



# Augmented Reality application for mobile platforms

By

Δημήτριος Κυριακάκης  
Διπλωματική Εργασία  
Επ. Καθηγήτρια : Κατερίνα Μανιά



## TABLE OF CONTENTS

Chapter 1 - Intro.....	4
1.1 The case .....	4
1.2 Case Description.....	4
1.3 Summary .....	5
Chapter 2 - Augmented Reality .....	6
2.1 Επαυξημένη πραγματικότητα .....	6
2.1.1 Η τεχνολογία Augmented Reality .....	6
2.1.2 Augmented Reality vs Virtual Reality .....	7
2.1.3 Current Uses of AR .....	7
2.2 Υλοποίηση AR σε κινητές συσκευές.....	12
2.2.1 Γενικά.....	12
2.2.2 Υπάρχουσες εφαρμογές AR .....	12
2.2.3 Υπάρχουσες Βιβλιοθήκες AR .....	14
2.3 Το HITLab και η Mobile AR Βιβλιοθήκη.....	16
2.3.1 Η Επαυξημένη πραγματικότητα και το HIT Lab.....	16
2.3.2 Mobile AR Framework.....	18
2.4 Τεχνική ανάλυση του Mobile AR Framework .....	19
2.4.1 Εισαγωγή.....	19
2.4.2 - Τα απαραίτητα μέρη του κινητού .....	20
2.4.3 Υλοποιώντας ένα AR View.....	24
2.4.4 Υλοποιώντας ένα Map View .....	26
Chapter 3 - Requirements .....	27
3.1 Υποθέσεις - παραδοχές .....	27
3.3 - Λειτουργικές Απαιτήσεις .....	28
3.3.1 Εγγραφή στο Σύστημα (User-Registration).....	28
3.3.2 Αρχική Σελίδα Χρηστη (User - Home Page) .....	29

3.3.3 Προσβαση στον χαρτη της πολης .....	29
3.3.4 Προσβαση στο περιβαλλον γριφου .....	30
3.3.5 Ενημέρωση για την πρόοδο άλλων παικτών και φίλων .....	31
3.4 - Οι οθόνες του συστήματος - Αρχική Μορφή .....	31
3.4.1 - Εισαγωγική Οθόνη .....	32
3.4.2 - Κεντρική Οθόνη .....	33
3.4.3 - Η οθόνη Χάρτη .....	34
3.4.4 - Η οθόνη ομάδας .....	35
3.4.5 - Το περιβάλλον γρίφου .....	36
3.5 - Αξιολόγηση αρχικού σχεδιασμού .....	38
3.5.1 - Ευχρηστία και κινητές εφαρμογές .....	38
3.5.2 - Αξιολογώντας τον αρχικό Σχεδιασμό .....	40
3.5.3 - Evaluation Process -Heuristic Evaluation .....	41
3.5.4 - Γενικά Συμπεράσματα Αξιολόγησης .....	44
3.6 - Οι οθόνες του συστήματος - Τελική μορφή .....	45
3.6.1 - Εισαγωγική Οθόνη .....	45
3.6.2 - Κεντρική Οθόνη .....	46
3.6.3 - Η οθόνη Χάρτη .....	47
3.6.4 - Η οθόνη ομάδας .....	48
3.6.5 - Το περιβάλλον γρίφου .....	49
3.6.6 - Η οθόνη ειδοποιήσεων .....	50
3.6.7 - Η οθόνη ρυθμίσεων .....	51
Chapter 4 - Developing the application .....	52
4.1 - The AR Libraries .....	52
4.1.1 Hitlab.NZ Android AR Library .....	52
4.1.2 Wikitude Plugin .....	52
4.2 - The AR Resources .....	53
4.2.1 - 3D Models .....	53

4.2.2 - AR Image Effects .....	54
4.3 - Προβλήματα κατά την ανάπτυξη της εφαρμογής.....	55
4.3.1 - Πρόβλημα Στην ακρίβεια των μετρήσεων του GPS.....	55
4.3.2 - Πρόβλημα φυσικών εμποδίων .....	56
Chapter 5 - System Architecture .....	58
5.1 Client-Server Architecture .....	58
5.1.1 Η αρχιτεκτονική Client-Server.....	58
5.2 Back-end Layer .....	60
5.2.1 - Web Services .....	60
5.2.2 - Η αρχιτεκτονική REST .....	61
5.2.3 - Web Service resources (Πόροι) .....	63
5.2.4 - Σχεδιασμός του Συστήματος .....	64
5.2.5 - Υλοποίηση του συστήματος .....	69
5.3 Front-end Layer .....	75
5.3.1 - Developing a Mobile application .....	75
5.3.2 Choosing the best solution .....	77
5.3.3 Phonegap .....	79
5.4 - Δομή της εφαρμογής .....	81
5.4.1 - Angular JS .....	82
5.4.2 - MAin Application .....	82
5.4.3 - Templates - View .....	86
5.4.4 - Controllers .....	90
5.4.5 - Services - Model.....	94
Chapter 6 - Use Cases .....	96
Chapter 7 - Future Work .....	99
Bibliography .....	100
References.....	100

### 1.1 THE CASE

Development of an augmented reality outdoors application (urban-rural) for mobile platforms.

### 1.2 CASE DESCRIPTION

Στα πλαίσια του της διπλωματικής μου εργασίας ζητήθηκε η δημιουργία και υλοποίηση μιας εφαρμογής επαυξημένης πραγματικότητας, εφαρμοσμένης σε εξωτερικούς χώρους, για κινητές συσκευές. Πρόκειται για μία εφαρμογή για κινητές συσκευές που θα χρησιμοποιεί στοιχεία επαυξημένης πραγματικότητας, προσπαθώντας να κινητοποιήσει τον χρήστη, να τον οδηγήσει σε μία διέξοδο από το αστικό περιβάλλον μέσω ενός παιχνιδιού. Ουσιαστικά μιλάμε για ένα παιχνίδι γρίφων σχεδιασμένο σε πραγματικούς χώρους με πρόσθετα στοιχεία που δεν υπάρχουν στον πραγματικό περιβάλλον. Η εφαρμογή θα χρησιμοποιεί και στοιχεία από μέσα κοινωνικής δικτύωσης έτσι ώστε ο χρήστης να συναναστρέφεται και με άλλους παίκτες.

Βασικό στοιχείο της είναι τα στοιχεία επαυξημένης πραγματικότητας(AR). Κατά τη διάρκεια της υλοποίησης σχεδιάστηκαν τρισδιάστατα μοντέλα τα οποία χρησιμοποιήθηκαν(τοποθετήθηκαν) σε πραγματικές τοποθεσίες προκειμένου να δημιουργηθεί ένα περιβάλλον γρίφου για τον παίκτη.

Στόχος της είναι η απομάκρυνση του χρήστη από το αστικό περιβάλλον, η κίνησή του-άσκησή του μέσα στην πόλη και η παρακίνηση να επισκεφτεί σημεία ενδιαφέροντος (πολιτιστικού και μη) που δεν ήξερε την ύπαρξή τους. Να δώσει τη δυνατότητα στους χρήστες κινητών συσκευών να εξερευνήσουν την πόλη τους, να ακονίσουν την σκέψη τους προσπαθώντας να λύσουν τους γρίφους και μέσα από αυτά να ξεφύγουν από την ρουτίνα και την πίεση που ζει ο μέσος πληθυσμός στις αστικές περιοχές. Η πορεία του παιχνιδιού οδηγεί τους χρήστες σε περιβάλλοντα με περισσότερο πράσινο και πιο κοντά στη φύση και επίσης τους

εντάσσει σε μία ομάδα με άλλους παίκτες στην οποία αναδεικνύεται και προάγεται η κοινωνικότητά τους.

Τα μέλη της θα έχουν το προσωπικό τους προφίλ όπου θα παρουσιάζουν πληροφορίες για τους ίδιους και την ομάδα στην οποία συμμετέχουν. Επίσης, θα μοιράζονται την δραστηριότητα τους και την πρόοδό τους στα πλαίσια του παιχνιδιού.

Το πληροφοριακό σύστημα που θα κατασκευαστεί θα περιλαμβάνει τα εξής:

1. Τη Βάση Δεδομένων του συστήματος.
2. Το back-end layer, που θα διαχειρίζεται τα δεδομένα.
3. Τις δυναμικές διεπαφές χρήστη (front-end layer), οι οποίες θα είναι ουσιαστικά η εφαρμογή, με τις οποίες οι χρήστες του συστήματος θα αλληλοεπιδρούν με αυτό.

---

### 1.3 SUMMARY

Η παρούσα εφαρμογή χρησιμοποιεί την τεχνολογία του AR για να παροτρύνει τον σύγχρονο άνθρωπο να ξεφύγει από τους έντονους ρυθμούς της καθημερινότητας του, να μεταβεί σε έναν παραπονημένο και πιο όμορφο κόσμο, χωρίς να χάσει την επαφή με την πραγματικότητα, να επισκεφτεί και να παρατηρήσει μέρη στα οποία ίσως δεν είχε ξαναβρεθεί, παίζοντας. Μέσα από το σενάριο της ευελπιστούμε να τον καθοδηγήσουμε μακριά από το αστικό περιβάλλον σε μέρη με περισσότερη φύση και πράσινο. Στα πλαίσια του κειμένου θα αναφερόμαστε πλέον στην εφαρμογή ως «**AR Game**».

## CHAPTER 2 - AUGMENTED REALITY

### 2.1 ΕΠΑΥΞΗΜΕΝΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

#### 2.1.1 Η ΤΕΧΝΟΛΟΓΙΑ AUGMENTED REALITY

Η τεχνολογία Augmented Reality ή αλλιώς επαυξημένη πραγματικότητα ή αλλιώς ενισχυμένη πραγματικότητα επιτρέπει την ζωντανή προβολή ενός φυσικού περιβάλλοντος του οποίου όμως η πραγματικότητα είναι επαυξημένη με την προβολή πληροφοριών αλλά και εικονικών προσώπων ή χώρων σχεδιασμένων μέσα έναν ηλεκτρονικό υπολογιστή. Ο συνδυασμός της κάμερας με το σύστημα GPS ενός κινητού τηλεφώνου επιτρέπουν την προβολή επιπλέον πληροφοριών για ένα γεωγραφικό σημείο, διαμορφώνοντας ένα επαυξημένο πληροφοριακά τελικό αποτέλεσμα. Παρέχονται πληροφορίες που περιλαμβάνουν κείμενα, ήχους και video και αφορούν ειδικά τη γεωγραφική θέση που βρίσκεται ο χρήστης και στοχεύει η κάμερα του. Οι προβολές δεδομένων είναι δυνατές είτε από τις οθόνες κινητών είτε από ειδικά γυαλιά προβολής Augmented Reality. Η συγκεκριμένη τεχνολογία μόλις πρόσφατα άρχισε να χρησιμοποιείται και στα κινητά τηλέφωνα.

Η επαυξημένη πραγματικότητα (AR) είναι μία ζωντανή άμεση ή έμμεση προβολή του φυσικού, πραγματικού κόσμου σε ένα περιβάλλον του οποίου τα στοιχεία έχουν εμπλουτιστεί (ή συμπληρωθεί) από ψηφιακά ερεθίσματα, όπως ήχος, βίντεο, γραφικά ή δεδομένα GPS. Σχετίζεται με μια γενικότερη έννοια που ονομάζεται ενδιάμεση πραγματικότητα, στην οποία μια όψη της πραγματικότητας είναι τροποποιημένη (πιθανώς μειωμένη παρά επαυξημένη) από έναν υπολογιστή.

Ως αποτέλεσμα, αυτή η τεχνολογία μπορεί να διαφοροποιήσει την τρέχουσα αντίληψη κάποιου για την πραγματικότητα. Αντίθετα, η εικονική πραγματικότητα αντικαθιστά τον πραγματικό κόσμο με μια προσομοίωση ενός άλλου. Η ενδιάμεση πραγματικότητα συμβατικά είναι ο εμπλουτισμός σε πραγματικό χρόνο του πραγματικού κόσμου με πρόσθετα στοιχεία, όπως είναι τα αθλητικά αποτελέσματα στην τηλεόραση κατά τη διάρκεια ενός αγώνα. Με τη βοήθεια προηγμένης τεχνολογίας AR (π.χ. προσθήκη υπολογιστικής όρασης και αναγνώρισης αντικειμένων) οι πληροφορίες σχετικά με τον πραγματικό κόσμο γύρω του χρήστη μπορεί να γίνει διαδραστική και ψηφιακά διαχειρίσιμη.

---

### 2.1.2 AUGMENTED REALITY VS VIRTUAL REALITY

Ήδη υπάρχουν ευρέως διαδεδομένες εφαρμογές εικονικής πραγματικότητας. Η διαφορά μεταξύ AR και VR εφαρμογών είναι ότι οι εφαρμογές VR δε χρησιμοποιούν δεδομένα από το πραγματικό περιβάλλον του χρήστη. Του παρουσιάζουν μία τελείως διαφορετική πραγματικότητα σχεδιασμένη πλήρως ψηφιακά. Αντίθετα στις εφαρμογές AR, αντικείμενα εικονικής πραγματικότητας συνδυάζονται με δεδομένα από το πραγματικό περιβάλλον του χρήστη και δημιουργούν ένα αποτέλεσμα «ενδιάμεσης πραγματικότητας».

---

### 2.1.3 CURRENT USES OF AR

Οι εφαρμογές AR μπορούν να καλύψουν πολλούς τομείς της αγοράς εφαρμογών. Υπάρχουν ήδη πολλές εφαρμογές που μπορούν να τις εκμεταλλευτούν οι μέσοι χρήστες κινητών συσκευών. Παιχνίδια , εξερευνητές του πραγματικού κόσμου, και εφαρμογές πλοήγησης είναι μερικά από τα είδη των εφαρμογών AR που κυκλοφορούν ήδη. Οι δυνατότητες που μας προσφέρει ο συνδυασμός στοιχείων του πραγματικού περιβάλλοντος, με ψηφιακά σχεδιασμένα μοντέλα μπορεί να οδηγήσει σε αξιοσημείωτες εφαρμογές που θα βελτιώνουν τον τρόπο ζωής μας και τον τρόπο που αντιλαμβανόμαστε το περιβάλλον. Η τεχνολογία ήδη εφαρμόζεται σε πολλούς τομείς οι οποίοι παρουσιάζονται παρακάτω.

