



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

# Σχεδίαση και Ανάπτυξη Ενοποιημένης Εφαρμογής Κινητών Συσκευών για Πλοήγηση με Χρήση Τεχνικών Επαυξημένης Πραγματικότητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θάλεια Γ. Σιάλδα

**Επιβλέπων :** Στέφανος Κόλλιας  
Καθηγητής Ε.Μ.Π

**Συνεπιβλέπουσα :** Παρασκευή Τζούβελη  
Μέλος ΕΔΙΠ

Αθήνα, Μάιος 2022





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

# Σχεδίαση και Ανάπτυξη Ενοποιημένης Εφαρμογής Κινητών Συσκευών για Πλοήγηση με Χρήση Τεχνικών Επαυξημένης Πραγματικότητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θάλεια Γ. Σιάλδα

**Επιβλέπων :** Στέφανος Κόλλιας  
Καθηγητής Ε.Μ.Π

**Συνεπιβλέπουσα :** Παρασκευή Τζούβελη  
Μέλος ΕΔΙΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11<sup>η</sup> Μαΐου 2022

.....  
Στέφανος Κόλλιας  
Καθηγητής Ε.Μ.Π

.....  
Ανδρέας-Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π

.....  
Γεώργιος Στάμου  
Αν.Καθηγητής Ε.Μ.Π

Αθήνα, Μάιος 2022

.....  
Θάλεια Γ. Σιάλδα

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Θάλεια Γ. Σιάλδα

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Τα τελευταία χρόνια η ανάπτυξη της τεχνολογίας και της επικοινωνίας έχει γνωρίσει μεγάλη άνοδο, πράγμα που οδηγεί ολοένα και περισσότερες εταιρείες να επενδύσουν σε τεχνικές και εργαλεία που θα βοηθήσουν στην στρατηγική ανάπτυξη του digital marketing τους. Συγκεκριμένα το Augmented Reality είναι μια τεχνολογία που αφήνει χώρο για την δημιουργία πολλών καινοτόμων ιδεών και η δημοτικότητα που έχει αποκτήσει είναι πρωτοφανής.

Η παρούσα διπλωματική εργασία έχει στόχο την εξοικείωση με το Augmented Reality. Στα πλαίσια της τεχνικής αυτής κατασκευάσαμε μία εφαρμογή που επιτρέπει στον χρήστη να πλοηγηθεί στον χώρο των νέων κτηρίων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, μέσω ενδείξεων AR.

Η συγκεκριμένη εφαρμογή αναπτύχθηκε στο περιβάλλον της Unity Game Engine. Προορίζεται αποκλειστικά για κινητά τηλέφωνα με λειτουργικό σύστημα Android κάνοντας χρήση του ARCore. Θέλοντας να εξαλείψουμε κάθε εξάρτηση, όπως πρόσβαση στο Internet ή συσκευές που απαιτούν πρόσθετο υλικό, και να τεστάrouμε κάθε βήμα της ανάπτυξης στον υπολογιστή μας, κάναμε χρήση του Image Tracking του Vuforia. Επιπλέον, για την πλοήγηση στον εσωτερικό χώρο της σχολής έγινε χρήση του NavMesh και των NavAgents. Όσον αφορά στο Augmented Reality κομμάτι της εφαρμογής, οι ενδείξεις AR κατασκευάστηκαν στο Blender και γίνονται ορατές στον χρήστη μέσω της κάμερας του κινητού του τηλεφώνου.

## Λέξεις κλειδιά

Εντοπισμός θέσης σε εσωτερικό χώρο, παρακολούθηση κινήσεων, πλοήγηση, τρισδιάστατο μοντέλο, διεπαφή χρήστη

## **Abstract**

In recent years the development of technology and communication has experienced great growth, which leads more and more companies to invest in techniques and tools that will help in the strategic development of their digital marketing. Specifically, Augmented Reality is a technology that leaves room for the creation of many innovative ideas and the popularity it has experienced is unprecedented.

This dissertation aims to familiarize you with Augmented Reality. As part of this technique, we built an application that allows the user to navigate through the new buildings of the School of Electrical and Computer Engineering of the National Technical University of Athens, through AR indicators.

This application was developed in the environment of Unity Game Engine. It is intended exclusively for mobile phones with Android operating system using ARCore. Wanting to eliminate any dependency, such as Internet access or devices that require additional hardware, and to test every step of the development on our computer, we made use of Vuforia's Image Tracking. In addition, NavMesh and NavAgents were used to navigate the school. As for the Augmented Reality part of the application, the AR indicators were made in Blender and are visible to the user through the camera of their mobile phone.

## **Keywords**

Indoor positioning, localization, tracking, navigation, 3d model, user interface

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την Διδάκτορα Παρασκευή Τζούβελη για όλες τις πολύτιμες συμβουλές και την καθοδήγηση που μου έδωσε όλο αυτό το διάστημα, καθώς επίσης και για τις καθοριστικές διορθώσεις και επισημάνσεις που μου υπέδειξε. Επιπλέον, θα ήθελα να την ευχαριστήσω για την ιδέα της εργασίας αυτής και την εμπιστοσύνη που μου έδειξε ως προς την διεκπεραίωσή της.

Θα ήθελα να ευχαριστήσω επίσης την οικογένεια μου που είναι δίπλα μου όλα αυτά τα χρόνια. Χωρίς την δική τους υλική και πνευματική υποστήριξη, αλλά και την εμπιστοσύνη που μου δείχνουν δεν θα μπορούσα να τα έχω καταφέρει. Τέλος, θέλω να ευχαριστήσω τους φίλους μου που ήταν δίπλα μου σε όλο αυτό το όμορφο ταξίδι της σχολής και που μαζί ζήσαμε εμπειρίες που θα με χαρακτηρίζουν ως άτομο από εδώ και στο εξής.

Αθήνα, Απρίλιος 2022

*Θάλεια Σιάλδα*





# Περιεχόμενα

Περίληψη .....	5
Abstract .....	6
Ευχαριστίες .....	7
<b>1 Εισαγωγή .....</b>	<b>11</b>
1.1 Κίνητρο .....	11
1.2 Προηγούμενες Μελέτες .....	12
1.3 Δομή Εργασίας .....	13
<b>I ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ.....</b>	<b>15</b>
<b>2 Εντοπισμός Θέσης Σε Εσωτερικούς Χώρους .....</b>	<b>16</b>
2.1 Εισαγωγή .....	16
2.2 Συστήματα Πλοήγησης σε Εξωτερικούς Χώρους .....	17
2.3 Συστήματα Πλοήγησης σε Εσωτερικούς Χώρους .....	17
2.4 Αναπαράσταση του Κτηρίου.....	18
2.5 Positioning / Tracking των Κινήσεων του Χρήστη .....	19
2.5.1 Εισαγωγή στο Indoor Positioning System .....	19
2.5.2 Τεχνολογίες, Τεχνικές και Μέθοδοι.....	19
2.5.3 Υπολογιστική Αρχιτεκτονική και Υποδομή.....	20
2.5.4 Συνεργατικά και Μη-Συνεργατικά IPS .....	21
2.5.5 Θετικά και Αρνητικά Indoor Positioning Τεχνολογιών .....	22
2.6 Αλγόριθμοι Εύρεσης Μονοπατιών .....	25
2.7 Πλοήγηση μέσω Ενδείξεων AR.....	26
<b>3 Augmented Reality .....</b>	<b>28</b>
3.1 Εισαγωγή .....	28
3.2 Είδη Επαύξησης.....	29
3.3 Παρακολούθηση, Βαθμονόμηση και Εγγραφή .....	30
3.4 Πλεονεκτήματα και Μειονεκτήματα του AR για Φορητές Συσκευές .....	31
3.5 Σχεδιαστικά Στοιχεία MAR Εφαρμογών .....	32
3.6 Τεχνολογίες για την επίτευξη AR.....	33
<b>4 Unity.....</b>	<b>34</b>
4.1 Εισαγωγή .....	34
4.2 ARkit και ARCore .....	34
4.3 ARFoundation και Vuforia.....	36

4.4	NavMesh .....	40
<b>II ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ .....</b>		<b>42</b>
<b>5</b>	<b>Μεθοδολογία .....</b>	<b>43</b>
5.1	Συλλογιστική Πορεία και Προτεινόμενο Σύστημα .....	43
5.2	Λειτουργία Εφαρμογής.....	44
<b>6</b>	<b>Ανάπτυξη Εφαρμογής .....</b>	<b>45</b>
6.1	3D Model του Κτηρίου.....	45
6.2	Τοποθέτηση των Πινάκιδων .....	51
6.3	Εντοπισμός του χρήστη .....	55
6.4	Επιλογή από τον Χρήστη του Τελικού Προορισμού .....	60
6.5	Εναλλαγή Ορόφων.....	64
6.5.1	OffMeshLink για σκάλες .....	64
6.5.2	NavMesh για σκάλες .....	71
6.6	Ενδείξεις AR.....	76
6.6.1	Arrow.....	76
6.6.2	Destination marker.....	80
6.7	UI .....	81
6.7.1	UI Scan Plate .....	81
6.7.2	UI Πληκτρολόγησης Τελικού Προορισμού .....	83
6.8	Scripts.....	86
<b>III ΕΠΙΛΟΓΟΣ.....</b>		<b>91</b>
<b>7</b>	<b>Επίλογος .....</b>	<b>92</b>
7.1	Σύνοψη Λειτουργίας Εφαρμογής .....	92
7.2	Προκλήσεις και Περιορισμοί .....	92
<b>Βιβλιογραφία.....</b>		<b>94</b>

# 1 Εισαγωγή

## 1.1 Κίνητρο

Κίνητρο για την μελέτη του συγκεκριμένου θέματος και την ανάπτυξης της παρούσας εφαρμογής αποτέλεσε η ραγδαία εξέλιξη του τομέα του Augmented Reality. Σήμερα, όλο και περισσότερες τεχνολογικές εταιρείες επενδύουν στον συγκεκριμένο τομέα. Εκτός από την «παραδοσιακή» χρήση του AR στα παιχνίδια, αρκετές νέες δυνατότητες και πλεονεκτήματα έχουν εμφανιστεί και σε άλλους τομείς όπως είναι ο εφοδιαστικός, ο κατασκευαστικός και αυτός του λιανικού εμπορίου. Το AR μπορεί να ωφελήσει οποιαδήποτε επιχειρηματική δραστηριότητα η οποία μπορεί να βελτιωθεί οπτικά.

Ταυτόχρονα με το ενδιαφέρον για το AR, τα τελευταία χρόνια η ανάπτυξη και η εκλαΐκευση της τεχνολογίας της επικοινωνίας επέτρεψε την ενσωμάτωση τεχνολογιών εντοπισμού θέσης σε εσωτερικούς χώρους (indoor positioning technologies) σε smartphones [1]. Η εφαρμογή συστημάτων πλοήγησης εσωτερικών χώρων είναι εξαιρετικά επωφελής για τον εντοπισμό και την εξαγωγή πληροφοριών σχετικά με σημεία ενδιαφέροντος μέσα στα κτήρια, ειδικά όταν οι χρήστες επισκέπτονται την τοποθεσία για πρώτη φορά, οι εσωτερικές πινακίδες δεν είναι πολύ σαφείς ή δεν υπάρχει διαθέσιμο προσωπικό για να παράσχει οδηγίες. Οι τουρίστες, για παράδειγμα, θα μπορούσαν να ελιχθούν με σιγουριά μέσα σε ένα τουριστικό αξιοθέατο χωρίς βοήθεια. Σε μέρη όπως μουσεία και γκαλερί τέχνης, η εφαρμογή θα μπορούσε να επεκταθεί για να συμπεριλάβει τα βέλτιστα ή «κοινά» μονοπάτια. Μια τέτοια συσκευή μπορεί επίσης να ενσωματωθεί στα αεροδρόμια για να κατευθύνει τους ταξιδιώτες στις πύλες εισόδου τους. Ομοίως, ένα σύστημα πλοήγησης εσωτερικού χώρου θα μπορούσε επίσης να ωφελήσει τους χρήστες που έχουν επισκεφτεί προηγουμένως την τοποθεσία αλλά εξακολουθούν να αγνοούν πού βρίσκονται ορισμένα από τα επιθυμητά αντικείμενα. Τέτοια μέρη είναι συνήθως supermarket, βιβλιοθήκες και εμπορικά κέντρα. Αυτή η τεχνολογία θα μπορούσε επίσης να “μάθει” τη συμπεριφορά των πελατών, να συλλέξει πολύτιμα δεδομένα γι’ αυτούς και να τα χρησιμοποιήσει για την προώθηση διαφημίσεων σε συγκεκριμένα μέρη ή σε συγκεκριμένες διαδρομές. Ωστόσο, λόγω της πολυπλοκότητας των εσωτερικών περιβαλλόντων, η χρήση μιας ενιαίας τεχνολογίας για την επίτευξη του positioning (εντοπισμού της τοποθεσίας) του χρήστη σε διάφορα σημεία αποτελεί πρόκληση.

## 1.2 Προηγούμενες Μελέτες

Για να βρούμε λύση στα προβλήματα που δημιουργούνται σε εσωτερικό περιβάλλον, όπως είναι αυτό της σχολής, μελετήσαμε διάφορες παρεμφερείς εργασίες που προσεγγίζουν το θέμα αυτό παρόμοια με εμάς. Οι παρακάτω δουλειές χρησιμοποιούν για το indoor navigation τα χαρακτηριστικά του επιταχυνσιόμετρου (accelerometer), της πυξίδας και της κάμερας των smartphones, σε συνδυασμό με το AR χωρίς την ανάγκη πρόσθετου υλικού ή μονάδας GPS.

Ο Link [58] και άλλοι χρησιμοποίησαν το επιταχυνσιόμετρο και την πυξίδα του smartphone για λογισμικό Android και ανέπτυξαν το σύστημα πλοήγησης Footpath που επιτρέπει το περπάτημα προς κάποιον στόχο από ένα σημείο εκκίνησης σε εσωτερικό χώρο. Αρχικά, δημιούργησαν τον χάρτη του εσωτερικού χώρου όπου εφαρμόζουν τον αλγόριθμο δρομολόγησης. Η διαδικασία δημιουργίας χάρτη έγινε στην εφαρμογή OpenStreetMap [93], η οποία είναι εύκολη στη χρήση και επιτρέπει πρόσθετες βελτιώσεις. Μετά τη διαδικασία δημιουργίας χάρτη, ο χρήστης αρχίζει να περπατά και τα βήματα αναγνωρίζονται χρησιμοποιώντας έναν αλγόριθμο που μετράει τα βήματα και οι χρήστες κατευθύνονται προς τον στόχο στο χάρτη. Εφόσον οι οδηγίες στην εφαρμογή Footpath είναι προκαθορισμένες στον χάρτη που ορίζονται στο OpenStreetMap, η συνέπεια παρέχεται συγκρίνοντας την αναμενόμενη κατεύθυνση και την κατεύθυνση προς την οποία πηγαίνει ο χρήστης. Τα εισερχόμενα δεδομένα από τη διαδικασία περιπάτου φιλτράρονται από τον αλγόριθμο First Fit.

Ο Ravi [59] και άλλοι στις μελέτες τους βρήκαν την τρέχουσα θέση του χρήστη σε εσωτερικό χώρο με τις κάμερες των smartphone. Στην εφαρμογή τους δεν χρησιμοποίησαν επιπλέον υλικό εκτός από τις δυνατότητες του κινητού τηλεφώνου. Τα μέρη που απαιτούν πρόσθετο κόστος είναι ο απομακρυσμένος διακομιστής, τα δεδομένα των εικόνων και μια βάση δεδομένων όπου αποθηκεύονται οι πληροφορίες θέσης των εικόνων. Οι φωτογραφίες του μετακινούμενου χρήστη που λαμβάνονται από ένα τηλέφωνο και οι πληροφορίες τοποθεσίας αποστέλλονται αμέσως στον απομακρυσμένο διακομιστή. Οι φωτογραφίες και οι πληροφορίες τοποθεσίας συγκρίνονται στη βάση δεδομένων και όταν υπάρχει αντιστοιχία, προσδιορίζεται η τοποθεσία. Επιπλέον, εξέτασαν το ύψος του ατόμου που επηρεάζει τη διαδικασία βάδισης και τράβηξαν φωτογραφίες από 10 διαφορετικές γωνίες και 10 διαφορετικά ύψη από κάθε σημείο και τις αποθήκευσαν στη βάση δεδομένων. Τα αποτελέσματά τους έδειξαν ότι η ακρίβεια της διαδικασίας ήταν 80%.

Ο Kolbe [60] επανεξέτασε τις μελέτες που σχετίζονται με την πλοήγηση για πεζούς και επικεντρώθηκε σε δύο νέες προσεγγίσεις αξιολογώντας τις προηγούμενες. Μία από τις ιδέες του είναι να καταγράψει τα βίντεο από τη διαδρομή που ακολουθεί ο πεζός και τον τρόπο που επιστρέφει. Μετά από αυτή τη διαδικασία, χρησιμοποιούν τις πληροφορίες σημαντικών τοποθεσιών στο δρόμο. Η άλλη είναι να αποθηκεύει τις πληροφορίες τοποθεσίας όλων των διασταυρώσεων στην πόλη για να χρησιμοποιηθούν από το σύστημα πλοήγησης. Στη συνέχεια, αυτές οι πληροφορίες ενσωματώνονται και δημιουργείται μια πανοραμική όψη της περιοχής. Ως αποτέλεσμα της έρευνας του Kolbe, υπάρχουν δύο γεγονότα που μπορούν να χρησιμοποιηθούν στη διαδικασία πλοήγησης για πεζούς. Μια πλοήγηση βάσει τοποθεσίας και μια πλοήγηση με βάση την επαυξημένη πραγματικότητα για υπαίθρια περιβάλλοντα. Στην πλοήγηση βάσει τοποθεσίας, οι πληροφορίες τοποθεσίας ενός ατόμου λαμβάνονται αμέσως και τα κτήρια, οι δρόμοι και τα αντικείμενα γύρω εμφανίζονται στο κινητό τηλέφωνο του χρήστη. Στην πλοήγηση

που βασίζεται στην επαυξημένη πραγματικότητα, η τρέχουσα τοποθεσία του ατόμου καθορίζεται από το GPS, την κάμερα και τις συσκευές πυξίδας που βρίσκονται στο κράνος του ατόμου. Οι πληροφορίες για την θέση των κτηρίων, των δρόμων και των αντικειμένων στο εξεταζόμενο πεδίο αποθηκεύονται στην εφαρμογή.

Οι Liu και Zlatonava [61] ανέπτυξαν έναν νέο τρόπο εύρεσης λύσεων για συστήματα πλοήγησης εσωτερικού χώρου. Σχεδίασαν τους χάρτες τους σε 2D και χρησιμοποίησαν τον αλγόριθμο του Dijkstra για να βρουν τη συντομότερη διαδρομή.

Οι Jimenez [62] και άλλοι προσπάθησαν να βρουν την καλύτερη λύση για κινητά τηλέφωνα με αισθητήρες MEMS συγκρίνοντας αλγόριθμους μέτρησης βημάτων που μπορούν να χρησιμοποιηθούν σε εξωτερικούς και εσωτερικούς χώρους. Χρησιμοποίησαν αισθητήρες MEMS αφού ήταν μικροί και ελαφριοί. Τοποθέτησαν αισθητήρες MEMS στο δεξί πόδι του χρήστη και ανέλυσαν τα εισερχόμενα δεδομένα.

Οι Kim και Jun [63] ανέπτυξαν σύστημα πλοήγησης σε εσωτερικούς χώρους χρησιμοποιώντας AR. Τοποθέτησαν μια οθόνη σε ένα κράνος, μαζί με κάμερα και δόθηκε στον χρήστη ένα tablet. Αποθήκευσαν τις εικόνες που είχαν ληφθεί προηγουμένως σε πολλές τοποθεσίες του κτηρίου και τις έστειλαν στον απομακρυσμένο διακομιστή. Εικόνες από την κάμερα κατά μήκος της διαδρομής του χρήστη που περπατούσε με πλοήγηση στάλθηκαν σε έναν απομακρυσμένο διακομιστή με ασύρματη σύνδεση και πριν από τη σύγκριση εικόνων στη βάση δεδομένων, αυτές οι εικόνες αναλύθηκαν με τεχνικές επεξεργασίας εικόνας. Εάν οι εικόνες στη βάση δεδομένων ταιριάζουν με την εικόνα της τοποθεσίας του χρήστη, αναγνώριζαν την τοποθεσία του χρήστη ως σωστή. Έδειχναν επίσης τις πληροφορίες κατεύθυνσης στην οθόνη για καθοδήγηση. Διαπίστωσαν ότι το ποσοστό επιτυχίας της μελέτης τους ήταν 89%.

### 1.3 Δομή Εργασίας

Τα υπόλοιπα κεφάλαια της εργασίας δομούνται ως εξής:

Στο Κεφάλαιο 2, παρουσιάζεται η διαφορά μεταξύ του positioning σε εσωτερικούς και εξωτερικούς χώρους καθώς και τα τέσσερα στάδια (η αναπαράσταση του κτηρίου όπου θα χρησιμοποιηθεί το σύστημα, η μέθοδος για την τοποθέτηση/παρακολούθηση των κινήσεων του χρήστη, η εφαρμογή αλγορίθμων εύρεσης μονοπατιών και παρουσίαση περιεχομένου πλοήγησης μέσω ενδείξεων AR) που απαιτούνται για την ανάπτυξη μιας εφαρμογής πλοήγησης σε εσωτερικούς χώρους μέσω ενδείξεων AR.

Στο Κεφάλαιο 3, γίνεται αρχικά αναφορά για την ιστορία του AR, τα είδη επαύξησης και τα στοιχεία που κρύβονται πίσω από την λειτουργικότητά του. Στην συνέχεια παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα του AR για τις φορητές συσκευές και οι τεχνολογίες που χρησιμοποιούνται για την επίτευξή του.

Στο Κεφάλαιο 4, γίνεται αναφορά στην Unity στην οποία θα γίνει στην συνέχεια η ανάπτυξη της εφαρμογής μας. Αναφέρονται αναλυτικά το ARKit της Apple και το ARCore της Google, καθώς και το ARFoundation που ενσωματώνει η Unity. Στην συνέχεια γίνεται παρουσίαση του Vuforia το οποίο χρησιμοποιήσαμε στην εφαρμογή μας έναντι του ARFoundation, αλλά και του NavMesh.

Στο Κεφάλαιο 5, παρουσιάζουμε αναλυτικά την συλλογιστική πορεία για την ανάπτυξη της τελικής εφαρμογής, το προτεινόμενο σύστημα καθώς και μια σύνοψη για την λειτουργία της. Παρουσιάζονται τα εργαλεία και οι τεχνολογίες που θα χρησιμοποιηθούν για κάθε βήμα της ανάπτυξής της.

Στο Κεφάλαιο 6, γίνεται αναλυτική παρουσίαση των βημάτων που ακολουθήθηκαν για την ανάπτυξη της εφαρμογής από το αρχικό μέχρι και το τελικό στάδιο. Συγκεκριμένα παρουσιάζουμε την κατασκευή του 3d model του κτηρίου, την τοποθέτηση όλων των πινακίδων στο κτήριο ώστε να μπορούν να σκαναριστούν από τον χρήστη, την επίτευξη της εναλλαγής των ορόφων, την κατασκευή των ενδείξεων AR και το τελικό UI.

Στο Κεφάλαιο 7, γίνεται η σύνοψη της λειτουργίας της εφαρμογής καθώς και παρουσίαση διάφορων προκλήσεων που αντιμετωπίσαμε και περιορισμών που προέκυψαν.

# **I ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ**

---

# 2 Εντοπισμός Θέσης Σε Εσωτερικούς Χώρους

## 2.1 Εισαγωγή

Ένας από τους πρωταρχικούς στόχους του Indoor Localization System (ILS) είναι ο εντοπισμός της στατικής θέσης ενός αντικειμένου (localization), η παρακολούθηση της πραγματικής ακολουθίας θέσεων οποιουδήποτε κινούμενου αντικειμένου (tracking) και η πλοήγηση (navigation) με ευκολία στο εσωτερικό περιβάλλον [2]. Στη διαχείριση τεχνικών έργων, αυτή η τεχνολογία έχει χρησιμοποιηθεί για την παρακολούθηση της εργασίας, των υλικών και των μηχανημάτων [3]. Άλλες εφαρμογές του ILS περιλαμβάνουν την παρακολούθηση ασθενών και άλλων οντοτήτων σε νοσοκομεία [4], εντοπισμό σε σκοτεινό περιβάλλον [5], Location-Based Services (LBS) εσωτερικού χώρου [6][7], geofencing και παρακολούθηση κατανομής πληρότητας σε πραγματικό χρόνο [7], πλοήγηση για άτομα με προβλήματα όρασης [8], παρακολούθηση σε αυτόνομα συστήματα [9] κ.λπ. Η επιτήρηση μπορεί να είναι ένα άλλο παράδειγμα όπου ένας ύποπτος μπορεί να εντοπιστεί σε εσωτερικούς χώρους. Σήμερα, το ILS έχει γίνει πολύ δημοφιλές για τη διαχείριση και παρακολούθηση περιουσιακών στοιχείων [10] [11]. Σε μια αποθήκη, η παρακολούθηση περιουσιακών στοιχείων όπως ο ηλεκτρονικός εξοπλισμός και ο κατασκευαστικός εξοπλισμός έχει μεγάλη ζήτηση [12][13]. Η παρακολούθηση ειδών σε ολόκληρη την αλυσίδα εφοδιασμού (supply chain) [14], το αυτόνομο σύστημα παρακολούθησης [15], η προστασία από κλοπή και άλλα ζητήματα που σχετίζονται με την ασφάλεια [11] είναι η κύρια απαίτηση της παρακολούθησης περιουσιακών στοιχείων εσωτερικού χώρου. Το IndoorAtlas2 είναι μια εφαρμογή παρακολούθησης στοιχείων εσωτερικών χώρων που παρακολουθεί τα ηλεκτρονικά στοιχεία, συμπεριλαμβανομένων φορητών υπολογιστών, smartphone, συσκευών IoT (Internet of Things) καθώς και βιομηχανικού εξοπλισμού και ιατρικών συσκευών.

Επιπλέον, ο εντοπισμός σε εσωτερικούς χώρους χρησιμοποιείται για την προώθηση της επιχείρησης. Για παράδειγμα, στο Ηνωμένο Βασίλειο, η Intu [16] ανέπτυξε μια τεχνολογία που παρέχει ευκαιρίες σε μικρούς λιανοπωλητές να εξερευνήσουν και να εκμεταλλευτούν ενδιαφέροντα επιχειρηματικά μοντέλα, όπως ειδοποιήσεις push που στέλνουν εκπαιδευτικά κουπόνια. Επιπλέον, κατά τη διάρκεια καταστροφών, όπως το ξέσπασμα πυρκαγιάς σε ένα κτήριο, ο εντοπισμός στους εσωτερικούς χώρους μπορεί να βοηθήσει πολύ τους πυροσβέστες για να σωθεί το μέγιστο πλήθος ζώων.



## 2.2 Συστήματα Πλοήγησης σε Εξωτερικούς Χώρους

Τα τελευταία χρόνια, η σημασία των συστημάτων πλοήγησης αυξάνεται λόγω της προσβασιμότητας και της λειτουργικότητάς τους, επιτρέποντας στους χρήστες να έχουν πρόσβαση από την κινητή συσκευή τους σε τοποθεσίες και πληροφορίες σχετικά με χώρους σε διαφορετικά σενάρια. Τα συστήματα πλοήγησης κατηγοριοποιούνται σε συστήματα για εξωτερικούς (outdoor) και εσωτερικούς (indoor) χώρους.

Όσον αφορά τα outdoor περιβάλλοντα, τα Global Navigation Satellite Systems (GNSSs)—π.χ. Global Positioning System (GPS), Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS), Galileo και BeiDou [17]— υιοθετούνται ευρέως για τον εντοπισμό θέσης σε παγκόσμιο επίπεδο [18]. Δηλαδή, το GNSS υποτίθεται ότι παρέχει ακριβή εντοπισμό θέσης οπουδήποτε στη Γη. Η ακρίβεια του GPS, όσον αφορά το σφάλμα εντοπισμού θέσης, στα smartphones είναι συνήθως ακτίνας 4,9 m σε συνθήκες καθαρού ουρανού, αλλά είναι ικανή για ακρίβεια εκατοστών όταν χρησιμοποιείται σε συνδυασμό με δέκτες διπλής συχνότητας ή/και συστήματα επαύξησης [19]. Παρά την υψηλή ακρίβεια και την παγκόσμια κάλυψη που παρέχουν τα GNSS, δεν μπορούν να λειτουργήσουν σωστά σε εσωτερικούς χώρους. Η ισχυρή εξασθένηση του σήματος, η παρουσία βαρέων σημάτων πολλαπλών διαδρομών και άλλες πηγές παρεμβολών ακυρώνουν το GNSS ως λύση εντοπισμού θέσης σε εσωτερικούς χώρους [18][20].

## 2.3 Συστήματα Πλοήγησης σε Εσωτερικούς Χώρους

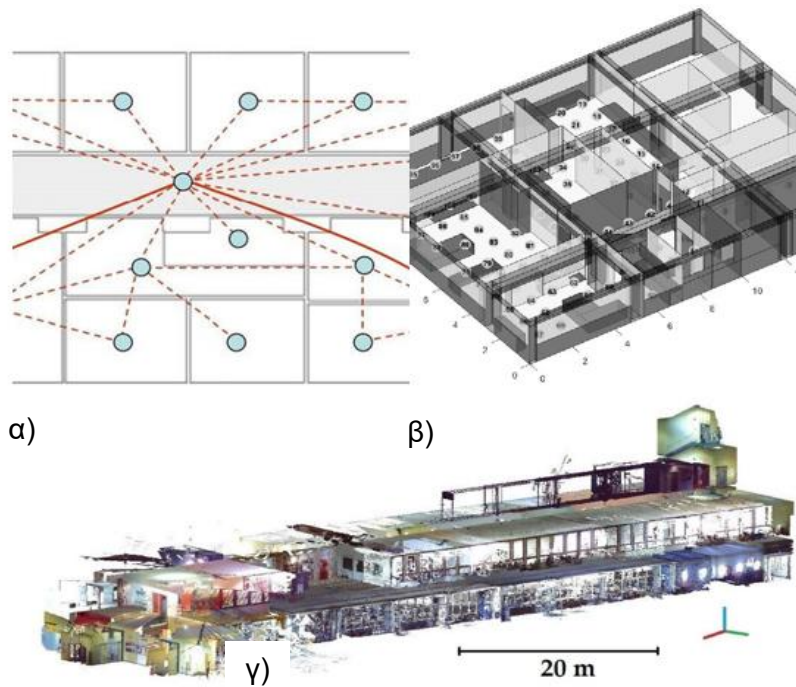
Το GPS δεν εφαρμόζεται για localization σε εσωτερικούς χώρους καθώς τα δορυφορικά σήματα δεν μπορούν να διεισδύσουν καλά στον πολύπλοκο εσωτερικό χώρο. Επιπλέον, για τις παραδοσιακές positioning techniques σε εξωτερικούς χώρους απαιτείται μέτρηση οπτικής επαφής, όπως είναι η τριγωνοποίηση και ο τριγωνισμός. Ως εκ τούτου, τέτοιες τεχνικές δεν είναι κατάλληλες σε εσωτερικούς χώρους με τοίχους, εμπόδια κ.λπ. [21]

Η ανάπτυξη συστημάτων πλοήγησης εσωτερικών χώρων έχει αποτελέσει αντικείμενο ποικίλων ερευνητικών εργασιών, οι οποίες έχουν οδηγήσει σε μια ποικιλία τεχνολογιών, μεθόδων και υποδομών για την συγκεκριμένη υλοποίηση. Αυτό, καθιστά δύσκολο να δημιουργηθεί ένα πρότυπο για την ανάπτυξή τους.

Για την εκτέλεση πλοήγησης σε εσωτερικούς χώρους που ενσωματώνουν AR (Augmented Reality), οι μέθοδοι (methods), οι τεχνικές (techniques) και οι τεχνολογίες (technologies) που χρησιμοποιούνται συνήθως για την ανάπτυξή τους αποτελούνται κυρίως από τέσσερα στάδια: μια αναπαράσταση του κτηρίου όπου θα χρησιμοποιηθεί το σύστημα, μια μέθοδος για την τοποθέτηση/παρακολούθηση των κινήσεων του χρήστη, εφαρμογή αλγορίθμων εύρεσης μονοπατιών και παρουσίαση περιεχομένου πλοήγησης μέσω ενδείξεων AR [22].

## 2.4 Αναπαράσταση του Κτηρίου

Όσον αφορά την αναπαράσταση του κτηρίου, χρησιμοποιούνται συχνά τεχνικές όπως γραφήματα, τρισδιάστατα μοντέλα (3d models) και σύννεφα σημείων (point clouds) (βλ. Εικόνα 2.1). Ωστόσο, η επιλογή μεταξύ της μίας ή της άλλης τεχνικής μοντελοποίησης συνήθως εξαρτάται από τη στρατηγική απόδοσης του κτηρίου που μας ενδιαφέρει και τη δημιουργία ενός χάρτη ή σχεδίου. Αυτό μπορεί να επιτευχθεί για παράδειγμα, καταγράφοντας σχέδια για τη δημιουργία γραφήματος [23] ή αποτυπώνοντας μια αναπαράσταση σκανάροντας την τοποθεσία και δημιουργώντας ένα 3d model ή point clouds [24].



Εικόνα 2.1 Αναπαράσταση κτηρίου με α) γράφημα β) 3d model γ) point clouds

Από την άλλη πλευρά, όσον αφορά τα μοντέλα δεδομένων, ιδιόκτητες μορφές χρησιμοποιούνται συχνά (και προσαρμόζονται στις απαιτήσεις της εφαρμογής) για την αναπαράσταση εσωτερικών χώρων κτηρίων, διαδρομών διαθέσιμες στον χρήστη (μπορεί να υπάρχουν περιορισμένης πρόσβασης περιοχές) και πληροφορίες σημείων ενδιαφέροντος (point of interest) (τακτικά με σχεσιακές βάσεις δεδομένων). Θα ήταν επιθυμητό και χρήσιμο να ενσωματωθούν μορφές για την ενοποίηση δεδομένων και την ανεξαρτησία της πλατφόρμας, όπως πληροφορίες που αντιπροσωπεύονται με πρότυπα του Σημασιολογικού Ιστού [25].

