το δεύτερο βέλος είναι ένα μετακινημένο και

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

κλιμακωμένο αντίγραφο του πρώτου. Ομοίως κατασκευάστηκε και το μονοπάτι που συνδέει τα δύο βέλη.

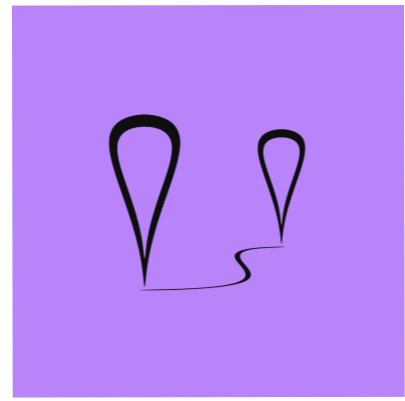
Κάνοντας render την σκηνή του Blender με το φόντο ως διαφανές (*Render Properties* → *Film* → *Transparent*) έχουμε το προσκήνιο του εικονιδίου. Το φόντο είναι ένα αρχείο png ίδιων διαστάσεων με το προσκήνιο, χρωματισμένο ολόκληρο μωβ. Εισάγοντάς τις εικόνες αυτές στις ρυθμίσεις εικονιδίου του project του Android Studio, το εικονίδιο της εφαρμογής είναι έτοιμο.