#### **Αρχαιολογία**

Το AR μπορεί να χρησιμοποιηθεί για να βοηθήσει την αρχαιολογική έρευνα, με προσθήκη αρχαιολογικών χαρακτηριστικών πάνω στο σύγχρονο τοπίο, που επιτρέπει στους αρχαιολόγους να διατυπώσουν χρήσιμα συμπεράσματα. Ένας άλλος τρόπος εφαρμογής AR σε αυτόν τον τομέα είναι η δυνατότητα των χρηστών να ανοικοδομήσουν τα ερείπια, κτίρια, ή ακόμα και τοπία που υπήρχαν στο παρελθόν.



## Αρχιτεκτονική

Το AR μπορεί να βοηθήσει στην οπτικοποίηση οικοδομικών έργων. Υπολογιστικά μοντέλα μιας οικοδομής μπορούν να τοποθετηθούν σε πραγματικό περιβάλλον προβάλλοντας μια άποψη της ιδιοκτησίας πριν το φυσικό κτίριο κατασκευαστεί εκεί. Αρχιτεκτονικά αξιοθέατα μπορούν να ενισχυθούν με εφαρμογές AR επιτρέποντας στους χρήστες να βλέπουν στο εξωτερικό ενός κτιρίου μέσα από τους τείχους του, την προβολή εσωτερικών χώρων και τη διάταξή του.

## Εμπόριο

Το AR μπορεί να ενισχύσει την προεπισκόπηση ενός προϊόντος. Ο πελάτης μπορεί να έχει τη δυνατότητα να δει τί υπάρχει στο εσωτερικό της συσκευασίας ενός προϊόντος χωρίς να το ανοίξει. Η τεχνολογία AR μπορεί επίσης να χρησιμοποιηθεί ως βοήθημα για την επιλογή των προϊόντων από έναν κατάλογο ή από ένα μενού. Σαρωμένες εικόνες των προϊόντων (2α) μπορούν να ενεργοποιήσουν επιπλέον περιεχόμενο, όπως επιλογές προσαρμογής και επιπλέον εικόνες του προϊόντος κατά τη χρήση του.

(2α) : Παράδειγμα εμπορικής εφαρμογής Augmented Reality σε κινητή συσκευή.



## Εκπαίδευση

Εφαρμογές επαυξημένης πραγματικότητας μπορούν να συμπληρώσουν ένα πρότυπο πρόγραμμα σπουδών. Κείμενα, γραφικά, βίντεο και ήχου μπορούν να τοποθετούνται σε περιβάλλον πραγματικού χρόνου του μαθητή. Συγγράμματα, flashcards και άλλο υλικό εκπαιδευτικής ανάγνωσης μπορούν να περιέχουν ενσωματωμένα «δείκτες» (2β) που, κατά τη σάρωση από μια συσκευή AR, παράγουν συμπληρωματικές πληροφορίες για το μαθητή που παρέχονται σε μορφή πολυμέσων. Οι μαθητές μπορούν να συμμετέχουν διαδραστικά με τον υπολογιστή σε προσομοιώσεις ιστορικών γεγονότων, εξερευνώντας

και μαθαίνοντας λεπτομέρειες για κάθε σημαντικό τομέα του γεγονότος. Άλλες εφαρμογές μπορούν να επιτρέψουν στους μαθητές να μαθαίνουν διαδραστικά έννοιες της μηχανικής, μαθηματικά, γεωγραφία (2γ) ή γεωμετρία. Αυτή είναι μια ενεργητική διαδικασία μάθησης όπου οι μαθητές μαθαίνουν να μαθαίνουν με την τεχνολογία. Η τεχνολογία AR μπορεί να βοηθήσει τους μαθητές στην κατανόηση της χημείας, επιτρέποντάς τους να απεικονίσουν τη χωρική δομή ενός μορίου και να αλληλεπιδρούν μαζί του. Μπορεί επίσης να επιτρέψει στους μαθητές της φυσιολογίας να απεικονίσουν τα διάφορα συστήματα του ανθρώπινου σώματος σε τρεις διαστάσεις. Η τεχνολογία της επαυξημένης πραγματικότητας και επιτρέπει τη μάθηση μέσω της εξ αποστάσεως συνεργασίας, στην οποία οι μαθητές και εκπαιδευτές όχι με την ίδια φυσική τοποθεσία μπορεί να μοιράζονται ένα κοινό εικονικό μαθησιακό περιβάλλον με εικονικά αντικείμενα και υλικά μάθησης και να αλληλεπιδρούν με πρόσθετα μέσα, σε αυτό το περιβάλλον.

(2β) : Το App iSkull, μια επαυξημένης πραγματικότητας εκπαιδευτική εφαρμογή η οποία διευκολύνει τη μελέτη του ανθρώπινου κρανίου.



(2γ) : App iWow, Μια φορητή συσκευή που βασίζεται σε μία επαυξημένης πραγματικότητας ενισχυμένη υδρόγειο σφαίρα για να απεικονίσει τον κόσμο.



## Gaming

Η επαυξημένη πραγματικότητα επιτρέπει στους παίκτες να βιώσουν ένα ψηφιακό παιχνίδι σε ένα πραγματικό περιβάλλον. Κατά τα τελευταία 10 χρόνια έχουν υπάρξει πολλές βελτιώσεις της τεχνολογίας, με αποτέλεσμα την καλύτερη ανίχνευση κίνησης σε τοποθεσίες αλλά και άμεση ανίχνευση των φυσικών κινήσεων του παίκτη. Επίσης, περιβάλλοντα παιχνιδιού παρουσιάζονται στο πραγματικό περιβάλλον με τη χρήση δεικτών (2δ).

(2δ) : Κινητό παιχνίδι το οποίο χρησιμοποιεί μια εικόνα για να δημιουργήσει επάνω της το περιβάλλον παιχνιδιού σε πραγματικό χώρο.



## Βιομηχανικός σχεδιασμός

Το AR μπορεί να βοηθήσει βιομηχανικούς σχεδιαστές στην οπτικοποίηση του σχεδιασμού και της λειτουργίας του προϊόντος πριν από την ολοκλήρωσή του. Η Volkswagen χρησιμοποιεί AR για τη σύγκριση υπολογιστικής και πραγματικής δοκιμής εικονικής πρόσκρουσης. Το AR μπορεί να χρησιμοποιηθεί για να απεικονίσει και να τροποποιήσει τη δομή αμαξώματος του αυτοκινήτου και τη διάταξη του κινητήρα.

## Ιατρική

Η τεχνολογία Augmented Reality μπορεί να παρέχει στο χειρουργό πληροφορίες, οι οποίες κατά τα άλλα είναι κρυμμένες, όπως να δείχνει το ρυθμό της καρδιάς, την αρτηριακή πίεση, την κατάσταση των οργάνων του ασθενούς, κλπ. Το AR μπορεί να χρησιμοποιηθεί για να αφήσει ένα γιατρό να κοιτάξει μέσα σε ασθενή χρησιμοποιώντας μία πηγή πληροφορίας, όπως μια ακτινογραφία ή μία μαγνητική τομογραφία.

## Στρατιωτική χρήση

Στην μάχη, το AR μπορεί να χρησιμεύσει ως ένα δικτυωμένο σύστημα επικοινωνίας που καθιστά χρήσιμα δεδομένα μάχης ορατά σε γυαλιά ενός στρατιώτη σε πραγματικό χρόνο. Από την πλευρά του στρατιώτη, άνθρωπος και διάφορα αντικείμενα μπορούν να επισημαίνονται με ειδικούς δείκτες και να προειδοποιούν για πιθανούς κινδύνους. Εικονικοί χάρτες και πανοραμική απεικόνιση του πεδίου μάχης μπορούν να αποδοθούν με συγκεκριμένους χειρισμούς του στρατιώτη, βελτιώνοντας την πλοήγηση και την προοπτική του, και αυτά τα δεδομένα μπορούν να μεταδοθούν σε στρατιωτικούς ηγέτες σε ένα απομακρυσμένο κέντρο διοίκησης.

## Πλοήγηση

Η τεχνολογία AR μπορεί να αυξήσει την αποτελεσματικότητα των συσκευών πλοήγησης. Οι πληροφορίες μπορούν να εμφανίζονται στο παρμπρίζ ενός αυτοκινήτου που δείχνει την αλλαγή κατεύθυνσης στον προορισμό και μετρητή απόστασης, τον καιρό, το έδαφος, τις οδικές συνθήκες και πληροφορίες για την κυκλοφορία, καθώς και προειδοποιήσεις για πιθανούς κινδύνους στην πορεία του.

## Τουρισμός

Εφαρμογές επαυξημένης πραγματικότητας μπορούν να βελτιώσουν την εμπειρία του χρήστη, όταν ταξιδεύουν με την παροχή, σε πραγματικό χρόνο, ενημερώσεων σχετικών με την τοποθεσία τους και τα χαρακτηριστικά της, συμπεριλαμβανομένων και σχολίων που έκαναν προηγούμενοι επισκέπτες. Οι AR εφαρμογές επιτρέπουν στους τουρίστες να γνωρίσουν προσομοιώσεις των ιστορικών γεγονότων, τόπων και αντικειμένων, καθιστώντας τες ορατές σε ένα πραγματικό τοπίο.

## 2.2 ΥΛΟΠΟΙΗΣΗ AR ΣΕ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

### 2.2.1 ΓΕΝΙΚΑ

Ο τομέας του Augmented Reality (A.R. ή επαυξημένη πραγματικότητα) είναι ένας σχετικά πρόσφατα ανεπτυγμένος τομέας. Δεν έχει μεγάλο μερίδιο της αγοράς εφαρμογών μέχρι στιγμής και ιδιαίτερα στην Ελληνική αγορά είναι κάτι σχεδόν άγνωστο. Οι περισσότερες εφαρμογές AR είναι ακόμα σε φάση πρωτοτύπων, δεν υπάρχει κάποια εφαρμογή ευρέως γνωστή στο αγοραστικό κοινό και είναι πολύ λίγες οι εφαρμογές στο Android Market που να υλοποιούν τεχνολογίες AR. Σκοπός της παρούσας διπλωματικής είναι να δημιουργήσει μία τέτοια εφαρμογή-παιχνίδι.

### 2.2.2 ΥΠΑΡΧΟΥΣΕΣ ΕΦΑΡΜΟΓΕΣ AR

#### IKEA Mobile Application



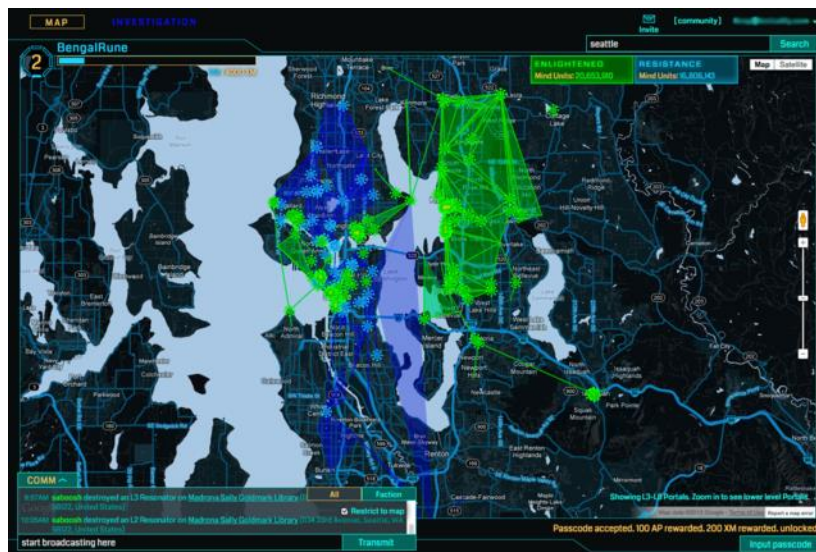
(2ε)

Ο κατάλογος IKEA, που είναι και η πιο διαδεδομένη AR mobile εφαρμογή, χρησιμοποιεί επαυξημένη πραγματικότητα για να δώσει μια εικονική προεπισκόπηση των επίπλων της εταιρείας IKEA σε ένα δωμάτιο. Η τοποθέτηση του επίπλου γίνεται με βάση την αναγνώριση ενός “marker” που υπάρχει στον τυπωμένο κατάλογο της εταιρείας.

## Ingress

Το μόνο διαδεδομένο AR παιχνίδι είναι το Ingress. Το Ingress είναι ένα GPS-based παιχνίδι όπου ο παίκτης δημιουργεί portals. Στο παιχνίδι μπορούμε να επιλέξουμε ανάμεσα σε δύο πλευρές (η αντίσταση και τη φωτισμένη) και να πολεμήσουμε την αντίπαλη πλευρά. Το Ingress λαμβάνει χώρα πάνω από ένα χάρτη του πραγματικού κόσμου και ο παίκτης ταξιδεύει σε πραγματικούς προορισμούς πυλών. Το παιχνίδι είναι παγκοσμίως γνωστό για την ικανότητά του να σηκώνει τον κόσμο από τον καναπέ και να τον οδηγεί στον έξω κόσμο.

(2ζ)



### 2.2.3 ΥΠΑΡΧΟΥΣΕΣ ΒΙΒΛΙΟΘΗΚΕΣ AR

Η εφαρμογή μας θα βασιστεί σε κάποια ήδη υπάρχουσα AR βιβλιοθήκη καθώς το να υλοποιηθεί από το μηδέν μια τέτοιου είδους εφαρμογή δε θα ήταν ευφικτό στα πλαίσια μιας διπλωματικής εργασίας. Παρακάτω θα αναλύσουμε τις υπάρχουσες βιβλιοθήκες AR και θα τεκμηριώσουμε την επιλογή μας.

#### AndAR

Website : <https://code.google.com/p/andar/>

Η AndAR είναι μια βιβλιοθήκη AR που δίνει τη δυνατότητα ανάπτυξης εφαρμογών Επαυξημένης Πραγματικότητας για την πλατφόρμα Android. Το όλο έργο, διατίθεται βάσει της GNU General Public License. Αυτό σημαίνει ότι μπορεί το χρησιμοποιηθεί ελεύθερα σε οποιαδήποτε εφαρμογή. Τα αρνητικά του είναι ότι παρέχεται ελάχιστη υποστήριξη και η τελευταία της έκδοση έχει βασιστεί σε ξεπερασμένη έκδοση Android. Επίσης, δεν παρέχεται υποστήριξη για περιβάλλοντα βασισμένα σε γεωγραφικές τοποθεσίες.