## 2.5 Positioning / Tracking των Κινήσεων του Χρήστη

### 2.5.1 Εισαγωγή στο Indoor Positioning System

Το IPS (Indoor Positioning System) λειτουργεί σε εσωτερικούς χώρους, όπου τα δορυφορικά συστήματα εντοπισμού θέσης (GPS, Galileo, Glonass) δεν παρέχουν ακρίβεια. Επιπλέον, η τεχνολογία IPS μπορεί να χρησιμοποιηθεί και για positioning σε εξωτερικό χώρο, αν και η ίδια τεχνολογία που χρησιμοποιείται στο εσωτερικό και στο εξωτερικό μπορεί να δώσει διαφορετικά αποτελέσματα [37]. Τα IPS χρησιμοποιούνται για τον προσδιορισμό της τρέχουσας θέσης ενός αντικείμενου ή ενός ατόμου. Ένα IPS αποτελείται από πομπούς (transmitters) που βρίσκονται στο αντικείμενο/άτομο που παρατηρείται και έναν ασύρματο αναγνώστη (reader) σημάτων που ανιχνεύει και διαβάζει τα λαμβανόμενα σήματα. Η θέση των πομπών και των δεκτών μπορεί να εναλλάσσεται, κάτι που εξαρτάται από τον τύπο της εφαρμογής. Διάφορες ασύρματες τεχνολογίες χρησιμοποιούνται για την επικοινωνία μεταξύ πομπών και αναγνώστων. Τα πιο δημοφιλή είναι τα συστήματα Ultra-Wideband (UWB), Wireless Fidelity (Wi-Fi), Bluetooth και ενεργά/παθητικά συστήματα RFID [38]. Η προσοχή που λαμβάνει το Visible Light Communication (VLC) αυξάνεται συνεχώς, καθώς ο υψηλός ρυθμός δεδομένων, η ασφάλειά του, η μη παρεμβολή στο φάσμα ραδιοσυχνοτήτων (Radio Frequency ή RF) και η υψηλή ανάλυση [39] το καθιστούν κατάλληλη επιλογή για μια εφαρμογή εντοπισμού.

Με την τοποθέτηση του reader σε σταθερές θέσεις μέσα σε ένα περιβάλλον (π.χ. μια αποθήκη), η θέση των κινητών αναμεταδοτών μπορεί να προσδιοριστεί με την ανάλυση των αλλαγών στις ιδιότητες του σήματος (καθυστέρηση, εξασθένηση, αλλαγές φάσης) και την ανταλλαγή δεδομένων μεταξύ του αναγνώστη και του αναμεταδότη [38]. Θα πρέπει να χρησιμοποιούνται εξελιγμένες μαθηματικές και ευρετικές μέθοδοι για την επεξεργασία εξαιρετικά ασταθών και θορυβωδών σημάτων που έρχονται από τους πομπούς [40].

### 2.5.2 Τεχνολογίες, Τεχνικές και Μέθοδοι

Όπως προαναφέραμε, ένα IPS χρησιμοποιείται για την παροχή εκτίμησης θέσης σε εσωτερικά περιβάλλοντα. Ωστόσο, ο σχεδιασμός ενός IPS εξαρτάται σε μεγάλο βαθμό από το γενικό πλαίσιο και είναι χτισμένο σύμφωνα με τρία κύρια στοιχεία:

- Πρώτον, η **τεχνολογία εντοπισμού θέσης σε εσωτερικούς χώρους** είναι ο πυρήνας του IPS και θα είναι κατά κάποιο τρόπο δείκτης του πλαισίου της ανάπτυξης, δηλαδή της αναμενόμενης ακρίβειας καθώς και τυχόν πρόσθετων απαιτήσεων και περιορισμών. Σε αντίθεση με τον εντοπισμό θέσης σε εξωτερικούς χώρους, όπου τα συγχρονισμένα, χρονικά σφραγισμένα ραδιοφωνικά σήματα μεταδίδονται από έναν αστερισμό δορυφόρων κατά μήκος της οπτικής επαφής με τον δέκτη, οι τεχνολογίες εντοπισμού θέσης σε εσωτερικούς χώρους είναι ποικίλης φύσης και περιλαμβάνουν γνωστές τεχνολογίες μέτρησης που βασίζονται στην όραση (π.χ. Επικοινωνία Ορατού Φωτός (Visible Light Communication ή VLC)), τη

ραδιοσυχνότητα (π.χ., Διαμόρφωση Συχνότητας (Frequency Modulation ή FM), Wi-Fi, BLE), την ακοή και την αδράνεια.

- Δεύτερον, η **τεχνική εντοπισμού θέσης σε εσωτερικούς χώρους** υποδεικνύει ποια δεδομένα/μετρήσεις ή πληροφορίες υποβάλλονται σε επεξεργασία για τον υπολογισμό της θέσης. Για παράδειγμα, η κατεύθυνση και η γωνία από την οποία λαμβάνεται ένα σήμα (Angle of Arrival (AoA)), ο χρόνος που έχει παρέλθει ώστε το σήμα να μεταδοθεί από έναν πομπό στον δέκτη (Time of Arrival (ToA) και παραλλαγές), οι ιδιότητες του καναλιού σε μια σύνδεση επικοινωνίας (Channel State Information (CSI)), η ισχύς του σήματος στην πλευρά του δέκτη (Received Signal Strength Indicator (RSSI)), ή ακόμα και το σύνολο των RSSI από πολλαπλούς εκπομπούς ως μπλοκ (δακτυλικό αποτύπωμα ή fingerprint).
- Τρίτον, η **μέθοδος εντοπισμού θέσης σε εσωτερικούς χώρους** είναι ο συγκεκριμένος αλγόριθμος που χρησιμοποιείται για την επεξεργασία των δεδομένων/μετρήσεων ή των πληροφοριών που συλλέγονται για την τοποθέτηση. Στη βιβλιογραφία, περιγράφεται ένα ευρύ φάσμα μεθόδων, από πολύ συγκεκριμένες παραλλαγές γνωστών αλγορίθμων (π.χ. k-Nearest Neighbors (k-NN)), έως και ασαφείς μέθοδοι που αναφέρονται με την τεχνική πάνω στην οποία είναι βασισμένοι (π.χ. fingerprint-based method). Επιπλέον, οι μέθοδοι εντοπισμού θέσης σε εσωτερικούς χώρους μπορεί να είναι συγκεκριμένες για μια συγκεκριμένη τεχνολογία και τεχνική (π.χ. PDR για αδρανειακές μετρήσεις) ή μπορεί να είναι καθολικοί αλγόριθμοι (π.χ. αλγόριθμοι Μηχανικής Μάθησης όπως k-NN ή Support Vector Machines). Συνοψίζοντας, παρόλο που ένα IPS μπορεί να είναι σχετικά απλό, όπως η εφαρμογή του αλγορίθμου k-NN πάνω από fingerprints σημάτων Wi-Fi [26], τα περισσότερα από τα προηγμένα συστήματα είναι πολύπλοκα, π.χ. εφαρμογή εκτεταμένων φίλτρων Kalman (EKF) ή φίλτρα σωματιδίων για συνδυασμό PDR μέσω δεδομένων IMU και δακτυλικών αποτυπωμάτων με βάση το ble [27][28].

### 2.5.3 Υπολογιστική Αρχιτεκτονική και Υποδομή

Με βάση την **υπολογιστική αρχιτεκτονική** (computational architecture), το IPS μπορεί να ταξινομηθεί σε δύο κατηγορίες: server-based, όπως στο Where@UM [29], και server-less/αυτόνομο, όπως στο AnyPlace [30], που υποδεικνύουν πού υπολογίζεται η εκτίμηση της θέσης. Σε μια server-based αρχιτεκτονική, ο διακομιστής επεξεργάζεται τα ακατέργαστα δεδομένα που παρέχονται από κάθε συσκευή, χωρίς να χρησιμοποιεί πληροφορίες από τις άλλες συσκευές, δηλαδή όλες οι εκτιμήσεις του localization πραγματοποιούνται σε έναν απομακρυσμένο διακομιστή. Στην server-less αρχιτεκτονική, κάθε συσκευή λαμβάνει τα ακατέργαστα δεδομένα από αισθητήρες και τα επεξεργάζεται για να αυτοπροσδιορίσει τη θέση, δηλαδή όλες οι εκτιμήσεις του localization πραγματοποιούνται τοπικά στη συσκευή. Και στις δύο περιπτώσεις, η θέση μιας συσκευής εκτιμάται χρησιμοποιώντας τα δεδομένα και τις πληροφορίες που παρέχονται από την ίδια τη συσκευή.

Όσον αφορά την **υποδομή** (infrastructure), η βιβλιογραφία κάνει γενικά διάκριση μεταξύ IPS χωρίς υποδομή (infrastructure-less) και IPS που βασίζεται σε υποδομή (infrastructure-based). Τα infrastructure-less συστήματα δεν απαιτούν την ανάπτυξη υποδομής στην περιοχή για να λειτουργήσουν, π.χ. IPS που βασίζονται σε μαγνητικό πεδίο. Αντίθετα, τα infrastructure-based

IPS απαιτούν μια υποδομή, δηλαδή ένα ή περισσότερα φυσικά στοιχεία που αναπτύσσονται στο περιβάλλον [31] (π.χ. ble beacons ή δέκτες υπερήχων). Για να διαφοροποιήσουν τα συστήματα όπου η υποδομή πρέπει να αναπτυχθεί σκόπιμα, από τα συστήματα που χρησιμοποιούν μια ήδη υπάρχουσα υποδομή (δηλαδή, ευκαιριακά σήματα ή signals of opportunity), ορισμένοι συγγραφείς εντόπισαν μια ενδιάμεση κατηγορία: ευκαιριακά IPS [32]. Για παράδειγμα, τα IPS που βασίζονται σε Wi-Fi θεωρούνται ευκαιριακά, εάν το περιβάλλον δεν έχει τροποποιηθεί για να επιτραπεί η λειτουργία τους (δηλαδή, δεν έχουν αναπτυχθεί σκόπιμα Wi-Fi Access Points (APs) για το IPS).

#### 2.5.4 Συνεργατικά και Μη-Συνεργατικά IPS

Λαμβάνοντας υπόψη τον ρόλο των διαφορετικών παραγόντων στα IPS, η βιβλιογραφία διακρίνει δύο κύριους τύπους: μη-συνεργατικούς (**non-collaborative**) και συνεργατικούς (**collaborative**) [33][34]. Αυτή η ορολογία αναφέρεται στη φάση της λειτουργίας (δηλαδή, εκτίμηση της θέσης), όχι στη φάση συλλογής δεδομένων (αναφοράς) (π.χ. κατασκευή ενός fingerprint radio map). Ως εκ τούτου, τα non-collaborative σχήματα αναφέρονται σε συστήματα που δεν λαμβάνουν υπόψη τη συμμετοχή άλλων χρηστών στον positioning algorithm τους [34]. Αντίθετα, ένα CIPS (Collaborative Indoor Positioning System) είναι ένα σχήμα στο οποίο η θέση προσδιορίζεται με βάση την άμεση ή έμμεση αλληλεπίδραση μεταξύ γειτονικών συσκευών ή διαφορετικών IPS. Να σημειωθεί, ότι οι collaborative προσεγγίσεις δεν πρέπει να συγχέονται με προσεγγίσεις συγχώνευσης δεδομένων ή αισθητήρων. Το collaborative positioning εστιάζει σε συστήματα των οποίων οι ανεξάρτητοι παράγοντες (χρήστες ή συσκευές) ανταλλάσσουν πληροφορίες και υπολογίζουν τις σχετικές αποστάσεις μεταξύ τους για να παρέχουν τη θέση του συνόλου των χρηστών [33][34], ενώ η συγχώνευση αισθητήρων συνδυάζει πληροφορίες από διάφορους αισθητήρες από έναν μόνο παράγοντα για την παροχή της θέσης ενός μόνο χρήστη [35].

Οι τεχνολογικές εξελίξεις και η ανάπτυξη τεχνικών και μεθόδων για τα παραδοσιακά IPS επαναχρησιμοποιούνται σε μεγάλο βαθμό από συνεργατικά συστήματα για τον προσδιορισμό της θέσης των συνεργατικών κόμβων στο CIPS. Ωστόσο, το CIPS εκμεταλλεύεται αυτές τις τεχνολογίες που όχι μόνο επιτρέπουν την εκτίμηση της θέσης αλλά και την ανταλλαγή πληροφοριών μεταξύ των κόμβων. Μέσα σε αυτές τις τεχνολογίες, μπορούμε να διακρίνουμε τις ασύρματες τεχνολογίες (π.χ. Wi-Fi, BLE, Ultra-wide band (UWB)) και τα cellular networks, τα οποία μπορούν να χρησιμοποιηθούν με διάφορα πρωτόκολλα επικοινωνίας, όπως το iBeacon και το Bluetooth, μεταξύ άλλων. Οι μέθοδοι των CIPS είναι πολύ διαφορετικές. Ωστόσο, μερικές από τις πιο γνωστές μεθόδους βασίζονται σε belief propagation και σε μη-Bayesian προσεγγίσεις όπως το Ελάχιστο Τετράγωνο (Least Square ή LS) και η μέγιστη πιθανότητα [36].

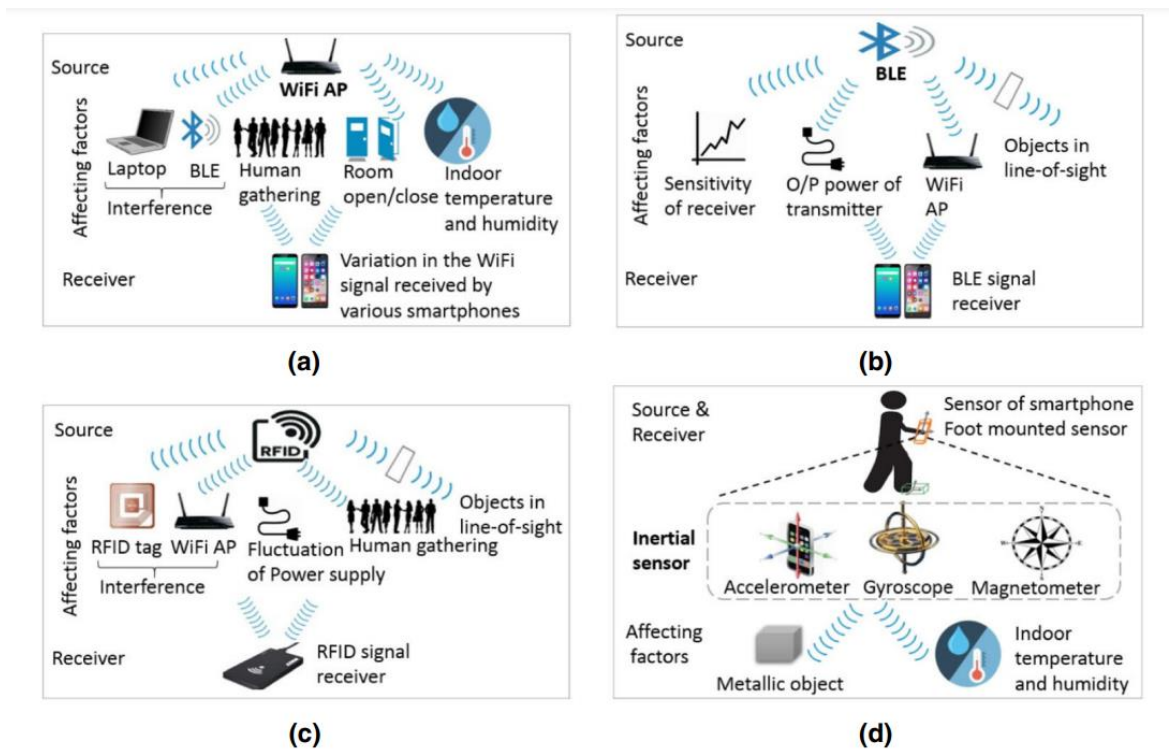
Σε αντίθεση με το μη συνεργατικό IPS, η υπολογιστική αρχιτεκτονική είναι πιο περίπλοκη καθώς η εκτίμηση θέσης δεν εξαρτάται μόνο από τα δεδομένα της συσκευής, αλλά και από τα δεδομένα που συλλέγονται από κοντινές συσκευές. Συνήθως, η υπολογιστική αρχιτεκτονική του CIPS μπορεί να ταξινομηθεί σε δύο κατηγορίες: κεντρική (**centralized**) και αποκεντρωμένη (**decentralized**). Σε μια centralized αρχιτεκτονική, οι κόμβοι/παράγοντες συλλέγουν τα μη επεξεργασμένα δεδομένα από τους αισθητήρες, τα οποία αποστέλλονται σε έναν κεντρικό κόμβο

που υπολογίζει το εκτιμώμενο position όλων των κόμβων. Σε μια decentralized αρχιτεκτονική, ο ρόλος των κόμβων συνίσταται στην απόκτηση και κοινή χρήση (ακατέργαστων ή επεξεργασμένων) σχετικών δεδομένων, αλλά και στην επεξεργασία τους προκειμένου, για παράδειγμα, να αυτοπροσδιορίσουν τη θέση τους. Και στις δύο περιπτώσεις, η τελική θέση μιας συσκευής εκτιμάται συλλογικά χρησιμοποιώντας τα δεδομένα και τις πληροφορίες που παρέχονται από αυτήν και άλλες συσκευές[34].

## 2.5.5 Θετικά και Αρνητικά Indoor Positioning Τεχνολογιών

Προκειμένου να προσδιοριστεί ποια τεχνολογία εντοπισμού θέσης είναι καταλληλότερη, συζητάμε τις πιο δημοφιλείς, συμπεριλαμβανομένου του GPS, του εντοπισμού θέσης βάσει Wi-Fi, του εντοπισμού θέσης βάσει φάρου (beacon) BLE (BLE-beacon), του εντοπισμού θέσης με βάση το μαγνητικό πεδίο (magnetic field based) και του εντοπισμού θέσης με βάση την όραση (vision-based positioning).

Μερικοί από τους παράγοντες που επηρεάζουν τα εκπεμπόμενα σήματα και τις ενδείξεις των αισθητήρων φαίνονται στην Εικόνα 2.2.



Εικόνα 2.2 Διάφοροι παράγοντες που επηρεάζουν α)τα σήματα WiFi β)τα Bluetooth σήματα γ)τα σήματα RFID δ) τιμές μέτρησης ενός αδρανειακού αισθητήρα

Τα σήματα GPS μπορούν να εξασθενήσουν όταν υπάρχουν εμπόδια, όπως οι τοίχοι και οι στέγες, μεταξύ των δεκτών και του δορυφόρου [71]. Επιπλέον, είναι επίσης εξαιρετικά δύσκολο να προσδιοριστεί το επίπεδο του δαπέδου [72]. Επομένως, το GPS είναι πολύ αδύναμο για να παρέχει χρήσιμες πληροφορίες εντοπισμού θέσης σε εσωτερικούς χώρους, ειδικά σε πολυώροφα κτήρια.

Τα σήματα RFID (Radio Frequency Identification) επηρεάζονται κυρίως από απόσβεση, εξασθένιση, σκίαση, ανθρώπινη κίνηση, συνθήκες εσωτερικού περιβάλλοντος, παρουσία συσκευών παρεμβολής, εμπόδια και ούτω καθεξής (όπως φαίνεται στην Εικόνα 2.2 a,b,c). Ως εκ τούτου, θόρυβοι συσσωρεύονται σε αυτά τα σήματα και λαμβάνονται λανθασμένα σήματα από τον δέκτη. Οι ίδιοι παράγοντες έχουν επίπτωση στα σήματα WiFi και BLE.

Οι μετρήσεις των αδρανειακών αισθητήρων (inertial sensors) επηρεάζονται κυρίως από τη διακύμανση της θερμοκρασίας και την παρουσία σιδηρομαγνητικών αντικειμένων στους εσωτερικούς χώρους όπως φαίνεται στην Εικόνα 2.2 d. Γενικά, η σταθερότητα πόλωσης και ο θερμομηχανικός θόρυβος είναι τα πιο συνηθισμένα σφάλματα που επηρεάζουν τους inertial sensors [56]. Το συστηματικό σφάλμα, που δημιουργείται από τους παράγοντες αυτούς, ενσωματώνεται με τις μετρήσεις των βημάτων, την εκτίμηση του μήκους του βήματος, της κατεύθυνσης, του προσανατολισμού, κ.λπ.

Το positioning που βασίζεται σε Wi-Fi έχει χαμηλό κόστος εγκατάστασης υλικού και υψηλή διαθεσιμότητα, αφού σήμερα εγκαθίσταται Wi-Fi μέσα σε κάθε κτήριο [43]. Ωστόσο, καθώς χρησιμοποιεί fingerprinting για τη συλλογή σημάτων Wi-Fi και τη συσχέτιση των σημάτων με την τοποθεσία, η διαδικασία είναι χρονοβόρα. Τα εμπόδια μέσα στα κτήρια καθώς και οι υπόλοιποι παράγοντες που προαναφέρθηκαν για τα σήματα RFID μπορεί να προκαλέσουν προβλήματα όπως για παράδειγμα φαινόμενα πολλαπλών διαδρομών που οδηγούν σε αναξιόπιστη ακρίβεια [44].

Το BLE beacon καταναλώνει λιγότερη ενέργεια, είναι μικρό σε μέγεθος, φθινό και εύκολο στη ρύθμιση [45]. Επιπλέον, όσον αφορά την κατανάλωση ενέργειας, το BLE είναι πιο ενεργειακά αποδοτικό από άλλα ασύρματα σήματα [46]. Ωστόσο, παρόμοια με τον εντοπισμό θέσης που βασίζεται σε Wi-Fi, η διαδικασία του fingerprinting είναι χρονοβόρα. Επίσης έχει τον κίνδυνο πολλαπλών διαδρομών και σε ακόμα μεγαλύτερο βαθμό από τον εντοπισμό θέσης που βασίζεται σε Wi-Fi [45].

Τα σήματα μαγνητικού πεδίου είναι φυσικοί πόροι, επομένως δεν απαιτείται πρόσθετη υποδομή. Επίσης, δεν επηρεάζεται από εμπόδια, όπως τοίχοι ή στέγες [47]. Ωστόσο, το μεταλλικό στοιχείο μπορεί να προκαλέσει εφέ σκληρού (hard-iron) και μαλακού (soft-iron) σιδήρου, γεγονός που επηρεάζει την ακρίβειά του [48]. Η διαδικασία fingerprinting είναι επίσης χρονοβόρα.

Το vision-based positioning μπορεί να κατηγοριοποιηθεί σε markerless tracking και σε ορατό marker-based tracking [49]. Το markerless tracking χρησιμοποιεί φυσικά χαρακτηριστικά, όπως γωνίες και υφές, του περιβάλλοντος για την παροχή πληροφοριών θέσης [50], επομένως, δεν απαιτείται πρόσθετη υποδομή. Ωστόσο, απαιτούνται υψηλοί υπολογιστικοί πόροι και πόροι μνήμης επειδή το σύστημα θα πρέπει να συνεχίσει να σαρώνει και να συγκρίνει το περιβάλλον με μεγάλης κλίμακας βάση δεδομένων [51]. Αυτή η τεχνική είναι επίσης αναξιόπιστη εάν χρησιμοποιείται σε δυναμικά μεταβαλλόμενο περιβάλλον [47]. Για παράδειγμα, όταν μετακινείται το τραπέζι ή η καρέκλα, το σύστημα ενδέχεται να μην αναγνωρίζει πλέον την τοποθεσία.

Επομένως, θα χρειαστεί να ενημερώνουμε συχνά τη βάση δεδομένων για να ταιριάζει με το περιβάλλον, κάτι που είναι δαπανηρό.

Ο ορατός marker αναφέρεται σε οποιοδήποτε κώδικα που βασίζεται σε εικόνα, όπως ο κωδικός QR (QR code) [52]. Ο δείκτης θα τοποθετηθεί σε συγκεκριμένη θέση για να παρέχει πληροφορίες τοποθεσίας κατά τη σάρωση. Με τη χρήση αυτής της τεχνικής απαιτείται πρόσθετο κόστος για την τοποθέτηση των markers στο κτήριο. Εκτός από αυτό, για να ενημερώσει τη θέση του, ο χρήστης θα πρέπει να συνεχίζει να βρίσκει και να σαρώνει τον marker. Και οι δύο μέθοδοι τοποθέτησης με βάση την όραση μπορούν να είναι σχετικά ακριβείς για την παρακολούθηση αντικειμένων σε χώρους με μικρές διαστάσεις, κάτι που είναι κατάλληλο για indoor positioning [53]. Ωστόσο, και οι δύο θα επηρεαστούν από πηγές, όπως το φως και ο καιρός [47]. Για παράδειγμα, όταν υπάρξει black out, και οι δύο τεχνικές δεν θα λειτουργήσουν.

Όπως προαναφέρθηκε, τα σήματα GPS από μόνα τους είναι πολύ αδύναμα για να παρέχουν indoor positioning. Τα σήματα Wi-Fi και BLE-beacon ενδέχεται να επηρεαστούν από εμπόδια μέσα στο κτήριο, ενώ το vision-based positioning επηρεάζεται επίσης από πηγές φωτός. Επομένως, η καλύτερη λύση θα ήταν ένα σύστημα positioning το οποίο συνδυάζει διαφορετικές τεχνολογίες για να αλληλοσυμπληρώνονται. Με βάση το [54], ο συνδυασμός σημάτων Wi-Fi και μαγνητικού πεδίου παρέχει μέση ακρίβεια 1,20 μέτρων, η οποία είναι καλύτερη από το να είχαμε μόνο του το Wi-Fi. Επιπλέον, η τεχνική που προτείνεται στο [55] με την ενσωμάτωση του Wi-Fi και του μαγνητικού πεδίου μπορεί να επιτύχει αξιόπιστα 0,836 μέτρα ακρίβειας στο πειραματικό περιβάλλον. Επίσης, καθώς τα περισσότερα κτήρια έχουν ήδη εγκατεστημένο Wi-Fi και το μαγνητικό πεδίο είναι φυσικός πόρος, το πρόσθετο κόστος υποδομής μπορεί να ελαχιστοποιηθεί. Ωστόσο, η τεχνική είναι χρονοβόρα γιατί απαιτεί fingerprinting.

## **Ετερογένεια συσκευής**

Πολλά smartphones από το ίδιο μέρος την ίδια στιγμή ενδέχεται να καταγράφουν διαφορετική ισχύ σήματος από την ίδια πηγή λόγω των διαφορετικών χαρακτηριστικών του hardware. Ως εκ τούτου, ο εντοπισμός μιας άγνωστης τοποθεσίας γίνεται μια δύσκολη εργασία για έναν αλγόριθμο μηχανικής μάθησης όταν οι συσκευές εκπαίδευσης και δοκιμής είναι διαφορετικές. Ωστόσο, στην πράξη, κατά τη φάση της εκπαίδευσης, δεν είναι δυνατή η συλλογή fingerprints από όλα τα χαρακτηριστικά της συσκευής. Ένας πιθανός τρόπος είναι να ενημερώσετε το μοντέλο εκπαίδευσης χρησιμοποιώντας διαδικτυακούς αλγόριθμους μάθησης [57].

Συνοψίζοντας τα παραπάνω οι εταιρείες για την υλοποίηση συστημάτων πλοήγησης σε εσωτερικούς χώρους χρησιμοποιούν κατά κύριο λόγο τεχνολογίες όπως Bluetooth beacons, Wi-Fi Fingerprinting και UWB. Ακόμη και αν αυτές οι τεχνολογίες χρησιμοποιούνται ευρέως υπάρχουν ορισμένα προβλήματα που καθιστούν αδύνατη την χρήση τους στην παρούσα εργασία. Τέτοια προβλήματα είναι:

- Στην τεχνολογία Bluetooth θα πρέπει να εγκατασταθεί ένας τεράστιος αριθμός beacon που είναι πολύ δαπανηροί. Και καθιστά δύσκολο τον εντοπισμό των συσκευών όταν αποστέλλονται περισσότερα αιτήματα συνδεσιμότητας.



· Στην τεχνολογία του Wi-Fi fingerprinting το κύριο πρόβλημα αφορά την ασφάλεια του δημόσιου Wi-Fi. Καθώς η συχνότητα ποικίλλει καθώς μετακινούμαστε από το ένα μέρος στο άλλο, δημιουργούνται προβλήματα συνδεσιμότητας πράγμα που οδηγεί περαιτέρω σε απώλεια του tracking.

· Στην τεχνολογία UWB το κύριο πρόβλημα είναι η ισχύς του σήματος που εκπέμπεται από τη συσκευή UWB. Αυτή η συσκευή εκπέμπει πολύ χαμηλά σήματα, γεγονός που δημιουργεί δυσκολίες στη σύλληψη σήματος και την παρακολούθηση της θέσης ενός ατόμου. Εκτός από αυτά τα προβλήματα που σχετίζονται με την τεχνολογία, υπάρχουν επίσης γενικότερα μειονεκτήματα όπως ακριβή εφαρμογή, ανάγκη εγκατάστασης υλικού, εξάρτηση από την πρόσβαση στο διαδίκτυο.

## 2.6 Αλγόριθμοι Εύρεσης Μονοπατιών

Τέλος, για την ανάπτυξη της εφαρμογής σημαντικό κομμάτι αποτελεί η επιλογή μιας διαδρομής πλοήγησης ανάμεσα σε δύο σημεία (σημείο εκκίνησης και τελικός προορισμός) και η αναπαράσταση του αποτελέσματος με τεχνικές AR (το οποίο θα εξετάσουμε στο επόμενο κεφάλαιο, κεφάλαιο 2.7). Οι δημοφιλείς τεχνικές εύρεσης μονοπατιών (pathfinding) περιλαμβάνουν τον αλγόριθμο του Dijkstra, τον αλγόριθμο A\* (A-star) και τον αλγόριθμο ACO (Ant Colony Optimization) [69]. Ο αλγόριθμος του Dijkstra είναι ένας αλγόριθμος εύρεσης συντομότερου μονοπατιού μεταξύ δύο κόμβων, ενώ ο αλγόριθμος A\* είναι μια επέκταση του αλγόριθμου του Dijkstra που συνδυάζει τα χαρακτηριστικά της αναζήτησης ομοιόμορφου κόστους και της καθαρής ευριστικής αναζήτησης για την αναζήτηση της βέλτιστης διαδρομής [70]. Από την άλλη πλευρά, ο αλγόριθμος ACO μιμείται τη συμπεριφορά των μυρμηγκιών που περπατούν γύρω από τον γράφο. Καθώς τα μυρμήγκια θα ακολουθήσουν το μονοπάτι με περισσότερες φερομόνες, έτσι ο αλγόριθμος ACO χρησιμοποιεί επίσης την ίδια προσέγγιση για τον υπολογισμό της διαδρομής [68].

Σύμφωνα με το [69], ο αλγόριθμος A\* είναι πιο αποτελεσματικός από τον αλγόριθμο του Dijkstra. Ωστόσο, όπως φαίνεται στα αποτελέσματα των δοκιμών στο [68], ο αλγόριθμος ACO έχει υψηλότερη ακρίβεια και μικρότερο κόστος χρόνου στην αναζήτηση διαδρομής συγκριτικά με τον αλγόριθμο A\*. Στο πλαίσιο του indoor περιβάλλον, το κόστος υπολογισμού του αλγορίθμου A\* θα είναι πολύ υψηλότερο από τον αλγόριθμο ACO. Σε λίγες περιπτώσεις, η διαδρομή που θα βρεθεί χρησιμοποιώντας τον αλγόριθμο A\* είναι λίγο μικρότερη από αυτή του αλγορίθμου ACO, ωστόσο, ο χρόνος που δαπανάται χρησιμοποιώντας τον αλγόριθμο A\* είναι 8 φορές μεγαλύτερος. Επομένως, το [68] δείχνει ότι ο αλγόριθμος ACO είναι ο κατάλληλος για εύρεση μονοπατιών σε indoor περιβάλλον.

Ο **αλγόριθμος A\*** είναι ένα είδος αλγορίθμου ευριστικής αναζήτησης. Χρησιμοποιεί μια συνάρτηση αξιολόγησης για να καθοδηγήσει την επιλογή του κόμβου:

$$f(n)=g(n)+h(n)$$

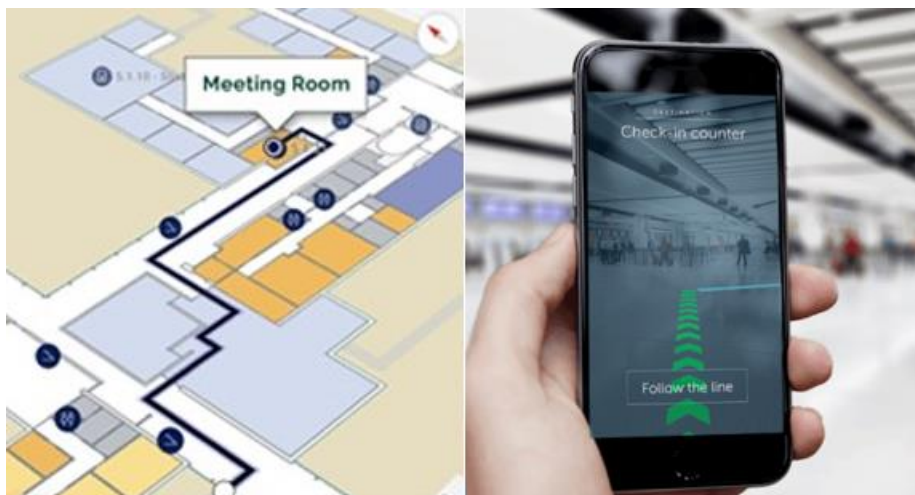
Στον τύπο, το  $g(n)$  αντιπροσωπεύει την τιμή της διαδρομής από το σημείο εκκίνησης έως τον κόμβο  $n$ . Το  $h(n)$  είναι το εκτιμώμενο κόστος της διαδρομής από τον κόμβο  $n$  στο σημείο στόχο. Είναι ευριστικές πληροφορίες που εξαρτώνται από τις προβληματικές περιοχές. Κάποια από τα πλεονεκτήματα του  $A^*$  είναι ότι μπορεί γρήγορα να συγκλίνει, αφού χρησιμοποιεί κανόνες διέλευσης κατά βάθος. Επίσης, έχει καλή καθολικότητα. Είναι κατάλληλο για οποιοδήποτε σχήμα και μέγεθος κτηρίου. Ωστόσο, η λύση του που βασίζεται σε μη κατευθυνόμενο γράφο ισχύει μόνο για ένα δεδομένο κτήριο. Για διαφορετικό κτήριο, πρέπει ο μη κατευθυνόμενος γράφος να επανασχεδιαστεί.