Εικόνα 5.21
Φόντο Εικονιδίου



Εικόνα 5.22
Προσκήνιο Εικονιδίου

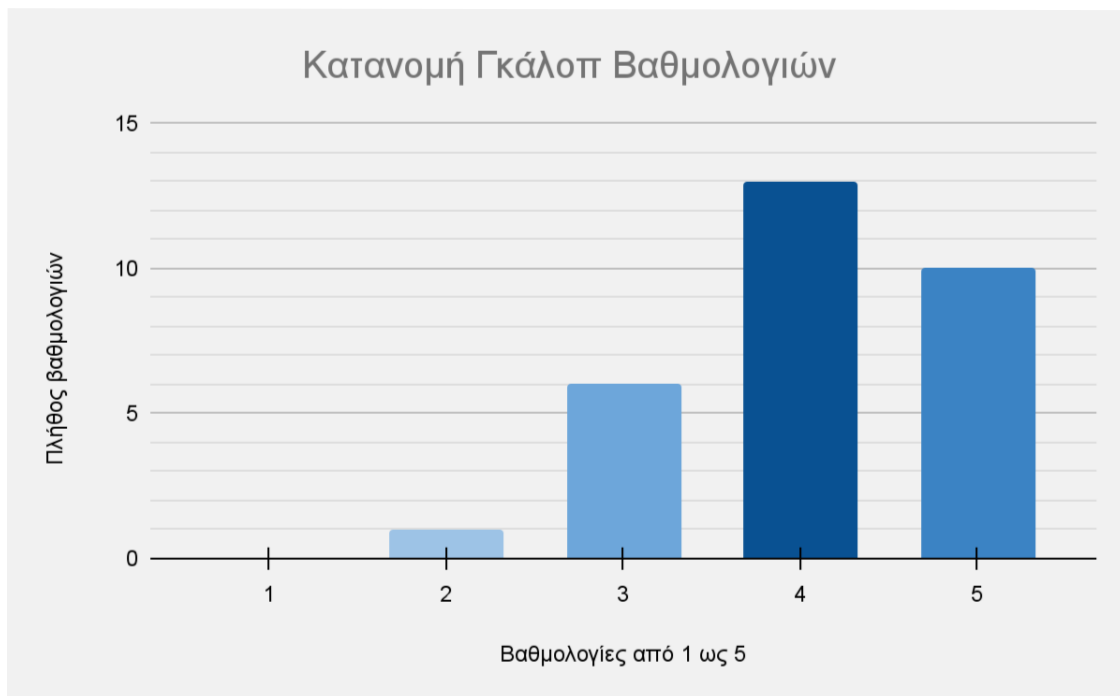


Εικόνα 5.23
Τελικό Εικονίδιο

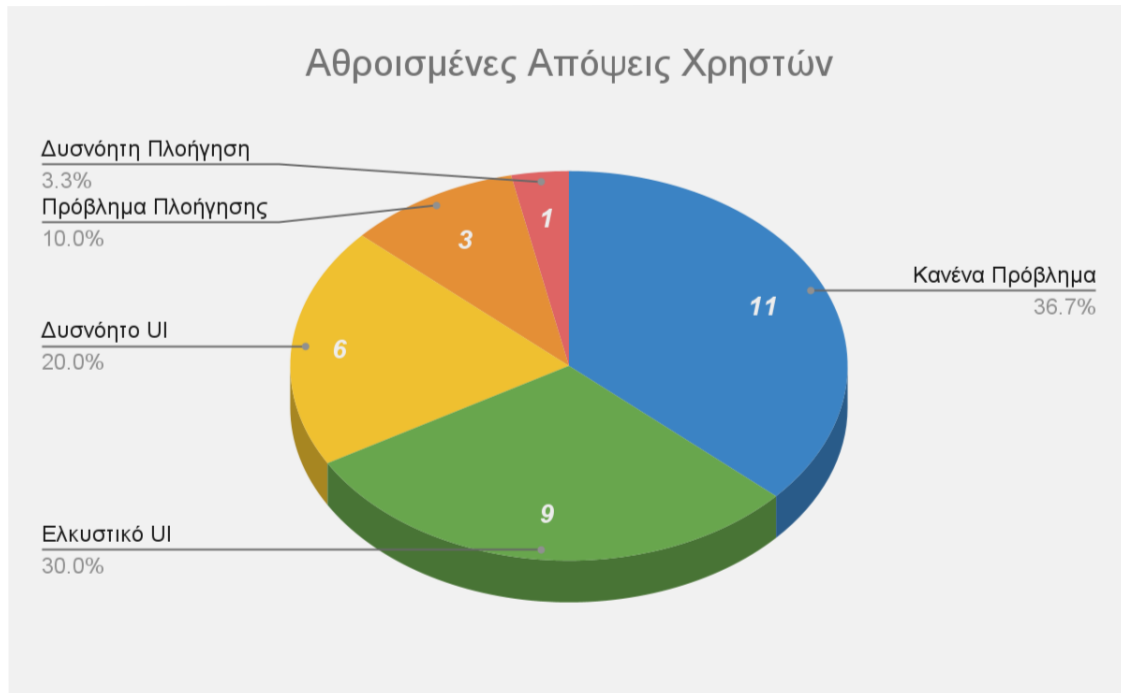
6. ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ

Με την ολοκλήρωση της εφαρμογής, δόθηκε σε φίλους, γνωστούς και φοιτητές του Πανεπιστημίου Δυτικής Αττικής για συλλογή απόψεων. Ζητήθηκε από κάθε άτομο μια βασική βαθμολόγηση της εφαρμογής σε μία κλίμακα μέτρησης από το 1 έως το 5, με το 1 να αντιπροσωπεύει τη χειρότερη βαθμολογία και το 5 να αντιπροσωπεύει την καλύτερη, συν επιπλέον να εκφράσουν την κυρίως άποψη ή το κυρίως πρόβλημα τους με την εφαρμογή αυτή.

Αξίζει να αναφερθεί πως χωρίς μια ανώνυμη διαδικασία βαθμολόγησης, οι βαθμολογητές συχνά δεν απαντούν με πλήρη σκληρότητα, προκειμένου να μην θίξουν τον συνομιλητή τους. Επίσης η βαθμολόγηση είναι ένα μέθοδος που μόνη της δεν προσφέρει αρκετές πληροφορίες, ούτε την πλήρη εικόνα, μα αν οι αριθμοί συσχετιστούν με επιπλέον πληροφορίες μπορούν να αντληθούν πολύ πιο χρήσιμα συμπεράσματα.



Πιο χρήσιμα δεδομένα είναι οι απόψεις των χρηστών περί της εφαρμογής, με επιπλέον προσοχή στις αρνητικές κριτικές. Για την παρουσίαση των δεδομένων αυτών, όμοιες κριτικές έχουν ομαδοποιηθεί και οι προτάσεις που τις περιέγραφαν έχουν συνοψισθεί σε σύντομες φράσεις.