#### DroidAR

Website : <http://bitstars.github.io/droidar/>

Το DroidAR είναι ένα πλαίσιο για ανάπτυξη Augmented Reality στο λογισμικό Android. Παρέχει υποστήριξη AR βασιζόμενου σε τοποθεσία όπως επίσης και βασιζόμενου σε δείκτη. Το πλαίσιο αυτό μπορεί να χρησιμοποιηθεί για πολλά διαφορετικά σενάρια, και υπάρχουν ήδη κάποιες εφαρμογές demo διαθέσιμες. Μπορεί να χρησιμοποιηθεί ελεύθερα, όμως δεν παρέχει επαρκές υλικό για εξοικείωση, και η τελευταία του ενημέρωση είναι βασισμένη σε ξεπερασμένη έκδοση Android.



## **Metaio SDK**

Website : <https://www.metaio.com/>

Το Metaio είναι από τα πιο διαδεδομένα πλαίσια AR παγκοσμίως. Πλέον παρέχεται υπό την αιγίδα της Apple και παρέχει υποστήριξη για όλες τις πλατφόρμες. Παρέχει πλήρεις πληροφορίες για εξοικείωση, πολύ δοκιμαστικό υλικό και υποστηρίζει όλες τις απαιτούμενες λειτουργικότητες, όμως η χρήση του προϋποθέτει άδεια χρήσης.

## **Wikitude SDK**

Website : <https://www.wikitude.com/>

Το wikitude είναι ένα ραγδαία αναπτυσσόμενο AR πλαίσιο, του οποίου ο κώδικας ενημερώνεται συνεχώς. Παρέχει υποστήριξη AR βασιζόμενου σε τοποθεσία όπως επίσης και βασιζόμενου σε δείκτη. Επίσης υποστηρίζει όλες τις πλατφόρμες αλλά η χρήση ορισμένων χαρακτηριστικών του προϋποθέτει μια ιδιαίτερα ακριβή άδεια χρήσης.

Κατά την υλοποίηση της διπλωματικής λόγω κάποιων τεχνικών περιορισμών αναγκαστήκαμε να χρησιμοποιήσουμε τη λειτουργικότητα AR με δείκτη της συγκεκριμένης βιβλιοθήκης.

## **HitLabNZ Mobile AR Framework**

Website : <http://www.hitlabnz.org/index.php/products/mobile-ar-framework>

Το Augmented Reality (AR) σε κινητές πλατφόρμες υπήρξε ένα από τα σημαντικότερα θέματα έρευνας στο HIT Lab τα τελευταία χρόνια. Η ερευνητική ομάδα αποφάσισε να ανοίξει και να μοιραστεί τις υποκείμενες τεχνολογίες, έτσι ώστε η ερευνητική κοινότητα να μπορέσει να επωφεληθεί από ό, τι έχει επιτευχθεί, και να επιταχύνει την ανάπτυξη της βιβλιοθήκης.

Το πλαίσιο Mobile AR είναι μια πλατφόρμα ανάπτυξης κινητών εφαρμογών AR. Το πλαίσιο αποτελείται από διάφορες βιβλιοθήκες και τα εργαλεία για να βοηθήσει τους προγραμματιστές να δημιουργήσουν τη δική τους εμπειρία κινητού AR. Παρέχεται



λειτουργικότητα για AR με βάση τοποθεσία και αλληλεπίδραση με Google Maps. Το επίπεδο υποστήριξης είναι πλήρως ικανοποιητικό και η χρήση του είναι ελεύθερη.

Με βάση τα παραπάνω δεδομένα αλλά και τις καλές σχέσεις τις επιβλέπουσας καθηγήτριας με τον Mark Billinghurst, βασικό συντελεστή του HIT Lab επιλέχθηκε να χρησιμοποιηθεί το Mobile AR Framework ως βάση της εφαρμογής.

## 2.3 ΤΟ HITLAB ΚΑΙ Η MOBILE AR ΒΙΒΛΙΟΘΗΚΗ

### 2.3.1 Η ΕΠΑΥΞΗΜΕΝΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ ΚΑΙ ΤΟ HIT LAB

Η επαυξημένη πραγματικότητα είναι μια τεχνολογία που περιλαμβάνει την απρόσκοπτη ενσωμάτωση εικονικών αντικειμένων στον πραγματικό κόσμο. Έχει πολλές πιθανές εφαρμογές όπως αναλύθηκε παραπάνω σε μια ευρεία ποικιλία πεδίων. Το HIT Lab ασχολείται με την έρευνα AR εδώ και 10 χρόνια και έχει μία από τις μεγαλύτερες ερευνητικές ομάδες στον κόσμο στον τομέα του AR.

Εκτός της βιβλιοθήκης που αφορά κινητές συσκευές η ομάδα του HIT Lab έχει ασχοληθεί και με άλλες πτυχές του AR. Παρακάτω θα παρουσιαστούν τα ορόσημα της ομάδας που οδήγησαν στην σταδιακή μεταφορά του AR σε κινητές πλατφόρμες.

#### **Magic Book (2002)**

Το MagicBook διερευνά την ομαλή μετάβαση μεταξύ της πραγματικότητας και της εικονικής πραγματικότητας. Όταν οι χρήστες περιηγούνται στις σελίδες ενός πραγματικού βιβλίου το οποίο βλέπουν μέσα από μία οθόνη μπορούν να δουν εικονικό περιεχόμενο τοποθετημένο πάνω από τις πραγματικές σελίδες. Όταν συναντήσουν μια επαυξημένης πραγματικότητας ( AR ) σκηνή που τους αρέσει , ο χρήστης μπορεί να εισέλθει στη σκηνή και να μεταφερθεί σε ένα virtual περιβάλλον μέσα από την οθόνη που κρατάει στο χέρι του. Επιπλέον, το βιβλίο χρησιμεύει ως βοήθημα για συνεργασία. Όταν αρκετοί χρήστες βρεθούν στην ίδια σελίδα του βιβλίου μπορούν να δουν την εικόνα AR από τη δική τους οπτική γωνία και εάν εισέλθουν στο virtual περιβάλλον, οι άλλοι χρήστες να τους δουν ως έναν εικονικό χαρακτήρα στη σκηνή. Με τον τρόπο αυτό, το MagicBook υποστηρίζει, μια κοινή εμπειρία AR και ένα multi-user περιβάλλον εικονικής

πραγματικότητας. Το MagicBook έχει πολλές πιθανές εφαρμογές στους τομείς της εκπαίδευσης, της αρχιτεκτονικής και της ψυχαγωγίας.

### **AR Tennis (2005)**

Η δυνατότητα χρήσης εντοπισμού μέσω κάμερας σε κινητά τηλέφωνα θα προσέφερε νέες ευκαιρίες σε διαδραστικές εφαρμογές AR.

Στο AR Tennis συμμετέχουν δύο παίκτες και στις οθόνες τους βλέπουν ένα εικονικό γήπεδο τένις επάνω σε έναν δείκτη (ο δείκτης βρίσκεται στο πραγματικό περιβάλλον). Τα κινητά τηλέφωνα ενεργούν ως εικονικές ρακέτες και χρησιμοποιούνται για να χτυπηθεί η μπάλα μπροστά και πίσω μεταξύ των παικτών. Η εφαρμογή βασίστηκε στο τότε ακμάζων λειτουργικό Symbian και χρησιμοποιούσε μεταξύ άλλων και το Bluetooth των κινητών για συγχρονισμό.

Αποτέλεσε την πρώτη διαδραστική εφαρμογή AR του εργαστηρίου για κινητά τηλέφωνα και προφανώς τη βάση για το Mobile AR Framework.

### **EmbeddedAR (2009)**

Το έργο EmbeddedAR στόχευσε να χρησιμοποιήσει τις τότε πρωτοεμφανιζόμενες πλατφόρμες κινητών συσκευών (Android , iOS) για να διευρηθούν τα όρια του τι είναι δυνατό στο πλαίσιο των κινητών συσκευών και της χαμηλής ισχύος επαυξημένης πραγματικότητας. Το έργο αυτό είχε ως στόχο να δείξει τις νέες δυνατότητες που αυτά τα συστήματα επιτρέπουν, με τη μόχλευση τόσο του hardware, όσο και του software, και την παρέμβαση στο firmware τους.

Σίγουρα αποτέλεσε τον πρόγονο της Βιβλιοθήκης Mobile AR.

## PhobiAR (2010)

Αυτό το έργο εστιάστηκε στην ανάπτυξη ενός προηγμένου διαδραστικού συστήματος θεραπείας συγκεκριμένων φοβιών, όπως ο φόβος για τις αράχνες. Επιτρέπει στους ασθενείς να δουν εικονικά ερεθίσματα φόβος ενσωματωμένα στον πραγματικό κόσμο και να αλληλεπιδρούν με τα ερεθίσματα σε πραγματικό χρόνο. Με τη δημιουργία ενός οπτικά ρεαλιστικού πλαισίου και στηρίζοντας την αλληλεπίδραση (μέσω της αναγνώρισης χειρονομιών) στόχος του έργου είναι να αναπτύξει ένα πολύ αληθοφανές σύστημα που επιτρέπει τον πλήρη έλεγχο της έντασης του ερεθίσματος και άλλες παραμέτρους που είναι αναγκαίες για την αποτελεσματική θεραπεία του ασθενούς.

Η ανάπτυξη του συγκεκριμένου έργου βοήθησε πολύ στην γεωτοποθεσιακή τοποθέτηση εικονικών αντικειμένων και στην αναγνώριση του πραγματικού περιβάλλοντος και των παραμέτρων του.

Η έρευνα στα παραπάνω έργα και η εξέλιξη της τεχνολογίας που οδήγησε στην ευρεία χρήση των κινητών συσκευών οδήγησε την ομάδα του HIT Lab στην ανάπτυξη του Mobile AR Framework, ενός Framework για ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας σε κινητές συσκευές.

---

### 2.3.2 MOBILE AR FRAMEWORK

Όπως αναλύθηκε και παραπάνω το συγκεκριμένο Framework επιλέθηκε ως βάση της παρούσας διπλωματικής.

Οι λόγοι που οδήγησαν στην επιλογή του συγκεκριμένου Framework και τα σημεία στα οποία διαφέρει από παρεμφερείς βιβλιοθήκες είναι τα εξής:

- 1) Δυνατότητα τοποθέτησης εικονικών αντικειμένων με βάση γεωγραφικά δεδομένα.
- 2) Δυνατότητα εντοπισμού των αντικειμένων σε χάρτη μέσω διαδραστικότητας με το Google Maps API.
- 3) Δυνατότητα αλληλεπίδρασης με τα εικονικά αντικείμενα.

- 4) Πλήρες documentation. Η ομάδα του HIT Lab έχει φροντίσει να καταστήσει το Framework και τις λειτουργίες του πλήρως κατανοητές σε προγραμματιστές που πρωτοέρχονται σε επαφή με τον τομέα της επαυξημένης πραγματικότητας παρέχοντας πλήρεις επεξηγήσεις για όλες τις δυνατότητες που προσφέρει και για τα σημεία που πρέπει να προσεχθούν κατά την ανάπτυξη μιας AR εφαρμογής.
- 5) Η χρήση του είναι ελεύθερη, δεν απαιτείται άδεια χρήσης επί πληρωμή.
- 6) Ο κώδικας είναι σχετικά ενημερωμένος και συμβατός με τις τελευταίες εκδόσεις Android.

## 2.4 ΤΕΧΝΙΚΗ ΑΝΑΛΥΣΗ ΤΟΥ MOBILE AR FRAMEWOK

### 2.4.1 ΕΙΣΑΓΩΓΗ

Το Mobile AR Framework παρέχει στον προγραμματιστή που θα έρθει σε επαφή μαζί του ένα εύρος χαρακτηριστικών. Για την ομαλή λειτουργία του χρειάζεται η παροχή πρόσβασης σε κάποια βασικά χαρακτηριστικά του κινητού την οποία παίρνουμε με δέσμευση των σχετικών αδειών. Πιο συγκεκριμένα χρειαζόμαστε πρόσβαση στις υπηρεσίες τοποθεσίας, στην κάμερα, και συνδεσιμότητα στο διαδίκτυο. Χρειαζόμαστε τον παρακάτω κώδικα στο αρχείο Manifest.xml για να δεσμεύσουμε τις σχετικές άδειες.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
```

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

```
<uses-permission android:name="android.permission.CAMERA"></uses-permission>
```