Το 1959, ο E. W. Dijkstra πρότεινε τον αλγόριθμο **Dijkstra**, είναι ένας κλασικός αλγόριθμος για την επίλυση του προβλήματος του συντομότερου μονοπατιού. Μπορεί να βρει τη συντομότερη διαδρομή από έναν κόμβο προς κάθε άλλο κόμβο του γράφου. Η χρονική του πολυπλοκότητα είναι  $O(n^2)$ . Το 'n' είναι το πλήθος των κόμβων στον γράφο. Αυτή η ελάχιστη απόσταση μπορεί να μετρηθεί μόνο με καθαρό μήκος, αλλά μπορεί επίσης να μετρηθεί με άλλους τρόπους, όπως γενικά έξοδα χρόνου, οικονομικό κόστος, απόδοση κ.λπ.

Ο αλγόριθμος Dijkstra είναι απλός και εύκολος, είναι μια εξαιρετική μέθοδος για τον προγραμματισμό διαδρομής ρομπότ και έχει χρησιμοποιηθεί ευρέως στη βελτιστοποίηση δικτύων, τις μεταφορές, τα logistics και άλλα πεδία (π.χ., τα συστήματα πλοήγησης GPS υιοθετούν γενικά αυτή τη μέθοδο για τον σχεδιασμό διαδρομής).

Ο σχεδιασμός διαδρομής περιλαμβάνει τον συνολικό σχεδιασμό διαδρομής και τον τοπικό σχεδιασμό διαδρομής, ο αλγόριθμος Dijkstra ανήκει στον πρώτο. Έτσι μπορεί να εγγυηθεί ότι η παραγόμενη διαδρομή είναι η συντομότερη διαδρομή. Ωστόσο, αυτή η λύση εξακολουθεί να έχει μια αντίφαση μεταξύ της επιβάρυνσης του χρόνου-χώρου και της ακρίβειας.

## 2.7 Πλοήγηση μέσω Ενδείξεων AR



Εικόνα 2.3 Αριστερά: digital map, Δεξιά: AR arrows indicator

Ο ψηφιακός χάρτης (digital map) (βλ. Εικόνα 2.3 αριστερά) χρησιμοποιείται συνήθως για την παροχή καθοδήγησης σε εσωτερικούς χώρους. Ωστόσο, έρευνες έχουν αποδείξει ότι τα γραφικά του υπολογιστή δεν επαρκούν για να αναπαραστήσουν τον πραγματικό κόσμο, επομένως είναι δύσκολο για τους χρήστες να ταιριάζουν τον χάρτη με τον πραγματικό κόσμο. Προκαλεί επίσης σύγχυση επειδή οι χρήστες θα πρέπει να συγκρίνουν το πραγματικό τοπίο και την κατεύθυνση με εκείνα του χάρτη [64]. Προκειμένου να βελτιωθεί η εμπειρία του χρήστη, η τεχνολογία του AR έχει προκύψει. Σε αντίθεση με τον ψηφιακό χάρτη που παρέχει μια third-person όψη του κόσμου, το AR δείχνει μια προβολή first-person της κατεύθυνσης του στόχου που επιτρέπει στους χρήστες να αλληλεπιδρούν με τη συσκευή φυσικά [65]. Το AR έχει επίσης αποδειχθεί ότι μειώνει τον χρόνο και τον κόπο που ο άνθρωπος καταναλώνει για να βρει τον τελικό προορισμό του [66]. Αυτό επιτυγχάνεται μέσω εικονικών αντικειμένων (virtual objects), όπως βέλη, γραμμές ή animated αντικείμενα (βλ. Εικόνα 2.3 δεξιά).

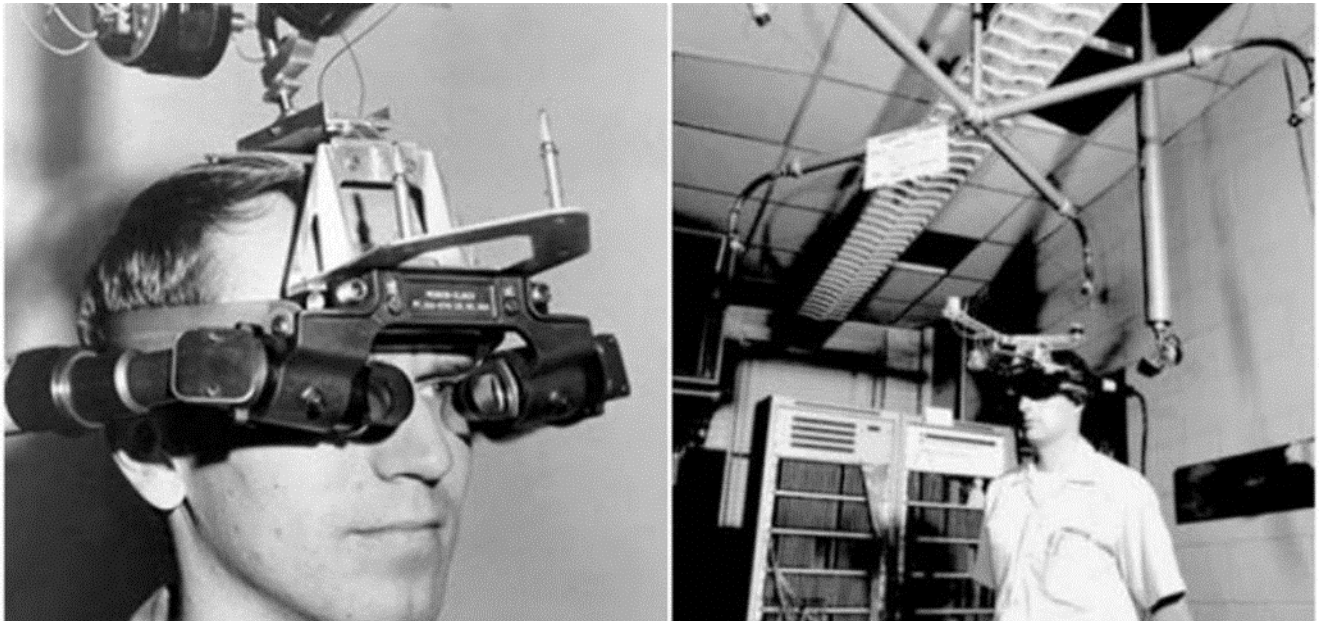
Σε μια πρόσφατη εργασία, ο Brush και άλλοι [67] συνέστησαν την activity-based πλοήγηση. Οι οδηγίες πλοήγησης διατυπώνονται ως μια σειρά από ενέργειες που ο χρήστης πρέπει να ακολουθήσει – π.χ. «περπατήστε 10 βήματα βόρεια» και μετά «περπατήστε 2 ορόφους κάτω». Όπως και στην πλοήγηση στροφή προς στροφή, οι οδηγίες είναι επικεντρωμένες στον χρήστη και τα αποτελέσματα έχουν δείξει ότι είναι μια εφικτή λύση για να βοηθηθούν οι χρήστες κατά την πλοήγησή τους μέσα σε ένα κτήριο, απουσία συνεχούς εντοπισμού της θέσης τους. Ωστόσο, η λύση τους δεν βοηθά στην περίπτωση που ο χρήστης αποκλίνει από την σωστή πορεία προς τον τελικό στόχο, αφού λόγω της έλλειψης εντοπισμού του, το σύστημα δεν είναι δυνατό να γνωρίζει την απόκλιση αυτή ώστε να τον οδηγήσει πίσω στην σωστή πορεία.

# 3

## Augmented Reality

### 3.1 Εισαγωγή

Η επαυξημένη πραγματικότητα (Augmented Reality ή AR) αναφέρεται σε μεθόδους και τεχνολογίες που χρησιμοποιούνται για την εισαγωγή εικονικών αντικειμένων σε μια ακολουθία πραγματικών εικόνων που λαμβάνονται μέσω της κάμερας μιας συσκευής με ρεαλιστικό τρόπο [71]. Σε αντίθεση με την εικονική πραγματικότητα (Virtual Reality ή VR), όπου ο χρήστης είναι πλήρως βυθισμένος σε έναν ψηφιακό κόσμο, η επαυξημένη πραγματικότητα στοχεύει να προσθέσει κάποιο εικονικό περιεχόμενο στον πραγματικό κόσμο και προσφέρει στον χρήστη δυνατότητες αλληλεπίδρασης σε πραγματικό χρόνο και γενικά βασίζεται σε ένα τρισδιάστατο (3D) περιβάλλον. Αυτός ο συνδυασμός είναι δυνατός με τη χρήση ηλεκτρονικών συσκευών που περιέχουν αισθητήρα κάμερας (γυαλιά, smartphone, tablet κ.λπ.). Τα smartphone και τα tablet είναι αυτή τη στιγμή τα πιο ευρέως χρησιμοποιούμενα. Οι virtual εικόνες προστίθενται στην πραγματικότητα που παρουσιάζεται στον χρήστη μέσα από την κάμερα του tablet ή του smartphone του [72].



Εικόνα 3.1 "Sword of Damocles" ήταν το ψευδώνυμο της πρώτης head-mounted οθόνης, που κατασκευάστηκε το 1968 [73]

Η πρώτη συσκευή AR αναπτύχθηκε από τον Sutherland το 1968. Δημιούργησε την πρώτη οθόνη που ήταν τοποθετημένη στο κεφάλι (head-mounted). Λόγω του βάρους της, χρειάστηκε να κρεμαστεί από το ταβάνι και ονομάστηκε “Sword of Damocles” (βλ. Εικόνα 3.1). Αργότερα, γύρω στο 1993, ο Feiner παρουσίασε το KARMA, ένα σύστημα που ενσωμάτωνε AR βασισμένο στη γνώση. Το AR ήταν χρήσιμο σε διάφορους τομείς από τα τέλη της δεκαετίας του '90 μέχρι και σήμερα. Το Εργαστήριο Ναυτικής Έρευνας (Naval Research Laboratory) των ΗΠΑ συμμετείχε σε ένα ερευνητικό πρόγραμμα διάρκειας μιας δεκαετίας που ονομάστηκε Σύστημα Επαυξημένης Πραγματικότητας στο Πεδίο της Μάχης (Battlefield Augmented Reality System) για να δημιουργήσει πρωτότυπα συστημάτων που οι στρατιώτες θα φορούσαν στο πεδίο της μάχης με σκοπό την επίγνωση της κατάστασης και την εκπαίδευση. Η πιο πρόσφατη εξέλιξη στο AR έγινε από τη Niantic με ένα παιχνίδι που ονομάζεται Pokemon-Go για iOS και Android, όπου οι παίκτες πρέπει να περιηγηθούν σε πραγματικό περιβάλλον για να βρουν στοχευμένα (targeted) αντικείμενα. Έγινε η πιο δημοφιλής εφαρμογή για smartphone και μπήκε στα παιχνίδια επαυξημένης πραγματικότητας [73].

## 3.2 Είδη Επαύξησης

Δεν υπάρχει μέχρι στιγμής σύστημα AR εσωτερικού χώρου που να συμπληρώνει τα μέρη που λείπουν από αντικείμενα, με αντίστοιχα εικονικά [74]. Η τοποθέτηση ενός marker σε αντικείμενα μπορεί να είναι ακατάλληλη για εκθέματα εσωτερικού χώρου, όπως μουσεία ή γκαλερί τέχνης. Τα εκθέματα αυτά μπορεί να είναι ευαίσθητα και, επομένως, δεν μπορούν να μετακινηθούν ή να τα αγγίξουν. Για την οπτικοποίηση αντικειμένων που λείπουν, έχουν προταθεί δύο απλές μέθοδοι, η μερική επαύξηση (Partial Augmentation) και η πλήρης επαύξηση (Complete Augmentation).

### Μερική Επαύξηση

Σε ένα σενάριο μερικής επαύξησης, ένα τρισδιάστατο σπασμένο αντικείμενο τοποθετείται σε μια σκηνή όπου οι διαστάσεις του είναι ακριβώς οι ίδιες με την πραγματικότητα. Ο αλγόριθμος καταχώρισης εικόνας βρίσκεται στην εργαλειοθήκη (toolkit) και βασίζεται στον marker και την απόσταση της κάμερας. Το πείραμα που πραγματοποιήθηκε από τον [75] και άλλους κατέληξε στο συμπέρασμα ότι ένα κενό 1–2,5 μέτρων αυξάνει το σφάλμα εγγραφής. Για να επιτευχθεί μερική επαύξηση, το σπασμένο αντικείμενο τοποθετείται κοντά στον marker (περίπου 20–30 cm) για να ελαχιστοποιηθεί το σφάλμα εγγραφής που παράγεται από το AR toolkit και να δώσει τα αναμενόμενα αποτελέσματα. Οι χρήστες μπορούν να έχουν ευελιξία τοποθετώντας τις κάμερες οπουδήποτε στην οπτική γωνία των marker καρτών, προσφέροντας τη δυνατότητα να μετακινούνται στον χώρο.

### Πλήρης Επαύξηση

Σε αντίθεση με τη μερική επαύξηση, η πλήρης επαύξηση δεν ασχολείται με την επεξεργασία ενός μικρού τμήματος πληροφοριών για την ολοκλήρωση του augmentation. Η πλήρης επαύξηση είναι όταν ολόκληρο το πραγματικό αντικείμενο αναπαρίσταται τρισδιάστατα σε ένα εκθεσιακό περιβάλλον. Για παράδειγμα, σε ένα μουσείο, που δεν είναι εφικτή η έκθεση κάποιων μνημείων λόγω της ευαισθησίας τους, μπορούν να χρησιμοποιηθούν πλήρεις τεχνικές επαύξησης για να τα δουν οι επισκέπτες. Σε προηγούμενη έρευνα, ο [76] προσπάθησε να αναπαραστήσει έναν τρισδιάστατο ελληνικό ναό όταν ο χρήστης δείχνει με την συσκευή του προς την τοποθεσία του ναού. Παρόλο που ο ναός δεν είναι σε αυτήν την τοποθεσία αυτή τη στιγμή, οι χρήστες μπορούν να τον δουν χρησιμοποιώντας την AR εφαρμογή για κινητά. Οι ίδιες τεχνικές μπορούν να εφαρμοστούν σε γκαλερί τέχνης και εκθέσεις. Στο προτεινόμενο σύστημα, τα εικονικά αντικείμενα επικαλύπτονται με τον marker των πραγματικών αντικειμένων και δίνουν την ψευδαίσθηση ότι το πραγματικό αντικείμενο δεν υπάρχει στην πραγματικότητα. Αυτό μπορεί να εφαρμοστεί σε περιπτώσεις όπου το αντικείμενο δεν μπορεί να έρθει σε φυσική επαφή με τον χρήστη. Το μειονέκτημα της πλήρους επαύξησης είναι το μεγάλο φορτίο κατά το rendering και κατά τη μοντελοποίηση του συστήματος. Όσο πιο ρεαλιστικό είναι το τρισδιάστατο επαυξημένο μοντέλο, τόσο μεγαλύτερο είναι το engagement των χρηστών. Οι συσκευές των χρηστών θα πρέπει να είναι αρκετά ικανές να χειρίζονται το βαρύ rendering που προκαλείται από την AR εφαρμογή και να μην κολλάνε λόγω έλλειψης ταχύτητας και ισχύος επεξεργασίας.

### 3.3 Παρακολούθηση, Βαθμονόμηση και Εγγραφή

Στα πλαίσια του AR, υπάρχουν κατά κύριο λόγο τρία στοιχεία που κρύβονται πίσω από τη λειτουργικότητά του – παρακολούθηση (tracking), βαθμονόμηση (calibration) και εγγραφή (registration) (βλ. Εικόνα 3.2). Κατά τη διαδικασία παρακολούθησης, το AR είναι υπεύθυνο για τη δυναμική εγγραφή. Τα αντικείμενα εγγράφονται στο AR και ευθυγραμμίζονται με το σύστημα συντεταγμένων. Ο στόχος του AR συστήματος είναι να καταγράψει τις εικονικές πληροφορίες ακριβώς όπως φαίνονται για το σύστημα. Η βαθμονόμηση διαβάζει την ακρίβεια του αισθητήρα και είναι υπεύθυνη για τη στατική εγγραφή για το σύστημα παρακολούθησης. [76]



Εικόνα 3.2 Τα τρία στοιχεία της λειτουργικότητας του AR

**Tracking** είναι ένας όρος που χρησιμοποιείται για να περιγράψει την ανίχνευση και τη μέτρηση των τιμών στο σύστημα AR. Για τη μετατροπή της τρισδιάστατης θέσης των εικονικών αντικειμένων στο σύστημα, απαιτούνται οι σχετικές θέσεις και οι προσανατολισμοί τους. Το AR λειτουργεί σε πραγματικό χρόνο, επομένως η μετάδοση οποιασδήποτε τιμής από το πραγματικό περιβάλλον θα πρέπει να γίνεται σε πραγματικό χρόνο και θα πρέπει να ενημερώνεται συνεχώς. Η παρακολούθηση δισδιάστατων αντικειμένων είναι συνηθισμένη στο σύστημα υπολογιστών, επομένως το tracking στο AR είναι διαφορετική τεχνολογία αφού στέλνει τις τρισδιάστατες συντεταγμένες οποιωνδήποτε αντικειμένων με βάση τη φύση τους. [76]

**Calibration** είναι η διαδικασία σύγκρισης των τιμών μεταξύ δύο συσκευών. Μία που είναι η συσκευή αναφοράς και η δεύτερη που πρέπει να βαθμονομηθεί. Στη συνέχεια, η συσκευή αναφοράς μπορεί να αντικατασταθεί με τις γνωστές συντεταγμένες από το πραγματικό περιβάλλον. Σε αντίθεση με το tracking που πρέπει να γίνεται συνεχώς, η βαθμονόμηση λειτουργεί μεταξύ διακριτών χρόνων. Η βαθμονόμηση πραγματοποιείται μόνο μία φορά σε μία συσκευή, εκτός εάν η συσκευή έχει αποκατασταθεί πλήρως. Η AR εφαρμογή λειτουργεί άψογα εάν η συσκευή έχει βαθμονομηθεί σωστά. [76]

Το **registration** αναφέρεται στην ευθυγράμμιση των συντεταγμένων μεταξύ εικονικών και πραγματικών αντικειμένων. Πιο συγκεκριμένα, η οθόνη AR θα πρέπει να δείχνει ή να ευθυγραμμίζει τέλεια το εικονικό αντικείμενο στον πραγματικό κόσμο. Αυτό απαιτεί tracking της κάμερας του χρήστη που παρέχει το φόντο. Η στατική εγγραφή συμβαίνει κατά τη διάρκεια της ακινητοποίησης της κάμερας για να υπολογιστεί η κοινή συντεταγμένη μεταξύ πραγματικών και εικονικών αντικειμένων, ενώ η δυναμική εγγραφή συμβαίνει όταν η κάμερα κινείται, κάτι που απαιτεί tracking. [76]

### 3.4 Πλεονεκτήματα και Μειονεκτήματα του AR για Φορητές Συσκευές



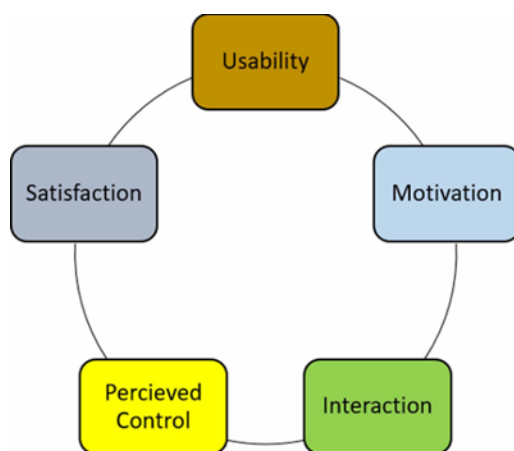
Εικόνα 3.3 Φορητές MAR συσκευές

Στην επαυξημένη πραγματικότητα για κινητά (Mobile Augmented Reality ή MAR) η συσκευή του χρήστη λειτουργεί ως hardware για όποια εφαρμογή χρησιμοποιεί λογισμικό AR [78]. Υπάρχει σύγχυση μεταξύ του φορητού AR και του MAR. Το φορητό AR επιτρέπει στον χρήστη να μετακινεί την τεχνολογία από το ένα μέρος στο άλλο. Είναι σαν τον ορισμό της ενέργειας: μπορεί να ανταλλάσσεται και να μεταφέρεται αλλά δεν μπορεί να δημιουργηθεί ή να καταστραφεί. Οι επιτραπέζιοι υπολογιστές και οι φορητοί υπολογιστές είναι μερικά παραδείγματα φορητών συσκευών AR, ενώ μια συσκευή MAR είναι συμπαγής, όπως ο φακός Microsoft Holo, το Magic Leap, το Oculus Rift, το Samsung Gear VR, το HTC Vive (βλ. Εικόνα 3.3).

- Πλεονεκτήματα του MAR: εμπειρία AR οπουδήποτε και οποτεδήποτε, κινητικότητα, εύκολη εγκατάσταση και γρήγορη συνδεσιμότητα.
- Μειονεκτήματα του MAR: Μπορεί να έχει πολλές δυσκολίες όσον αφορά τη συμβατότητα, δυσκολίες λόγω περιβαλλοντικών περιορισμών, περιορισμούς τεχνολογίας, σύνθετη κατανόηση από τους χρήστες και περίπλοκη αρχιτεκτονική συστήματος. Μία στις δέκα συσκευές MAR δεν επισκευάζεται.

### 3.5 Σχεδιαστικά Στοιχεία MAR Εφαρμογών

Υπάρχουν πολλά σημαντικά σχεδιαστικά στοιχεία τα οποία είναι απαραίτητα για μια AR εφαρμογή ώστε ο χρήστης να έχει μια ικανοποιητική αλληλεπίδραση και μεγαλύτερο engagement [79]. Τα πιο σημαντικά από αυτά είναι η αλληλεπίδραση, η χρησιμότητα, το κίνητρο, η ικανοποίηση, η εστιασμένη προσοχή και ο αντιληπτός έλεγχος. Τα άλλα στοιχεία που δεν τονίστηκαν λόγω μικρότερης αξιοπιστίας είναι η αισθητική, η αυτο-αποτελεσματικότητα, η απόλαυση και το ενδιαφέρον (βλ. Εικόνα 3.4).



Εικόνα 3.4 Σχεδιαστικά στοιχεία MAR εφαρμογών για μεγαλύτερο engagement του χρήστη



- **Αλληλεπίδραση:** Αυτό το στοιχείο παίζει ζωτικό ρόλο αφού η εφαρμογή θα πρέπει να είναι ικανή μέσω των γραφικών της να δεσμεύσει τον χρήστη και να του προσφέρει μια εύκολη και κατανοητή περιήγηση στις δυνατότητες που προσφέρει. Η φύση και ο σχεδιασμός της εφαρμογής εξαρτάται άμεσα από την αλληλεπίδραση που ο χρήστης θα έχει με αυτή.
- **Ευχρηστία:** Η ευχρηστία είναι το στοιχείο που σχετίζεται κυρίως με την ικανοποίηση των χρηστών από την εφαρμογή. Θα πρέπει να υπάρχει ευκολία στη χρήση της εφαρμογής για την προώθηση του engagement των χρηστών.
- **Κίνητρο:** Αυτό αναφέρεται στην ικανότητα και την προθυμία των χρηστών να συμμετάσχουν. Μόλις οι χρήστες ασχοληθούν με μια εφαρμογή, θα πρέπει να θέλουν να προσπαθήσουν “αριστεύσουν” και να μάθουν όσα περισσότερα γίνεται για αυτήν.
- **Ικανοποίηση:** Απολαυστικές στιγμές με την εφαρμογή προκύπτουν όταν οι χρήστες εκπληρώνουν τις προσδοκίες τους γι’ αυτή. Κάθε χρήστης αναμένει μια συγκεκριμένη συμπεριφορά από την εφαρμογή. Η εφαρμογή δεν πληροί τα κριτήρια ικανοποίησης εάν δεν ανταποκρίνεται σε αυτήν την προσδοκία.
- **Εστιασμένη προσοχή:** Η δέσμευση της προσοχής των χρηστών αποτελεί θεμέλιο για μια εφαρμογή ώστε να καταφέρει να αυξήσει το engagement των χρηστών της. Πολλές προηγούμενες μελέτες (π.χ. [80]) διαπίστωσαν ότι η εφαρμογή που τράβηξε την προσοχή των χρηστών ήταν σε θέση να επιτύχει engagement.
- **Αντιλαμβανόμενος έλεγχος:** Προκαλώντας την αίσθηση στους χρήστες ότι η εφαρμογή βρίσκεται υπό τον έλεγχό τους, δημιουργείται ένα περιβάλλον όπου οι χρήστες αισθάνονται επιρροή πάνω στην εφαρμογή και την ικανότητα να επιφέρουν τα επιθυμητά αποτελέσματα. Αυτό το στοιχείο περιλαμβάνει εν μέρει και άλλους παράγοντες, όπως η αισθητική, η προσοχή, η απόλαυση και το ενδιαφέρον.

### 3.6 Τεχνολογίες για την επίτευξη AR

Το AR λειτουργεί καλά σε εσωτερικούς ή κλειστούς χώρους. Η εγκατάσταση ασύρματων αισθητήρων, δεικτών (markers) τοποθεσίας, καμερών παρακολούθησης και GPS markers μπορεί να ευνοήσει τις συνθήκες λειτουργίας των AR εφαρμογών όσον αφορά τον εντοπισμό και την συνδεσιμότητα του δικτύου [81]. Η τεχνική του AR που βασίζεται σε markers, προορίζεται για οπτική παρακολούθηση (visual tracking), έχει αποδειχθεί κατάλληλη για εσωτερικούς χώρους.

Διακυμάνσεις και ασυνέπειες στις συνθήκες φωτισμού μπορεί να επηρεάσουν την οπτική παρακολούθηση. Η χρήση οπτικών markers είναι πάντα η καλύτερη επιλογή, αλλά αυτό εμποδίζει τους χρήστες να ασχοληθούν με την εφαρμογή και έμμεσα οδηγεί σε αποτυχία όσον αφορά την εμπειρία του χρήστη. Η απόκρυψη των markers από τους χρήστες για να αποφευχθεί η απόσπαση της προσοχής τους κατά την προβολή οπτικού περιεχομένου μπορεί να είναι αποτελεσματική [82]. Μελέτες έχουν εξετάσει τις νέες τεχνολογίες που χρησιμοποιούν την όραση υπολογιστή (computer vision), τη βαθιά μάθηση (deep learning) και ένα μείγμα και των δύο για την εξάλειψη της χρήσης των markers. Το Google Tango, που τώρα έχει επίσης αναβαθμιστεί σε ARCore, είναι η πιο πρόσφατη τεχνολογία για τη μείωση της χρήσης ετικετών.

# 4 Unity

## 4.1 Εισαγωγή

Η Unity είναι μια cross-platform μηχανή κατασκευής παιχνιδιών (game engine) γραμμένη σε C#, που αναπτύχθηκε και σχεδιάστηκε από την Unity Technologies. Πολλά δημοφιλή παιχνίδια έχουν αναπτυχθεί στην Unity και το Unreal από το 2008 και πολλά παιχνίδια 2D και 3D συνεχίζουν να αναπτύσσονται. Παρέχει υποστήριξη για πολλά κρίσιμα χαρακτηριστικά, όπως live video streaming και απεικόνιση 2D/3D, προηγμένη επεξεργασία εικόνας και βίντεο, μοντελοποίηση και οπτικοποίηση δυναμικών αντικειμένων, επικοινωνία πελάτη-διακομιστή, βάση δεδομένων στο cloud. Τα τελευταία χρόνια, η Unity εργάστηκε σκληρά για να ενσωματώσει το ARCore της Google και το ARKit της Apple στο περιβάλλον και ανακοίνωσε αργότερα το AR Foundation. Λίγο μετά το AR Foundation, το Vuforia Engine αναπτύχθηκε και συναγωνίστηκε με τα υπόλοιπα για να γίνει η πιο ευρέως χρησιμοποιούμενη πλατφόρμα. Σε συνδυασμό με την Unity, το Vuforia 3D επιτρέπει στους χρήστες να προσθέτουν προηγμένη λειτουργικότητα υπολογιστικής όρασης (computer vision) σε εφαρμογές Android και iOS. Το Vuforia μπορεί να ενσωματωθεί στην Unity απλά, μέσω της εγκατάστασης ενός τερματικού SDK ή NPM. Το Vuforia υποστηρίζει C# API για Unity και άλλες πλατφόρμες όπως το Android, το iOS και το UWP.

## 4.2 ARkit και ARCore

### ARKit

Το ARKit είναι το κιτ ανάπτυξης εφαρμογών επαυξημένης πραγματικότητας της Apple που κυκλοφόρησε τον Ιούνιο του 2017. Αυτό το κιτ επιτρέπει τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας για iPhone και iPad με iOS 11 ή νεότερη έκδοση. Το Apple ARKit SDK είναι διαθέσιμο για λήψη για όλους τους προγραμματιστές iOS που διαθέτουν λογαριασμό προγραμματιστή Apple (Apple Developer Account). Κύρια χαρακτηριστικά [83]:

**Tracking:** Το Apple ARKit σας επιτρέπει να παρακολουθείτε την τοποθεσία του smartphone στον πραγματικό κόσμο, σε πραγματικό χρόνο. Για να γίνει αυτό, το SDK συνδυάζει το VIO (Visual Inertial Odometer) με δεδομένα από την κάμερα και τον αισθητήρα κίνησης της συσκευής.

**Κατανόηση του περιβάλλοντος:** Με το ARKit, το iPhone και το iPad μπορούν να αναγνωρίσουν πραγματικές, κάθετες και οριζόντιες, επίπεδες επιφάνειες σε πραγματικό χρόνο για ρεαλιστική ενσωμάτωση εικονικών στοιχείων.

**Στατική 2D αναγνώριση εικόνας:** αυτή η δυνατότητα επιτρέπει την αποθήκευση εικόνων στην εφαρμογή και την ανίχνευση τους στο πραγματικό περιβάλλον, την εμφάνιση και την υπέρθεση εικονικών πληροφοριών σε αυτές τις εικόνες.

**Εκτίμηση φωτεινότητας:** Το ARKit μπορεί επίσης να χρησιμοποιήσει τον αισθητήρα φωτεινότητας για να προσδιορίσει την ποσότητα φωτός στο περιβάλλον ώστε να ρυθμιστεί ο βαθμός φωτεινότητας στο εικονικό αντικείμενο. Το ARKit μπορεί επίσης να δημιουργήσει αντανάκλασεις πραγματικών αντικειμένων σε εικονικό μέταλλο.

## ARCore

Το 2014, η Google κυκλοφόρησε το πρώτο της Tango SDK αφιερωμένο στην Επαυξημένη Πραγματικότητα, απαιτούσε ένα smartphone εξοπλισμένο με κάμερα ανίχνευσης βάθους (depth sensing camera) για να λειτουργήσει. Η ιδιαιτερότητα του Tango ήταν να επιτρέπει στους χρήστες του να χαρτογραφούν περιβάλλοντα σε 2D και 3D. Το Lenovo phab 2 pro ήταν το πρώτο smartphone που ενσωμάτωσε την τεχνολογία Tango, επιτρέποντας στους χρήστες του να επωφεληθούν από λειτουργίες που σχετίζονται με την επαυξημένη πραγματικότητα. Αυτό το έργο δεν κράτησε πολύ γιατί περιοριζόταν σε συσκευές με βασικούς αισθητήρες που κανένα smartphone δεν είχε από προεπιλογή. Η Google συνεχίζει στον τομέα της επαυξημένης πραγματικότητας με το λανσάρισμα του ARCore τον Μάρτιο του 2018. Σε αντίθεση με το Tango, το ARCore μπορεί να χρησιμοποιηθεί σε πολλά smartphone με Android Nougat (7.0) και συσκευές IOS που υποστηρίζουν το ARKit. Το ARCore βασίζεται στο Tango με λιγότερες δυνατότητες που δεν εξαρτώνται από εξειδικευμένες κάμερες, επιτρέποντας σε μεγαλύτερο αριθμό συσκευών να είναι συμβατές με την τεχνολογία του. Βρίσκουμε εδώ τις ίδιες τεχνολογίες όπως στο ARKit, δηλαδή tracking της κίνησης, κατανόηση χώρου και εκτίμηση φωτεινότητας. Κύρια χαρακτηριστικά [84]:

**Tracking:** Το ARCore χρησιμοποιεί την κάμερα του τηλεφώνου για να παρατηρήσει τα χαρακτηριστικά σημεία της σκηνής και τα δεδομένα από τον αισθητήρα της μονάδας μέτρησης αδράνειας (Inertial Measurement Unit ή IMU), καθορίζει τη θέση και τον προσανατολισμό της συσκευής καθώς κινείται. Αυτό επιτρέπει στα εικονικά αντικείμενα να τοποθετηθούν σωστά.

**Κατανόηση του περιβάλλοντος:** Το ARCore χρησιμοποιεί τους διαφορετικούς αισθητήρες στο smartphone για να ανιχνεύει οριζόντιες επιφάνειες χρησιμοποιώντας τα ίδια σημεία χαρακτηριστικών που χρησιμοποιεί για την παρακολούθηση κίνησης.

**Εκτίμηση φωτεινότητας:** Η εκτίμηση φωτός επιτρέπει στο smartphone να εκτιμήσει το φως του περιβάλλοντος προκειμένου να ενσωματώσει τέλεια τα τρισδιάστατα στοιχεία στον πραγματικό κόσμο.

Το ARKit και το ARCore μοιράζονται τις ακόλουθες δυνατότητες για την εισαγωγή ενός εικονικού αντικειμένου στον πραγματικό κόσμο:

- Παρακολούθηση κίνησης (Motion Tracking) : παρακολούθηση της θέσης της συσκευής στον πραγματικό κόσμο
- Ανίχνευση επιπέδων (Planes detection) : ανίχνευση οριζόντιων και κάθετων επιφανειών
- Εκτίμηση φωτός: για ανίχνευση πληροφοριών σχετικά με το φωτισμό
- Image Tracking: ανίχνευση και παρακολούθηση (tracking) μιας εικόνας σε μια σκηνή
- Ανίχνευση προσώπου (Face Tracking): ανίχνευση και παρακολούθηση προσώπου σε μια σκηνή

#### **Περιβάλλον ανάπτυξης:**

Η επαυξημένη πραγματικότητα βασίζεται σε μεθόδους επεξεργασίας εικόνας και 3D rendering. Για την ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας, είναι απαραίτητο λογισμικό που επιτρέπει το 3D rendering.

### **4.3 ARFoundation και Vuforia**

#### **ARFoundation**

Η Unity εισήγαγε το ARFoundation [85] που συνδυάζει τις λειτουργίες του ARKit και του ARCore σε μια ενιαία ενότητα, η οποία επιτρέπει στον προγραμματιστή να στοχεύει πλατφόρμες IOS και Android με τον ίδιο source code. Αυτό επιτρέπει σε μια εφαρμογή να αναπτυχθεί μία φορά και να εγκατασταθεί και στις δύο συσκευές χωρίς καμία τροποποίηση.