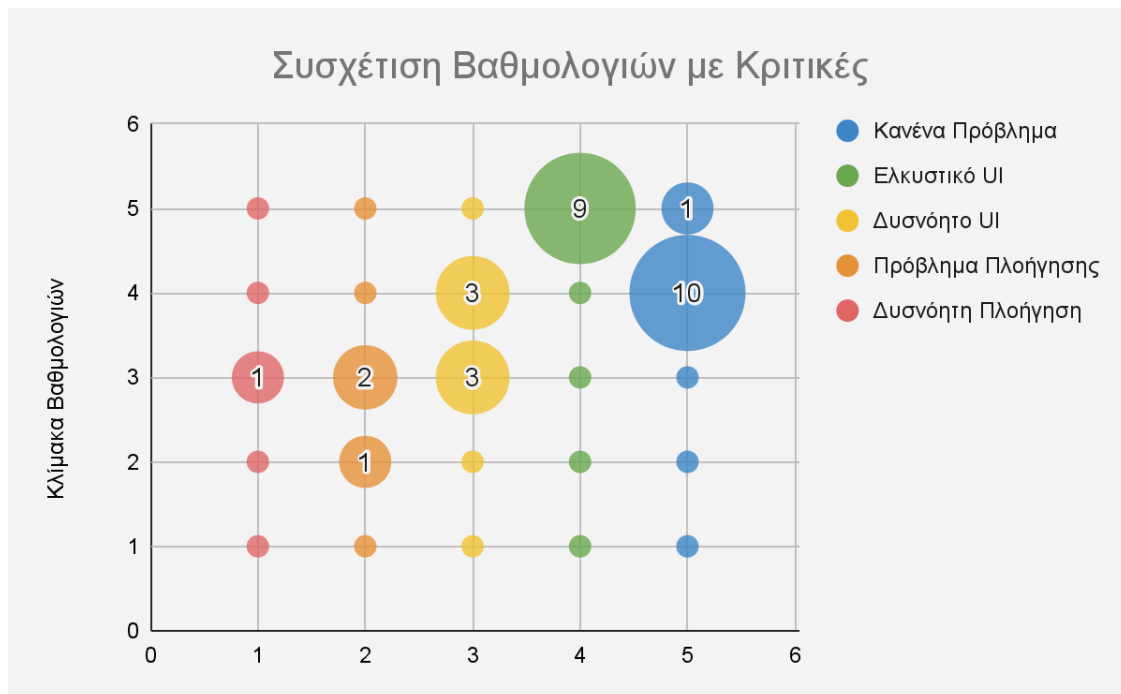


- Η κατηγορία "Κανένα Πρόβλημα" περιλαμβάνει τις περισσότερες και γενικά πιο θετικές απαντήσεις.
- Η κατηγορία "Ελκυστικό UI" περιλαμβάνει τις απαντήσεις που επικεντρώνονται στα εμφανισιακά στοιχεία της εφαρμογής.
- Η κατηγορία "Δυσνόητο UI" περιλαμβάνει απαντήσεις χρηστών που είχαν δυσκολίες με την επιλογή τωρινής τοποθεσίας, κυρίως στοχευμένες στην λίστα επιλογής και την μη ορθή αντιστοίχιση της φυσικής τοποθεσίας του

χρήστη με την τοποθεσία του στον ψηφιακό χώρο. Χρήστες που επέλεξαν την τοποθεσία τους μέσω QR Code δεν εξέφρασαν το πρόβλημα αυτό.

- Η κατηγορία "Πρόβλημα Πλοήγησης" περιλαμβάνει απαντήσεις χρηστών στους οποίους προέκυψαν σφάλματα πλοήγησης, με τις λειτουργίες AR να χάνουν την τοποθεσία και να τους τοποθετούν λάθος στον ψηφιακό χώρο.
- Η κατηγορία "Δυσνόητη Πλοήγηση" αποτελείται από έναν μόνο χρήστη, που εξέφρασε δυσκολία προσδιορισμού του τέλους της γραμμής διαδρομής.