```
<uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
```

Τα παραπάνω χαρακτηριστικά είναι απαραίτητα καθώς για να έχουμε εφαρμογή επαυξημένης πραγματικότητας πρέπει να εκτελέσουμε παρεμβάσεις στο πραγματικό περιβάλλον του χρήστη. Έτσι, για να το πετύχουμε αυτό χρειαζόμαστε κάποιους αισθητήρες της συσκευής για να λάβουμε δεδομένα που αφορούν τη θέση και τη

κατάσταση του χρήστη σε σχέση με τον πραγματικό κόσμο. Τα βασικά μέρη του κινητού που χρησιμοποιούνται σε μια εφαρμογή AR είναι:

- Camera για τη λήψη εικόνας.
- Orientation Sensor για τη διάγνωση του προσανατολισμού.
- Accelerometer
- Global Positioning System

---

## 2.4.2 - ΤΑ ΑΠΑΡΑΙΤΗΤΑ ΜΕΡΗ ΤΟΥ ΚΙΝΗΤΟΥ

### Camera

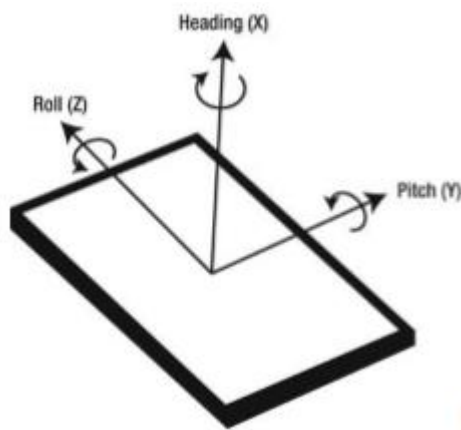
Το πιο βασικό στοιχείο της εφαρμογής AR είναι η κάμερα του κινητού, η οποία είναι απαραίτητη για να δημιουργήσουμε ένα περιβάλλον επαυξημένης πραγματικότητας. Λαμβάνοντας την εικόνα από την κάμερα, την επεξεργαζόμαστε μέσα στην εφαρμογή, συνυπολογίζοντας τις μετρήσεις προσανατολισμού και θέσης της συσκευής. Με βάση τις μετρήσεις αυτές κάνουμε τις απαραίτητες παρεμβάσεις στην εικόνα για να έχουμε τα επιθυμητά αποτελέσματα στην οθόνη της συσκευής.

### Orientation sensor

Για να λειτουργήσει η εφαρμογή AR, βασική προϋπόθεση είναι η διάγνωση του προσανατολισμού της συσκευής. Εφόσον η κίνηση της συσκευής γίνεται σε 3 διαστάσεις θα χρειαστούμε 3 άξονες για να καταγράψουμε την ακριβή της θέση. Η καταγραφή γίνεται με τη βοήθεια δύο αισθητήρων που παρέχονται σε όλα τα σύγχρονα smartphones, του επιταχυνσιόμετρου (accelerometer sensor) και του αισθητήρα μαγνητικού πεδίου (magnetic field sensor). Εδώ να σημειώσουμε ότι όπως και σε κάθε πυξίδα, οι αισθητήρες μαγνητικού πεδίου δεν τα πηγαίνουν και πολύ καλά όταν βρίσκονται πολύ κοντά σε άλλα μεταλλικά αντικείμενα, κάτι που θα πρέπει να προσέξουμε στις μετρήσεις μας. Συλλέγοντας τα δεδομένα από αυτούς τους 2 αισθητήρες και χρησιμοποιώντας λίγη τριγωνομετρία υπολογίζουμε με ακρίβεια τον προσανατολισμό της συσκευής.

Καθορίζουμε 3 μεταβλητές που θα μας δείχνουν τον ακριβή προσανατολισμό (2η).

- Heading (X) Κάτι σαν πυξίδα. Μας δείχνει αν η συσκευή μας είναι προσανατολισμένη στο Βορρά, στο Νότο κλπ. Τιμή 0 ή 360 αντιστοιχεί στο Βορρά, 90 στα Ανατολικά, 180 Νότια, 270 Δυτικά.
- Pitch (Y) Εδώ μετράμε την περιστροφή της συσκευής σε σχέση με τον άξονά της. Η τιμή του pitch θα είναι 0 αν η συσκευή είναι σε οριζόντια θέση, -90 αν το πάνω μέρος της συσκευής κοιτάει την οροφή, 90 αν το πάνω μέρος κοιτάει το πάτωμα.
- Roll (Z) Η επόμενη απαραίτητη μέτρηση για την περιστροφή της συσκευής σε σχέση με τον άξονά της. Η τιμή της είναι 0 όταν η συσκευή είναι ξαπλωμένη με την οθόνη προς τα πάνω, -90 όταν η οθόνη κοιτάει αριστερά, 90 όταν η οθόνη κοιτάει δεξιά.



(2η) : αναπαράσταση των μεταβλητών Roll, Heading, Pitch στο επίπεδο της συσκευής

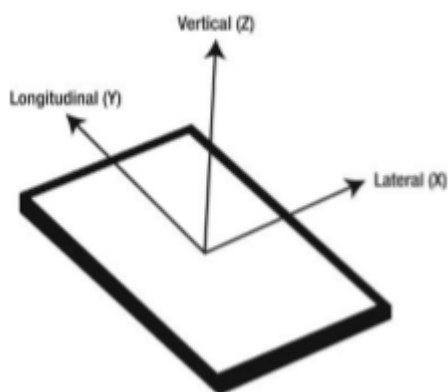
Η Βιβλιοθήκη παρέχει σε κώδικα το μέρος του *Sensor Manager* που θα διαχειρίζεται τον προσανατολισμό της συσκευής και οι αντίστοιχες τιμές θα είναι *headingAngle*, *pitchAngle*, *rollAngle*.

Το android μας στέλνει από μόνο του μετρήσεις από τους αισθητήρες ανά συγκεκριμένα χρονικά διαστήματα, καθώς οι συνεχείς μετρήσεις επιβαρύνουν πολύ τον επεξεργαστή και τη μπαταρία της συσκευής. Όμως, στην περίπτωσή μας πιο πολύ μας ενδιαφέρει η ικανοποίηση του χρήστη από την απόκριση του AR παιχνιδιού παρά η εξοικονόμηση πόρων. Έτσι, θα χρησιμοποιήσουμε την πιο απαιτητική ρύθμιση στους αισθητήρες “*SENSOR\_DELAY\_FASTEST*” προκειμένου να παίρνουμε αποτελέσματα όσο το δυνατόν ταχύτερα.

## Accelerometer

Το επιταχυνσιόμετρο (accelerometer) δίνει μετρήσεις σχετικές με την κίνηση της συσκευής. Οι μετρήσεις είναι πάνω σε 3 κατευθυντήριους άξονες (2θ).

- Lateral(x) Οι κινήσεις σε μήκος. Αυτό σημαίνει ότι εάν θεωρήσουμε ότι η συσκευή μας είναι πάνω σε ένα τραπέζι και μετακινείται από τα αριστερά προς τα δεξιά η τιμή του lateral θα είναι θετική και ανάλογη με την ταχύτητα, ενώ σε αντίθετη κίνηση η τιμή είναι αρνητική και πάλι ανάλογη με την ταχύτητα της κίνησης.
- Longitudinal(y) Οι κινήσεις σε πλάτος. Αυτό σημαίνει ότι εάν η ίδια συσκευή βρίσκεται στο τραπέζι μετακινηθεί κάθετα (εμπρός-πίσω) η τιμή του longitudinal θα είναι θετική όταν η συσκευή κινηθεί προς το πάνω μέρος της και αρνητική στην αντίθετη περίπτωση. Οι μετρήσεις είναι και πάλι ανάλογες της ταχύτητας της κίνησης.
- Vertical(z) Οι κινήσεις σε ύψος. Σε αυτόν τον άξονα μετριοούνται οι μεταβολές του ύψους της συσκευής από το έδαφος δηλαδή με βάση τον κάθετο άξονα από την επιφάνεια της Γης. Σε ακινησία η τιμή που επιστρέφεται είναι  $-9,8 \text{ m/s}^2$ , λόγω της βαρύτητας, κάτι το οποίο θα παίξει σημαντικό ρόλο στις μετρήσεις μας.



(2θ) : αναπαράσταση των μεταβλητών που παίρνουμε από τις μετρήσεις του GPS στο επίπεδο της συσκευής

## Global Positioning System (GPS)

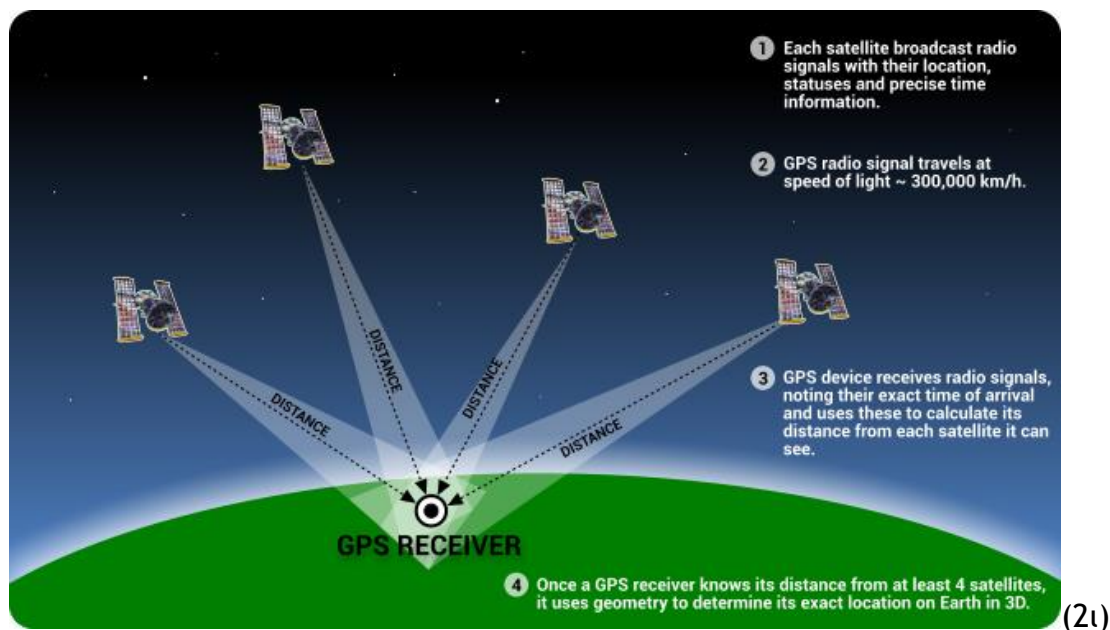
Το γνωστό σε όλους μας GPS είναι ένα σύστημα λήψης τοποθεσίας, το οποίο μπορεί να δώσει ακριβείς μετρήσεις για την τοποθεσία της συσκευής μέσω δορυφόρων.

Αυτή τη στιγμή υπάρχουν 31 δορυφόροι με τους οποίους θα μπορούσε να επικοινωνήσει μία συσκευή παγκοσμίως. Σε οποιαδήποτε στιγμή το λιγότερο 9 δορυφόροι είναι ορατοί από οποιοδήποτε σημείο του εδάφους.

Για να έχουμε ασφαλή δεδομένα μέτρησης η συσκευή πρέπει να επικοινωνήσει με το λιγότερο 4 δορυφόρους. Οι δορυφόροι στέλνουν 3 ειδών δεδομένα στη συσκευή (21) τα οποία περνάνε μέσα από πολύπλοκους αλγορίθμους προκειμένου να ανακτήσουμε τη θέση της.

- Η ακριβής ώρα εκπομπής
- Η ακριβής θέση του εκάστοτε δορυφόρου
- Η σχετική θέση των άλλων 3 δορυφόρων

Η τοποθεσία της συσκευής, τελικά υπολογίζεται με αρχές τριγωνομετρίας. Αυτό ίσως θα μας έκανε να νομίζουμε ότι 3 δορυφόροι θα ήταν αρκετοί για να έχουμε ανάκτηση τοποθεσίας, όμως ο τέταρτος δορυφόρος χρησιμεύει στο να παρακάμψουμε σφάλματα χρονισμού, τα οποία θα μπορούσαν να αποβούν καταστροφικά στο τελικό αποτέλεσμα.





### 2.4.3 ΥΛΟΠΟΙΩΝΤΑΣ ΕΝΑ AR VIEW

Βασικό χαρακτηριστικό μιας εφαρμογής επαυξημένης πραγματικότητας είναι η προβολή των εικονικών αντικειμένων στο πραγματικό περιβάλλον. Αρχικά μεταφέρουμε τα εικονικά αντικείμενα που θέλουμε να χρησιμοποιήσουμε (τρισδιάστατα μοντέλα συγκεκριμένης διαμόρφωσης) στον σχετικό φάκελο “Assets”.

Στη συνέχεια, δηλώνουμε την ύπαρξη της κλάσης που θα διαχειρίζεται το AR View στο αρχείο Manifest.xml

```
<activity android:label="@string/app_name" android:name=".ARViewActivity"
android:screenOrientation="landscape"
android:theme="@android:style/Theme.NoTitleBar.Fullscreen" > </activity>
```

Δε μένει παρά να υλοποιήσουμε την κλάση ARViewActivity.java η οποία θα κληρονομεί τις μεθόδους που είναι απαραίτητες από το Mobile AR Framework. Θέτουμε μια εικονική τοποθεσία για τις αρχικές μας δοκιμές(MockLocation).

```
public class ARViewActivity extends OARComponentBase {

    @Override    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);                // setup fake location
        getSensorManager().enableMockLocation(-43.522189, 172.58292);

    }

}
```

Στη συνέχεια κάνουμε την απαραίτη προεργασία προκειμένου να πάρουμε τα εικονικά αντικείμενα από τον φάκελο Assets δημιουργώντας μία σχετική μέθοδο.

```
protected OADDataManager setupDataManager() {

    OADDataManagerAssets dm = new OADDataManagerAssets(this);

    return dm;

}
```

Το κάθε εικονικό αντικείμενο εμφανίζεται στο AR View σαν «σκηνή». Το AR View μπορεί να αποτελείται από πολλές σκηνές καθεμιά με τη δικιά της παραμετροποίηση. Τα απαραίτητα δεδομένα για τις σκηνές μας όταν έχουμε πολλές σκηνές τα κρατάμε

στο αρχείο scenes.xml . Η παρακάτω υπερμέθοδος του Mobile AR Framework τραβάει τα δεδομένα για τις σκηνές του AR View από το συγκεκριμένο αρχείο.

```
protected void setupScenes(OADataManager dataManager) {  
    super.setupScenes(dataManager);  
}
```

Εάν έχουμε λίγα αντικείμενα για το AR View μπορούμε να τα περάσουμε κατευθείαν στη μέθοδο setupScenes ως εξής. Ο παρακάτω κώδικας τοποθετεί στο AR View 3 αντικείμενα. Τα ορίσματα στον κατασκευαστή OAScene είναι Latitude, Longitude, ετικέτα αντικειμένου, όνομα αρχείου στον φάκελο Assets.

```
OAScene scene1 = new OAScene(-43.522695, 172.582303, "Box", "tbox");  
OAScene scene2 = new OAScene(-43.522686, 172.583333, "Cat", "MrMeow");  
OAScene scene3 = new OAScene(-43.522313, 172.583333, "Ball", "15");  
dataManager.addScene(scene1);  
dataManager.addScene(scene2);  
dataManager.addScene(scene3);
```

Συμπεριλαμβάνουμε και την παρακάτω μέθοδο με την οποία αφαιρούμε από το AR View τον εικονικό ορίζοντα.