Το AR Foundation είναι ένα σύνολο MonoBehaviours και APIs, για συσκευές που υποστηρίζουν τα ακόλουθα:

- Tracking συσκευής: παρακολούθηση της θέσης και του προσανατολισμού της συσκευής στο φυσικό χώρο.
- Ανίχνευση επιπέδου: ανίχνευση οριζόντιων και κάθετων επιφανειών.
- Point clouds, γνωστά και ως feature points.
- Anchor: μια αυθαίρετη θέση και προσανατολισμός που παρακολουθεί η συσκευή.
- Εκτίμηση φωτός: εκτιμήσεις για τη μέση θερμοκρασία χρώματος και φωτεινότητα στο φυσικό χώρο.
- Environment probe: ένα μέσο για τη δημιουργία ενός χάρτη κύβου για την αναπαράσταση μιας συγκεκριμένης περιοχής του φυσικού περιβάλλοντος.
- Παρακολούθηση προσώπου: ανίχνευση και παρακολούθηση ανθρώπινων προσώπων.

- 2D Image Tracking: ανίχνευση και παρακολούθηση εικόνων 2D.
- 3D Object Tracking: ανίχνευση τρισδιάστατων αντικειμένων.
- Πλέγμα (Mesh) : δημιουργία τριγωνικών πλεγμάτων που αντιστοιχούν στον φυσικό χώρο.
- Body Tracking: 2D και 3D αναπαραστάσεις ανθρώπων που αναγνωρίζονται στο φυσικό χώρο.
- Συνεργατικοί συμμετέχοντες: παρακολούθηση της θέσης και του προσανατολισμού άλλων συσκευών σε μια κοινή εμπειρία AR.
- Ανθρώπινη τμηματοποίηση: καθορίζει την υφή του στένσιλ και τον χάρτη βάθους των ανθρώπων που εντοπίζονται στην εικόνα της κάμερας.
- Raycast: ερωτά το φυσικό περιβάλλον για ανιχνευμένα επίπεδα και point clouds.
- Pass-through video: βελτιστοποιημένο rendering της εικόνας της κάμερας του κινητού τηλεφώνου στην οθόνη αφής ως φόντο για περιεχόμενο AR.
- Διαχείριση περιόδου σύνδεσης: χειρισμός της διαμόρφωσης σε επίπεδο πλατφόρμας αυτόματα όταν ενεργοποιούνται ή απενεργοποιούνται οι λειτουργίες AR.
- Απόφραξη: επιτρέπει την απόφραξη εικονικού περιεχομένου από το ανιχνευμένο περιβαλλοντικό βάθος (περιβαλλοντική απόφραξη) ή από ανιχνευμένο ανθρώπινο βάθος (ανθρώπινη απόφραξη).

## Vuforia

Το Vuforia είναι ένα kit ανάπτυξης λογισμικού επαυξημένης πραγματικότητας για κινητές συσκευές για τη δημιουργία εφαρμογών AR. Οι προγραμματιστές μπορούν εύκολα να προσθέσουν προηγμένες λειτουργίες όρασης υπολογιστή σε οποιαδήποτε εφαρμογή, επιτρέποντάς της να αναγνωρίζει εικόνες και αντικείμενα και να αλληλεπιδρά με χώρους στον πραγματικό κόσμο. Αυτό το SDK προσφέρεται από την Qualcomm, η οποία έχει αυξήσει το μερίδιο της στη βιομηχανία επαυξημένης πραγματικότητας. [86]

Το Vuforia που το 2006 ήταν στην κατοχή της Qualcomm, ανήκει πλέον στην PTC από τον Νοέμβριο του 2015. Χρησιμοποιεί την όραση της κάμερας για να αναγνωρίζει και να κάνει track εικόνες και τρισδιάστατα αντικείμενα σε πραγματικό χρόνο. Σύμφωνα με δοκιμές, έχει αποδειχθεί ότι είναι ο ταχύτερος tracking αλγόριθμος και με μεγαλύτερη σταθερότητα εικόνας. Η διεπαφή χρήστη (user interface) και οι λειτουργίες είναι εύχρηστες, γεγονός που κάνει ακόμα και τους αρχάριους να αισθάνονται άνετα με αυτό το νέο εργαλείο. Το πιο σημαντικό χαρακτηριστικό για τους ερασιτέχνες προγραμματιστές είναι το κόστος των εργαλείων ανάπτυξης. Τα open source projects έχουν αυξήσει τον αριθμό των συμμετεχόντων στην ανάπτυξη εφαρμογών. Επίσης, το Vuforia προσφέρεται δωρεάν για χρήση με περιορισμένες δυνατότητες [86]. Σήμερα, το μεγαλύτερο βοήθημα στην ανάπτυξη εφαρμογών είναι το Διαδίκτυο και πιο συγκεκριμένα τα φόρουμ και οι κοινότητες. Υπάρχουν φόρουμ σε διαφορετικούς ιστότοπους και στον επίσημο ιστότοπο της Vuforia, όπου οι προγραμματιστές μπορούν να συζητήσουν και να μοιραστούν προβλήματα που αντιμετωπίζουν κατά την ανάπτυξη.

Το Vuforia προσφέρει εύκολη αλληλεπίδραση μεταξύ του τελικού χρήστη και του πραγματικού περιβάλλοντος. Χρησιμοποιεί τα στοιχεία της ARCamera για να παρέχει ζωντανή ανατροφοδότηση βίντεο από το πραγματικό περιβάλλον, μαζί με ανιχνεύσιμες συντεταγμένες από τη στοχευμένη εικόνα και αντικείμενο. Η ARCamera παρακολουθεί επίσης τον προσανατολισμό των συσκευών και των αντικειμένων για να πάρει την ακριβή τιμή, γεγονός που απλοποιεί την εμπειρία της επαυξημένης πραγματικότητας. Η Vuforia παρέχει API με γλώσσες προγραμματισμού Java, C++ και .Net μέσω της επέκτασης του Unity 3D engine.

Το Vuforia περιλαμβάνει πολλά στοιχεία που καθιστούν την ανάπτυξη εφαρμογών εύκολη και ρεαλιστική. Τα ακόλουθα στοιχεία είναι οι βασικές δυνατότητες που είναι διαθέσιμες στο SDK . Υπάρχουν επίσης πολλά άλλα στοιχεία διαθέσιμα ως plugin που βρίσκονται στο κατάστημα του Vuforia.

ARCamera, Target Image, Video Background Renderer and Frame Marker, Virtual Buttons.

Το Vuforia SDK της Unity είναι η πιο ευρέως χρησιμοποιούμενη πλατφόρμα για ανάπτυξη AR, με υποστήριξη για κορυφαία τηλέφωνα, tablet και γυαλιά. Οι προγραμματιστές μπορούν εύκολα να προσθέσουν προηγμένες λειτουργίες computer vision σε εφαρμογές Android, iOS και UWP, για να δημιουργήσουν εμπειρίες AR που αλληλεπιδρούν ρεαλιστικά με αντικείμενα και με το περιβάλλον [87]. Οι εφαρμογές Vuforia Engine για επαυξημένη πραγματικότητα ενσωματώνουν εξελιγμένες μεθόδους και αλγόριθμους για την ανίχνευση και την αξιολόγηση των Points Of Interest [88], των οπτικών ροών [89] και των markers [90]. Αυτά, μαζί με άλλα τυπικά στοιχεία της πλατφόρμας Unity, όπως το World Anchor και η ικανότητα scripting, παρουσιάζουν έναν πυρήνα λειτουργιών για την ανάπτυξη ολοκληρωμένων προϊόντων AR. Η τεχνολογία του Vuforia SDK Cloud δίνει τη δυνατότητα σε πολλαπλές smart συσκευές να παρατηρούν το ίδιο αντικείμενο από διαφορετικές γωνίες ταυτόχρονα, κάτι που επιτρέπει σε πολλούς συμμετέχοντες να εμπλακούν. Το Vuforia Engine Library περιέχει ένα πλούσιο API documentation, το οποίο περιλαμβάνει C# APIs για Unity, C++ APIs για iOS και UWP και Java για Android.

## Σημαντικές Παρατηρήσεις

Πρέπει να σημειωθεί ότι το Vuforia είναι μια πολύ πιο σύνθετη λύση από το AR Foundation:

- Προσφέρει περισσότερες επιλογές όσον αφορά τις δυνατότητες AR, οι οποίες λειτουργούν αμέσως με λίγες μόνο αλλαγές.
- Βασίζεται στη δική του πολύπλοκη τεχνολογία που χρησιμοποιείται από πολύ μεγάλες τεχνολογικές εταιρείες.
- Μπορεί να δοκιμαστεί απευθείας μέσα στην Unity χρησιμοποιώντας μία web κάμερα, πράγμα που μειώνει σημαντικά τον χρόνο ανάπτυξης της εφαρμογής.
- Λειτουργεί σε όλες τις πλατφόρμες.
- Ως μέρος των έτοιμων λύσεων της, εκμεταλλεύεται τα πλεονεκτήματα της τεχνολογίας του Lidar scanner που είναι διαθέσιμη σε κινητές συσκευές τελευταίας τεχνολογίας (για παράδειγμα, τα πιο πρόσφατα iPads Pro).

Το μειονέκτημα του Vuforia είναι η τιμή. Κάποιος μπορεί να το δοκιμάσει δωρεάν και να δημιουργήσει την εφαρμογή του σε μια συσκευή, αλλά δεν θα μπορούσε να την δημοσιεύσει στο App Store ή στο Google Play χωρίς έγκυρη εμπορική άδεια. Αυτό, δεν επηρεάζει την ανάπτυξη της δικής μας εφαρμογής αφού ο σκοπός της συγκεκριμένης διπλωματικής είναι εκπαιδευτικός.

Το Vuforia προσφέρει μια λύση προσαρμοσμένη ειδικά για πλοήγηση σε εσωτερικούς χώρους, η οποία βασίζεται στην τεχνολογία σάρωσης Lidar που αναφέρθηκε προηγουμένως. Λειτουργεί ως εξής:

1. Σάρωση ενός δεδομένου περιβάλλοντος (π.χ. πανεπιστημιούπολη, εργοστάσιο, εμπορικό κέντρο) στο σύνολό του με τον Lidar scanner (χρησιμοποιώντας το τελευταίο iPad Pro, για παράδειγμα).
2. Εισαγωγή της σάρωσης στο project της Unity χρησιμοποιώντας τα εργαλεία του Vuforia.
3. Προσαρμογή των ρυθμίσεων του project για την επίτευξη της επιθυμητής λειτουργίας.

Να σημειωθεί ότι η πλοήγηση σε εσωτερικούς χώρους με το Vuforia κάνει χρήση των Navmesh και NavAgents της Unity. Ωστόσο, αυτή η προσέγγιση απαιτεί πρόσβαση στην πιο πρόσφατη τεχνολογία (το πιο πρόσφατο iPad Pro ή έναν εμπορικό 3D scanner), την οποία δεν διαθέτουμε. Η λύση για το συγκεκριμένο πρόβλημα περιλαμβάνει το image tracking.

Το image tracking λειτουργεί ως εξής: έχουμε μια εικόνα που μπορεί να αναγνωριστεί από την κάμερα, εκτυπώνουμε την εικόνα σε ένα χαρτί και τοποθετούμε αυτό το χαρτί οπουδήποτε στο περιβάλλον μας. Στρέφουμε την κάμερα στην εικόνα, η εφαρμογή μας αναγνωρίζει την εικόνα, υπολογίζει τη θέση και τον προσανατολισμό της εικόνας σε σχέση με τη δική της θέση και προσανατολισμό και τοποθετεί ένα τρισδιάστατο αντικείμενο που έχουμε καθορίσει εκ των προτέρων στην οθόνη. Μόλις το αντικείμενο τοποθετηθεί στην οθόνη, η εφαρμογή ενημερώνει συνεχώς τη θέση και τον προσανατολισμό αυτού του αντικειμένου, ενώ μπορούμε να περπατήσουμε και να επιθεωρήσουμε το αντικείμενο από κάθε πλευρά. Είναι βασικά σαν ένας κωδικός QR με επιπλέον βήματα. Το image tracking είναι μια δυνατότητα που προσφέρεται από οποιοδήποτε SDK: ARCore, ARKit, AR Foundation, Vuforia, κ.λπ. Επιλέξαμε το Vuforia επειδή ήταν εύκολο να δοκιμαστεί χρησιμοποιώντας μια web κάμερα.

Στο 100% των περιπτώσεων το image tracking χρησιμοποιείται για την εμφάνιση μικρών ή μεσαίων τρισδιάστατων αντικειμένων στην οθόνη. Συνήθως ένα αντικείμενο θα χωρούσε ολόκληρο στην οθόνη και θα μπορούσαμε να περπατήσουμε γύρω του. Ωστόσο, τι θα συνέβαινε εάν χρησιμοποιούσαμε το image tracking για έναν σκοπό για τον οποίο δεν προορίζεται; Δηλαδή, τι θα γινόταν αν φτιάχναμε ένα τρισδιάστατο αντικείμενο που θα ήταν ακριβώς 1:1 αντίγραφο ενός πραγματικού περιβάλλοντος (π.χ. σχολείο) και αλλάζαμε το μέγεθος αυτού του τρισδιάστατου αντικειμένου με τέτοιο τρόπο ώστε να αντιπροσωπεύει τον πραγματικό κόσμο, στον εικονικό; Ιδού η απάντηση:

- Αποδεικνύεται ότι μπορούμε να κάνουμε scale ένα τρισδιάστατο αντικείμενο με τέτοιο τρόπο ώστε να ευθυγραμμίζεται με το πραγματικό περιβάλλον.

- Η εφαρμογή δεν έχει κανένα πρόβλημα να παρακολουθεί αυτό το εικονικό περιβάλλον και δεν το "χάνει" από τα μάτια της ακόμα κι αν απομακρυνθούμε 50-100 μέτρα από το αντικείμενο.

Το Vuforia διαθέτει αναγνώριση κειμένου (text recognition), αλλά αυτή η δυνατότητα έχει καταργηθεί. Ωστόσο, η αναγνώριση εικόνας λειτουργεί εκπληκτικά καλά κάνοντας διάκριση μεταξύ παρόμοιων εικόνων. Ίσως, αυτός είναι ο λόγος για την κατάργηση της δυνατότητας αναγνώρισης κειμένου.

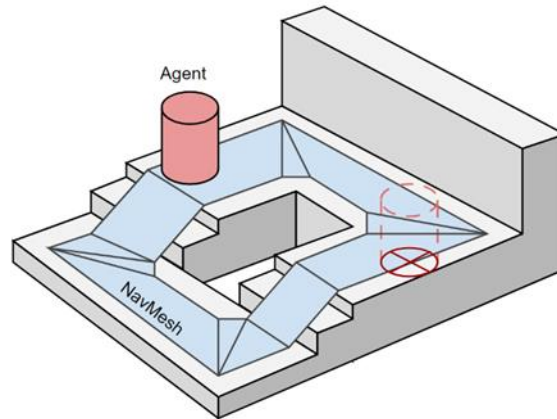
## 4.4 NavMesh

Ο A\* είναι ένας σημαντικός αλγόριθμος όσον αφορά την εύρεση μονοπατιών, αλλά η Unity μας παρέχει επίσης μερικές δυνατότητες, όπως η αυτόματη δημιουργία πλέγματος πλοήγησης (Navigation Mesh) και ο NavMesh agent. Αυτές οι δυνατότητες κάνουν την εύρεση μονοπατιών στα παιχνίδια πολύ εύκολη. Είτε επιλέξουμε να εφαρμόσουμε τη δική μας λύση A\* είτε απλώς να χρησιμοποιήσουμε την ενσωματωμένη λειτουργία Nav Mesh της Unity εξαρτάται από τις ανάγκες του πρότζεκτ μας.

Η χρήση ενός απλού πλέγματος στον A\* απαιτεί αρκετούς υπολογισμούς ώστε να ληφθεί μια διαδρομή που να είναι η συντομότερη προς τον στόχο αλλά, ταυτόχρονα, να αποφεύγονται και τα εμπόδια. Έτσι, για να είναι φθηνότερο και ευκολότερο για τους AI characters να βρίσκουν μια διαδρομή, οι άνθρωποι σκέφτηκαν να χρησιμοποιήσουν waypoints ως οδηγό για τη μετακίνηση των AI characters από το σημείο εκκίνησης στο σημείο στόχο.

Τα περισσότερα παιχνίδια χρησιμοποιούν waypoints για την εύρεση μονοπατιών επειδή είναι απλά και αρκετά αποτελεσματικά όσον αφορά τη χρήση λιγότερων υπολογιστικών πόρων. Ωστόσο, έχουν κάποια προβλήματα. Τι γίνεται αν θέλουμε να ενημερώσουμε τα εμπόδια στον χάρτη μας; Θα πρέπει να τοποθετήσουμε καινούρια σημεία για τον ενημερωμένο χάρτη. Το να πρέπει να αλλάζουμε με μη αυτόματο τρόπο τα σημεία κάθε φορά που αλλάζει η διάταξη των επιπέδων μας μπορεί να είναι δύσκολο και πολύ χρονοβόρο. Επιπλέον, ακολουθώντας κάθε κόμβο προς τον στόχο, σημαίνει ότι ο AI character κινείται σε μια σειρά από ευθείες γραμμές από τον έναν κόμβο στον επόμενο. Έτσι, είναι πολύ πιθανό ο AI character να συγκρουστεί με κάποιον τοίχο όπου η διαδρομή είναι κοντά σε αυτόν. Εάν συμβεί αυτό, θα συνεχίσει να προσπαθεί να περάσει μέσα από τον τοίχο για να φτάσει στον επόμενο στόχο, αλλά δεν θα τα καταφέρει και θα κολλήσει εκεί. Παρόλο που μπορούμε να εξομαλύνουμε το μονοπάτι μετατρέποντάς το σε spline και κάνοντας κάποιες προσαρμογές για να αποφύγουμε τέτοια εμπόδια, το πρόβλημα είναι ότι τα waypoints δεν μας δίνουν καμία πληροφορία για το περιβάλλον, εκτός από το spline που συνδέει δύο κόμβους. Έτσι, για να μπορέσουν οι AI οντότητές μας να διασχίσουν αποτελεσματικά ολόκληρο το επίπεδο, θα χρειαστούμε έναν τεράστιο αριθμό από waypoints, τα οποία θα διαχειρίζονται με δυσκολία. [91]





Εικόνα 4.1 NavMesh in Unity

Λύση για το συγκεκριμένο πρόβλημα αποτελεί το NavMesh. Το NavMesh είναι μια άλλη δομή γράφου που μπορεί να χρησιμοποιηθεί για να αναπαραστήσει τον κόσμο μας. Το σύστημα πλοήγησης χρειάζεται τα δικά του δεδομένα για να αναπαραστήσει τις περιοχές που μπορούν να “περπατηθούν” (walkable areas) σε μια σκηνή παιχνιδιού. Οι walkable areas καθορίζουν τις θέσεις στη σκηνή όπου ο πράκτορας μπορεί να σταθεί και να κινηθεί. Στην Unity οι πράκτορες περιγράφονται ως κύλινδροι. Η βατή περιοχή κατασκευάζεται αυτόματα από τη γεωμετρία στη σκηνή δοκιμάζοντας τις τοποθεσίες όπου μπορεί να σταθεί ο πράκτορας. Στη συνέχεια, οι τοποθεσίες συνδέονται με μια επιφάνεια που βρίσκεται πάνω από τη γεωμετρία της σκηνής. Αυτή η επιφάνεια ονομάζεται πλέγμα πλοήγησης (Navigation Mesh ή NavMesh) (βλ. Εικόνα 4.1). Το NavMesh αποθηκεύει αυτήν την επιφάνεια ως κυρτά πολύγωνα (convex polygons). Τα κυρτά πολύγωνα είναι μια χρήσιμη αναπαράσταση, αφού γνωρίζουμε ότι δεν υπάρχουν εμπόδια μεταξύ δύο σημείων μέσα σε ένα πολύγωνο. Εκτός από τα όρια του πολυγώνου, αποθηκεύουμε πληροφορίες σχετικά με το ποια πολύγωνα είναι γειτονικά μεταξύ τους. Αυτό μας επιτρέπει να βγάλουμε συμπεράσματα για ολόκληρη την βατή περιοχή [92]. Μπορούμε έτσι να προσαρμόσουμε την πορεία μας με ασφάλεια επειδή γνωρίζουμε την ασφαλή περιοχή στην οποία μπορούν να ταξιδέψουν οι πράκτορές μας. Ένα άλλο πλεονέκτημα της χρήσης ενός Navigation Mesh είναι ότι μπορούμε να χρησιμοποιήσουμε το ίδιο πλέγμα για διαφορετικούς τύπους οντοτήτων AI. Διαφορετικές οντότητες AI μπορεί να έχουν διαφορετικές ιδιότητες όπως μέγεθος, ταχύτητα και ικανότητες κίνησης. Η δημιουργία ενός Navigation Mesh μέσω προγραμματισμού μπορεί να είναι μια κάπως περίπλοκη διαδικασία. Ευτυχώς, η Unity 3.5 εισήγαγε μια ενσωματωμένη γεννήτρια Navigation Mesh ως pro-only feature, αλλά τώρα προσφέρεται δωρεάν προς όλους από την έκδοση Unity 5 και μετά. Η υλοποίηση αυτή της Unity παρέχει πολλές πρόσθετες λειτουργίες. Όχι μόνο την δημιουργία του ίδιου του NavMesh, αλλά και την σύγκρουση (agent collision) και την εύρεση διαδρομής (pathfinding) στον παραγόμενο γράφο (μέσω του A\*). [91]

## **II ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ**

---

# 5

## Μεθοδολογία

### 5.1 Συλλογιστική Πορεία και Προτεινόμενο Σύστημα

Αυτή η ενότητα παρουσιάζει την προτεινόμενη μεθοδολογία για την ανάπτυξη συστημάτων πλοήγησης εσωτερικού χώρου για κινητές συσκευές. Ο πρωταρχικός στόχος της εφαρμογής είναι να εξαλείψει όλες τις εξαρτήσεις όπως σημεία πρόσβασης Wi-Fi, bluetooth beacons, Internet και άλλες συσκευές που απαιτούν πρόσθετο υλικό, στο οποίο δεν έχουμε πρόσβαση, για την υλοποίηση της πλοήγησης σε εσωτερικό χώρο.

Πήραμε πραγματικές κατόψεις και φτιάξαμε μια πραγματική εκδοχή του κτηρίου, χρησιμοποιώντας τυπικούς κύβους. Ένας κύβος στο Unity είναι 1x1x1 μέτρα, γεγονός που διευκόλυνε τη σωστή κλίμακα του περιβάλλοντος. Μετρήσαμε έναν από τους τοίχους του κτηρίου και χρησιμοποιήσαμε τον τυπικό κύβο για να αναπαραστήσουμε σωστά το περιβάλλον, δηλαδή ο τοίχος ήταν 308 cm, ρυθμίσαμε την κλίμακα του κύβου στο 3,08 και φροντίσαμε το μέγεθός του να ταιριάζει με το υπόλοιπο περιβάλλον.

Όσον αφορά τις μεταβάσεις μεταξύ των ορόφων, μετρήσαμε ένα μεμονωμένο σκαλοπάτι και, προσθέτοντας τις διαστάσεις τους, μπορούσαμε να δημιουργήσουμε μια πραγματική τρισδιάστατη αναπαράσταση αυτών.

Η πλοήγηση γίνεται με τις προεπιλεγμένες ρυθμίσεις, τίποτα το ιδιαίτερο εκεί.

Ο δείκτης γεωγραφικής θέσης που δημιουργήθηκε στο Blender τοποθετείται στο τέλος της διαδρομής που δημιουργείται από το NavMesh.

Τέλος, υπάρχει ένα τρισδιάστατο βέλος που δημιουργήθηκε με χρήση του Blender, το οποίο είναι απλώς γονικό στην κάμερα, έτσι ώστε να παραμένει πάντα μπροστά του και σε περιβάλλον AR. Περιστρέφεται προς τα σημεία διαδρομής που δείχνουν την κατεύθυνση πλοήγησης (χρησιμοποιώντας προσαρμοσμένη δέσμη ενεργειών).

Η εφαρμογή μας κάνει χρήση του AR Core για την εμφάνιση της διαδρομής πλοήγησης στο κινητό τηλέφωνο του χρήστη με τη βοήθεια των Anchors που δημιουργούνται με βάση τη διαδρομή πλοήγησης η οποία διαμορφώνεται με τη βοήθεια του navmesh. Εδώ το navmesh δημιουργείται με τη βοήθεια μιας κάτοψης της περιοχής προς πλοήγηση. Για να παρακολουθήσουμε τη θέση του χρήστη, χρησιμοποιούμε μερικούς απλούς υπολογισμούς της απόστασης και της γωνίας από την προηγούμενη θέση στην τρέχουσα θέση. Για την εκτέλεση εφαρμογών AR Core, η συσκευή πρέπει να είναι πιστοποιημένη ως συμβατή με AR Core. Η πιστοποίηση παρέχεται από την Google, όπου αποφασίζει ποια συσκευή θα πιστοποιήσει με βάση την ποιότητα της κάμερας,

τους αισθητήρες κίνησης και την αρχιτεκτονική σχεδίασης που αναμένεται να έχει ένα πρότυπο. Επίσης, η συσκευή πρέπει να διαθέτει αρκετά ισχυρή CPU που να ενσωματώνεται με τη σχεδίαση υλικού για να διασφαλίζει καλή απόδοση και αποτελεσματικούς υπολογισμούς σε πραγματικό χρόνο.

Η διεπαφή χρήστη (User Interface ή UI) είναι ο κύριος τρόπος μέσω του οποίου ο χρήστης αλληλεπιδρά με την εφαρμογή Android όπου εκτελείται ολόκληρο το backend. Στο παρασκήνιο, τόσο ο πυρήνας επαυξημένης πραγματικότητας (ARCore) όσο και το Android SDK είναι συνδεδεμένα στην Unity Engine. Η καθοδήγηση πλοήγησης παρέχεται από επαυξημένα αντικείμενα τα οποία είναι ορατά μέσω της οθόνης της κινητής συσκευής, διασφαλίζοντας ότι ο χρήστης μπορεί να ακολουθήσει τις οδηγίες, ενώ εξακολουθεί να δίνει προσοχή στο βήμα του.

## 5.2 Λειτουργία Εφαρμογής

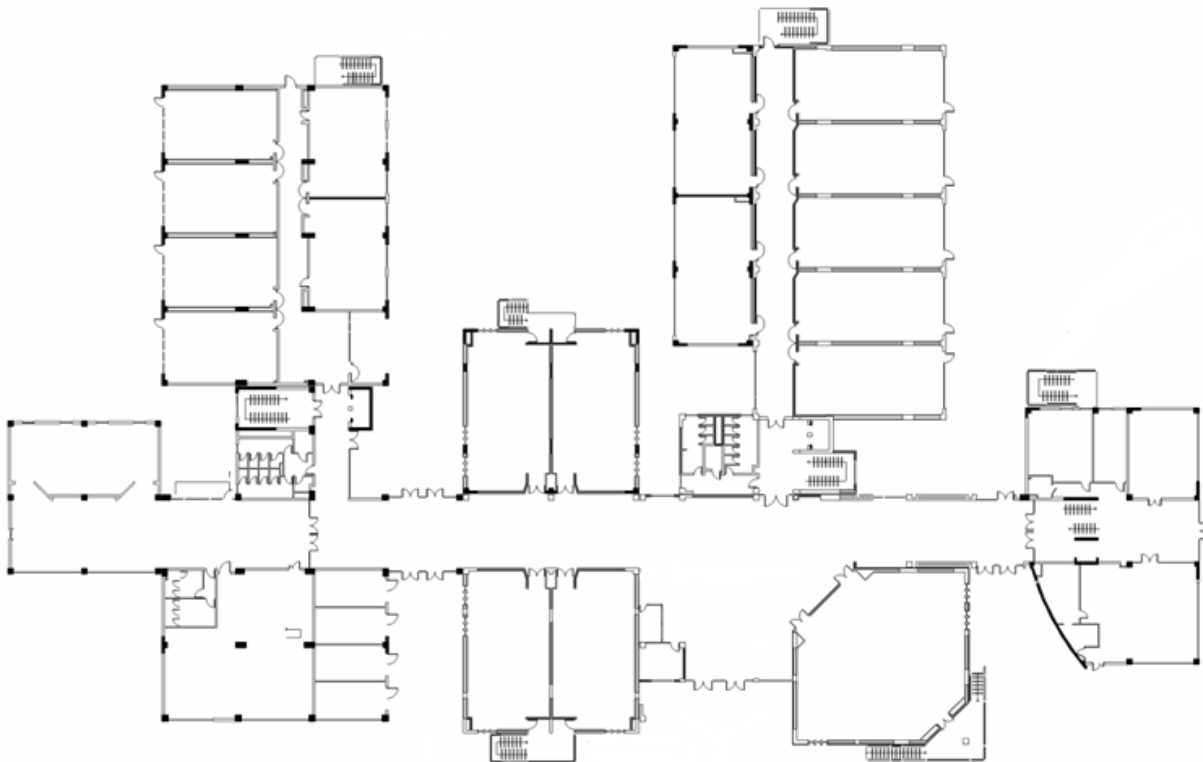
Η εφαρμογή θα αναπτυχθεί στην Unity για λειτουργικό σύστημα Android κάνοντας χρήση του ARCore. Η παρουσίαση του κτηρίου θα γίνει φτιάχνοντας ένα 3D model το οποίο θα είναι ακριβές αντίγραφο (σχεδιασμός και διαστάσεις) της σχολής. Η αρχική θέση του χρήστη θα εντοπιστεί σκανάροντας με την κάμερα του κινητού του την πινακίδα (plate) που υπάρχει έξω από την αίθουσα/αμφιθέατρο που βρίσκεται, με την χρήση του Image Recognition. Ο υπολογισμός της διαδρομής θα γίνεται με την χρήση του NavMesh και η θέση του στον χώρο ενημερώνεται από ένα σημείο σε ένα γειτονικό σημείο μέσω των αισθητήρων κίνησης της κινητής συσκευής (π.χ. επιταχυνσιόμετρο, γυροσκόπιο ) και τα οπτικά χαρακτηριστικά του περιβάλλοντος. Όσον αφορά το UI θα χρησιμοποιηθεί για την καθοδήγηση του χρήστη μία γραμμή (LineRenderer) η οποία δείχνει την διαδρομή που πρέπει να ακολουθήσει ο χρήστης ενώνοντας το current position του με τον τελικό στόχο και επιπλέον ένα βέλος (Arrow) το οποίο δείχνει την κατεύθυνση που πρέπει να κινηθεί στον χώρο. Τέλος, όταν ο χρήστης φτάσει στον στόχο εμφανίζεται ένα pin το οποίο καθορίζει και το τέλος της διαδρομής.

# 6

## Ανάπτυξη Εφαρμογής

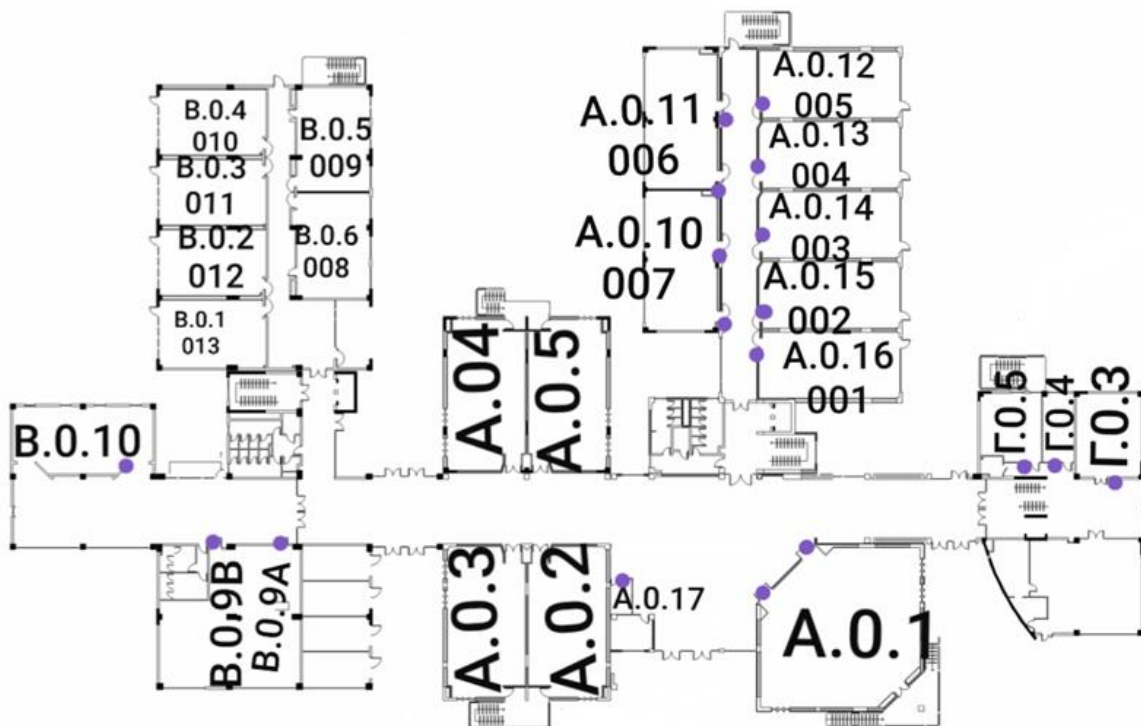
### 6.1 3D Model του Κτηρίου

Για να φέρουμε σε πέρας την εφαρμογή χρειαζόμασταν τα αρχιτεκτονικά σχέδια (floor plans) της σχολής τα οποία και ζητήσαμε, αφού για να τα χρησιμοποιήσουμε χρειαζόμαστε άδεια από την διεύθυνση. Τα αρχιτεκτονικά σχέδια φαίνονται στις Εικόνα 6.1, για τον όροφο 0.