Συσχετίζοντας τις βαθμολογίες με τις κριτικές των χρηστών, μπορούμε να δούμε μια πιο καθαρή εικόνα της κατάστασης της εφαρμογής και να δώσουμε προτεραιότητα σε αλλαγές και προσθήκες που θα βελτιώσουν σημαντικά την μέση βαθμολογία της εφαρμογής.



Το μέγεθος δείγματος δεν είναι επαρκές για να παραχθούν σίγουρα αποτελέσματα, αλλά βάσει των μέσων βαθμολογιών 4,1 στην κατηγορία "Κανένα Πρόβλημα" και 5 στην κατηγορία "Ελκυστικό UI", μπορούμε να υποθέσουμε ότι η εφαρμογή είναι σε σωστό δρόμο.

Με μέση βαθμολογία 3.5 στην κατηγορία "Δυσνόητο UI", 2,7 στην κατηγορία "Προβλήματα Πλοήγησης" και 3 στην κατηγορία "Δυσνόητη Πλοήγηση", γνωρίζουμε που πρέπει να στοχευθεί η μελλοντική ανάπτυξη της εφαρμογής προκειμένου να βελτιωθεί η εμπειρία των χρηστών.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Συνοπτικά και εν κατακλείδι, η εφαρμογή που αναλύθηκε στην διπλωματική αυτή, έχει ως σκοπό την πλοήγηση χρηστών σε εσωτερικούς χώρους με ελάχιστο κόστος εγκατάστασης.

Άλλες προσεγγίσεις εντοπισμού και ιχνηλάτησης τοποθεσίας, όπως μέσω Bluetooth Lite (BLE) Beacons, απαιτούν σημαντικό κόστος υλικού και εγκατάστασης, σε αντίθεση με το κόστος εκτύπωσης και τοποθέτησης κωδικών QR για τον εντοπισμό τοποθεσίας και χρήση δυνατοτήτων επαυξημένης πραγματικότητας για την ιχνηλάτηση τοποθεσίας.

Παρόλα αυτά, η περιστασιακή ανακρίβεια και δυσλειτουργία της ιχνηλάτησης τοποθεσίας μέσω AR θα μπορούσε να μειωθεί ή και να εξλειφθεί με την χρήση κάποιου επιπλέον μηχανισμού εντοπισμού και ιχνηλάτησης τοποθεσίας. Για εγκαταστάσεις της εφαρμογής με αξιόλογο budget εγκατάστασης, η προσθήκη δυνατοτήτων εντοπισμού και ιχνηλάτησης τοποθεσίας μέσω BLE, σε συνεργασία με το υπάρχων σύστημα AR, θα αποτελούσε πολύ σημαντική προσθήκη και βελτίωση.

Η επιλογή προορισμού του χρήστη μέσω λίστας είναι η πιο απλή και ευνόητη προσέγγιση. Στην πειραματική έκδοση της εφαρμογής αυτής, οι εν δυνάμει προορισμοί ήταν αρκετά λίγοι που μια κυλιόμενη λίστα ήταν επαρκής, αλλά η προσθήκη πεδίου αναζήτησης σε συνεργασία με την λίστα θα ήταν απαραίτητη για εγκαταστάσεις μεγάλης κλίμακας.

Επιπλέον η λίστα προορισμών θα μπορούσε να ταξινομηθεί βάσει περισσότερων παραγόντων πέραν του αλφαβητικού, όπως π.χ. κοινό κτίριο με Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών – Λογισμικού & Πληροφοριακών Συστημάτων

την τοποθεσία, κοινό όροφο με την τοποθεσία, απόσταση από τοποθεσία, προηγούμενες επιλογές του χρήστη, ή και συχνές επιλογές όλων των χρηστών εάν προστεθούν δυνατότητες συλλογής και επεξεργασίας δεδομένων σε server.