```
protected int setupOptions() {  
    return OAARComponentBase.OPTION_HIDE_GRID;  
}
```

Αυτή είναι η βασική δομή για την κλάση που διαχειρίζεται ένα AR View. Φυσικά υπάρχει η δυνατότητα για πιο αναλυτικές παρεμβάσεις και λεπτομέρειες οι οποίες είναι εύκολο να υλοποιηθούν μέσα απο documentation του Mobile AR Framework.

#### 2.4.4 ΥΛΟΠΟΙΩΝΤΑΣ ΕΝΑ MAP VIEW

Μία δραστική καινοτομία του Mobile AR Framework είναι ότι παρέχει τη δυνατότητα για περιήγηση στο περιβάλλον επαυξημένης πραγματικότητας μέσω χάρτη. Δηλαδή ο χρήστης μπορεί να δει την απόστασή του από κάποιο εικονικό αντικείμενο σε χάρτη και να το προσεγγίσει πολύ πιο εύκολα από ότι αν είχε στη διάθεσή του μόνο το AR View.

Όπως και για το AR View ορίζουμε τη σχετική κλάση στο Manifest.xml

```
<activity android:label="@string/app_name" android:name=".MapViewActivity"
android:screenOrientation="landscape"
android:theme="@android:style/Theme.NoTitleBar.Fullscreen" > </activity>
```

Έπειτα δημιουργούμε την κλάση MapViewActivity.java η οποία κληρονομεί τις απαραίτητες μεθόδους από το Mobile AR Framework.

Επίσης χρειαζόμαστε το Google Maps API Key το οποίο προμηθευόμαστε από το Google Developer Console.

```
public class MapViewActivity extends OAMapComponentBase {

    protected String setupGoogleMapApiKey() {

        return

        "here we put the API Key";