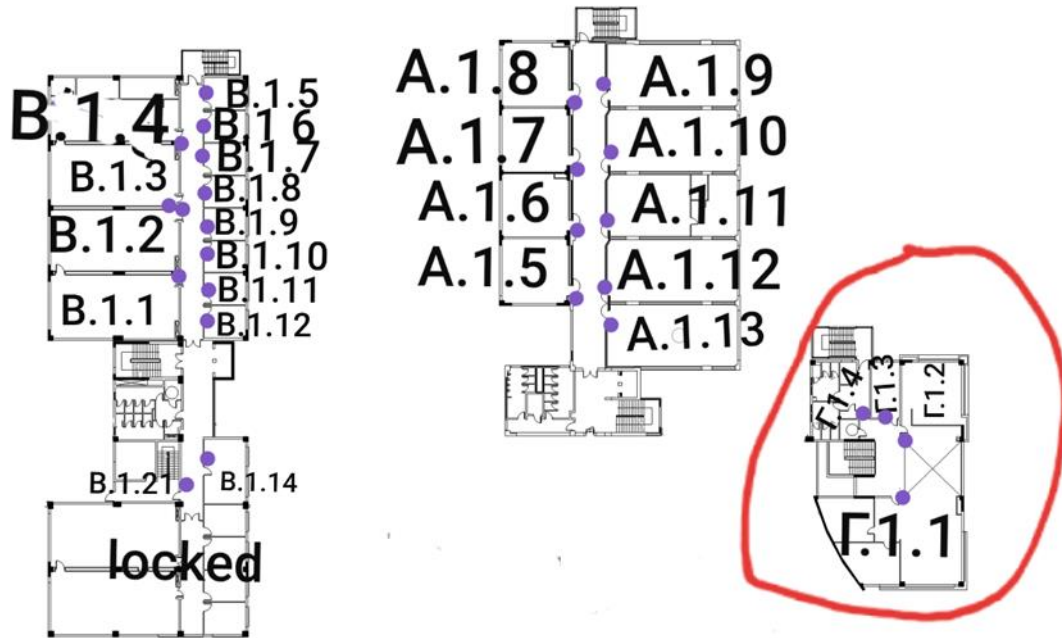
Εικόνα 6.1 Αρχιτεκτονικά σχέδια Νέα Κτήρια ΗΜΜΥ, όροφος 0

Στα οποία για διευκόλυνσή μας κατά την ανάπτυξη μέσα στο περιβάλλον της Unity καταγράψαμε την ονομασία της κάθε αίθουσας και αμφιθεάτρου, καθώς και χοντρικά που βρίσκεται η πινακίδα (plate) όπου αναγράφεται η ονομασία/αρίθμηση του (μωβ τελείες στην Εικόνα 6.2).



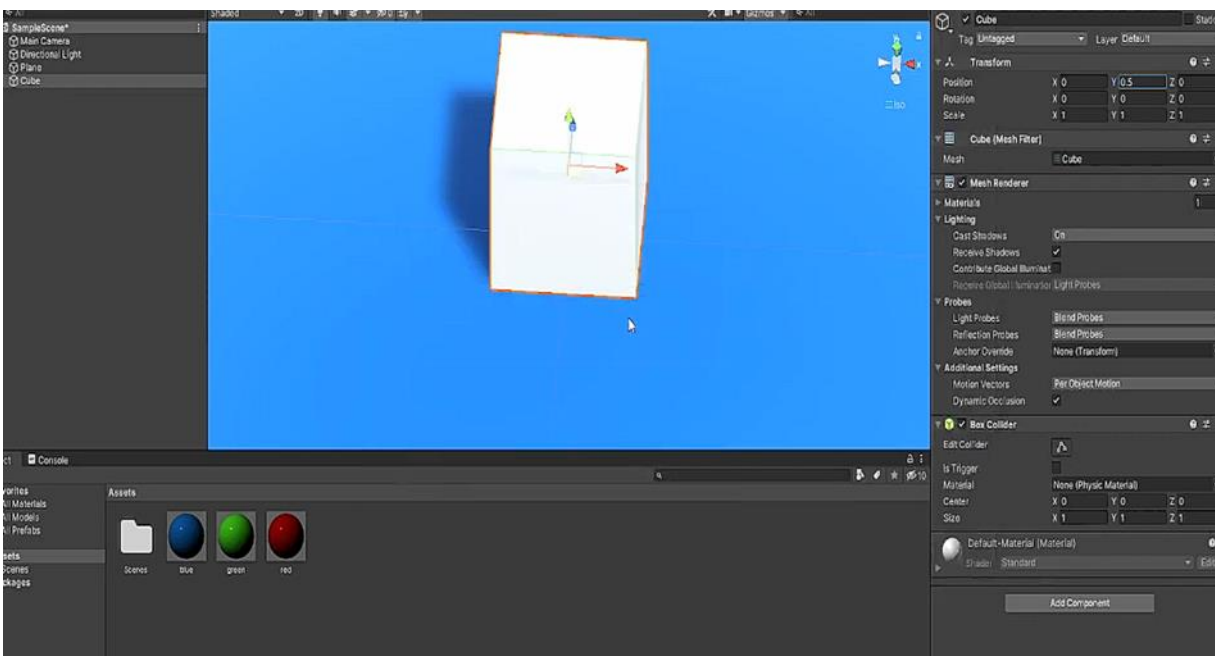
Εικόνα 6.2 Αρχιτεκτονικά σχέδια ορόφου 0 μαζί με ονομασίες αιθουσών/αμφιθεάτρων και τοποθεσίες πινακίδων

Το ίδιο κάναμε και για τον δεύτερο όροφο. Το κομμάτι που καλύπτει η εφαρμογή είναι το κυκλωμένο στην Εικόνα 6.3, το οποίο αντιπροσωπεύει την πτέρυγα Γ. Δεν καλύψαμε όλον τον δεύτερο όροφο διότι σκοπός ήταν να ανακαλύψουμε πώς θα κατασκευάσουμε με επιτυχία την μετάβαση από τον έναν όροφο στον άλλον, και αυτό το σημείο του κτηρίου ήταν αρκετό. Οι σκάλες που φαίνονται στο σημείο που είναι κυκλωμένο είναι δύο, μία που ενώνει τον όροφο 0 με την πτέρυγα Γ του ορόφου 1 και μία που ενώνει τις Α και Β πτέρυγες (ή φάσεις) με την πτέρυγα Γ του ορόφου 0 η οποία είναι υπόγειος ημιόροφος (θα γίνει πιο κατανοητή παρακάτω η συγκεκριμένη αρχιτεκτονική).



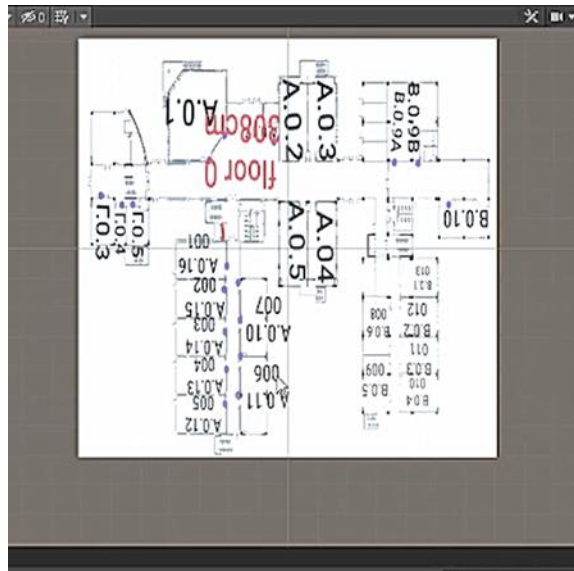
Εικόνα 6.3 Αρχιτεκτονικά σχέδια 1ου ορόφου

Για την αναπαράσταση του κτηρίου φτιάξαμε ένα 3d model του στην Unity. Αυτό επιτυγχάνεται φτιάχνοντας ένα Plane το οποίο θα εκπροσωπεί το έδαφος και του κάνουμε assign ένα μπλε material. Φτιάξαμε ένα cube του οποίου το Position στο y είναι 0.5 ώστε να ακουμπάει ακριβώς στο έδαφος μιας που το μέγεθος των κύβων είναι 1x1x1 (meters) by default (βλ. Εικόνα 6.4).



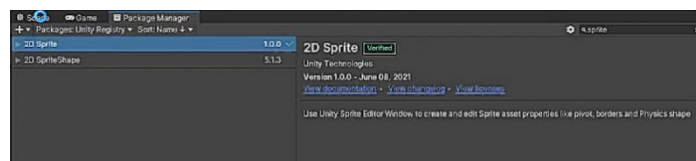
Εικόνα 6.4 Plane του εδάφους και το Cube για τους τοίχους στην Unity

Κάναμε import τα floor plans σε μορφή JPG File. Αρχικά προσπαθήσαμε να κάνουμε assign σε αυτήν την μορφή τα floor plans στο Plane του εδάφους, αλλά αυτό δεν επιφέρει το επιθυμητό αποτέλεσμα αφού εμείς θέλουμε οι διαστάσεις του κτηρίου να είναι ακριβώς οι ίδιες με την πραγματικότητα και αν προσπαθούσαμε να μεγαλώσουμε το plane με τέτοιον τρόπο ώστε η φωτογραφία των floor plans να μην είναι παραμορφωμένη δεν θα καταλήγαμε σε ακριβή αντιστοιχία. Η Εικόνα 6.5 δείχνει τι γίνεται αν προσπαθήσουμε τον εξής τρόπο. Το ίδιο ακριβώς πρόβλημα θα είχαμε αν κάναμε assign τα floor plans σε new material και στην συνέχεια κάναμε assign αυτό το material στο plane του εδάφους.



Εικόνα 6.5 Λάθος εισαγωγή των floor plans στην σκηνή μας στην Unity

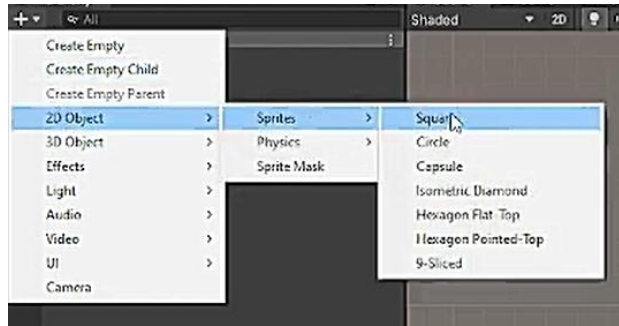
Αντί λοιπόν για την παραπάνω τεχνική κάναμε install το 2D Sprite από το Package Manager της Unity (βλ. Εικόνα 6.6).



Εικόνα 6.6 Install 2D Sprite από Package Manager στην Unity

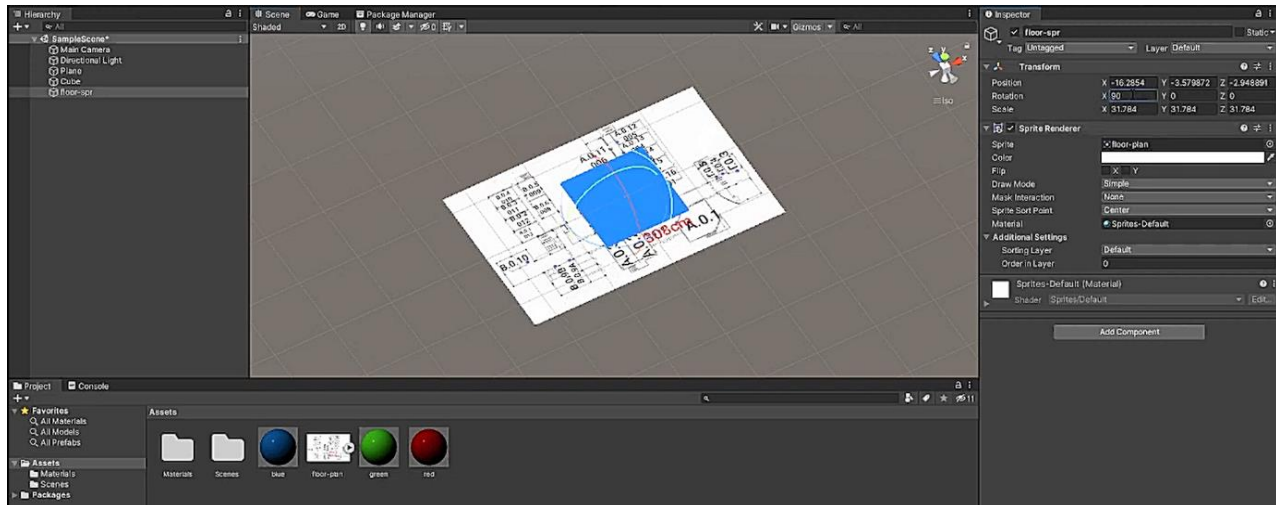
Στην σκηνή μας στην Unity : Create New > 2D Object > Sprites > Square (βλ. Εικόνα 6.7)





Εικόνα 6.7 Δημιουργία ενός 2D Sprite Square στην Unity

Αλλάξαμε το Texture Type του JPG File σε Sprite (2D and UI) και το κάναμε import στο 2D Sprite Square. Τέλος, διορθώνουμε το rotation (90 μοίρες) ώστε να είναι παράλληλο με το plane του εδάφους (βλ. Εικόνα 6.8).



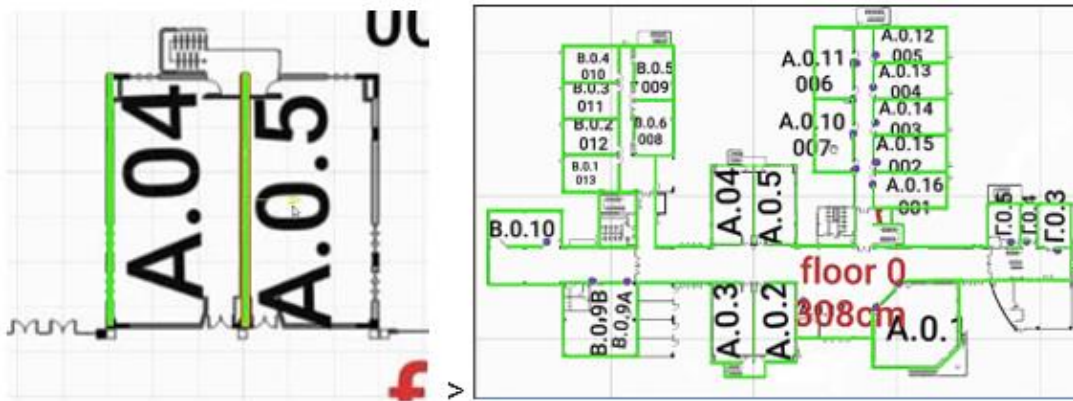
Εικόνα 6.8 Εισαγωγή floor plans με σωστές διαστάσεις

Αλλάζουμε το scale του plane με τέτοιο τρόπο ώστε να είναι ίσο με αυτό των floor plans και τοποθετούμε τα floor plans από πάνω. (βλ. Εικόνα 6.9)



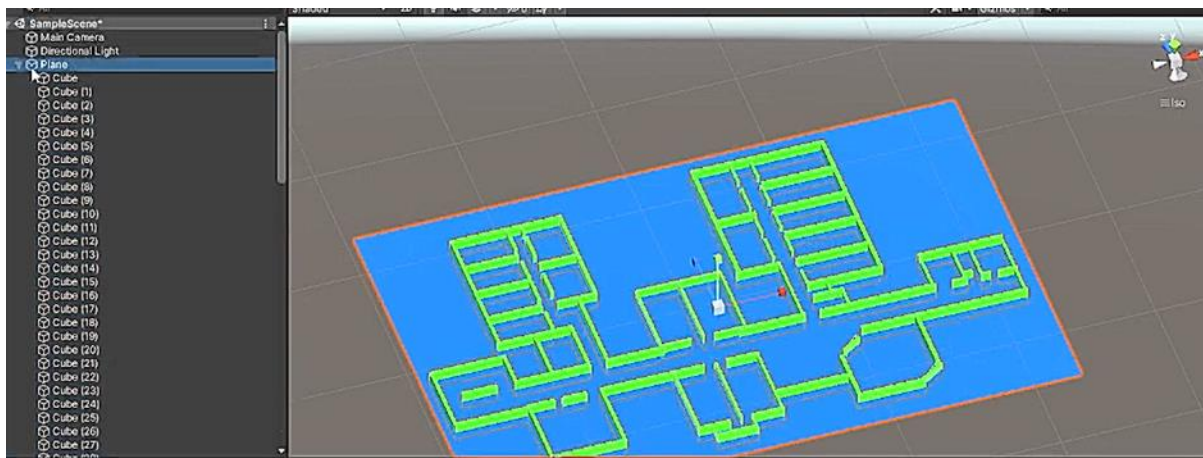
Εικόνα 6.9 Προσαρμογή scale του plane του εδάφους ώστε να ταιριάζει με αυτό των floor plans

Στην συνέχεια φτιάξαμε τους τοίχους με cubes τα οποία προσαρμόσαμε σε scale έτσι ώστε να ταιριάζουν με τους τοίχους των floor plans. Το γεγονός ότι ένα cube στην Unity είναι 1x1x1 meters καθιστά την κατασκευή του ακριβούς αντίγραφου του κτηρίου εύκολη. Μετρήσαμε έναν από τους τοίχους του πραγματικού κτηρίου της σχολής και χρησιμοποιήσαμε το standard cube για να προσαρμόσουμε κατάλληλα το περιβάλλον. Έτσι έχοντας μετρήσει τον τοίχο ο οποίος ήταν 308cm, θέσαμε το μέγεθος του κύβου να είναι 3.08 και σιγουρευτήκαμε ότι αυτό επαληθεύεται και για το υπόλοιπο κτήριο. Κάθε τοίχος είναι ένα cube. “Ξεπατικώνουμε” ουσιαστικά τους τοίχους που φαίνονται στην κάτοψη του κτηρίου. Το τελικό αποτέλεσμα φαίνεται στην δεξιά πλευρά της Εικόνα 6.10.



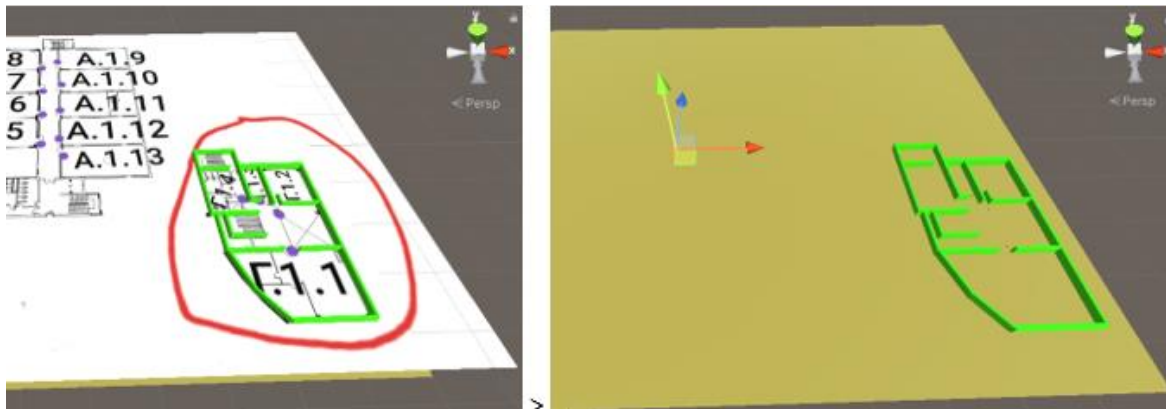
Εικόνα 6.10 Δημιουργία των τοίχων του κτηρίου

Κάναμε hide τα floor plans και κάναμε και όλα τα cubes (τοίχοι) παιδιά του plane. Αυτό είναι το τελικό 3D Model του κτηρίου που θα χρησιμοποιήσουμε για την ανάπτυξη της υπόλοιπης εφαρμογής (βλ. Εικόνα 6.11)



Εικόνα 6.11 3D Model ορόφου 0 του κτηρίου

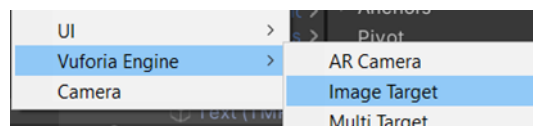
Το ίδιο κάναμε και για το κομμάτι του πρώτου ορόφου που μας ενδιαφέρει. (βλ Εικόνα 6.12)



Εικόνα 6.12 3D Model 1ου ορόφου του κτηρίου

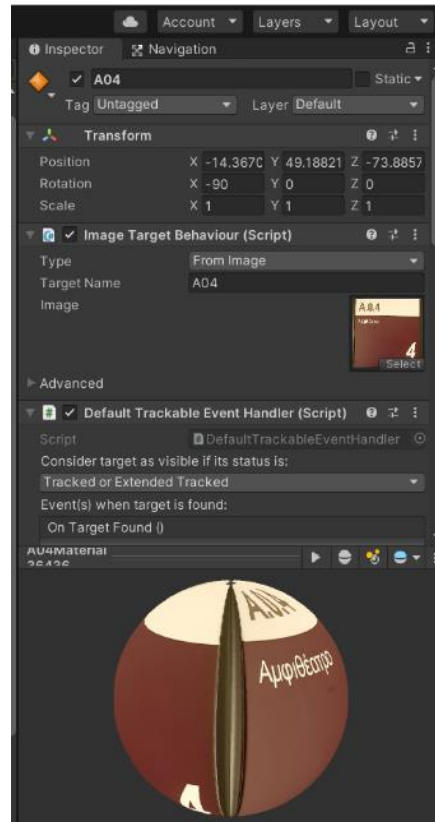
## 6.2 Τοποθέτηση των Πινακίδων

Για να πετύχουμε το Image Recognition φτιάξαμε αρχικά ένα Image Target (βλ. Εικόνα 6.13) για κάθε plate. Τα plates, όπως αναφέρθηκε και προηγουμένως, είναι οι πινακίδες έξω από κάθε αίθουσα/αμφιθέατρο όπου αναγράφεται η ονομασία τους, η ονομασία της κάθε αίθουσας αποτελείται από τον τομέα στον οποίο βρίσκεται: Α ή Β ή Γ, τον όροφο 0 ή 1 και την αρίθμηση της 1-16 πχ Α.0.4 .



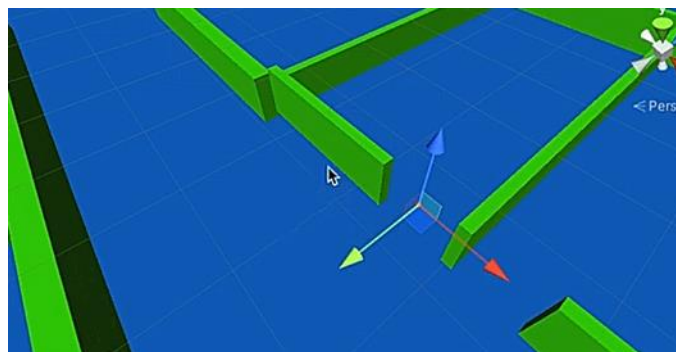
Εικόνα 6.13 Δημιουργία Image Target στην Unity

Κάθε Image Target έχει Type>From Image, του δώσαμε Target Name το όνομα του plate το οποίο εκπροσωπεί και του αναθέσαμε την φωτογραφία του plate που έχουμε τραβήξει. (βλ. Εικόνα 6.14)



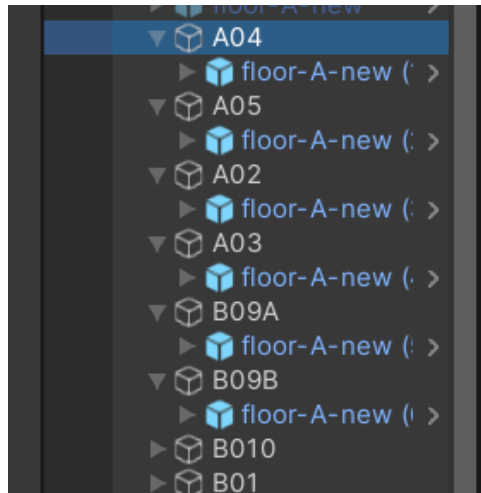
Εικόνα 6.14 Inspector για την δημιουργία Image Target για κάθε plate

Το κάθε plate το τοποθετούμε στο “σωστό” σημείο στην σκηνή (scene) μας, εκεί δηλαδή που είναι και στο πραγματικό κτήριο της σχολής. (βλ. Εικόνα 6.15)



Εικόνα 6.15 Τοποθέτηση του plate στο 3d model του κτηρίου

Κάθε plate έχει παιδί του από ένα 3d model των floor plans. (βλ. Εικόνα 6.16)



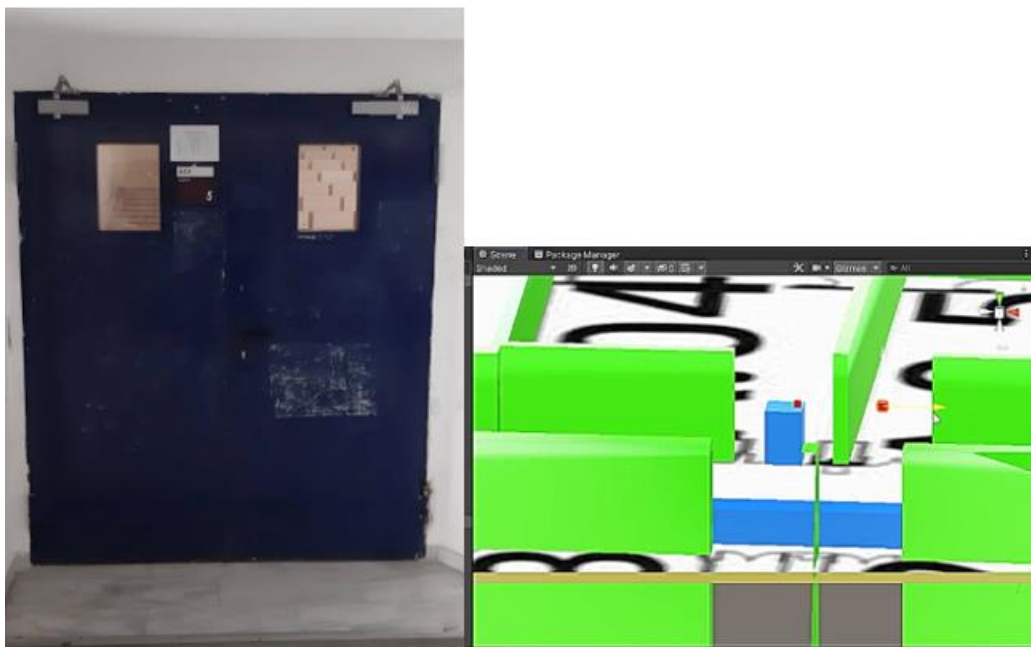
Εικόνα 6.16 Hierarchy για τα Image Targets των plates

Φτιάξαμε έναν κύβο ύψους 1.5m όπου 1.5 m το ύψος των plates από το έδαφος στην πραγματικότητα. Στην συνέχεια φτιάξαμε έναν κύβο για κάθε plate και το τοποθετούμε πάνω ακριβώς από τον κύβο και στο σημείο που βρίσκεται έξω από κάθε αίθουσα/αμφιθέατρο, έχοντας ως αναφορά φωτογραφίες που έχουμε τραβήξει από τον χώρο της σχολής. Στις Εικόνα 6.17 και Εικόνα 6.18 φαίνονται δύο παραδείγματα για τα αμφιθέατρα A.0.4 και A.0.5 αντίστοιχα.



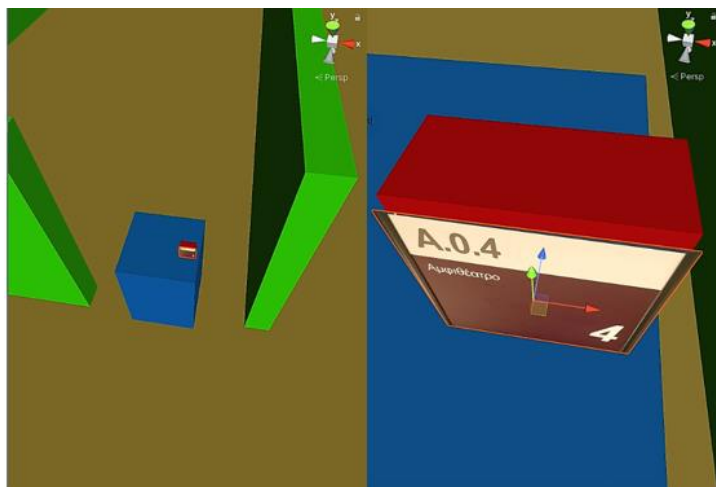
Εικόνα 6.17 Τοποθέτηση plate A.0.4

Μετά την τοποθέτηση του πρώτου plate (A.0.4) κάνουμε duplicate το τελευταίο plate που έχουμε τοποθετήσει και κρατώντας σταθερό το ύψος (y) τοποθετούμε στην σωστή θέση και το επόμενο plate.



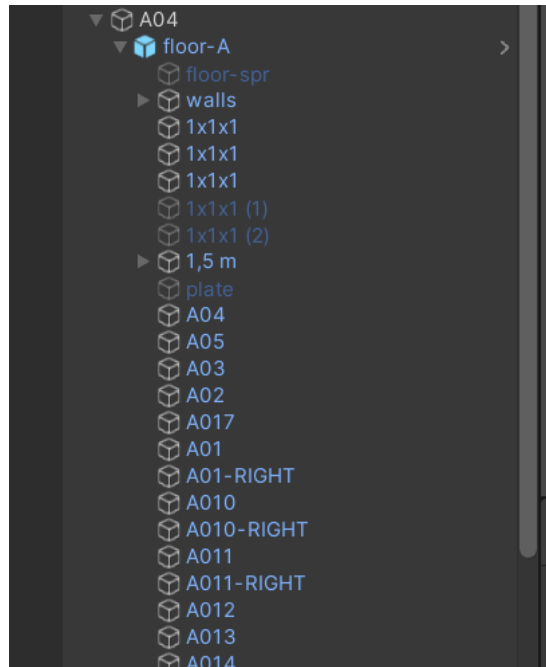
Εικόνα 6.18 Τοποθέτηση plate A.0.5

Άρα για κάθε αίθουσα/αμφιθέατρο έχουμε φτιάξει έναν κύβο και ένα ImageTarget τα οποία έχουμε τοποθετήσει στο ίδιο σημείο. Τοποθετούμε το ImageTarget ακριβώς μπροστά από τον κύβο. (βλ. Εικόνα 6.19)



Εικόνα 6.19 Τελική τοποθέτηση plates στην σκηνή

Στο hierarchy οι κύβοι είναι παιδιά του 3d model του κτηρίου (βλ. Εικόνα 6.20). Ακολουθούμε αυτήν την διαδικασία για όλες τα plates και έτσι έχουμε ένα room Object για κάθε αίθουσα/αμφιθέατρο.

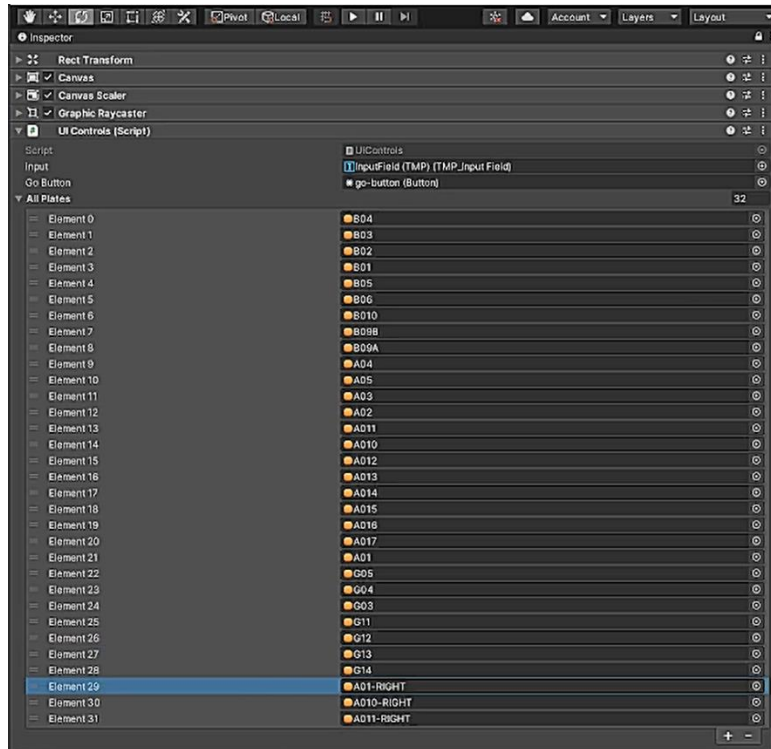


Εικόνα 6.20 Τελικό hierarchy για plates

### 6.3 Εντοπισμός του χρήστη

#### TEST #1

Για να δούμε αν δουλεύει η τοποθέτηση των plates, φτιάξαμε στο UIControls.cs μια λίστα στην οποία προσθέσαμε όλα τα plates. Η προσθήκη όλων των plates έγινε στον Inspector του UI. (βλ. Εικόνα 6.21)



Εικόνα 6.21 Εισαγωγή των plates στον Inspector του UI

Τεστάρουμε το navigation ξεκινώντας από κάποιο τυχαίο (random) plate. Κώδικας στο UIControls.cs φαίνεται στην Εικόνα 6.22.

```

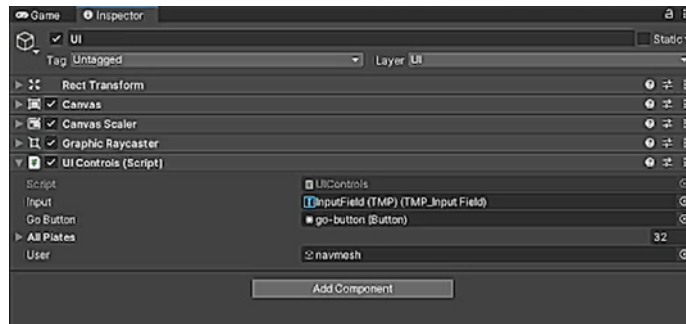
63  Unity Message | 0 references
64  private void Start()
65  {
66      StartFromRandomPlate();
67  }
68
69  public List<GameObject> AllPlates;
70  public GameObject User;
71
72  1 reference
73  public void StartFromRandomPlate()
74  {
75      GameObject randomPlate = AllPlates[Random.Range(0, AllPlates.Count - 1)];
76      User.transform.position = randomPlate.transform.position;
77      User.SetActive(true);
78  }

```

Εικόνα 6.22 Code στο UIControls.cs για εκκίνηση από random plate

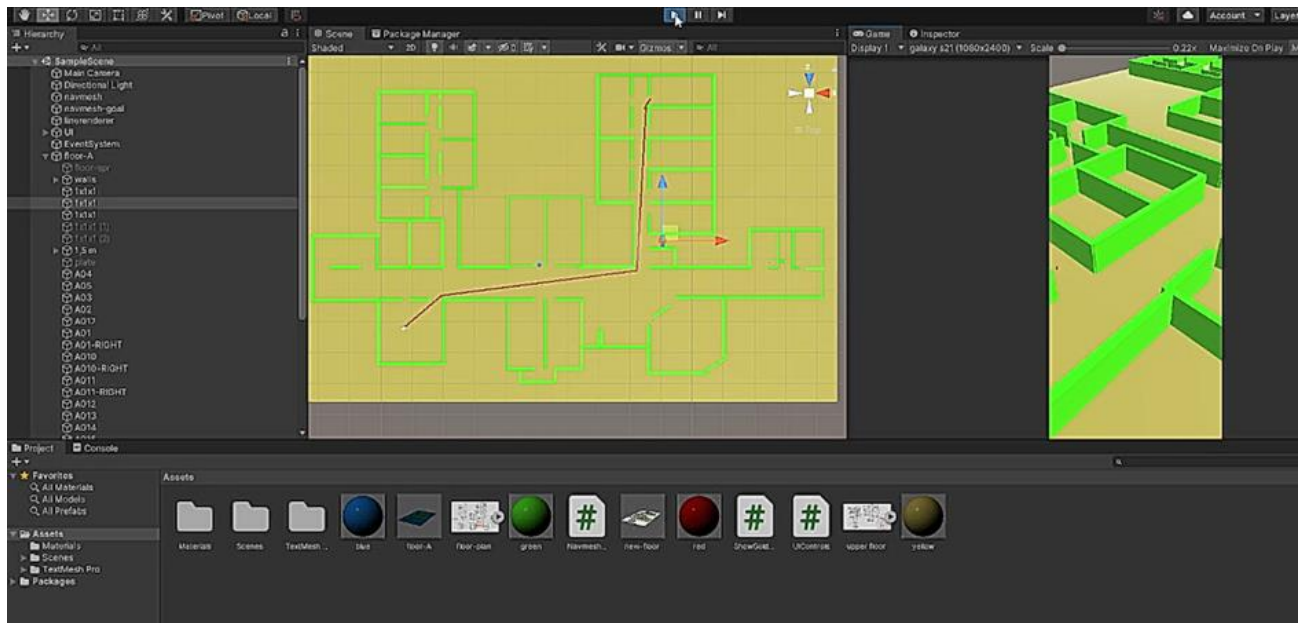
Βάλαμε ως user το navmesh (βλ. Εικόνα 6.23)





Εικόνα 6.23 Τοποθέτηση στον Inspector του UI το navmesh ως User

Ένα τυχαίο τρέξιμο φαίνεται στην Εικόνα 6.24 όπου το σημείο εκκίνησης επιλέγεται τυχαία και ο τελικός προορισμός είναι σταθερός.