Αντίθετα, η χρήση λίστας για την επιλογή τωρινής τοποθεσίας ήταν εξαιρετικά χρήσιμη για τα στάδια πειραματισμού, αλλά πλήρως δύσχρηστη πέραν του ελεγχόμενου πειραματικού περιβάλλοντος. Επιτρέπει στον χρήστη να επιλέξει οποιαδήποτε τοποθεσία ως την τωρινή του, χωρίς να έχει τρόπο να επιβεβαιώσει την εγκυρότητα της επιλογής αυτής, οδηγώντας σε πλήρως λανθασμένη αντιστοίχιση τοποθεσίας στον φυσικό και τον ψηφιακό κόσμο. Η αφαίρεση της λίστας και η τροποποίηση της εφαρμογής ώστε να βασίζεται πλήρως σε κωδικούς QR για τον εντοπισμό τοποθεσίας είναι απαραίτητες αλλαγές προτού δοθεί σε κοινούς χρήστες.

Ο ψηφιακός χώρος κατασκευασμένος μέσω προγραμμάτων δημιουργίας τρισδιάστατων γραφικών δούλεψε εξαιρετικά για την εφαρμογή αυτή, επιτρέποντας της να υπολογίσει και να χαράξει βέλτιστες διαδρομές χωρίς να έχουν προϋπολογιστεί όπως π.χ. χρησιμοποιώντας λίστες γειτονικών κόμβων, ή χωρίς να έχει "περπατηθεί" από προγραμματιστές όπως π.χ. με χρήση εφαρμογών επαυξημένης πραγματικότητας.

Παρόλα αυτά η διαδικασία αυτή είναι σημαντικά χρονοβόρα. Η χρήση προϋπάρχοντων κατόψεων σε μορφή SVG αποτελεί ένα εξαιρετικό μέτρο εξοικονόμησης χρόνου, αλλά ακολουθούν πολλαπλά βήματα τα οποία θα μπορούσαν να αυτοματοποιηθούν. Η δημιουργία ενός Python plugin για το εργαλείο δημιουργίας τρισδιάστατων γραφικών Blender που να αυτοματοποιεί

όσα περισσότερα βήματα γίνεται, θα επιτάχυνε εξαιρετικά την προσαρμογή της εφαρμογής σε νέους χώρους.

Απαραίτητη μελλοντική επέκταση θα είναι η προσθήκη υποστήριξης πλοήγησης μεταξύ πολλαπλών ορόφων. Η προσθήκη αυτή είναι σχετικά απλή, αλλά προσθέτει μια επιπλέον στρώση πολυπλοκότητας στην εφαρμογή και αυξάνει τις εργατοώρες που χρειάζονται για την χαρτογράφηση του τυχόν πολυώροφου εσωτερικού χώρου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Official Unity Documentation, από:
<https://docs.unity3d.com/>
2. Official ARFoundation Documentation, από:
<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html>
3. Official ARCore Documentation, από:
<https://developers.google.com/ar/develop>
4. Official Android Studio Documentation, από:
<https://developer.android.com/docs>
5. Official ZXing Documentation, από:
<https://zxing.github.io/zxing/>
6. Francesco Pascale, Ennio Andrea Adinolfi, Massimiliano Avagliano, Venanzio Giannella and Andres Salas, A Low Energy IoT Application Using Beacon for Indoor Localization, από:
<https://www.mdpi.com/2076-3417/11/11/4902>
7. Chuan-Bi Lin, An Improved Weighted KNN Algorithm for Indoor Positioning using Beacon Devices, από:
https://www.researchgate.net/publication/339664856_An_Improved_Weighted_KNN_Algorithm_for_Indoor_Positioning_using_Beacon_Devices
8. Maani Ghaffari Jadidi, Mitesh Patel, and Jaime Valls Miro, Gaussian Processes Online Observation Classification for RSSI-based Low-cost Indoor Positioning

Systems, από:

<https://ui.adsabs.harvard.edu/abs/2016arXiv160903130G/abstract>

9. Abhijith Ajith, Albin Mathew Thomas, Jestin Antony, Joel JosephSuma R, Indoor Mapping Based on Augmented Reality Using Unity Engine, από:

<https://ijisrt.com/assets/upload/files/IJISRT20MAY756.pdf>

10. "Unity source code access", από:

<https://store.unity.com/front-page>

11. GSMarena, Weekly poll: what's the best position for a punch hole camera, από:

https://www.gsmarena.com/weekly_poll_whats_the_best_position_for_a_punch_hole_camera-news-37721.php