    }

}
```

Το Mobile AR Framework φορτώνει αυτόματα τις συντεταγμένες των σκηνών που δηλώθηκαν για το AR View και τις αναπαριστά στον χάρτη.

Η βασική υλοποίηση περιλαμβάνει έναν απλό χάρτη Google Maps με τα εικονικά αντικείμενα απεικονισμένα πάνω του.

## CHAPTER 3 - REQUIREMENTS

### 3.1 ΥΠΟΘΕΣΕΙΣ - ΠΑΡΑΔΟΧΕΣ

Πριν αναλύσουμε τις λειτουργικές απαιτήσεις του συστήματος θα πρέπει να αναφέρουμε τις υποθέσεις που κάναμε προκειμένου να καθορίσουμε τις απαιτήσεις.

Η εφαρμογή θα είναι διαθέσιμη μέσω της αγοράς εφαρμογών Android “Play Store”. Η εφαρμογή θα περιλαμβάνει όλη την απαιτούμενη υλοποίηση για να παρέχει μια ολοκληρωμένη εμπειρία στον εκάστοτε χρήστη. Θα συνδέεται σε ένα API(application programming interface), το οποίο θα διαχειρίζεται τη βάση δεδομένων μας και οι διαθέσιμες λειτουργίες του συστήματος θα παρέχονται στους χρήστες μέσω των διεπαφών της εφαρμογής που θα αναλυθούν αργότερα.

Θα είναι ανοιχτή σε όλους τους κατόχους κινητών συσκευών με λειτουργικό Android, όμως το περιεχόμενό της θα είναι προσβάσιμο μόνο σε όσους κάνουν εγγραφή. Η εγγραφή θα γίνεται δωρεάν καταχωρώντας απλώς κάποια στοιχεία όπως ονοματεπώνυμο, e-mail κλπ, ή μέσω κάποιου λογαριασμού του χρήστη στο δημοφιλές κοινωνικό δίκτυο facebook.

Ένα εγγεγραμμένο μέλος θα μπορεί να συμμετέχει στο «παιχνίδι», το οποίο θα διεξάγεται στην πόλη του. Θα έχει πρόσβαση στα σημεία ενδιαφέροντος, στα οποία θα διεξάγονται οι γρίφοι μέσω χάρτη και μέσω της κάμερας του κινητού του όταν θα βρίσκεται σε κοντινή απόσταση. Σε επίπεδο χάρτη θα έχει μία ολοκληρωμένη εικόνα της πόλης του, και σε επίπεδο κάμερας θα έχει μια εμπειρία επαυξημένης πραγματικότητας(augmented reality). Το περιβάλλον επαυξημένης πραγματικότητας θα δημιουργείται με τη βοήθεια ειδικών εφέ και τρισδιάστατων μοντέλων.

Ένα μέλος θα μπορεί επίσης, να παρακολουθεί τη δραστηριότητα άλλων χρηστών της ομάδας στην οποία θα συμμετάσχει και άλλων ομάδων που συμμετέχουν στο παιχνίδι σε ίδιο επίπεδο πόλης.

Βασικός στόχος της εφαρμογής θα είναι η απομάκρυνση του χρήστη από το αστικό περιβάλλον, η κίνησή του-άσκησή του μέσα στην πόλη και η παρακίνηση να επισκεφτεί σημεία ενδιαφέροντος (πολιτιστικού και μη) που δεν ήξερε την ύπαρξή τους.

Πέρα από το βασικό της στόχο η εφαρμογή **AR Game** θα εκπληρώνει και στόχους κοινωνικής δικτύωσης. Ο χρήστης μέσα από την πορεία του στο παιχνίδι θα «συναναστρέφεται» και με άλλους παίκτες. Εντάσσεται σε ομάδα και προχωράει στο σενάριο βρισκόμενος σε συνεργασία με άλλους χρήστες. Μοιράζεται την πρόοδό του και τα επιτεύγματά του με τους συμπαίκτες του και ενημερώνεται και για τη δική τους δραστηριότητα. Επίσης, θα έχει τη δυνατότητα σύνδεσης και με φίλους του από το facebook, και διαμοιρασμού της δραστηριότητάς του στο κοινωνικό δίκτυο.

### 3.3 - ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

#### 3.3.1 ΕΓΓΡΑΦΗ ΣΤΟ ΣΥΣΤΗΜΑ (USER-REGISTRATION)

Η εφαρμογή θα επιτρέπει την εγγραφή χρηστών στο σύστημα. Η εγγραφή είναι απαραίτητη για την πρόσβαση στο περιεχόμενο της εφαρμογής και στο διαδραστικό παιχνίδι. Με την είσοδο στην εισαγωγική σελίδα της εφαρμογής θα είναι διαθέσιμη η επιλογή εγγραφής(sign up) όπου στη σχετική φόρμα απαραίτητα στοιχεία καταχώρησης θα είναι: Όνομα, Επώνυμο(ή τουλάχιστον το ονοματεπώνυμο που θέλει ο χρήστης να φαίνεται), e-mail, κωδικός πρόσβασης στο λογαριασμό, φύλο, ημερομηνία γέννησης, πόλη.

Όλα τα δηλωθέντα στοιχεία μπορούν να αλλάξουν από το χρήστη κατά τη χρήση του συστήματος. Επίσης, θα δίνεται η επιλογή εγγραφής μέσω Facebook (Log in with Facebook) όπου τα στοιχεία του θα παίρνονται αυτόματα από το Graph API του κοινωνικού δικτύου.

Μαζί με τη συμπλήρωση της φόρμας εγγραφής το σύστημα θα ζητάει από το χρήστη να ανεβάσει κάποια φωτογραφία η οποία θα χρησιμοποιείται ως “avatar”(συνήθως φωτογραφία προσώπου ή κάτι χαρακτηριστικό) και θα συνοδεύει το όνομά του στο προφίλ του και στις δημοσιεύσεις που θα αφορούν την πρόοδό του. Στην περίπτωση

εισόδου μέσω του κοινωνικού δικτύου η φωτογραφία προφίλ του χρήστη στο Facebook θα χρησιμοποιείται αυτόματα.

Μετά την ολοκλήρωση της διαδικασίας εγγραφής ο χρήστης μπορεί να έχει πρόσβαση στο περιεχόμενο της εφαρμογής κάνοντας σύνδεση(login) με χρησιμοποιώντας το e-mail και τον κωδικό που καταχώρησε ο ίδιος κατά τη διαδικασία εγγραφής, ή ξανακάνοντας είσοδο μέσω facebook.

---

### 3.3.2 ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΧΡΗΣΤΗ (USER - HOME PAGE)

Με την ολοκλήρωση της διαδικασίας εγγραφής ο χρήστης οδηγείται στην αρχική σελίδα χρήστη(user-home page). Εκεί θα είναι το βασικό interface με το οποίο ο εγγεγραμμένος πλέον, χρήστης θα χρησιμοποιεί τις υπηρεσίες του συστήματος. Η αρχική σελίδα χρήστη κατ'αρχήν θα του παρέχει τη δυνατότητα αλλαγής των προσωπικών του στοιχείων και άλλων βασικών ρυθμίσεων μέσω ενός ξεχωριστού πλαισίου.

Το κυρίως περιεχόμενο της αρχικής οθόνης θα περιέχει την πρόσφατη δραστηριότητα στα πλαίσια της εφαρμογής, από συμπαίκτες του, ή από άλλους χρήστες που βρίσκονται στην ίδια πόλη. Μέσα από την αρχική οθόνη έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής. (Στον χάρτη, στην πρόοδο και στα στατιστικά του, στα στατιστικά της ομάδας του, και των άλλων ομάδων της ίδιας πόλης. Ίσως και στη δραστηριότητα φίλων του από το facebook.

---

### 3.3.3 ΠΡΟΣΒΑΣΗ ΣΤΟΝ ΧΑΡΤΗ ΤΗΣ ΠΟΛΗΣ

Θα του παρέχεται η δυνατότητα πρόσβασης στον Χάρτη της πόλης (Google Maps), όπου θα βλέπει απομακρυσμένες τις περιοχές στις οποίες θα διαδραματίζονται οι δραστηριότητες-γρίφοι. Όσο βρίσκεται στο χάρτη, πλησιάζοντας σε σημεία ενδιαφέροντος, ο χάρτης εστιάζει και τα αντικείμενα, στα οποία πρέπει να βρεθεί δίπλα περπατώντας, γίνονται ορατά.

Η λειτουργία του χάρτη συνεχίζεται μέχρι ο χρήστης να επιστρέψει στην αρχική οθόνη ή μέχρι να βρεθεί στην ακτίνα κάποιου AR αντικειμένου. Ο χάρτης παρέχει γενικότερες πληροφορίες για την πορεία της ομάδας του μέσα στην πόλη αλλά και για το ποιά σημεία έχουν επισκεφτεί αντίπαλες ομάδες ή και φίλοι του από το facebook.

Όταν ο χρήστης τοποθετεί το κινητό σε οριζόντια θέση με την οθόνη προς τα πάνω, η εφαρμογή θα μεταβαίνει αυτόματα στην οθόνη του χάρτη. Εάν πάλι βρεθεί σε καθορισμένη ακτίνα από κάποιο συγκεκριμένο AR αντικείμενο η εφαρμογή περνάει σε λειτουργία κάμερας ή αλλιώς στο περιβάλλον γρίφου.

---

#### 3.3.4 ΠΡΟΣΒΑΣΗ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΓΡΙΦΟΥ

Όταν ο χρήστης φτάσει στην ακτίνα των αντικειμένων-γρίφων, η εφαρμογή μπαίνει στο περιβάλλον γρίφου. Το περιβάλλον γρίφου κατασκευάζεται μέσα από την κάμερα του κινητού. Παρουσιάζεται το πραγματικό περιβάλλον όπως φαίνεται μέσα από την κάμερα του κινητού μαζί με στοιχεία AR που προσθέτονται προγραμματιστικά. Τα στοιχεία AR που παρουσιάζονται στο περιβάλλον γρίφου είναι εφφέ, ήχοι και κυρίως τα αντικείμενα που πρωταγωνιστούνε στον γρίφο. Τα αντικείμενα είναι διαφορετικά σε κάθε σημείο ενδιαφέροντος και συνάδουν με το σενάριο του γρίφου. Υπάρχει μετρητής χρόνου που υπολογίζει πόσο χρειάστηκε ο χρήστης για να «λύσει» τον γρίφο.

Ο χρήστης επεξεργάζεται τα αντικείμενα προκειμένου να φτάσει στο επιθυμητό σενάριο. Χρησιμοποιώντας το δάχτυλό του στο περιβάλλον γρίφου που αντικρύζει στο κινητό του μπορεί να επεξεργαστεί το μέγεθος και τη θέση του αντικειμένου AR. Το σενάριο λύσης του γρίφου έχει οριστεί ως μία καθορισμένη διαδικασία τροποποίησης του αντικειμένου-γρίφου όσον αφορά το μέγεθός του και τη σχετική του θέση στην εικόνα που βλέπει ο χρήστης.

Όταν ο χρήστης επιλύσει επιτυχώς τον γρίφο βλέπει το σκορ που έπιασε και ποιοί συμπαίκτες ή φίλοι του από το facebook έχουν βρεθεί στο ίδιο σημείο μαζί με την επίδοσή τους.

### 3.3.5 ΕΝΗΜΕΡΩΣΗ ΓΙΑ ΤΗΝ ΠΡΟΟΔΟ ΑΛΛΩΝ ΠΑΙΚΤΩΝ ΚΑΙ ΦΙΛΩΝ

Ένα σημαντικό κομμάτι της εφαρμογής είναι η κοινωνικοποίηση του χρήστη. Ο χρήστης θα συμμετέχει σε ομάδα στο σενάριο που θα διαδραματίζεται στην πόλη του και θα ενημερώνεται για την πρόοδο και τη δραστηριότητα των συμπαικτών του. Επίσης, εάν έχει κάνει την εγγραφή του μέσω του κοινωνικού δικτύου facebook θα βλέπει και τη δραστηριότητα των φίλων του. Και αυτός αλλά και ο «κύκλος» του θα έχει τη δυνατότητα να μοιράζεται φωτογραφίες από τα σημεία των γρίφων και όχι μόνο.

Κάθε φορά που κάποιος παίκτης θα ολοκληρώνει κάποιο μέρος του σεναρίου θα κοινοποιείται αυτόματα στο “feed” της εφαρμογής μια ενημέρωση, για παράδειγμα «Ο τάδε μόλις ολοκλήρωσε το γρίφο στο παλιό λιμάνι σε 3 λεπτά και έδωσε 50 πόντους στην ομάδα του!». Θα υπάρχει η δυνατότητα για “like” και σχολιασμό αυτών των ενημερώσεων. Τέλος θα υπάρχει οθόνη στατιστικών, στην οποία θα φαίνεται η βαθμολογία των υπόλοιπων ομάδων που κινητοποιούνται στην ίδια πόλη.

## 3.4 - ΟΙ ΟΘΟΝΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ - ΑΡΧΙΚΗ ΜΟΡΦΗ

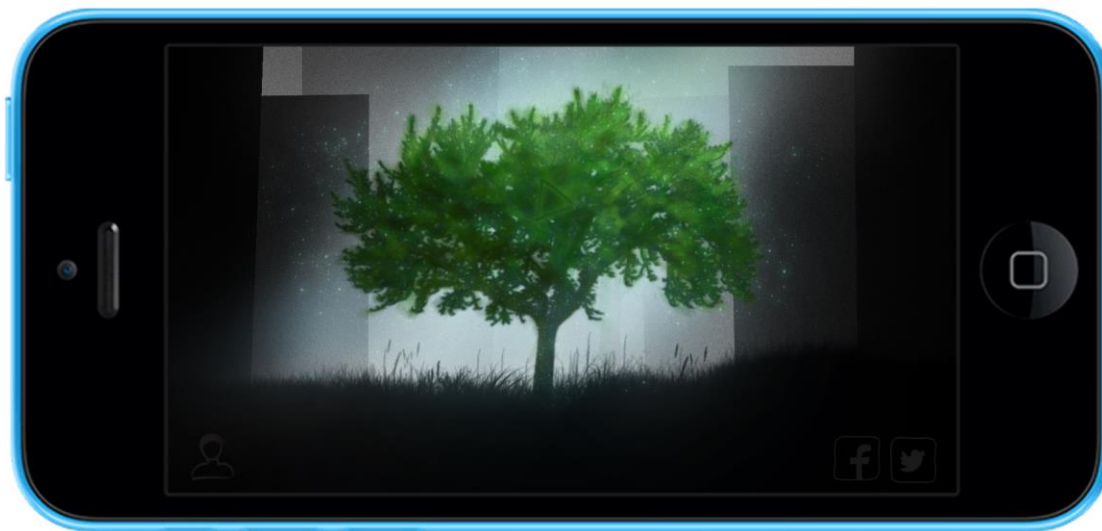