Εικόνα 6.24 TEST εφαρμογής με τυχαία επιλογή σημείου εκκίνησης

## TEST #2

Τεστάρουμε αν δουλεύει ο εντοπισμός (localization) του χρήστη χρησιμοποιώντας ξανά την τυχαία επιλογή σκαναρισμένου plate. Ο κώδικας που προσθέσαμε στο UIControls.cs φαίνεται στην Εικόνα 6.25.

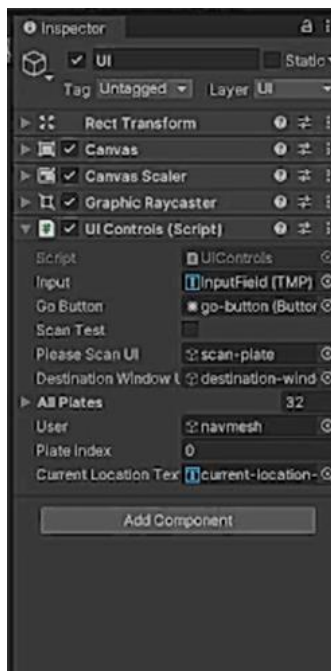
```

67
68     public bool ScanTest;
69     public GameObject PleaseScanUI, DestinationWindowUI;
70
71     @ Unity Message | 0 references
72     private void Update()
73     {
74         if (ScanTest)
75         {
76             ScanTest = false;
77             StartFromRandomPlate();
78             PleaseScanUI.SetActive(false);
79             DestinationWindowUI.SetActive(true);
80         }
81     }
82
83     public List<GameObject> AllPlates;
84     public GameObject User;
85     public int PlateIndex;
86     public TextMeshProUGUI CurrentLocationText;
87
88     2 references
89     public void StartFromRandomPlate()
90     {
91         PlateIndex = Random.Range(0, AllPlates.Count - 1);
92         GameObject randomPlate = AllPlates[PlateIndex];
93         User.transform.position = randomPlate.transform.position;
94         CurrentLocationText.text = "Current location: " + rooms[PlateIndex];
95         //User.SetActive(true);
96     }

```

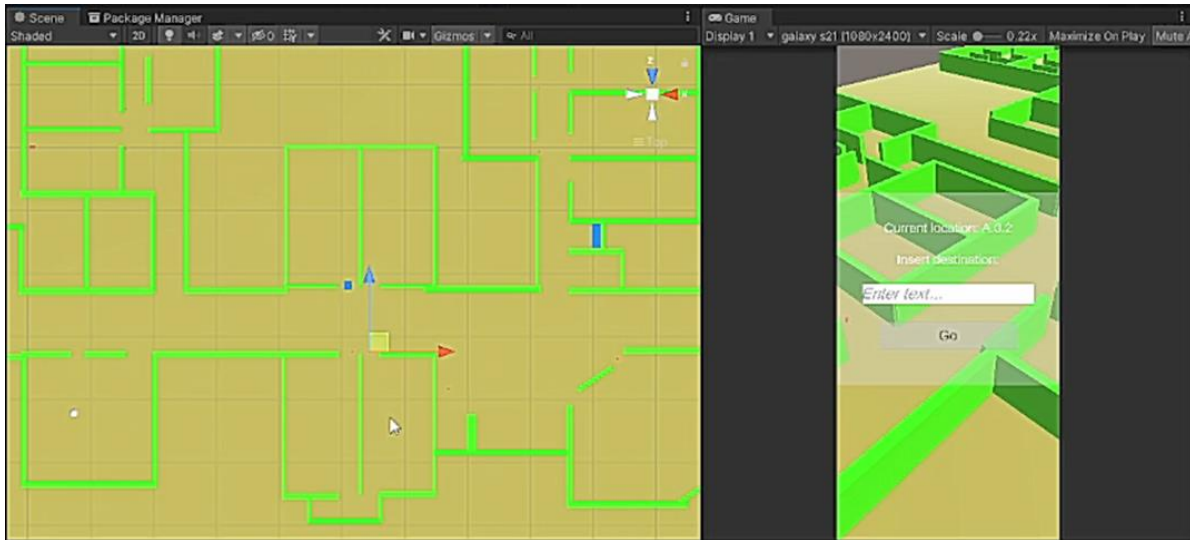
Εικόνα 6.25 Code στο UIControls.cs για επαλήθευση του σωστού localization του χρήστη

Ο Inspector του UI φαίνεται στην Εικόνα 6.26.



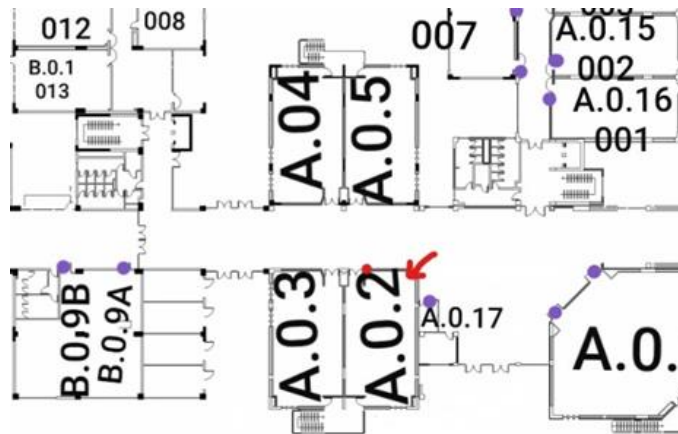
Εικόνα 6.26 Inspector του UI για TEST #2

Το αποτέλεσμα από τυχαίο τρέξιμο φαίνεται στην Εικόνα 6.27 όπου το Current location είναι το αμφιθέατρο A.0.2 όπως φαίνεται δεξιά και το αποτέλεσμα το για το localization του χρήστη φαίνεται αριστερά.



Εικόνα 6.27 TEST εφαρμογής για επαλήθευση της τυχαίας επιλογής σημείου εκκίνησης

Το localization για το σημείο εκκίνησης του χρήστη ήταν σωστό αφού το plate του αμφιθέατρου A.0.2 βρίσκεται στο σημείο που είναι ο κέρσοντας όπως φαίνεται στην Εικόνα 6.28.



Εικόνα 6.28 Floor plans της σχολής όπου σημειώνεται με κόκκινο χρώμα η πινακίδα του αμφιθέατρου A.0.2 και η τοποθεσία του

## 6.4 Επιλογή από τον Χρήστη του Τελικού Προορισμού

### TEST #3

Η επιλογή της αρχικής θέσης συνεχίζει να είναι τυχαία. Στο σημείο αυτό στο `UIControls.cs` κάναμε τις κατάλληλες αλλαγές (βλ. Εικόνα 6.29) ώστε ο τελικός προορισμός να επιλέγεται από τον χρήστη πληκτρολογώντας το “όνομα” της αίθουσας ή του αμφιθεάτρου που επιθυμεί να πάει.

```
        "0.1.4",
        "0.1.3",
        "0.1.4"
    };

    public TMP_InputField input;
    public Button GoButton;

    0 references
    public void CheckRoom()
    {
        string temp = input.text;
        for (int i = 0; i < rooms.Length; i++)
        {
            //Debug.Log(input.text);
            if(rooms[i] == temp)
            {
                DestinationPlateIndex = i;

                GoButton.interactable = true;

                // stops the loop, should be the last line
                i = rooms.Length;
            }
            else
            {
                GoButton.interactable = false;
            }
        }
    }

    public bool ScanTest;
    public GameObject PleaseScanUI, DestinationWindowUI;

    0 Unity Message | 0 references
    private void Update()
    {
        if (ScanTest)
        {
            ScanTest = false;
            StartFromRandomPlate();
            PleaseScanUI.SetActive(false);
            DestinationWindowUI.SetActive(true);
        }
    }

    public List<GameObject> AllPlates;
    public GameObject User;
    public int PlateIndex, DestinationPlateIndex;
    public TextMeshProUGUI CurrentLocationText;
    public GameObject TargetObject;

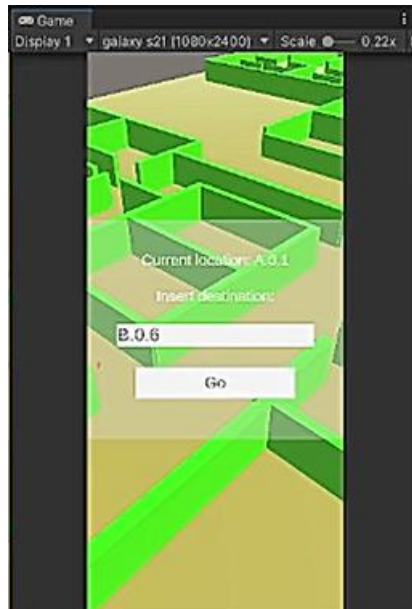
    2 references
    public void StartFromRandomPlate()
    {
        PlateIndex = Random.Range(0, AllPlates.Count - 1); // possible bug here
        GameObject randomPlate = AllPlates[PlateIndex];
        User.transform.position = randomPlate.transform.position;
        CurrentLocationText.text = "Current location: " + rooms[PlateIndex];
        //User.SetActive(true);
    }

    0 references
    public void OnGoButtonPressed()
    {
        TargetObject.transform.position = AllPlates[DestinationPlateIndex].transform.position;
        DestinationWindowUI.SetActive(false);
        User.SetActive(true);
    }
}
```

Εικόνα 6.29 Code στο `UIControls.cs` για επιλογή από τον χρήστη του τελικού προορισμού

Παρακάτω παρατίθενται δύο παραδείγματα για την επαλήθευση της σωστής λειτουργίας της εφαρμογής.

Παράδειγμα 1:



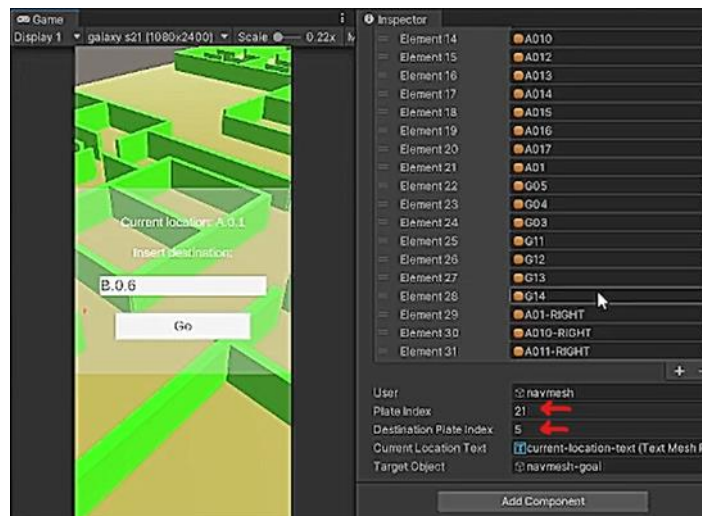
Εικόνα 6.30 Παράδειγμα 1

Στο παράδειγμα αυτό η random επιλογή του σημείου εκκίνησης ήταν το plate A.0.1. Επιλέξαμε να εισάγουμε ως τελικό προορισμό την αίθουσα B.0.6 (βλ. Εικόνα 6.30). Το index (δείκτης στην λίστα με όλα τα plates) για το κάθε plate φαίνεται στην Εικόνα 6.31.



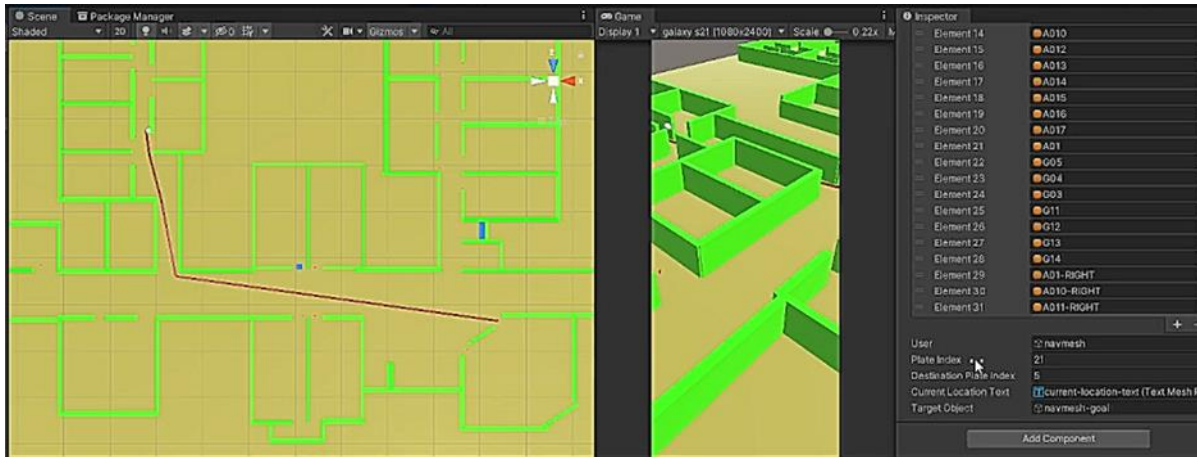
Εικόνα 6.31 Ο Inspector του UI όπου φαίνεται το index του κάθε plate

Όπου το index για το αμφιθέατρο A.0.1 είναι 21 και για την αίθουσα B.0.6 είναι 5, τα οποία συμπίπτουν με το αποτέλεσμα όπως φαίνεται και στην Εικόνα 6.32.



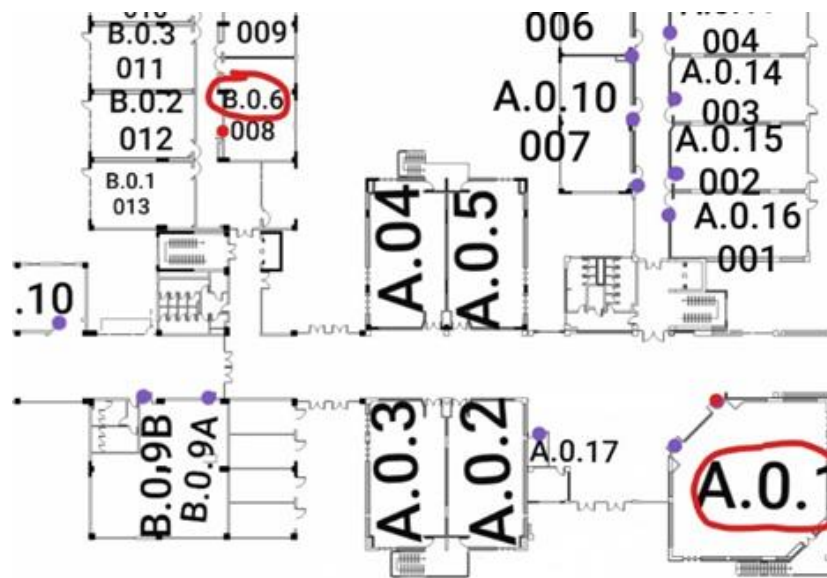
Εικόνα 6.32 Αποτελέσματα στον Inspector του UI

Πατώντας το go button η διαδρομή που εμφανίστηκε φαίνεται στην Εικόνα 6.33.



Εικόνα 6.33 Το αποτέλεσμα της εφαρμογής μετά το πάτημα του go button

Το αποτέλεσμα είναι σωστό και επαληθεύεται στην Εικόνα 6.34.



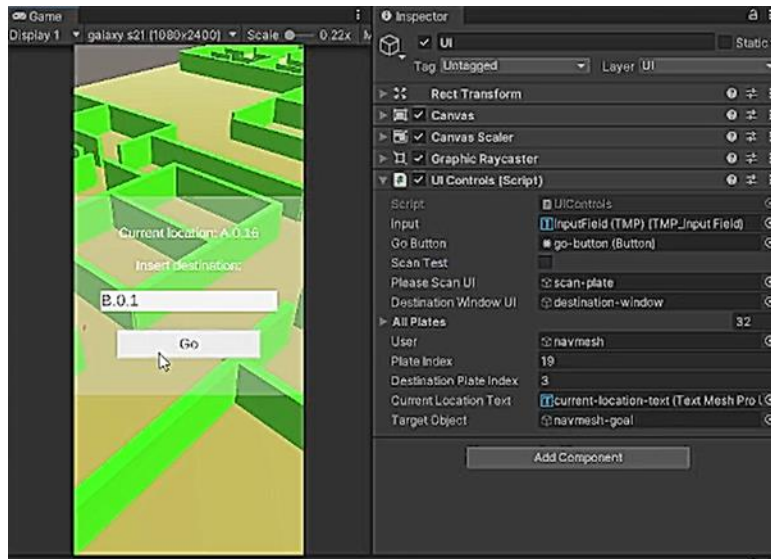
Εικόνα 6.34 Floor plans της σχολής για επαλήθευση του Παραδείγματος 1

Παράδειγμα 2:

Στο παράδειγμα αυτό έχουμε (βλ. Εικόνα 6.35):

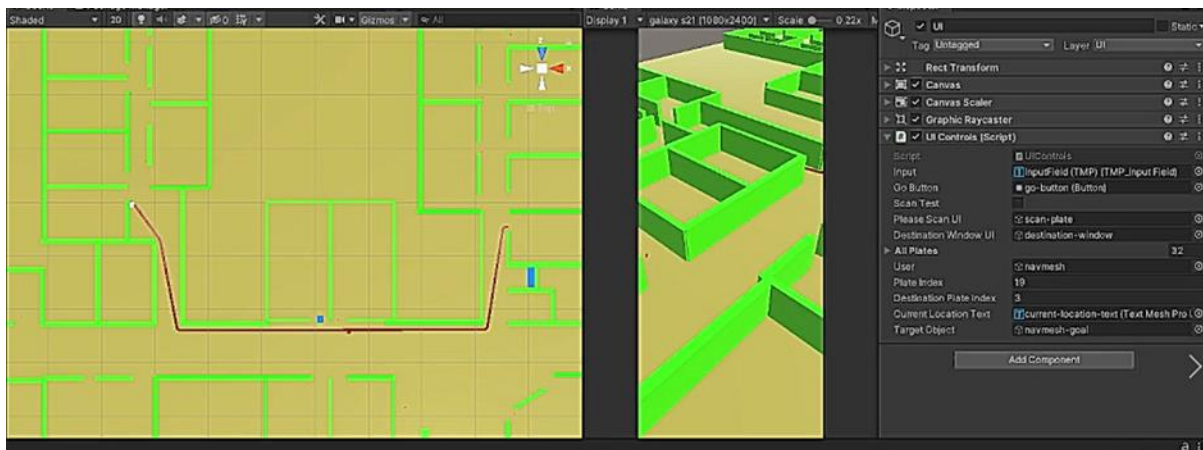
Random plate selection: αίθουσα A.0.16 με Plate Index=19 και

Επιλογή ως τελικό προορισμό: αίθουσα B.0.1 με Destination Plate Index=3



Εικόνα 6.35 Παράδειγμα 2

Σωστά σύμφωνα και με την Εικόνα 6.31. Πατώντας το go button η διαδρομή που εμφανίστηκε φαίνεται στην Εικόνα 6.36. Το αποτέλεσμα είναι σωστό σύμφωνα με την εικόνα των floor plans στο πρώτο παράδειγμα (βλ. Εικόνα 6.34).



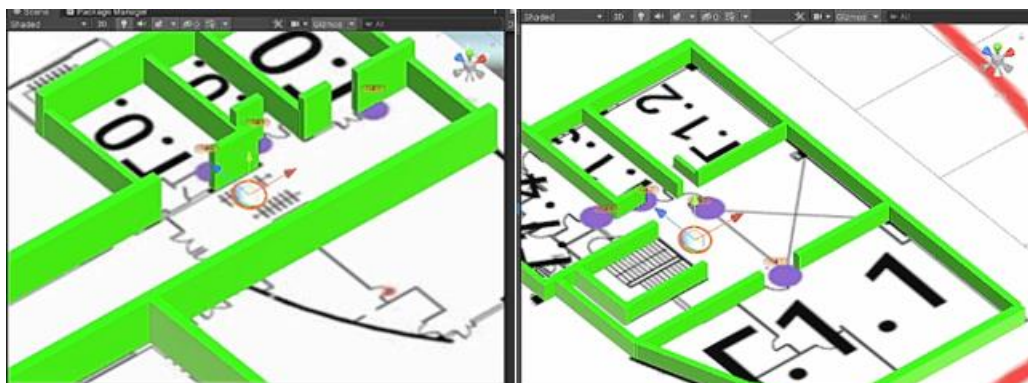
Εικόνα 6.36 Το αποτέλεσμα της εφαρμογής μετά το πάτημα του go button

## 6.5 Εναλλαγή Ορόφων

### 6.5.1 OffMeshLink για σκάλες



Η πρώτη σκέψη για να λύσουμε το πρόβλημα με την εναλλαγή ορόφων σε ένα navigation session ήταν ο χρήστης με το που φτάνει στις σκάλες που πρέπει να ανέβει (ή να κατέβει) να του εμφανίζεται ένα μήνυμα στην οθόνη το οποίο να τον προειδοποιεί ότι πρέπει να χρησιμοποιήσει τις σκάλες για να μεταφερθεί σε άλλον όροφο και αφότου τις διασχίσει να πατάει ένα κουμπί για να ενημερώνει την εφαρμογή ότι βρίσκεται πλέον στον άλλον όροφο και να συνεχίσει το navigation από το σημείο εκείνο. Για να το πετύχουμε αυτό φτιάξαμε ένα sphere 3D Object (floor-transition-A) που τοποθετήσαμε στην αρχή των σκαλιών (και το θέσαμε ως παιδί του floor-A) και ένα (floor-transition-G) στο τέλος αυτών (και το θέσαμε ως παιδί του floor-G). Τα δύο 3D Objects φαίνονται στην Εικόνα 6.37.



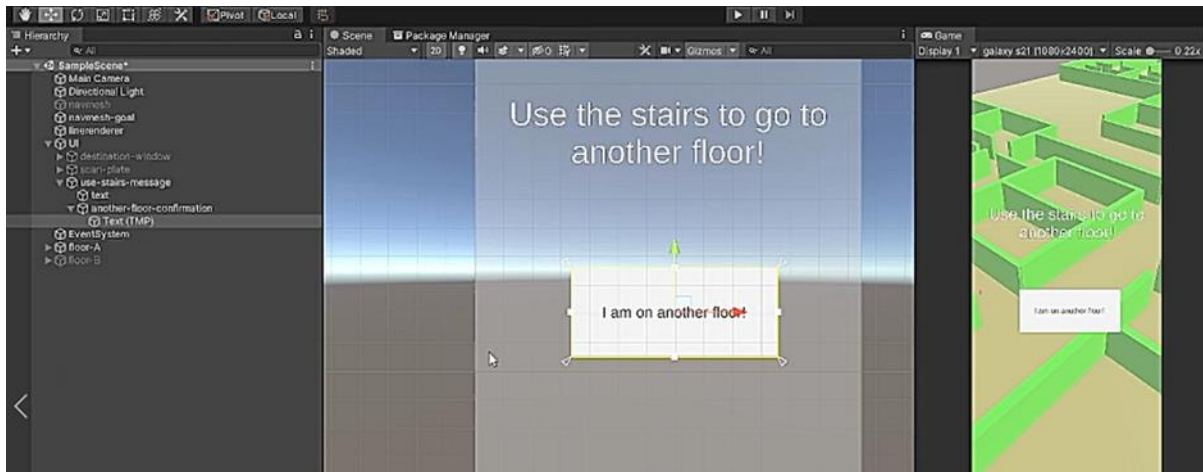
Εικόνα 6.37 3D Objects στον όροφο 0 (αριστερά) και στον όροφο 1 (δεξιά) όπου θα εμφανίζεται το μήνυμα για την εναλλαγή ορόφων

Ο Inspector του UI για τα transition points φαίνεται στην Εικόνα 6.38.



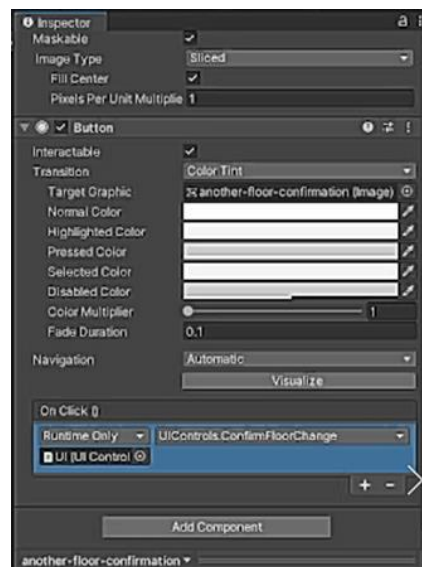
Εικόνα 6.38 Inspector του UI για τα δύο transition points

Το UI για τις σκάλες αποτελείται από το μήνυμα που θα εμφανίζεται στον χρήστη και ένα κουμπί (button) που θα πατάει με το που έχει φτάσει στον άλλον όροφο. (βλ. Εικόνα 6.39)



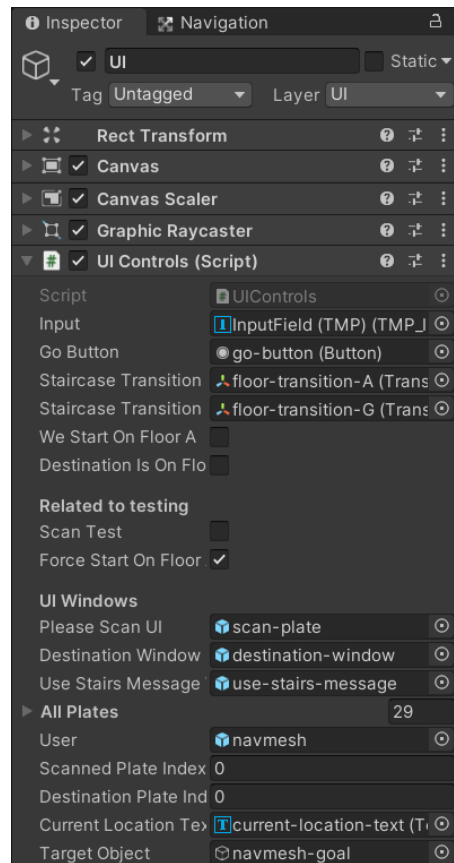
Εικόνα 6.39 UI για τις σκάλες

Ο Inspector για το another-floor-confirmation φαίνεται στην Εικόνα 6.40.



Εικόνα 6.40 Inspector για το another-floor-confirmation

Ο Inspector για το UI φαίνεται στην Εικόνα 6.41.



Εικόνα 6.41 Inspector για το UI

Οι αλλαγές στο UIControls.cs που φαίνονται στην Εικόνα 6.42 ενημερώνουν την εφαρμογή αν θα πρέπει να χρησιμοποιήσει ο χρήστης τις σκάλες ή όχι.

```

public void ConfirmFloorChange()
{
    UseStairsMessageWindowUI.SetActive(false);

    //activate another floor navigation
    //start from StaircaseTransition spot
}

public void CheckRoom()
{
    string temp = input.text;
    for (int i = 0; i < rooms.Length; i++)
    {
        //Debug.Log(input.text);
        if (rooms[i] == temp)
        {
            DestinationPlateIndex = i;
            GoButton.interactable = true;

            if(ScannedPlateIndex < 25)
            {
                //we start on floor A.0.x
                WeStartOnFloorA = true;
            }
            else
            {
                //we start on floor G.1.x
                WeStartOnFloorA = false;
            }

            if(temp == "G11" || temp == "G12" || temp == "G13" || temp == "G14")
            {
                //our destination is on G.1.x floor
                DestinationIsOnFloorG = true;
            }
            else
            {
                //our destination is on A.0.x floor
                DestinationIsOnFloorG = false;
            }

            //stops the loop
            i = rooms.Length;
        }
        else
        {
            GoButton.interactable = false;
        }
    }
}

```

Εικόνα 6.42 UIControls.cs αλλαγή στην CheckRoom() για να ενημέρωνει την εφαρμογή αν θα γίνει χρήση της σκάλας

Αν οι σκάλες πρέπει να χρησιμοποιηθούν από τον χρήστη, Target γίνεται ένα από τα δύο 3D Objects που φτιάξαμε προηγουμένως (ανάλογα με τον όροφο που βρίσκεται ήδη), όπου θα του εμφανιστεί το μήνυμα για την χρήση της σκάλας. Φτιάξαμε επίσης και την ForceStartOnFloorA για διευκόλυνση στο testing. (βλ. Εικόνα 6.43)

```

[Header("Related to testing")]
public bool ScanTest;
public bool ForceStartOnFloorA;

[Header("UI windows")]
public GameObject PleaseScanUI;
public GameObject DestinationWindowUI;
public GameObject UseStairsMessageWindowUI;

@ Unity Message | 0 references
private void Update()
{
    if (ScanTest) // what happens after we scan
    {
        ScanTest = false;
        StartFromRandomPlate();
        PleaseScanUI.SetActive(false);
        DestinationWindowUI.SetActive(true);
    }

    public List<GameObject> AllPlates;
    public GameObject User;
    public int ScannedPlateIndex, DestinationPlateIndex;
    public TextMeshProUGUI CurrentLocationText;
    public GameObject TargetObject;

    public GameObject randomPlate;
    1 reference
    public void StartFromRandomPlate()
    {
        if (ForceStartOnFloorA)
        {
            //ScannedPlateIndex = Random.Range(25, 29);
            ScannedPlateIndex = Random.Range(0, 25);
        }
        else
        {
            ScannedPlateIndex = Random.Range(0, AllPlates.Count - 1); // possible bug here, ONLY relates to random number
        }

        randomPlate = AllPlates[ScannedPlateIndex];
        User.transform.position = randomPlate.transform.position;
        CurrentLocationText.text = "Current location: " + rooms[ScannedPlateIndex];
        //User.SetActive(true);
    }

    0 references
    public void OnGoButtonPressed()
    {
        if (WeStartOnFloorA && DestinationIsOnFloor0)
        {
            // make transition point our target
            TargetObject.transform.position = StaircaseTransitionA.position;
        }
        else if (!WeStartOnFloorA && !DestinationIsOnFloor0)
        {
            // make another transition point our target
            TargetObject.transform.position = StaircaseTransition0.position;
        }
        else if (WeStartOnFloorA && !DestinationIsOnFloor0)
        {
            // destination is on the same floor
            TargetObject.transform.position = AllPlates[DestinationPlateIndex].transform.position;
        }
        else if (!WeStartOnFloorA && DestinationIsOnFloor0)
        {
            // destination is on the same floor
            TargetObject.transform.position = AllPlates[DestinationPlateIndex].transform.position;
        }

        DestinationWindowUI.SetActive(false);
        User.SetActive(true);
    }
}

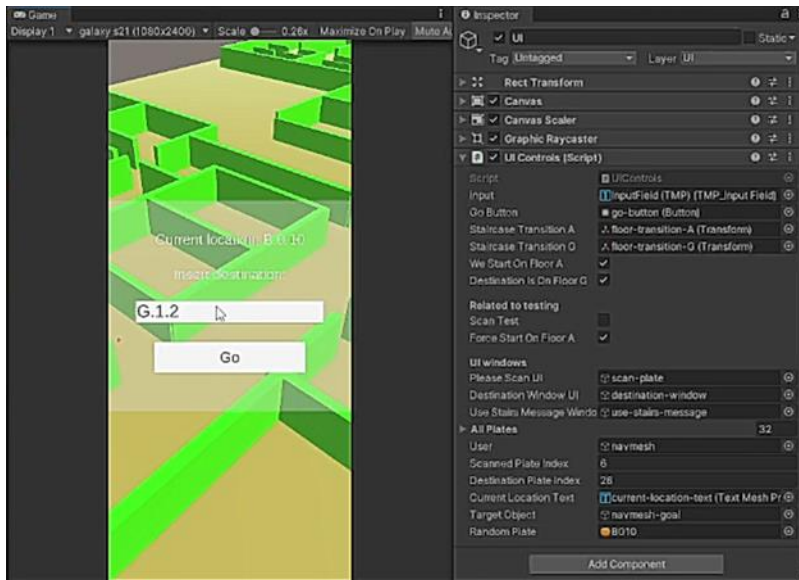
```

Εικόνα 6.43 UIControls.cs προσθήκη ForceStartOnFloorA για testing και αλλαγή του Target ανάλογα με την ανάγκη για χρήση σκάλας

Παράδειγμα:

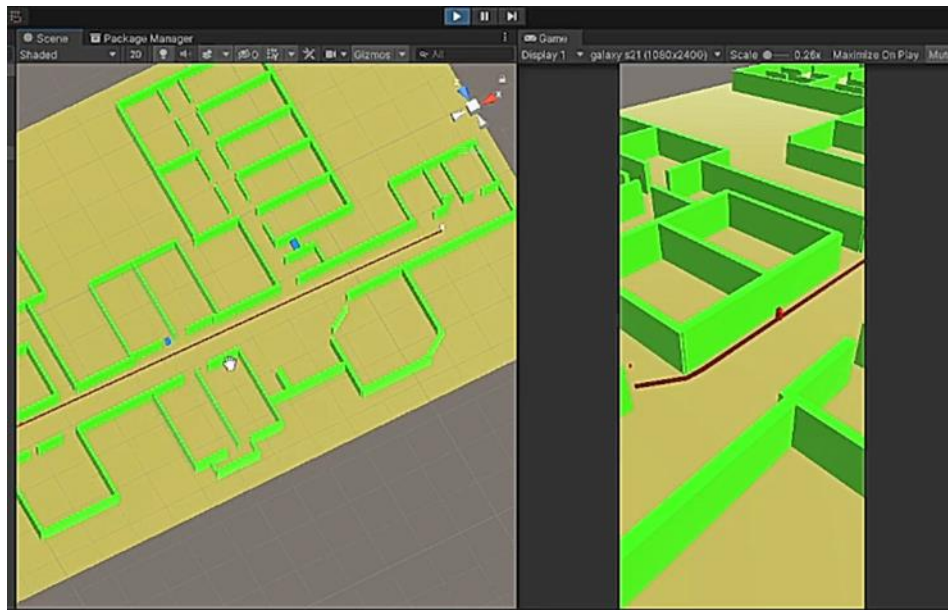
Πατώντας το κουτάκι του Scan Test η random επιλογή σκαναρισμένου Plate είναι η αίθουσα B.0.10 (έχουμε βάλει για το συγκεκριμένο run η random επιλογή να είναι πάντα κάποιο από τα

plates του ορόφου 0, ForceStartOnFloorA=True). Πληκτρολογούμε κάποιο από τα plates που βρίσκονται στον 1<sup>ο</sup> όροφο, το G.1.2 στην συγκεκριμένη περίπτωση. (βλ. Εικόνα 6.44)



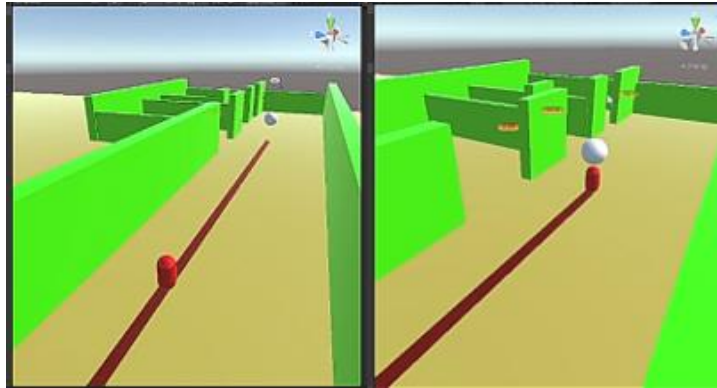
Εικόνα 6.44 Παράδειγμα για την λειτουργία της λογικής για τις σκάλες

Πατώντας το Go Button εμφανίζεται η διαδρομή όπως δείχνει η Εικόνα 6.45.