12. Jeremy Norman, History of Information, Timeline of Augmented or Virtual Reality, από:

<https://www.historyofinformation.com/index.php?cat=89>

13. Christos Papakosta, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou, Exploration of Augmented Reality in Spatial Abilities Training: A Systematic Literature Review for the Last Decade, από:

<https://eric.ed.gov/?id=EJ1289174>

14. Christos Papakostas, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou, On the Development of a Personalized Augmented Reality Spatial Ability Training Mobile Application, από:

<https://books.google.gr/books?hl=en&lr=&id=I4xJEAAQBAJ&oi=fnd&pg=PA>

[75&dq=info:5HUlzm-mPd4J:scholar.google.com&ots=V67dGrzqdh&sig=iUYZxG7QicCjC0yCLJ1JTBucAmc&redir_esc=y#v=onepage&q&f=false](https://scholar.google.com/lookup?q=info:5HUlzm-mPd4J:scholar.google.com&ots=V67dGrzqdh&sig=iUYZxG7QicCjC0yCLJ1JTBucAmc&redir_esc=y#v=onepage&q&f=false)

15. Andrianthi Kapetanaki, Akrivi Krouska, Christos Troussas, Cleo Sgouropoulou, A Novel Framework Incorporating Augmented Reality and Pedagogy for Improving Reading Comprehension in Special Education, από:

https://books.google.gr/books?hl=en&lr=&id=I4xJEAAAQBAJ&oi=fnd&pg=PA105&dq=info:zysoV9VZA88J:scholar.google.com&ots=V67dGrzrej&sig=SjoxDXQcYqoAKICXkntT27ZLclk&redir_esc=y#v=onepage&q&f=false

16. Christos Papakostas, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou, Personalization of the Learning Path within an Augmented Reality Spatial Ability Training Application Based on Fuzzy Weights, από:

<https://www.mdpi.com/1424-8220/22/18/7059>

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

```

private String from;
private String to;

#if (!UNITY_EDITOR && UNITY_ANDROID)
    AndroidJavaObject currentActivity =
        UnityPlayer.GetStatic<AndroidJavaObject> ("currentActivity");
    AndroidJavaObject intent =
        currentActivity.Call<AndroidJavaObject> ("getIntent");

    AndroidJavaObject extras = null;
    extras = intent.Call<AndroidJavaObject>("getExtras");

    if (extras != null) {
        string s = string.Empty;
        from = extras.Call<string>("getString", "intentFrom");
        to = extras.Call<string>("getString", "intentTo");
    }
#endif

```

Κώδικας Unity 1 - Λήψη Intent από Android Studio

```

private void Update() {
    if (Application.platform == RuntimePlatform.Android)
        if (Input.GetKeyDown(KeyCode.Escape))
            Application.Unload();
}

```

Κώδικας Unity 2 - Unload Unity και επιστροφή σε Android Studio


```

[SerializeField] private ARSession session;
[SerializeField] private ARSessionOrigin sessionOrigin;

[SerializeField]
private List<T> navigationQRObjects = new List<Target>();
Target currentPosition =
    navigationQRObjects.Find(x => x.Name.Equals(from));
sessionOrigin.transform.position =
    currentTarget.PositionObject.transform.position;
sessionOrigin.transform.rotation =
    currentTarget.PositionObject.transform.rotation;
session.Reset();

[SerializeField]
private List<Target> navigationTargetObjects = new List<Target>();
Target currentTarget =
    navigationTargetObjects.Find(x => x.Name.Equals(to));
private Vector3 targetPosition =
    currentTarget.PositionObject.transform.position;

```

Κώδικας Unity 3 - Αρχικοποίηση λιστών τοποθεσιών και προορισμών

```

private NavMeshPath path = new NavMeshPath();
NavMesh.CalculatePath(
    transform.position, targetPosition, NavMesh.AllAreas, path);

private LineRenderer line = transform.GetComponent<LineRenderer>();
line.enabled = true;
line.positionCount = path.corners.Length;
line.SetPositions(path.corners);

```

Κώδικας Unity 4 - Υπολογισμός και χάραξη διαδρομής