Με βάση την ανάλυση των απαιτήσεων που αναλύθηκε παραπάνω προχωρήσαμε στον αρχικό σχεδιασμό των οθονών του συστήματος. Ο αρχικός σχεδιασμός έγινε σε photoshop με καθαρά εικαστική προτεραιότητα, αφήνοντας την ευχρηστία στο περιθώριο. Προτιμήθηκαν σκοτεινά χρώματα και κάποιες λεπτομέρειες σε πράσινο, προκειμένου να αποκτήσει μυστηριακό χαρακτήρα αλλά και να υπενθυμίζει τον σκοπό του που είναι να φέρει τον χρήστη πιο κοντά στο περιβάλλον και στη φύση. Στον αρχικό σχεδιασμό υιοθετήθηκε ο οριζόντιος προσανατολισμός. Ο αρχικός σχεδιασμός είναι αποτέλεσμα της δουλειάς της Δέσποινας Σαπουντζή.



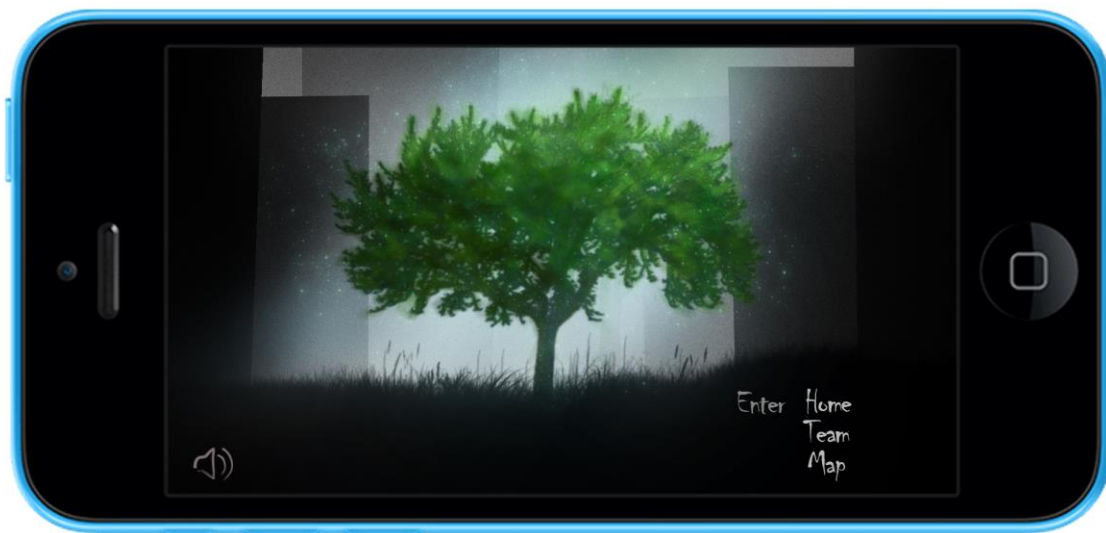
### 3.4.1 - ΕΙΣΑΓΩΓΙΚΗ ΟΘΟΝΗ



(3α)



(3β)

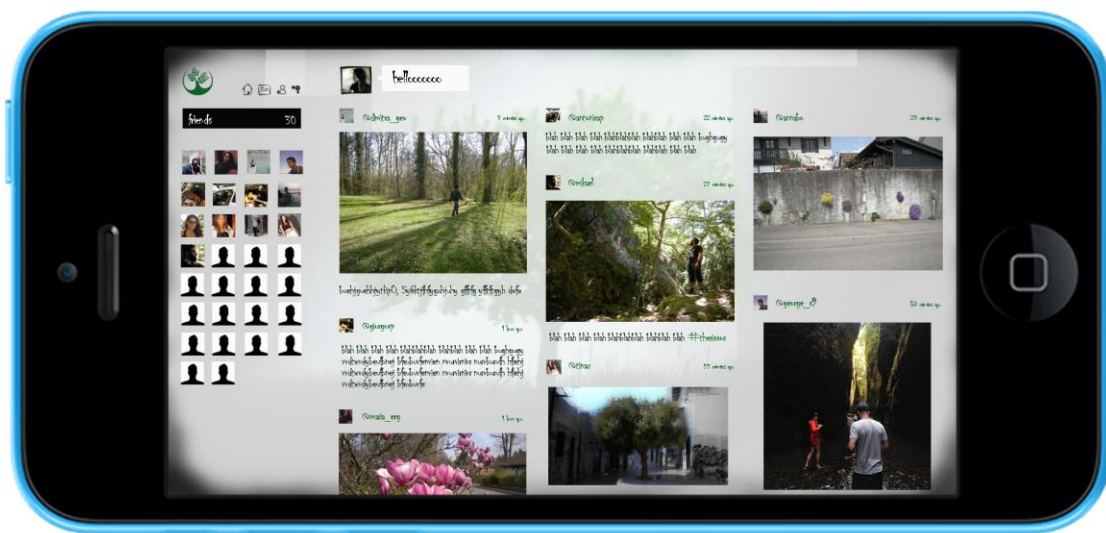


(3γ)

Στην εισαγωγική οθόνη συμπεριλήφθηκαν λειτουργίες εισόδου , δηλαδή το social login, βασικές επιλογές πλοήγησης και η ρύθμιση του ήχου.

### 3.4.2 - ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ

Η κεντρική οθόνη περιέχει το feed της εφαρμογής την δραστηριότητα δηλαδή άλλων χρηστών, κάποιες βασικές λειτουργίες, την ομάδα στην οποία συμμετέχει ο χρήστης.



(3δ)

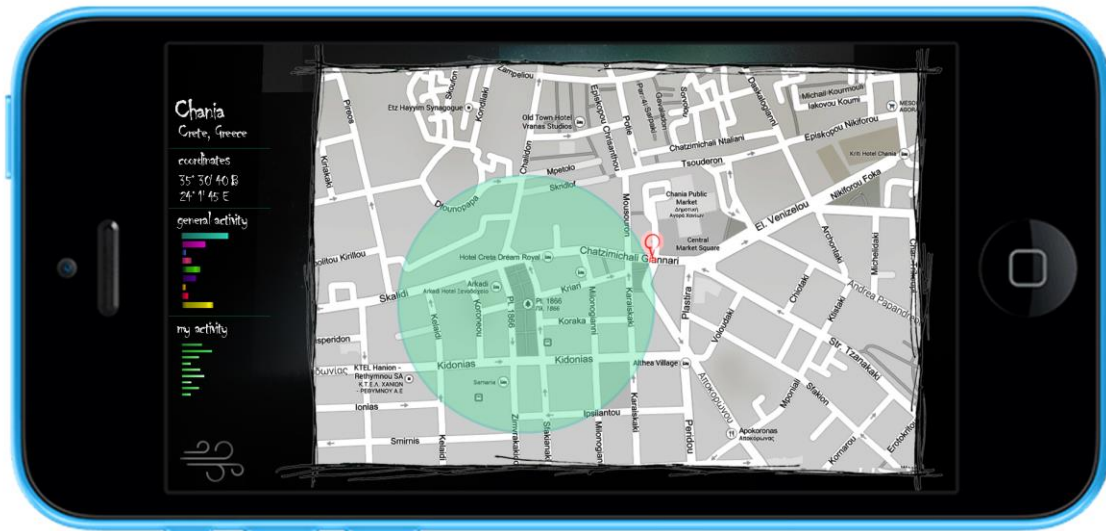
### 3.4.3 - Η ΟΘΟΝΗ ΧΑΡΤΗ

Η οθόνη χάρτη περιέχει την απαραίτητη λειτουργικότητα σε επίπεδο χάρτη. Ο χρήστης περιηγείται στην πόλη έχοντας πανοραμική άποψη των σημείων- γρίφων. Επίσης, βλέπει την τοποθεσία του και στατιστικά στοιχεία, σχετικά με την πρόοδό του.



(3ε)

Όταν πλησιάζει σε τοποθεσία γρίφου, μαρκάρεται η περιοχή στην οποία γίνεται προσβάσιμη η κάμερα, και τονίζεται με διαφορετικό χρώμα.



(3ζ)

Σε εναλλακτική έκδοση, πλησιάζοντας την τοποθεσία γρίφου, η όψη του χάρτη αλλάζει και γίνεται με τρισδιάστατη αίσθηση η απεικόνιση της περιοχής.



(3η)

#### 3.4.4 - Η ΟΘΟΝΗ ΟΜΑΔΑΣ

Η οθόνη ομάδας παρέχει πληροφορίες για τη δραστηριότητα της ομάδας του παίκτη αλλά και για τη δραστηριότητα των άλλων ομάδων. Τα δεδομένα απεικονίζονται με γραφικά σχήματα στον χάρτη.



(3θ)



### 3.4.5 - ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΓΡΙΦΟΥ

Όταν ο χρήστης μπει στην ακτίνα του αντικειμένου του γρίφου αρκετά, τότε ανοίγει η κάμερα του κινητού. Τα αντικείμενα του περιβάλλοντος παρουσιάζονται παραμορφωμένα και η εικόνα φιλτράρεται με διάφορους τρόπους που προσδίδουν μυστηριακό χαρακτήρα στην εικόνα.



(3ι)

Διάφορες πληροφορίες και βοήθεια για τον χρήστη εμφανίζονται σε συνεφέκια και παράθυρα διαλόγου.



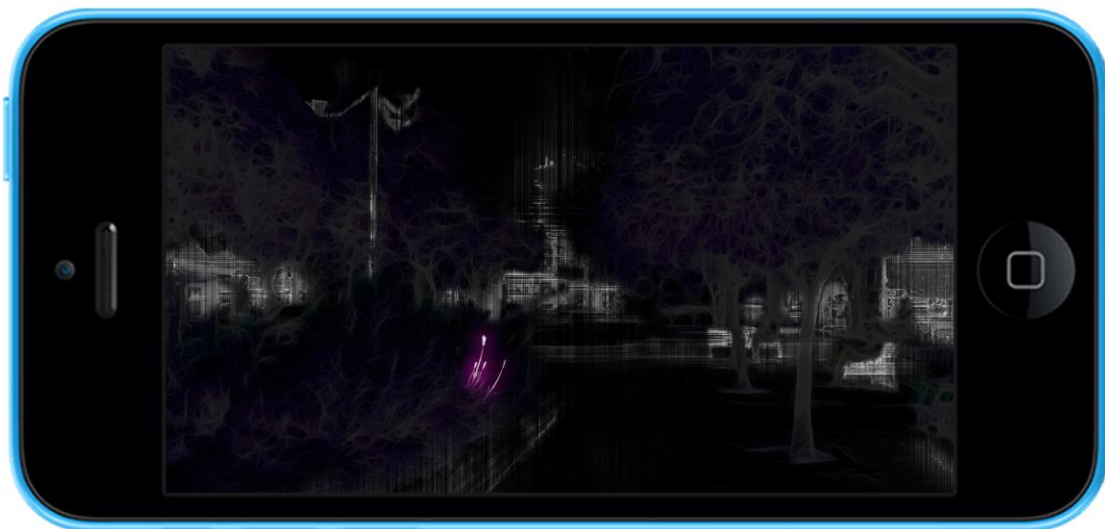
(3κ)

Στο κάτω μέρος της οθόνης εμφανίζεται το inventory, με τα αντικείμενα που «έχει συλλέξει» ο χρήστης, επιλογές για ρυθμίσεις και hints σχετικά με το σενάριο του γρίφου.



(3λ)

Αν περάσει χρόνος και ο χρήστης δεν αλληλοεπιδράσει με το ενεργό αντικείμενο του γρίφου, τότε η οθόνη φιλτράρεται κατάλληλα για να οδηγηθεί το μάτι του χρήστη στο ενεργό αντικείμενο.



(3μ)

## 3.5 - ΑΞΙΟΛΟΓΗΣΗ ΑΡΧΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ

### 3.5.1 - ΕΥΧΡΗΣΤΙΑ ΚΑΙ ΚΙΝΗΤΕΣ ΕΦΑΡΜΟΓΕΣ

Οι πρόοδοι στην τεχνολογία κινητών συσκευών έχουν δημιουργήσει ένα ευρύ φάσμα εφαρμογών που αναπτύσσονται και που μπορούν να χρησιμοποιηθούν από τους ανθρώπους σε διάφορες καθημερινές τους δραστηριότητες. Οι προγραμματιστές μερικές φορές παραβλέπουν το γεγονός ότι οι χρήστες θα θέλουν να αλληλεπιδράσουν με τέτοιες συσκευές ενώ βρίσκονται εν κινήσει. Μικρό μέγεθος οθόνης, περιορισμένη συνδεσιμότητα, υψηλά ποσοστά κατανάλωσης ενέργειας και περιορισμένη μορφές εισόδου είναι μερικά μόνο από τα θέματα που ανακύπτουν κατά το σχεδιασμό για μικρές, φορητές συσκευές. Η εύκολη πρόσβαση δεδομένων και λειτουργιών από τον χρήστη εν κινήσει, είναι ένας κρίσιμος παράγοντας για την επιτυχία ή την αποτυχία της εφαρμογής, για το αν η εφαρμογή θα χαρακτηριστεί ως **εύχρηστη** ή όχι.

Ο Nielsen ορίζει την ευχρηστία ως την ικανότητα του συστήματος να καλύψει τις ανάγκες του χρήστη. Εάν ένα προϊόν δεν παρέχει ευχρηστία, στη συνέχεια, δεν προσφέρει τις δυνατότητες και τις λειτουργίες που σχεδιάστηκε για να παρέχει. Η χρηστικότητα της εφαρμογής καθίσταται περιττή, δεδομένου ότι δεν θα επιτρέπει στο χρήστη να επιτύχει τους στόχους του. Ομοίως, ο Διεθνής Οργανισμός Τυποποίησης (ISO) ορίζει την ευχρηστία ως το "βαθμό στον οποίο μπορεί ένα προϊόν να βοηθήσει τους χρήστες να επιτύχουν συγκεκριμένους στόχους με αποτελεσματικότητα, αποδοτικότητα και ικανοποίηση σε ένα συγκεκριμένο πλαίσιο χρήσης".

Ο ορισμός αυτός προσδιορίζει τρεις παράγοντες ότι θα πρέπει να λαμβάνονται υπόψη κατά την αξιολόγηση της χρηστικότητας.

**Χρήστης:** Άτομο που αλληλεπιδρά με το προϊόν

**Στόχος:** Σκοπός αποτέλεσμα

**Πλαίσιο χρήσης:** Χρήστες, καθήκοντα, εξοπλισμός (hardware, λογισμικό και υλικό), και τα φυσικά και κοινωνικά περιβάλλοντα στα οποία μία εφαρμογή χρησιμοποιείται.

Κάθε ένας από τους παραπάνω παράγοντες μπορεί να έχουν αντίκτυπο στο συνολικό σχεδιασμό της εφαρμογής και ιδίως θα επηρεάσει το πώς ο χρήστης θα αλληλεπιδρά με το σύστημα. Για να μετρήσει πόσο εύχρηστο είναι ένα σύστημα, το πρότυπο ISO ορίζει τρεις μετρήσιμες ιδιότητες:

**Αποτελεσματικότητα:** Η ακρίβεια και η πληρότητα με την οποία οι χρήστες οδηγούνται στην επίτευξη ειδικών στόχων.

**Αποδοτικότητα:** Οι πόροι που δαπανήθηκαν σε σχέση με την ακρίβεια και την πληρότητα με την οποία οι χρήστες εκπληρώνουν τους στόχους τους.

**Ικανοποίηση:** Η ελευθερία από την ταλαιπωρία, και θετική στάση απέναντι στη χρήση του προϊόντος.

### **Περιορισμοί για κινητές εφαρμογές**

Τα μοντέλα που παρουσιάζονται παραπάνω προέρχονται σε μεγάλο βαθμό από παραδοσιακές εφαρμογές desktop. Για παράδειγμα, το έργο του Nielsen βασίστηκε σε μεγάλο βαθμό στο σχεδιασμό συστημάτων τηλεπικοινωνιών και λογισμικού ηλεκτρονικών υπολογιστών. Η έλευση των κινητών συσκευών παρουσίασε νέες προκλήσεις στην ευχρηστία που είναι δύσκολο να ενσωματωθεί στη χρήση των παραδοσιακών μοντέλων.

Ο Zhang και ο Adipat υπογράμμισαν μια σειρά θεμάτων που έχουν εισαχθεί από την έλευση των κινητών συσκευών:

**Κινητό Πλαίσιο:** Κατά τη χρήση κινητών εφαρμογών ο χρήστης δεν συνδέεται με μια συγκεκριμένη τοποθεσία. Μπορεί επίσης να αλληλεπιδρά με κοντινούς ανθρώπους, αντικείμενα και περιβαλλοντικά στοιχεία που μπορεί να αποσπάσει την προσοχή του.

**Συνδεσιμότητα:** Η συνδεσιμότητα είναι συχνά αργή και αναξιόπιστη σε κινητές συσκευές. Αυτό επηρεάζει την απόδοση των κινητών εφαρμογών που χρησιμοποιούν αυτά τα χαρακτηριστικά.

**Μικρό μέγεθος οθόνης:** Για την παροχή φορητότητας οι κινητές συσκευές περιέχουν πολύ περιορισμένο μέγεθος οθόνης και έτσι η ποσότητα των πληροφοριών που μπορούν να εμφανιστούν είναι περιορισμένη.



**Διαφορετικές Αναλύσεις οθόνης:** Η ανάλυση των κινητών συσκευών είναι μειωμένη σε σχέση με εκείνη της επιφάνειας εργασίας υπολογιστή με αποτέλεσμα χαμηλότερη ποιότητα εικόνας. Επίσης, υπάρχει ποικιλία στις αναλύσεις οθόνης ανάμεσα σε διαφορετικές κινητές συσκευές.

**Περιορισμένη ικανότητα επεξεργασίας και τροφοδοσίας:** Για να παρέχουν φορητότητα, οι φορητές συσκευές συχνά παρέχουν μικρότερη ικανότητα επεξεργασίας. Αυτό περιορίζει το είδος των εφαρμογών που είναι κατάλληλες για κινητές συσκευές.

**Διαφορετικές Μέθοδοι εισαγωγής δεδομένων:** Οι μέθοδοι εισαγωγής διαθέσιμες για φορητές συσκευές είναι διαφορετικές από εκείνες για επιτραπέζιους υπολογιστές και απαιτούν ένα ορισμένο επίπεδο επάρκειας. Το πρόβλημα αυτό αυξάνει την πιθανότητα της λανθασμένης εισόδου και καθιστά επιτακτική την σωστή δόμηση μεταξύ των περιοχών εισόδου.

Από την επισκόπηση μας είναι προφανές ότι πολλά από τα υπάρχοντα μοντέλα για την χρηστικότητα δεν λαμβάνουν υπόψιν τους την φορητότητα και τις συνέπειές της. Αυτό περιπλέκει την διαδικασία αξιολόγησης της ευχρηστίας καθώς πρέπει, κατά συνέπεια, να συμπεριληφθεί στους πυλώνες αξιολόγησης και η φορητότητα.

---

### 3.5.2 - ΑΞΙΟΛΟΓΩΝΤΑΣ ΤΟΝ ΑΡΧΙΚΟ ΣΧΕΔΙΑΣΜΟ

Ο αρχικός σχεδιασμός έδωσε ένα πολύ ωραίο αποτέλεσμα οπτικά, όμως δεν είναι μόνο αυτή η προτεραιότητά μας. Η οθόνες των κινητών συσκευών, πληθαίνουν συνεχώς γύρω μας, και πλέον χρησιμοποιούνται πολύ περισσότερο από οθόνες ηλεκτρονικών υπολογιστών στις καθημερινές μας δραστηριότητες. Έτσι, όταν υλοποιούμε τον σχεδιασμό μιας mobile εφαρμογής πρέπει να κινηθούμε μέσα σε συγκεκριμένα πλαίσια για να έχουμε ένα εύχρηστο αποτέλεσμα.

Έρευνες έχουν δείξει πως τα περιττά και εντυπωσιακά γραφικά πολλές φορές αποσπούν την προσοχή του χρήστη ή στην καλύτερη περίπτωση, τον καθυστερούν ακόμα περισσότερο από την σωστή επιλογή. Αν τα γραφικά δεν έχουν συγκεκριμένα και κατανοητή σχέση με την οθόνη της εφαρμογής κάνουν περισσότερο κακό παρά καλό. Πόσο μάλλον σε μία περιορισμένη οθόνη όπως αυτή ενός κινητού τηλεφώνου, οι πληροφορίες πρέπει να είναι ιδιαίτερα διακριτές και ευανάγνωστες.