Εικόνα 6.45 Πατώντας το Go Button στο παράδειγμα για τις σκάλες

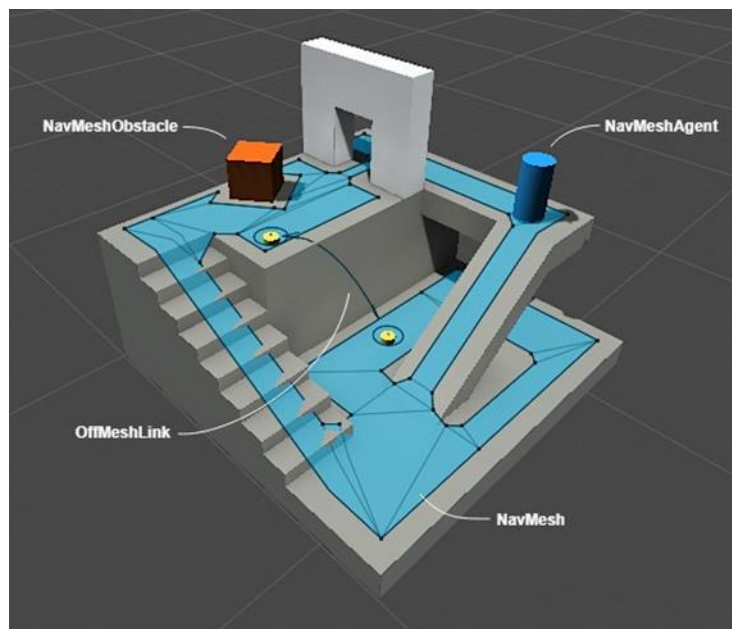
Όντως ο NavMesh Agent ακολουθεί την διαδρομή μέχρι να φτάσει στο floor-transition-A 3D Object. (βλ. Εικόνα 6.46)



Εικόνα 6.46 Ο NavMesh Agent ακολουθεί την διαδρομή μέχρι το floor-transition-A 3D Object

## 6.5.2 NavMesh για σκάλες

Η αρχική σκέψη για την αλλαγή ορόφων, όπως εξηγήσαμε και παραπάνω, ήταν να φτιάξουμε ένα OffMeshLink, όμως θα χανόταν το AR part της εφαρμογής μας για λίγο διάστημα πράγμα που δεν θέλαμε. Αντί αυτού και για τις σκάλες φτιάξαμε NavMesh όπως ακριβώς είχαμε κάνει και για το υπόλοιπο floor του κτηρίου. Όπως φαίνεται και στην Εικόνα 6.47 αυτό είναι εφικτό.



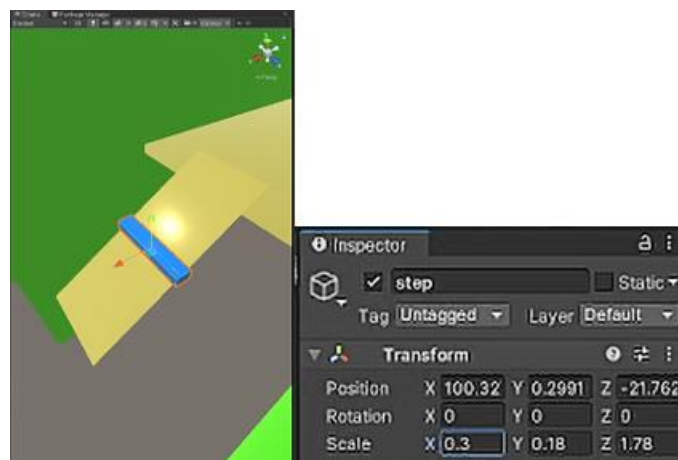
Εικόνα 6.47 Unity NavMesh parts

Για να το καταφέρουμε αυτό με ακρίβεια, μιας που το 3D Model είναι ακριβής replica του πραγματικού κτηρίου, μετρήσαμε τις διαστάσεις κάθε σκαλοπατιού και το πλήθος αυτών. Στο κτήριο μας έχουμε 2 σκάλες η μία δίπλα στην άλλη. Η μία οδηγεί στον όροφο -1 ο οποίος περιέχει την Γ πτέρυγα του ορόφου 0, και αποτελείται από 11 σκαλοπάτια. Και η άλλη οδηγεί στην Γ πτέρυγα του 1<sup>ου</sup> ορόφου, και αποτελείται από 12 σκαλοπάτια. Οι διαστάσεις των σκαλοπατιών και για τις δύο σκάλες φαίνονται στην Εικόνα 6.48.



Εικόνα 6.48 Διαστάσεις ενός σκαλοπατιού στις σκάλες της σχολής

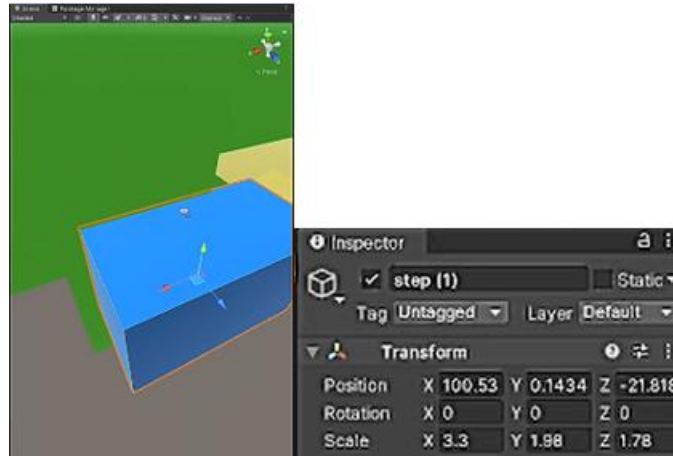
Άρα στην Unity για να φτιάξουμε το 3D Model της κάθε σκάλας, φτιάξαμε αρχικά έναν κύβο για το ένα σκαλοπάτι με τις διαστάσεις που πήραμε. (βλ. Εικόνα 6.49)



Εικόνα 6.49 Δεξιά το 3D model ενός σκαλοπατιού στην Unity και αριστερά οι διαστάσεις του

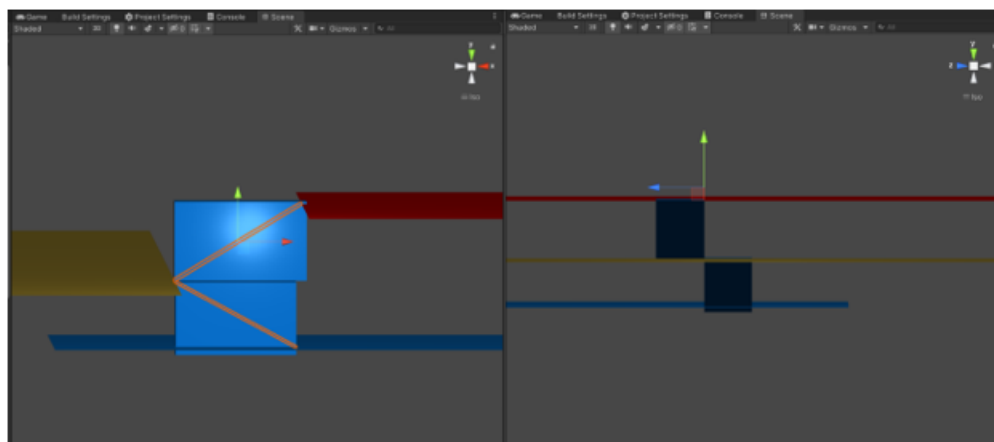
Πολλαπλασιάσαμε τις διαστάσεις των x και y με το πλήθος των σκαλοπατιών για να βρούμε το συνολικό μέγεθος της σκάλας. Π.χ. με \*11 για την πρώτη σκάλα. (βλ. Εικόνα 6.50)





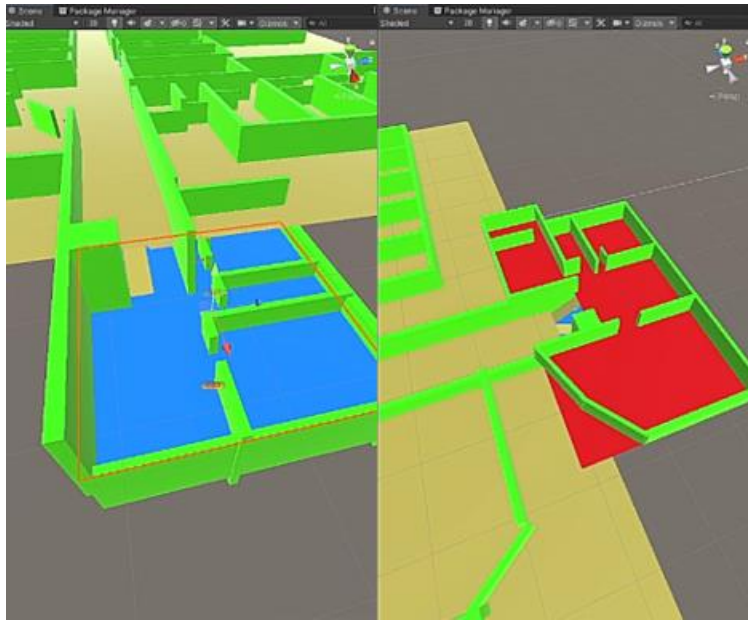
Εικόνα 6.50 Τελικές διαστάσεις σκάλας

Κάναμε το ίδιο και για τις δεύτερες σκάλες και φτιάξαμε ένα Plane για την κάθε σκάλα με μήκος ίσο με την διαγώνιο του cube. (βλ. Εικόνα 6.51)



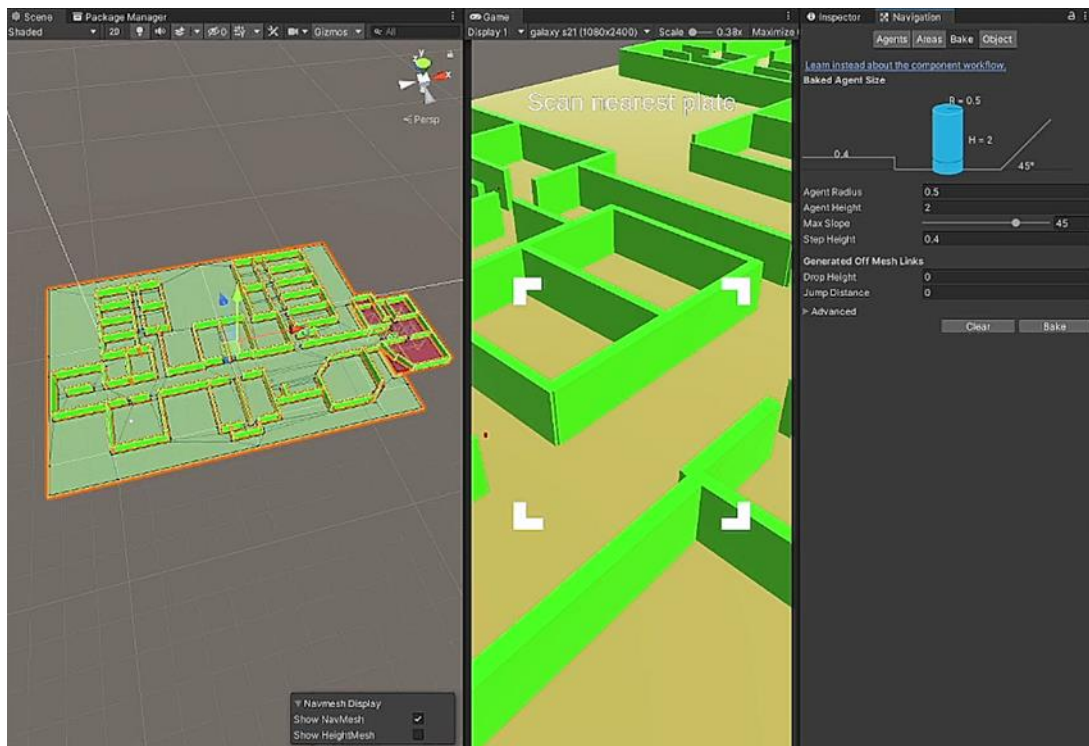
Εικόνα 6.51 Τα δύο Planes που αντιπροσωπεύουν τις σκάλες και η τοποθέτηση των τριών ορόφων. Όροφος 0 με κίτρινο χρώμα, όροφος 1 με κόκκινο και όροφος -1 με μπλε.

Μια καλύτερη όψη του τελικού 3d Model των τριών ορόφων της σχολής μαζί με τους τοίχους φαίνεται στην Εικόνα 6.52.



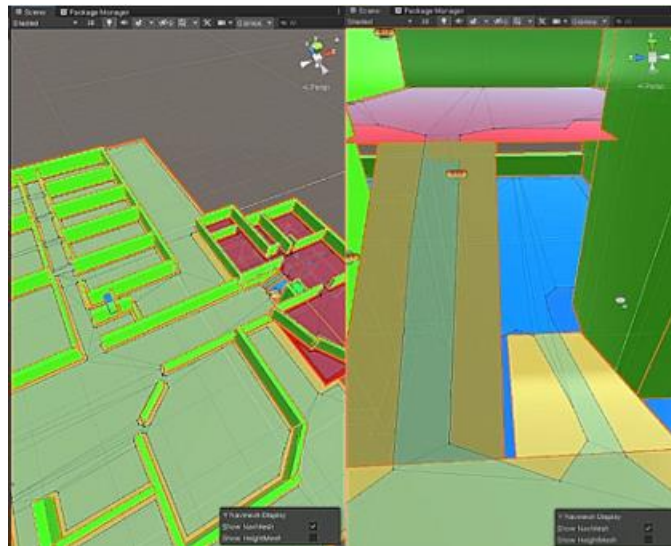
Εικόνα 6.52 3D model των τριών επιπέδων της σχολής

Τέλος, κάναμε bake για το navigation στην σκηνή μας μαζί με τις “σκάλες” αυτή τη φορά.(βλ. Εικόνα 6.53)



Εικόνα 6.53 Τελικό bake του 3d model μαζί με τις σκάλες

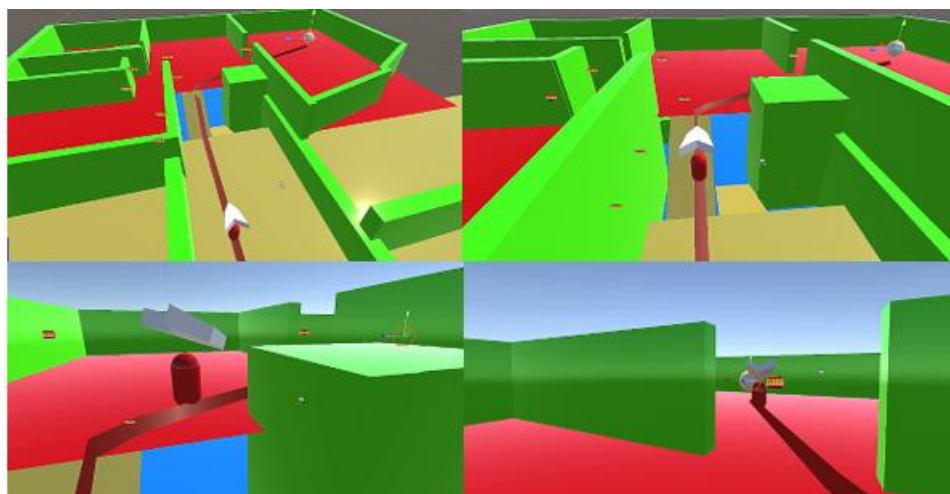
Walkable είναι όλα τα planes των floors, και non-walkable είναι οι τοίχοι. Μία πιο κοντινή εικόνα του baked mesh φαίνεται στην Εικόνα 6.54.



Εικόνα 6.54 Κοντινή εικόνα του baked mesh

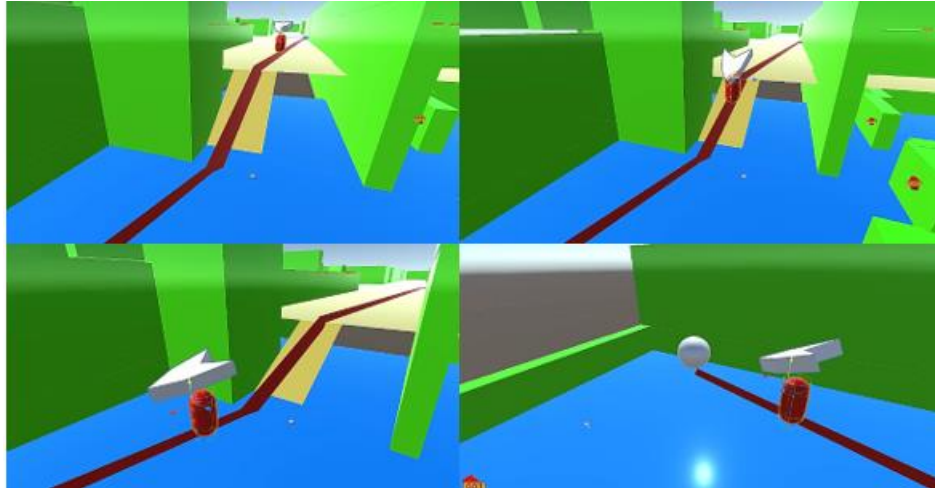
Το κομμάτι των floors που λαμβάνεται υπόψιν για να βρεθεί το NavMeshPath είναι αυτό μέσα στο πράσινο πλαίσιο. Κάναμε τεστ για να δούμε αν δουλεύει η μετάβαση στους δύο ορόφους με τις σκάλες.

Παράδειγμα για τον όροφο 1 φαίνεται στην Εικόνα 6.55.



Εικόνα 6.55 Λειτουργία εφαρμογής για μετάβαση στον όροφο 1

Παράδειγμα για τον όροφο -1 φαίνεται στην Εικόνα 6.56.

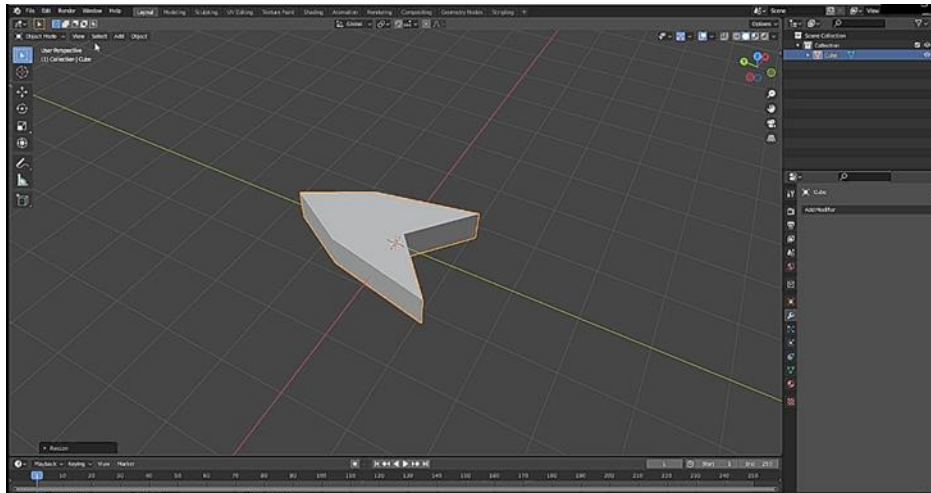


Εικόνα 6.56 Λειτουργία εφαρμογής για μετάβαση στον όροφο -1

## 6.6 Ενδείξεις AR

### 6.6.1 Arrow

Φτιάξαμε ένα βέλος (arrow) το οποίο θα βρίσκεται πάντα μπροστά στην οθόνη του χρήστη και θα τον καθοδηγεί προς την σωστή κατεύθυνση, όπως φαίνεται και στις Εικόνα 6.55 και Εικόνα 6.56. Το βέλος θα λειτουργεί συμπληρωματικά με την κόκκινη γραμμή που εμφανίζεται στην οθόνη και ενώνει την αρχική θέση του χρήστη με τον τελικό προορισμό. Η δημιουργία του βέλους έγινε στο blender (βλ.) και το κάναμε export ως .fbx file.



Εικόνα 6.57 Δημιουργία 3d βέλους στο blender

Για να καταφέρουμε το βέλος που φτιάξαμε στο blender να δείχνει πάντα στην κατεύθυνση που πρέπει να πάει ο χρήστης γράψαμε το ArrowLookAt.cs (βλ. Εικόνα 6.58).

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [Unity Script (1 asset reference)] 1 reference
6 public class ArrowLookAt : MonoBehaviour
7 {
8     public Vector3[] pathways;
9     public int index;
10    public float Threshold;
11
12    [Unity Message | 0 references]
13    void Update()
14    {
15        if(pathways.Length > 0)
16        {
17            transform.LookAt(pathways[index]);
18
19            if(Vector3.Distance(transform.position, pathways[index]) < Threshold && index + 1 != pathways.Length)
20            {
21                index++;
22            }
23        }
24    }

```

Εικόνα 6.58 ArrowLookAt.cs

Και προσθέσαμε στο ShowGoldenPath.cs (βλ. Εικόνα 6.59):

```
ArrowLookAt.pathways = path.corners;
```

```
ArrowLookAt.index = 0;
```

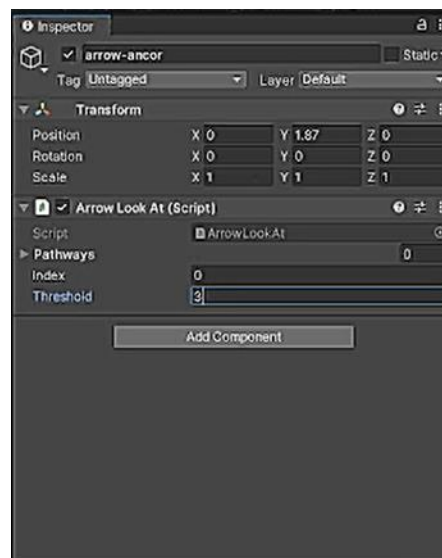
```

1  using UnityEngine;
2  using UnityEngine.AI;
3
4  public class ShowGoldenPath : MonoBehaviour
5  {
6      public Transform target;
7      private NavMeshPath path;
8      private float elapsed = 0.0f;
9      public NavMeshAgent NavMeshAgent;
10     public LineRenderer LineRenderer;
11
12     public ArrowLookAt ArrowLookAt;
13
14     void Start()
15     {
16         path = new NavMeshPath();
17         elapsed = 0.0f;
18         NavMeshAgent.destination = target.position;
19         NavMeshAgent.CalculatePath(NavMeshAgent.destination, path);
20         LineRenderer.positionCount = path.corners.Length;
21         LineRenderer.SetPositions(path.corners);
22
23         ArrowLookAt.pathways = path.corners;
24         ArrowLookAt.index = 0;
25     }
26
27 }

```

Εικόνα 6.59 ShowGoldenPath.cs update για το βέλος

Επιπλέον στο Hierarchy θέσαμε ως παιδί του navmesh ένα GameObject που ονομάσαμε arrow-anchor στο οποίο προσθήσαμε το ArrowLookAt.sc (βλ Εικόνα 6.60).



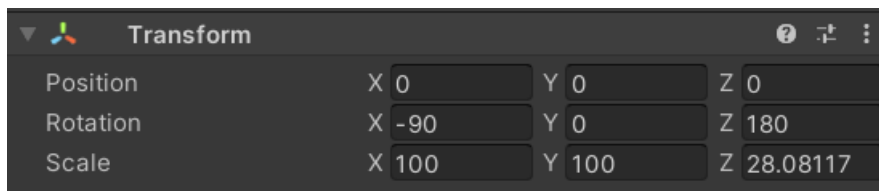
Εικόνα 6.60 Inspector για arrow-anchor

Παιδί του arrow-anchor, το blender αρχείο του arrow (βλ. Εικόνα 6.61).



Εικόνα 6.61 Hierarchy για το arrow

Αλλάξαμε το rotation του arrow στον Z σε 180 μοίρες για να δείχνει προς την σωστή κατεύθυνση (βλ. Εικόνα 6.62).



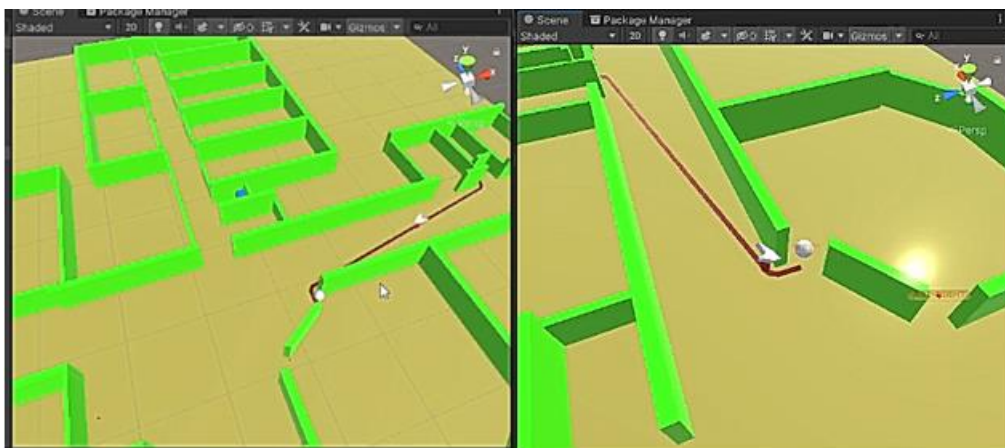
Εικόνα 6.62 Rotation του arrow

Ο Inspector του navmesh φαίνεται στην Εικόνα 6.63.



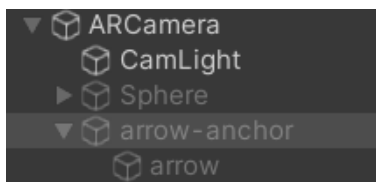
Εικόνα 6.63 Inspector του navmesh

Η επαλήθευση ότι το βέλος λειτουργεί σωστά φαίνεται στην Εικόνα 6.64.



Εικόνα 6.64 Παράδειγμα για την σωστή λειτουργία του βέλους

Στο τελικό build της εφαρμογής το βέλος προκειμένου να μένει πάντα μπροστά από την κάμερα του χρήστη είναι parented στην ARCamera (βλ. Εικόνα 6.65)

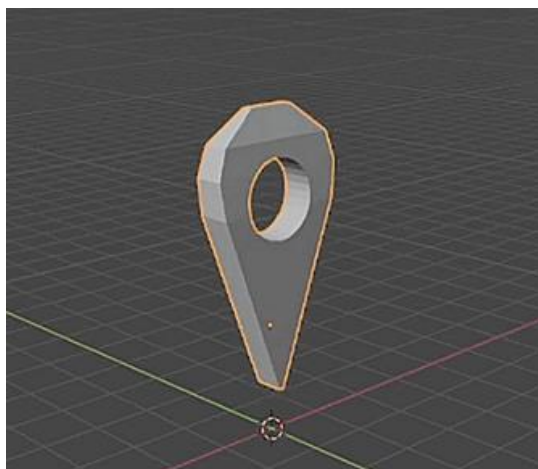


Εικόνα 6.65 Hierarchy τελικού build όπου το βέλος είναι parented στην ARCamera του χρήστη

## 6.6.2 Destination marker

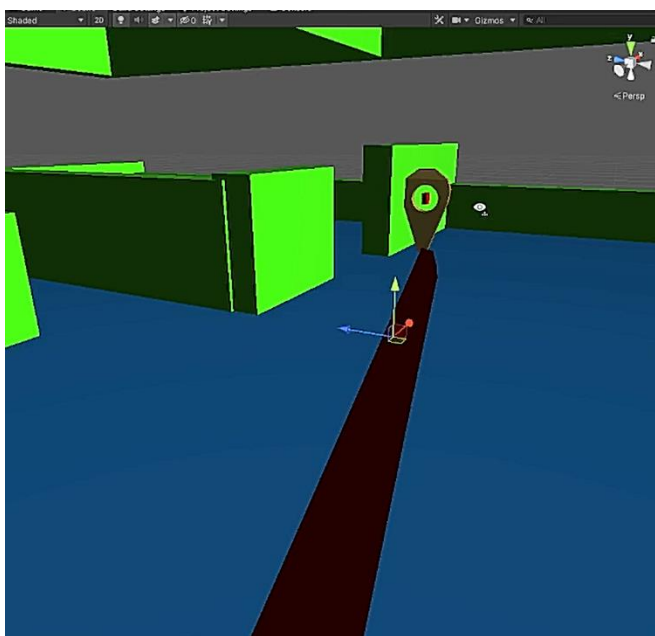
Φτιάξαμε ένα destination marker (pin) στο blender, το οποίο θα εμφανίζεται όταν φτάνουμε στον τελικό προορισμό, και το κάναμε export ως .fbx file. (βλ. Εικόνα 6.66)





Εικόνα 6.66 Δημιουργία destination marker στο blender

Στην Εικόνα 6.67 φαίνεται ο destination marker να εμφανίζεται στον τελικό προορισμό.



Εικόνα 6.67 Παράδειγμα για εμφάνιση του rip στον τελικό προορισμό

## 6.7 UI

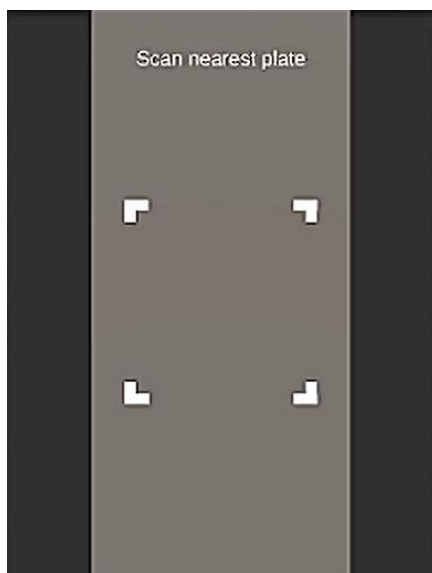
### 6.7.1 UI Scan Plate

Το hierarchy του UI για το σκανάρισμα του κοντινότερου plate στον χρήστη φαίνεται στην Εικόνα 6.68.



Εικόνα 6.68 Hierarchy για UI του scan plate

Το UI για το scan nearest plate όπου το πλαίσιο που σχηματίζεται καθορίζει τα όρια μέσα στα οποία ο χρήστης πρέπει να τοποθετήσει το plate που σκανάρει φαίνεται στην Εικόνα 6.69.



Εικόνα 6.69 UI για scan plate

Στην Εικόνα 6.70 φαίνεται το κομμάτι του UIControls.cs για το σκανάρισμα.

```

public void OnImageScanned(GameObject img)
{
    CurrentLocationText.text = "Current location: " + img.GetComponent<ImageTargetBehaviour>().ImageTarget.Name;

    //disable all trackable images
    ImageTargetBehaviour[] itbs = FindObjectsOfType<ImageTargetBehaviour>();
    foreach (var t in itbs)
    {
        t.enabled = false;
    }
    PleaseScanUI.SetActive(false);
    DestinationWindowUI.SetActive(true);
    User.transform.position = img.transform.position;
}

```

Εικόνα 6.70 UIControls.cs για το Image Recognition

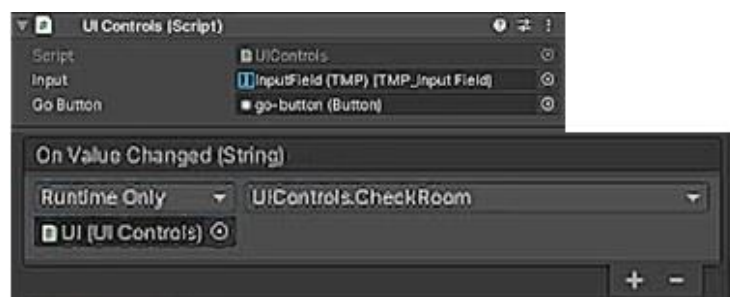
## 6.7.2 UI Πληκτρολόγησης Τελικού Προορισμού

Για το UI της επιλογής του τελικού προορισμού ακολουθήσαμε παρόμοια διαδικασία με την κατασκευή του UI για το σκανάρισμα του πλησιέστερου plate. Το Hierarchy φαίνεται στην Εικόνα 6.71.



Εικόνα 6.71 Hierarchy για το UI της επιλογής τελικού προορισμού

Κάναμε αλλαγές στον Inspector του UI (βλ. Εικόνα 6.72 πάνω) και του InputField(TMP) (βλ. Εικόνα 6.72 κάτω).



Εικόνα 6.72 Πάνω: Inspector του UI, κάτω: Inspector του InputField(TMP)

Το check για την εισαγωγή επιτρεπτής εισόδου στο InputField γίνεται στο UIControls.cs (βλ. Εικόνα 6.73).

```
public void CheckRoom()
{
    string temp = input.text;
    for (int i = 0; i < rooms.Length; i++)
    {
        //Debug.Log(input.text);
        if (rooms[i] == temp)
        {
            DestinationPlateIndex = i;
            GoButton.interactable = true;
            //stops the loop
            i = rooms.Length;
        }
        else
        {
            GoButton.interactable = false;
        }
    }
}
```

Εικόνα 6.73 UIControls.cs check σωστής εισόδου προορισμού

Όπου το Input Text συγκρίνεται με τα περιεχόμενα του array rooms (βλ. Εικόνα 6.74).

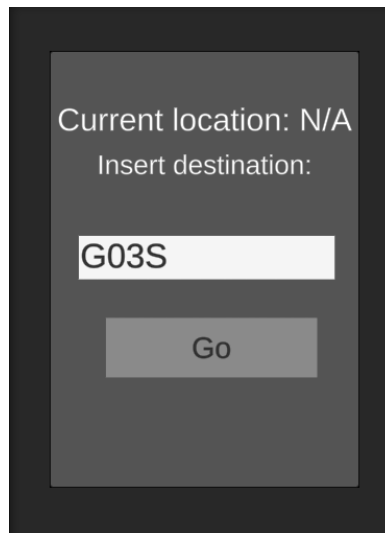
```

public class UIControls : MonoBehaviour
{
    string[] rooms = new string[]
    {
        "B04", // 0
        "B03",
        "B02",
        "B01",
        "B05",
        "B06", // 5
        "B010",
        "B09B",
        "B09A",
        "A04",
        "A05", // 10
        "A03",
        "A02",
        "A011",
        "A010",
        "A012", // 15
        "A013",
        "A014",
        "A015",
        "A016",
        "A017", // 20
        "A01",
        "G05",
        "G04",
        "G03",
        // the ones below are on the G.1.x floor
        "G11", // 25
        "G12",
        "G13",
        "G14"
    };
}

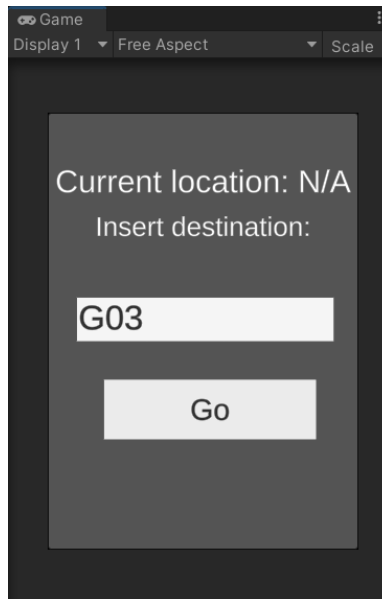
```

Εικόνα 6.74 UIControls.cs τα επιτρεπτά ονόματα δωματίων που ο χρήστης μπορεί να πληκτρολογήσει

Η Εικόνα 6.75 δείχνει ανενεργό το Go button στην οθόνη του χρήστη, αφού έχει πληκτρολογήσει μη αποδεκτή είσοδο στο InputField. Η Εικόνα 6.76 δείχνει ενεργό το Go button αφού ο χρήστης έχει βάλει αποδεκτή είσοδο στο InputField.



Εικόνα 6.75 Μη επιτρεπτή είσοδος στο InputField, Go button ανενεργό



Εικόνα 6.76 Επιτρεπτή είσοδος στο InputField, Go button ενεργό

## 6.8 Scripts

Τα τελικά scripts που κάνουν και κάτι μέσα στην εφαρμογή είναι 3, τα υπόλοιπα είναι είτε για testing, είτε trash. Τα 3 αυτά scripts παρατίθενται παρακάτω:

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

public class ArrowLookAt : MonoBehaviour
{
    public Vector3[] pathways;
    public int index;
    public float Threshold;

    void Update()
    {
        if(pathways.Length > 0)
        {
            transform.LookAt(pathways[index]);

            if(Vector3.Distance(transform.position, pathways[index]) < Threshold && index + 1 != pathways.Length)
            {
                index++;
            }
        }
    }
}
```

Εικόνα 6.77 ArrowLookAt.cs

To *ArrowLookAt.cs* (βλ. Εικόνα 6.77):

- Points the arrow towards the next point in the path specified by *ShowGoldenPath*

Όταν ο χρήστης έρχεται αρκετά κοντά στο σημείο, παίρνει το επόμενο σημείο από την λίστα και δείχνει προς εκείνο κ.ο.κ.

```
using UnityEngine;
using UnityEngine.AI;

public class ShowGoldenPath : MonoBehaviour
{
    public Transform target;
    private NavMeshPath path;
    private float elapsed = 0.0f;
    public NavMeshAgent NavMeshAgent;
    public LineRenderer LineRenderer;

    public ArrowLookAt ArrowLookAt;

    void Start()
    {
        path = new NavMeshPath();
        elapsed = 0.0f;
        NavMeshAgent.destination = target.position;
        NavMeshAgent.CalculatePath(NavMeshAgent.destination, path);
        LineRenderer.positionCount = path.corners.Length;
        LineRenderer.SetPositions(path.corners);

        ArrowLookAt.pathways = path.corners;
        ArrowLookAt.index = 0;
    }
}
```

Εικόνα 6.78 *ShowGoldenPath.cs*

To *ShowGoldenPath.cs* (βλ. Εικόνα 6.78) :

- Sets NavMesh destination to the target object (geolocation 3D model)
- Calculates NavMesh path
- Places LineRenderer along the calculated NavMesh path
- Takes the arrow (*ArrowLookAt.cs*) and send it the list of all path points from the calculated path

```

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

public class UIControls : MonoBehaviour
{
    string[] rooms = new string[]
    {
        "B04", // 0
        "B03",
        "B02",
        "B01",
        "B05",
        "B06", // 5
        "B010",
        "B09B",
        "B09A",
        "A04",
        "A05", // 10
        "A03",
        "A02",
        "A011",
        "A010",
        "A012", // 15
        "A013",
        "A014",
        "A015",
        "A016",
        "A017", // 20
        "A01",
        "G05",
        "G04",
        "G03",
        // the ones below are on the G.1.x floor
        "G11", // 25
        "G12",
        "G13",
        "G14"
    };

    [Header("Related to testing")]
    public bool ScanTest;
    public bool ForceStartOnFloorA;
    public bool WeStartOnFloorA, DestinationIsOnFloorG; // for testing only

    public string ChosenDestinationName;

    [Header("UI windows")]
    public GameObject PleaseScanUI;
    public GameObject DestinationWindowUI;
    public GameObject UseStairsMessageWindowUI;
    public TMP_InputField input;
    public Button GoButton;

    public GameObject User;
    public int ScannedPlateIndex, DestinationPlateIndex;
    public TextMeshProUGUI CurrentLocationText;
    public GameObject TargetObject;
    public GameObject randomPlate;

    // to activate when "go" is pressed
    public GameObject arrowObject;

    private void Update()
    {
        // for testing in editor
        if (ScanTest)
        {
            ScanTest = false;
            StartFromRandomPlate();
        }
    }
}

```



```

public void CheckRoom()
{
    string temp = input.text;
    for (int i = 0; i < rooms.Length; i++)
    {
        if(rooms[i] == temp)
        {
            ChosenDestinationName = rooms[i];

            DestinationPlateIndex = i;

            GoButton.interactable = true;

            // for testing:
            if(ScannedPlateIndex < 25)
            {
                // we start on floor A.0.x
                WeStartOnFloorA = true;
            }
            else
            {
                // we start on floor G.1.x
                WeStartOnFloorA = false;
            }

            // for testing:
            if (temp == "G11" || temp == "G12" || temp == "G13" || temp == "G14")
            {
                // our destination is on the G.1.x floor
                DestinationIsOnFloorG = true;
            }
            else
            {
                // our destination is on the A.0.x floor
                DestinationIsOnFloorG = false;
            }

            // stop the loop, should be the last line!
            i = rooms.Length;
        }
        else
        {
            GoButton.interactable = false;
        }
    }
}

public void OnImageScanned(GameObject img)
{
    CurrentLocationText.text = "Current location: " + img.GetComponent<ImageTargetBehaviour>().ImageTarget.Name;

    //disable all trackable images
    ImageTargetBehaviour[] itbs = FindObjectsOfType<ImageTargetBehaviour>();
    foreach (var t in itbs)
    {
        t.enabled = false;
    }
    PleaseScanUI.SetActive(false);
    DestinationWindowUI.SetActive(true);
    User.transform.position = img.transform.position;
}

// for testing only
public void StartFromRandomPlate()
{
    if (ForceStartOnFloorA)
    {
        ScannedPlateIndex = Random.Range(0, 25);
    }
    else
    {
        ScannedPlateIndex = Random.Range(0, rooms.Length - 1);
    }

    randomPlate = GameObject.Find(rooms[ScannedPlateIndex]);
    User.transform.position = randomPlate.transform.position;
    CurrentLocationText.text = "Current location: " + rooms[ScannedPlateIndex];
    PleaseScanUI.SetActive(false);
}

```

```

        DestinationWindowUI.SetActive(true);
    }

    public void OnGoButtonPressed()
    {
        Transform newTarget = GameObject.Find(ChosenDestinationName).transform;
        User.GetComponent<ShowGoldenPath>().target.transform.position = newTarget.position;
        DestinationWindowUI.SetActive(false);
        User.SetActive(true);
        arrowObject.SetActive(true);
    }
}

```

Εικόνα 6.79 *UIControls.cs*

Το *UIControls.cs* (βλ. Εικόνα 6.79):

- Displays the room number as text in the UI
- Disables other “building objects” as soon as any image is scanned
- Enables all other UIs
- Places NavMesh at the proper start location
- Checks that the text that the user typed has an actual corresponding room, then enables the ‘Go’ button
- Enables other scripts (that do the rest of calculations and handle arrow rotation)
- Has all methods for testing navigation and UI directly in the editor

# **III ΕΠΙΛΟΓΟΣ**

---

# 7 Επίλογος

## 7.1 Σύνοψη Λειτουργίας Εφαρμογής

Για να γίνει κατανοητή η εφαρμογή μας παρουσιάζουμε στο κεφάλαιο αυτό συνοπτικά την λειτουργία της. Αρχικά κάθε φωτογραφία plate περικόπτεται και τοποθετείται στον τρισδιάστατο κόσμο στην ίδια θέση και με τον ίδιο προσανατολισμό όπως στον πραγματικό κόσμο. Κάθε πλάκα έχει ένα τρισδιάστατο αντίγραφο του κτηρίου ως child object, οπότε όταν κινείται στο AR περιβάλλον, ολόκληρο το τρισδιάστατο κτήριο κινείται μαζί του. Αυτό σημαίνει ότι υπάρχουν πολλά αόρατα τρισδιάστατα κτήρια στη σκηνή όταν ανοίγει. Αλλά όταν μια εικόνα σαρωθεί με επιτυχία, η αναγνώριση εικόνας (image recognition) διακόπτεται και αντ' αυτού πραγματοποιείται image tracking.

Μόλις σαρωθεί το plate, εμφανίζεται ένας διάλογος διεπαφής χρήστη (UI):

- Κείμενο που επιβεβαιώνει την τοποθεσία του χρήστη (εδώ ο χρήστης μπορεί να ελέγξει αν το δωμάτιο είναι το σωστό).
- Πεδίο εισαγωγής όπου ο χρήστης μπορεί να εισάγει τον προορισμό του (πρέπει να είναι αριθμοί δωματίων και κεφαλαία λατινικά γράμματα, χωρίς τελείες).
- Το κουμπί "Go" που είναι ανενεργό έως ότου ο χρήστης εισάγει έγκυρη είσοδο (όνομα δωματίου που υπάρχει στην πραγματικότητα).

Εάν ο χρήστης πληκτρολογήσει έγκυρο δωμάτιο, το κουμπί "Go" γίνεται ενεργό. Πατώντας το κουμπί τοποθετείται ένας NavMesh agent στη θέση του plate που σαρώθηκε, ενώ το NavMesh target τοποθετείται στον προορισμό. Υπολογίζεται η διαδρομή πλοήγησης χρησιμοποιώντας το NavMeshPath και τοποθετείται ένα line για να την οπτικοποιήσουμε.

Τέλος, ενεργοποιείται το τρισδιάστατο βέλος και του "στέλνονται" τα navigation path points που έχουν δημιουργηθεί ώστε να τα ακολουθήσει. Το βέλος δείχνει στο πρώτο navigation path point και αλλάζει στο επόμενο από αυτό μόλις ο χρήστης το πλησιάσει αρκετά κοντά.

## 7.2 Προκλήσεις και Περιορισμοί

Διάφορες προκλήσεις και περιορισμοί που αφορούν στην ανάπτυξη αλλά και στην λειτουργία της εφαρμογής παρατίθενται παρακάτω:

- Είναι πολύ σημαντικό το scale του 3d model του κτηρίου να αντικατοπτρίζει την πραγματικότητα με μεγάλη ακρίβεια, δεν θα πρέπει να είναι μικρότερο ή μεγαλύτερο από το πραγματικό περιβάλλον.
- Το Image Recognition δεν λειτουργεί σε περιβάλλοντα χαμηλού φωτισμού ή τα βράδια.
- Εφόσον το Image Tracking ελέγχει όχι μόνο τη θέση οποιασδήποτε σκαναρισμένης εικόνας, αλλά και τον προσανατολισμό της (rotation), μερικές φορές ολόκληρο το περιβάλλον λαμβάνεται περιστραμένο ελαφρώς, γεγονός που παράγει περίεργα αποτελέσματα στην πλοήγηση (π.χ. μονοπάτι που οδηγεί προς τα κάτω ή προς τα πάνω). Αυτό οφείλεται σε ένα μικρό σφάλμα στην τεχνολογία του Image Tracking.
- Μερικές φορές, το Image Recognition μπερδεύει το ένα plate με το άλλο επειδή οι πλάκες μοιάζουν πολύ μεταξύ τους (π.χ. A.0.3 και A.0.4), με αποτέλεσμα μια λανθασμένη τοποθέτηση "κόσμου" στην πλοήγηση.
- Για το build της εφαρμογής, είναι σημαντικό όλα τα image objects με τα κτήρια να διατηρούνται ενεργά (A01, A02, A03, κ.λπ.).
- Για να δοκιμάσουμε την εφαρμογή στον editor της Unity, απενεργοποιούμε αυτά τα objects και δοκιμάζουμε την πλοήγηση σε ένα προκατασκευασμένο δάπεδο («floor-A-new» στην περίπτωση μας). Αυτό το floor prefab θα πρέπει να είναι ανενεργό στο πραγματικό build της εφαρμογής.
- Unity version που χρησιμοποιήθηκε: 2020.3.12f1.
- Tested on Samsung Galaxy A7 (2018).
- Video με την χρήση της εφαρμογής στον χώρο της σχολής υπάρχουν στους παρακάτω συνδέσμους:

<https://youtu.be/4TIH6NiiIQI> (με εναλλαγή ορόφων)

<https://youtu.be/5uy7Cr2L8cU> (χωρίς εναλλαγή ορόφων)

## Βιβλιογραφία

- [1] Liu, J.; Chen, R.; Pei, L.; Guinness, R.; Kuusniemi, H. A hybrid smartphone indoor positioning solution for mobile LBS. *Sensors* 2012, 12, 17208–17233.
- [2] Abbas, M., Elhamshary, M., Rizk, H., Torki, M., Youssef, M.: WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1– 10 (2019).
- [3] Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M., Heo, J.: Application of WiFi-based Indoor Positioning System for Labor Tracking at Construction sites: A Case Study in Guangzhou MTR. *Autom. Constr.* 20(1), 3–13 (2011).
- [4] Calderoni, L., Ferrara, M., Franco, A., Maio, D.: Indoor Localization in a Hospital Environment using Random Forest Classifiers. *Expert Syst. Appl.* 42(1), 125–134 (2015).
- [5] Lu, G., Yan, Y., Ren, L., Saponaro, P., Sebe, N., Kambhamettu, C.: Where am I in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. *Neurocomputing* 173, 83–92 (2016). <https://doi.org/10.1016/j.neucom.2015.07.106>
- [6] Farid, Z., Nordin, R., Ismail, M., Abdullah, N.F.: Hybrid Indoorbased WLAN-WSN localization scheme for improving accuracy based on artificial neural network. *Mob. Inf. Syst.* 2016, 1–11 (2016). <https://doi.org/10.1155/2016/6923931>
- [7] Zou, H., Huang, B., Lu, X., Jiang, H., Xie, L.: A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine. *IEEE Trans. Wireless Commun.* 15(2), 1252–1266 (2016).
- [8] Poggi, M., Mattoccia, S.: A wearable mobility aid for the visually impaired based on embedded 3d vision and deep learning. In: 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 208–213. IEEE (2016).
- [9] Shenoy, M.V., Karuppiah, A., Manjarekar, N.: A Lightweight ANN based Robust Localization Technique for Rapid Deployment of Autonomous Systems. *J. Ambient. Intell. Humaniz. Comput.* 1–16. <https://doi.org/10.1007/s12652-019-01331-0> (2019)
- [10] Emanuel, D.C., Mahan, L.G., Ungerbuehler, R.H.: Apparatus and method for asset tracking. US Patent 8,565,913 (2013)
- [11] Zou, H., Wang, H., Xie, L., Jia, Q.S.: An RFID Indoor positioning system by using weighted path loss and extreme learning machine. In: 2013 IEEE 1St International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), pp. 66–71. IEEE (2013).
- [12] Ma, H., Wang, K.: Fusion of RSS and Phase Shift using the Kalman Filter for RFID Tracking. *IEEE Sensors J.* 17(11), 3551–3558 (2017).
- [13] Yang, Z., Zhang, P., Chen, L.: RFID-Enabled Indoor Positioning Method for a Real-time Manufacturing Execution System using OS-ELM. *Neurocomputing* 174, 121–133 (2016).
- [14] Loffler, C., Riechel, S., Fischer, J., Mutschler, C.: Evaluation " criteria for inside-out indoor positioning systems based on machine learning. In: 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-8. IEEE (2018).
- [15] Ma, H., Wang, Y., Wang, K.: Automatic detection of false positive RFID readings using machine learning algorithms. *Expert Syst. Appl.* 91, 442–451 (2018).
- [16] <https://www.telegraph.co.uk/finance/newsbysector/retailandconsumer/11858755/Trafford-Centre-owner-unveils-indoor-mapping-and-deals-app.html>

- [17] Vasile, V.C.; Nafornita, C.; Borda, M. Comparative Study of Satellite Navigation Systems. In Proceedings of the 2018 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 8–9 November 2018; pp. 1–6.
- [18] Basiri, A.; Lohan, E.; Moore, T.; Winstanley, A.; Peltola, P.; Hill, C.; Amirian, P.; Figueiredo e Silva, P. Indoor location based services challenges, requirements and usability of current solutions. *Comput. Sci. Rev.* 2017, 24, 1–12.
- [19] Elmezayen, A.; El-Rabbany, A. Precise Point Positioning Using World’s First Dual-Frequency GPS/GALILEO Smartphone. *Sensors* 2019, 19, 2593.
- [20] Puricer, P.; Kovar, P. Technical Limitations of GNSS Receivers in Indoor Positioning. In Proceedings of the 17th International Conference Radioelektronika, Brno, Czech Republic, 24–25 April 2007; pp. 1–5.
- [21] Zafari, F.; Gkelias, A.; Leung, K.: A survey of indoor localization systems and technologies. *IEEE Commun. Surveys Tutor.* 21(3), 2568–2599 (2017).
- [22] Huang, H.; Gartner, G. A survey of mobile indoor navigation systems. In *Cartography in Central and Eastern Europe*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 305–319.
- [23] Sato, D.; Oh, U.; Guerreiro, J.; Ahmetovic, D.; Naito, K.; Takagi, H.; Kitani, K.M.; Asakawa, C. NavCog3 in the Wild: Large-scale Blind Indoor Navigation Assistant with Semantic Features. *ACM Trans. Access. Comput.* 2019, 12, 1–30.
- [24] Chidsin, W.; Gu, Y.; Goncharenko, I. AR-Based Navigation Using RGB-D Camera and Hybrid Map. *Sustainability* 2021, 13, 5585.
- [25] Schmalstieg, D.; Reitmayr, G. The World as a User Interface: Augmented Reality for Ubiquitous Computing. In *Location Based Services and TeleCartography*; Gartner, G., Cartwright, W.E., Peterson, M.P., Eds.; Lecture Notes in Geoinformation and Cartography; Springer: Berlin/Heidelberg, Germany, 2007; pp. 369–391.
- [26] Belmonte-Fernández, Ó.; Puertas-Cabedo, A.; Torres-Sospedra, J.; Montoliu-Colás, R.; Trilles-Oliver, S. An indoor positioning system based on wearables for ambient-assisted living. *Sensors* 2017, 17, 36.
- [27] Karmacharya, A.; Mendoza-Silva, G.M.; Torres-Sospedra, J. Sensor Fusion and Well Condition Triangle Approach for BLE-based Indoor Positioning. In Proceedings of the Tenth International Conference on Localization and GNSS (ICL-GNSS-WIP 2020) Co-Located with the Tenth International Conference on Indoor Positioning and Indoor Navigation (ICL-GNSS 2020), Tampere, Finland, 2–4 June 2020.
- [28] Ahmetovic, D.; Murata, M.; Gleason, C.; Brady, E.; Takagi, H.; Kitani, K.; Asakawa, C. Achieving Practical and Accurate Indoor Navigation for People with Visual Impairments. In Proceedings of the 14th Web for All Conference on the Future of Accessible Work; Association for Computing Machinery: New York, NY, USA, 2017.
- [29] Karmacharya, A.; Mendoza-Silva, G.M.; Torres-Sospedra, J. Sensor Fusion and Well Condition Triangle Approach for BLE-based Indoor Positioning. In Proceedings of the Tenth International Conference on Localization and GNSS (ICL-GNSS-WIP 2020) Co-Located with the Tenth International Conference on Indoor Positioning and Indoor Navigation (ICL-GNSS 2020), Tampere, Finland, 2–4 June 2020.
- [30] Moreira, A.; Meneses, F. Where@UM-Dependable organic radio maps. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–9.

- [31] Noh, Y.; Yamaguchi, H.; Lee, U. Infrastructure-Free Collaborative Indoor Positioning Scheme for Time-Critical Team Operations. *IEEE Trans. Syst. Man Cybern. Syst.* 2018, 48, 418–432.
- [32] Merry, L.A.; Faragher, R.M.; Scheduling, S. Comparison of Opportunistic Signals for Localisation. *IFAC Proc. Vol.* 2010, 43, 109–114.
- [33] Keskin, M.; Erdem, O.; Gezici, S. Cooperative Localization in Hybrid Infrared/Visible Light Networks: Theoretical Limits and Distributed Algorithms. *IEEE Trans. Signal Inf. Process. Netw.* 2019, 5, 181–197.
- [34] Wymeersch, H.; Lien, J.; Win, M.Z. Cooperative Localization in Wireless Networks. *Proc. IEEE* 2009, 97, 427–450.
- [35] Gallagher, T.; Wise, E.; Li, B.; Dempster, A.G.; Rizos, C.; Ramsey-Stewart, E. Indoor positioning system based on sensor fusion for the Blind and Visually Impaired. In *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, 13–15 November 2012; pp. 1–9.
- [36] Zekavat, R.; Buehrer, R.M. Collaborative Position Location. In *Handbook of Position Location: Theory, Practice, and Advances*; IEEE: New York, NY, USA, 2019; pp. 885–941.
- [37] Fadzilla, M.A.; Harun, A.; Shahrman, A.B. Localization Assessment for Asset Tracking Deployment by Comparing an Indoor Localization System with a Possible Outdoor Localization System. In *Proceedings of the 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, Kuching, Malaysia, 15–17 August 2018; pp. 1–6.
- [38] Mendoza-Silva, G.M.; Torres-Sospedra, J.; Huerta, J. A Meta-Review of Indoor Positioning Systems. *Sensors* 2019, 19, 4507.
- [39] Khan, L.U. Visible light communication: Applications, architecture, standardization and research challenges. *Digit. Commun. Netw.* 2017, 3, 78–88.
- [40] Lovon-Melgarejo, J.; Castillo-Cara, M.; Huarcaya-Canal, O.; Orozco-Barbosa, L.; Garcia-Varea, I. Comparative Study of Supervised Learning and Metaheuristic Algorithms for the Development of Bluetooth-Based Indoor Localization Mechanisms. *IEEE Access* 2019, 7, 26123–26135.
- [41] M. B. Kjær, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, and K. Grønbæk, “Indoor Positioning Using GPS Revisited,” in *Pervasive 2010*, 2010, pp. 38–56.
- [42] D. Macias-Valadez, R. Santerre, S. Larochele, and R. Landry, “Improving vertical GPS precision with a GPS-over-fiber architecture and real-time relative delay calibration,” *GPS Solutions*, vol. 16, no. 4, pp. 449–462, 2012.
- [43] S. He and S. H. G. Chan, “Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.
- [44] C. Yang and H. R. Shao, “WiFi-based indoor positioning,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.
- [45] Y. C. Pu and P. C. You, “Indoor positioning system based on BLE location fingerprinting with classification approach,” *Applied Mathematical Modelling*, 2018.
- [46] C. Zhou, J. Yuan, H. Liu, and J. Qiu, “Bluetooth Indoor Positioning Based on RSSI and Kalman Filter,” *Wireless Personal Communications*, vol. 96, no. 3, pp. 4115–4130, 2017.



- [47] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.
- [48] W. Shao et al., "Location Fingerprint Extraction for Magnetic Field Magnitude Based Indoor Positioning," *Journal of Sensors*, vol. 2016, 2016.
- [49] G. Kim and E. M. Petriu, "Fiducial marker indoor localization with Artificial Neural Network," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 961–966, 2010.
- [50] I. Igo Barandiaran, C. Paloc, and M. Grañ, "Real-time optical markerless tracking for augmented reality applications."
- [51] K.-W. Chen et al., "To Know Where We Are: Vision-Based Positioning in Outdoor Environments."
- [52] H. Subakti and J. R. Jiang, "A marker-based cyber-physical augmentedreality indoor guidance system for smart campuses," *Proceedings - 18<sup>th</sup> IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, pp. 1373–1379, 2017.
- [53] B. Hartmann, N. Link, and G. F. Trommer, "Indoor 3D position estimation using low-cost inertial sensors and marker-based videotracking," *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 319–326, 2010.
- [54] J. P. Grisales Campeon, S. Lopez, S. R. De Jesus Melean, H. Moldovan, D. R. Parisi, and P. I. Fierens, "Fusion of magnetic and WiFi fingerprints for indoor positioning," in *Congreso Argentino de Ciencias de la Informatica y Desarrollos de Investigacion, CACIDI 2018*, 2018.
- [55] E. Wang, M. Wang, Z. Meng, and X. Xu, "A Study of WiFi-Aided Magnetic Matching Indoor Positioning Algorithm," *Journal of Computer and Communications*, vol. 05, no. 03, pp. 91–101, 2017.
- [56] Yang, Z., Wu, C., Zhou, Z., Zhang, X., Wang, X., Liu, Y.: Mobility Increases localizability: A Survey on Wireless Indoor Localization using Inertial Sensors. *ACM Computing Surveys (CSUR)* 47(3), 54 (2015).
- [57] Zou, H., Lu, X., Jiang, H., Xie, L.: A fast and precise indoor localization algorithm based on an online sequential extreme learning machine. *Sensors* 15(1), 1804–1824 (2015).
- [58] J. B. Link, P. Smith, N. Viol and K. Wehrle, "Footpath: Accurate map-based indoor navigation using smartphones", *Indoor Positioning and Indoor Navigation (IPIN) International Conference IEEE*, pp. 1-8, September 21-23, 2011.
- [59] N. Ravi, P. Shankar, A. Frankel, A. Elgammal and L. Iftode, "Indoor localization using camera phones In *Mobile Computing Systems and Applications*", *WMCSA '06*, pp. 1-7, August 2005.
- [60] T. H. Kolbe, "Augmented videos and panoramas for pedestrian navigation", *2nd Symposium Location Based Services and TeleCartography*, January 28-29, 2004.
- [61] L. Liu and S. Zlatanova, "A "door-to-door" Path-finding Approach for Indoor Navigation", *Proceedings Gi4DM: GeoInformation for Disaster Management*, May 3-8, 2011.

- [62] A. R. Jimenez, F. Seco, C. Prieto and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU", Intelligent Signal Processing WISP IEEE International Symposium, August 26-28, 2009.
- [63] J. Kim and H. Jun, "Vision-based location positioning using augmented reality for indoor navigation", Consumer Electronics IEEE Transactions on, vol. 54, no. 3, pp. 954-962, 2008.
- [64] K. Akaho, T. Nakagawa, Y. Yamaguchi, K. Kawai, H. Kato, and S. Nishida, "Route guidance by a car navigation system based on augmented reality," Electrical Engineering in Japan (English translation of Denki Gakkai Ronbunshi), vol. 180, no. 2, pp. 43–54, 2012.
- [65] G. A. Lee, A. Dunser, A. Nassani, and M. Billinghurst, "AntarcticAR: An outdoor AR experience of a virtual tour to Antarctica," 2013 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities, ISMAR-AMH 2013, no. Cci, pp. 29–38, 2013.
- [66] M. J. Kim, X. Wang, S. Han, and Y. Wang, "Implementing an augmented reality-enabled wayfinding system through studying user experience and requirements in complex environments," 2015.
- [67] Brush, A.J.B. et al. 2010. User experiences with activity-based navigation on mobile devices. Proceedings of MobileHCI 2010, 73–82.
- [68] K. Liu, G. Motta, T. Ma and T. Guo, Multi-floor Indoor Navigation with Geomagnetic Field Positioning and Ant Colony Optimization Algorithm, 2016.
- [69] H. Zhang, W. Lin and A. Chen, "Path Planning for the Mobile Robot: A Review", Symmetry, vol. 10, no. 10, pp. 450, Oct. 2018.
- [70] P. T. Mahida, S. Shahrestani and H. Cheung, "Comparison of pathfinding algorithms for visually impaired people in IoT based smart buildings".
- [71] R. T. Azuma, A Survey of Augmented Reality, pp. 48, 1997.
- [72] D. Chatzopoulos, C. Bermejo, Z. Huang and P. Hui, "Mobile Augmented Reality Survey: From Where We Are to Where We Go", IEEE Access, vol. 5, pp. 6917-6950, 2017.
- [73] Schmalstieg & Höllerer. Augmented Reality. A Brief History of Augmented Reality, p. 4-12.
- [74] Liarokapis, F. (2002). Augmented reality interfaces—Architectures for visualising and interacting with virtual information [PhD thesis]. University of Sussex.
- [75] Malbezin, P., Piekarski, W. and Thomas, B. (2002) 'Measuring ARToolKit accuracy in long distance tracking experiments', 1st IEEE International Workshop on Augmented Reality, 29 September, Germany
- [76] Stricker, D., & Daehne, P. (2001). Design and development issues for ARCHEOGUIDE: An augmented reality based cultural heritage on-site guide. Proceedings of the International Conference on Augmented Virtual Environments and Three-Dimensional Imaging.
- [77] Siltanen, S. Theory and applications of marker-based augmented reality. Toolkits and libraries, p. 35- 36.
- [78] Craig, A. (2013). Understanding Augmented Reality. Elsevier.
- [79] Baker, S., Bakar, J., & Zulkifli, A. (2017). Elements of museum mobile augmented reality for engaging hearingimpaired visitors. AIP Conference Proceedings, 020033. doi:10.1

- [80] O'Brien, H. L., & Toms, E. G. (2010). The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1), 50–69. doi:10.1002/asi.21229
- [81] Behzadan, A. H., Timm, B. W., & Kamat, V. R. (2008). General Purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering. *Advanced Engineering Informatics*, 22(1), 90–105. doi:10.1016/j.aei.2007.08.005
- [82] Ellsworth, J. J., Gossett, C. P., & Clements, K. (2016). US patent application no. 15/160,996. US Patent Office.
- [83] "<https://docs.unity3d.com/Packages/com.unity.xr.arkit@4.1/manual/index.html#depth-subsystem>"
- [84] "<https://developers.google.com/ar/develop>"
- [85] "<https://unity.com/unity/features/arfoundation>"
- [86] <https://library.vuforia.com/articles/Solution/How-Toadd-a-License-Key-to-your-Vuforia-App>.
- [87] <https://library.vuforia.com/getting-started/overview.html>
- [88] Schmid, C., R. Mohr, C. Bauckhage. Evaluation of Interest Point Detectors. *International. – Journal of Computer Vision*, Springer Verlag, Vol. 37, 2000, No 2, pp. 151-172.
- [89] Fleet, D.J., Y. Weiss. Optical Flow Estimation. – *Handbook of Mathematical Models in Computer Vision*, Springer, 2006, pp. 237-257.
- [90] Jurado, S., Garrido, R. Muñoz-Salinas F.J., Madrid-Cuevas M.J., MarínJiménez. Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion. – *J. Pattern Recognition*, Vol. 47, June 2014, Issue 6, pp. 2280-2292.
- [91] [https://books.google.gr/books?hl=en&lr=&id=qtRJDwAAQBAJ&oi=fnd&pg=PP1&dq=nav mesh+in+unity&ots=5UKvBfbbAQ&sig=reIK7jxNcBlornigj03ycofyU9w&redir\\_esc=y#v=onepage&q&f=false](https://books.google.gr/books?hl=en&lr=&id=qtRJDwAAQBAJ&oi=fnd&pg=PP1&dq=nav+mesh+in+unity&ots=5UKvBfbbAQ&sig=reIK7jxNcBlornigj03ycofyU9w&redir_esc=y#v=onepage&q&f=false)
- [92] <https://docs.unity3d.com/Manual/nav-InnerWorkings.html>
- [93] <https://www.openstreetmap.org/#map=6/38.359/23.810>