Στις οθόνες που προέκυψαν από τον αρχικό σχεδιασμό, βλέπουμε εκ πρώτης όψεως πολλά θέματα ευχρηστίας. Σε μερικές έχουμε μεγάλο όγκο πληροφορίας μαζεμένο σε μικρές περιοχές, κάτι το οποίο σίγουρα θα δημιουργήσει πρόβλημα σε χρήστες της εφαρμογής. Μεγάλο μέρος της πληροφορίας πρέπει να ανακαταταξινομηθεί ανάμεσα στις οθόνες, και μερικές λειτουργίες ίσως να πρέπει να μεταφερθούν σε άλλες οθόνες, καινούριες. Προκειμένου το αποτέλεσμα να είναι ικανοποιητικό περάσαμε στη διαδικασία της *ευρυστικής αξιολόγησης*.

### 3.5.3 - EVALUATION PROCESS -HEURISTIC EVALUATION

Οι οθόνες του συστήματος υποβλήθηκαν σε ευρυστική αξιολόγηση. Παρακάτω παραθέτονται τα αποτελέσματα χωρισμένα ανά οθόνη και λειτουργικότητα.

#### Εισαγωγική οθόνη

Ο χρήστης καθώς εισέρχεται στο σύστημα δεν πληροφορείται για τη λειτουργικότητα και τις δυνατότητες που αυτό παρέχει. Εάν δεν ενημερωθεί από κάποιον που ήδη γνωρίζει και έχει έρθει σε επαφή με το σύστημα δεν μπορεί να προχωρήσει. Ευριστικός Κανόνας: Nielsen-Κανόνας 10 : Help and Documentation.

Ο χρήστης δεν γνωρίζει ποιος δημιούργησε την εφαρμογή και πόσο αξιόπιστος είναι. Αυτό τον κάνει δύσπιστο στο να την χρησιμοποιήσει. Πρέπει να δίνεται στο χρήστη τουλάχιστον η επιλογή να ενημερωθεί, σε “help” ή “about us” section. Ευριστικός Κανόνας: Nielsen-Κανόνας 10 : Help and Documentation.

Η εφαρμογή θα λειτουργήσει με social account του χρήστη από facebook αποκλειστικά. Η προσαρμογή του API του facebook αποτελεί μεγάλο όγκο δουλειάς για να γίνει προσαρμογή και για το twitter. Επίσης, με τα νέα guidelines του facebook για να γίνει αποδεκτή εφαρμογή από το facebook πρέπει να έχει ευδιάκριτο facebook login button.

Υπάρχει δυσκολία του χρήστη να συνειδητοποιήσει ότι εισήλθε επιτυχώς στο σύστημα. Απαραίτητο ένα μήνυμα που να δηλώνει ότι η είσοδος έγινε επιτυχώς, ή μία πιο ευδιάκριτη αλλαγή οθόνης. Ευριστικός Κανόνας: Nielsen, 1, Visibility of System Status.

Εφ' όσον ο χρήστης εισέλθει στην εφαρμογή με τον προσωπικό του λογαριασμό, πρέπει να έχει και τη δυνατότητα να αποσυνδεθεί. Πρέπει να προστεθεί και η επιλογή “Logout” ή “Settings” και να δίνεται η δυνατότητα αποσύνδεσης. Ευριστικός Κανόνας: Nielsen, 9, Help users recognize, diagnose and recover from errors.

Το κουμπιά της αρχικής οθόνης είναι σε διάταξη στα 4:3 της οθόνης ενώ το αντίστοιχο που φαίνεται να χειρίζεται τον ήχο είναι σε 16:9. Επίσης, λόγω της ιδιαιτερότητας της οθόνης του κινητού πρέπει να υπάρξουν μεγαλύτερα κενά κάθετα μεταξύ των επιλογών, προκειμένου να μην υπάρξουν λάθος ανιχνεύσεις αφής. Ευριστικός Κανόνας: Mayhew, General Layout 3, Group items logically.

Στην αρχική οθόνη μερικές επιλογές (ήχος, facebook account, twitter account) είναι εικονίδια ενώ οι υπόλοιπες είναι κείμενο. Καλό θα ήταν να υπάρξει ένα σταθερό grid π.χ. εικονίδιο και κείμενο. Ευριστικός Κανόνας: Nielsen 4, Consistency and standards.

Οι επιλογές στο δεξί μέρος της οθόνης είναι (Enter Home, Team, Map). Η δεύτερη λέξη (Enter) στην πρώτη επιλογή δημιουργεί σύγχυση σε κάποιον χρήστη που έχει πρώτη επαφή με την εφαρμογή καθώς έχει συνηθίσει απλώς “Home” ή θα περίμενε “Enter Map” και “Enter Team” αντίστοιχα. Προτείνεται η αφαίρεση της λέξης “Enter”. Ευριστικός Κανόνας: Nielsen 4, Consistency and standards.

### Αρχική Οθόνη

Το παρόν grid της κεντρικής οθόνης (home) δεν ενδεικνύεται για χρήση, ειδικά σε κινητές συσκευές.

Τα posts πρέπει να μπαίνουν το ένα κάτω από το άλλο σε μία στήλη και η πληροφορία τους να είναι ευδιάκριτη. Προτείνεται η συγχώνευση των 3 στηλών σε μία και το αριστερό μενού να γίνει sidebar και να εμφανίζεται όταν ο χρήστης το επιθυμεί και όχι πάντα. Ευριστικός Κανόνας: Nielsen, 7, Flexibility and efficiency of use.

Η διαδρομή που πρέπει να ακολουθήσει ο χρήστης για επιστροφή στην αρχική σελίδα (Home, Team, Map) δεν είναι εμφανής. Πρέπει να δωθεί εμφανής επιλογή επιστροφής ή η προηγούμενη οθόνη να συνενωθεί με την παρούσα. Ευριστικός Κανόνας: Nielsen, 3, User Control and Freedom.

Όλο το αριστερό μέρος της κεντρικής οθόνης δεν ενδεικνύεται για χρήση σε κινητή συσκευή. Οι επιλογές είναι ιδιαίτερα συμπυκνωμένες, ο χρήστης εύκολα θέλοντας να επιλέξει κάτι μπορεί να επιλέξει κάτι άλλο κατά λάθος. Οι βασικές επιλογές στο πάνω μέρος (Home, Profile κλπ) πρέπει να είναι πολύ μεγαλύτερες και τα προφίλ χρηστών

παρακάτω ίσως να μούνε σε διαφορετική οθόνη, (π.χ. Friends). Ευριστικός Κανόνας: Nielsen 5, Error Prevention.

Δε φαίνεται να έχει προβλεφθεί μετάβαση του χρήστη από την κεντρική οθόνη στην οθόνη χάρτη. Το να πρέπει να επιστρέψει στην προηγούμενη οθόνη και να επιλέξει “Map” του στερεί την ευελιξία και χρόνο καθώς ο χάρτης είναι βασικό σημείο της εφαρμογής. Η πρόσβαση στο χάρτη θα γίνεται και με οριζόντια τοποθέτηση του κινητού, όμως πρέπει να έχει προβλεφθεί και η περίπτωση που ο χρήστης δεν το κατανοήσει. Ευριστικός Κανόνας: Nielsen 3, User Control and freedom.

### **Οθόνη χάρτη**

Όλο το αριστερό μέρος της οθόνης χάρτη δεν ενδεικνύεται για χρήση σε κινητή συσκευή. Οι πληροφορίες είναι ιδιαίτερα συμπυκνωμένες, ο χρήστης εύκολα θα δυσκολευτεί ιδιαίτερα να αναγνωρίσει τους χαρακτήρες. Ενδεικνύεται η επιλογή πιο ευδιάκριτης γραμματοσειράς, και η μεγέθυνσή της. Επίσης, και η πινέζες πρέπει να μεγαλώσουν. Ευριστικός Κανόνας: Nielsen, 6, Recognition Rather Than Recall.

Περιορίζεται ο έλεγχος του χρήστη όσον αφορά την προβολή του χάρτη. Όλες οι οθόνες χάρτη δεν παρέχουν δυνατότητα επιστροφής στην αρχική σελίδα ή απλά επιστροφής στην προηγούμενη οθόνη αν και θα έπρεπε. Ευριστικός Κανόνας: Nielsen, 3, User control and freedom.

### **Οθόνη ομάδων**

Πρόβλημα εμφανίζεται στο γεγονός ότι ο χρήστης είναι δύσκολο να καταλάβει αυτή την πληροφορία καθώς δεν υπάρχει καμία πληροφορία περί τίνος πρόκειται και καμία επεξήγηση γιατί βρίσκεται πάνω στον χάρτη που βλέπει. Αυτό μπορεί να τον κάνει να νομίσει ότι δεν πρέπει να βρεθεί σε κάποια χρωματισμένη από άλλη ομάδα περιοχή. Προτείνεται η εμφάνιση ενός πληροφοριακού μηνύματος ή ενός Leaderboard με στατιστικά για την κάθε ομάδα. Ευριστικός Κανόνας: Nielsen, 1, Visibility of system status.

#### 3.5.4 - ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ

Στις επί μέρους οθόνες τις εφαρμογής, η πληροφορία πρέπει να είναι συγκεντρωμένη σε σταθερό μέρος π.χ. στο αριστερό μέρος της οθόνης. Επίσης, πρέπει να υπεραπλουστευθούν οι οθόνες, καθώς η μεγάλη συγκέντρωση πληροφορίας τις καθιστά δυσανάγνωστες. Ίσως να υπάρχει ένα σταθερό μέρος τις οθόνης όπως το αριστερό στο οποίο να εμφανίζονται κάποιες πληροφορίες και με επιλογή τους ο χρήστης να έχει πρόσβαση και σε περισσότερες λεπτομέρειες. Για παράδειγμα να εμφανίζεται η επιλογή “my team” και με την επιλογή της να εμφανίζονται οι παίκτες, ευδιάκριτα σε pop-up οθόνη και ούτω καθεξής. Ευριστικός Κανόνας: Nielsen, 4, Consistency and Standards.

Το σύστημα δεν υποστηρίζει λειτουργικότητα για αναίρεση ενεργειών ή επιστροφής σε προηγούμενη οθόνη. Κάθε οθόνη πρέπει να παρέχει τη δυνατότητα επιστροφής σε προηγούμενη κατάσταση. Ευριστικός Κανόνας: Mayhew, Error Handling, Error Recovery, 2, Provide an undo function.

Στις οθόνες δεν βρέθηκε βοήθεια χρήστη πέρα από το περιβάλλον γρίφου που υπάρχει η λάμπα που θα παρέχει το hint. Οποιαδήποτε ενέργεια γίνεται είτε εμπειρικά είτε δοκιμάζοντας πράγματα. Ευριστικός Κανόνας: Nielsen, 10, Documentation.

## 3.6 - ΟΙ ΟΘΟΝΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ - ΤΕΛΙΚΗ ΜΟΡΦΗ

Εδώ παρουσιάζονται οι τελικές οθόνες του συστήματος όπως περιλήφθηκαν στην τελική υλοποίηση.

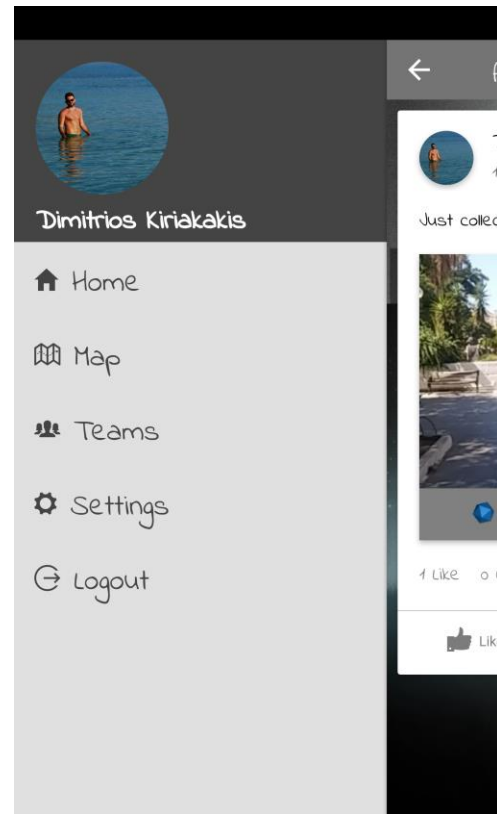
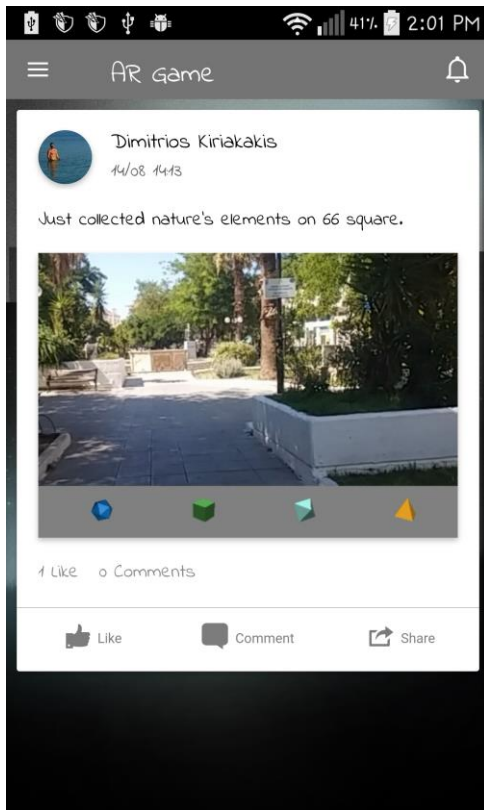
### 3.6.1 - ΕΙΣΑΓΩΓΙΚΗ ΟΘΟΝΗ



Στην εισαγωγική οθόνη που υλοποιήθηκε συμπεριλήφθηκαν λειτουργίες εισόδου , δηλαδή το social login.

### 3.6.2 - ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ

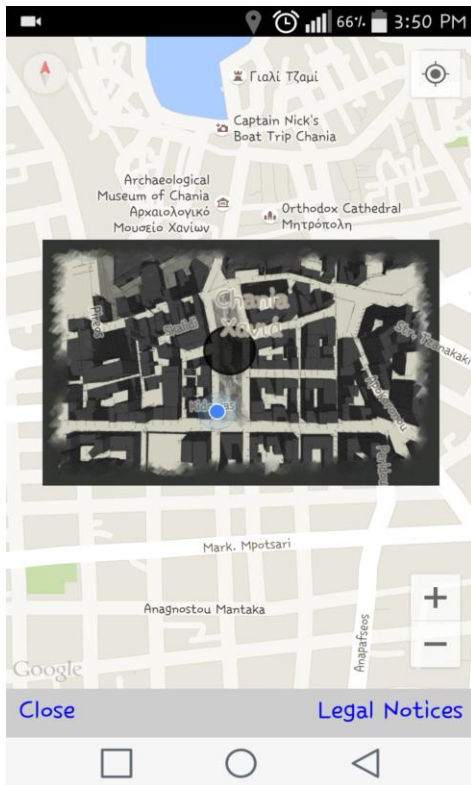
Η κεντρική οθόνη τελικά περιέχει το feed της εφαρμογής την δραστηριότητα δηλαδή άλλων χρηστών, και παρέχει πρόσβαση στις ειδοποιήσεις του χρήστη και στο βασικό μενού πλοήγησης.



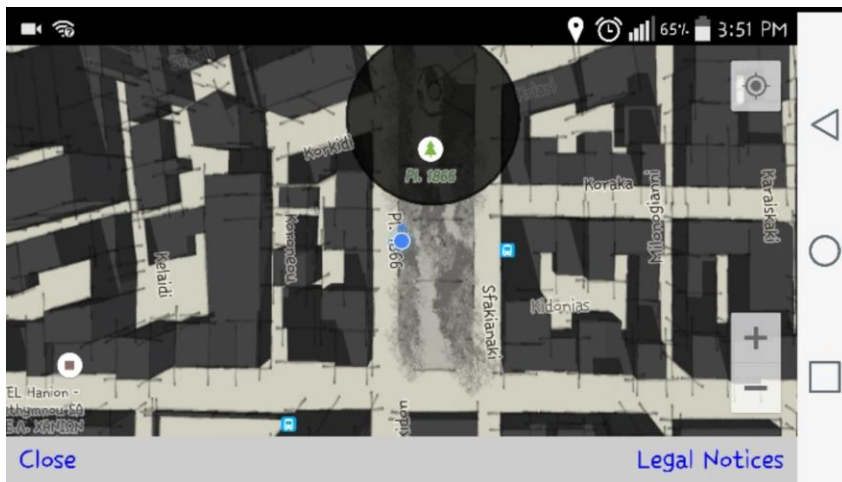
Μέσα από το βασικό μενού πλοήγησης ο χρήστης έχει πρόσβαση στην οθόνη χάρτη, στην οθόνη ομάδας, στις ρυθμίσεις. Επίσης, μπορεί να επιστρέψει στην κεντρική οθόνη ή να αποσυνδεθεί.

### 3.6.3 - Η ΟΘΟΝΗ ΧΑΡΤΗ

Η οθόνη χάρτη περιέχει την απαραίτητη λειτουργικότητα σε επίπεδο χάρτη. Ο χρήστης περιηγείται στην πόλη έχοντας πανοραμική άποψη των σημείων- γρίφων. Επίσης, βλέπει την τοποθεσία του και οι περιοχές γρίφων έχουν σκιαγραφηθεί με ιδιαίτερες λεπτομέρειες.



Πλησιάζοντας την τοποθεσία γρίφου, η όψη του χάρτη αλλάζει και γίνεται με τρισδιάστατη αίσθηση η απεικόνιση της περιοχής.

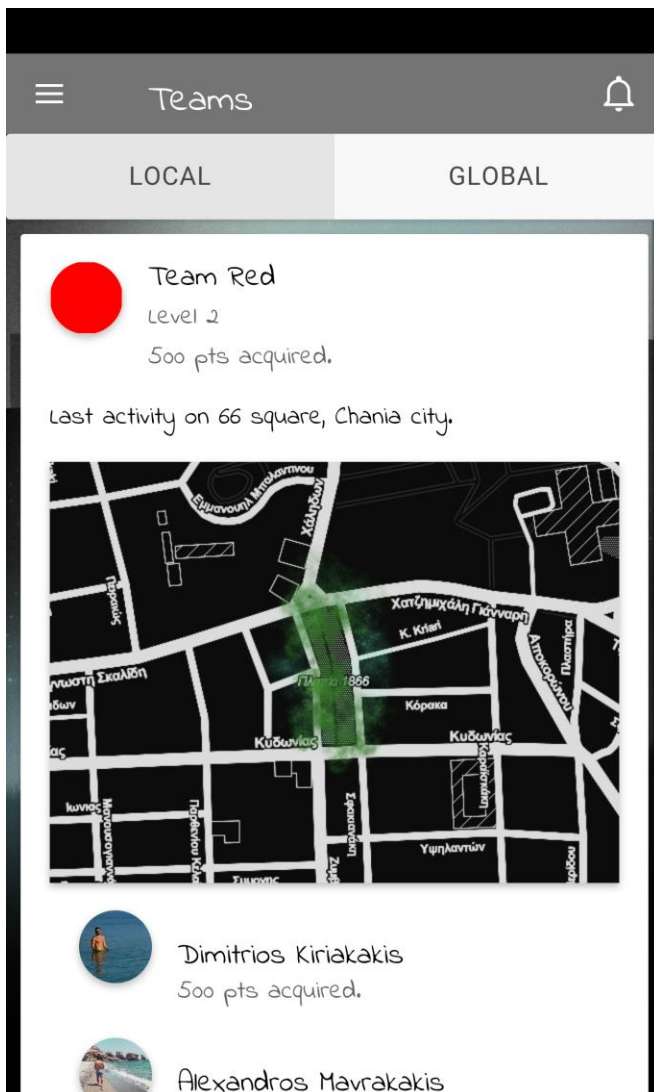




Οι περιοχές στις οποίες πρέπει να εισέλθει κατά τη διάρκεια του γρίφου είναι μαρκαρισμένες με γεωμετρικά σχήματα.

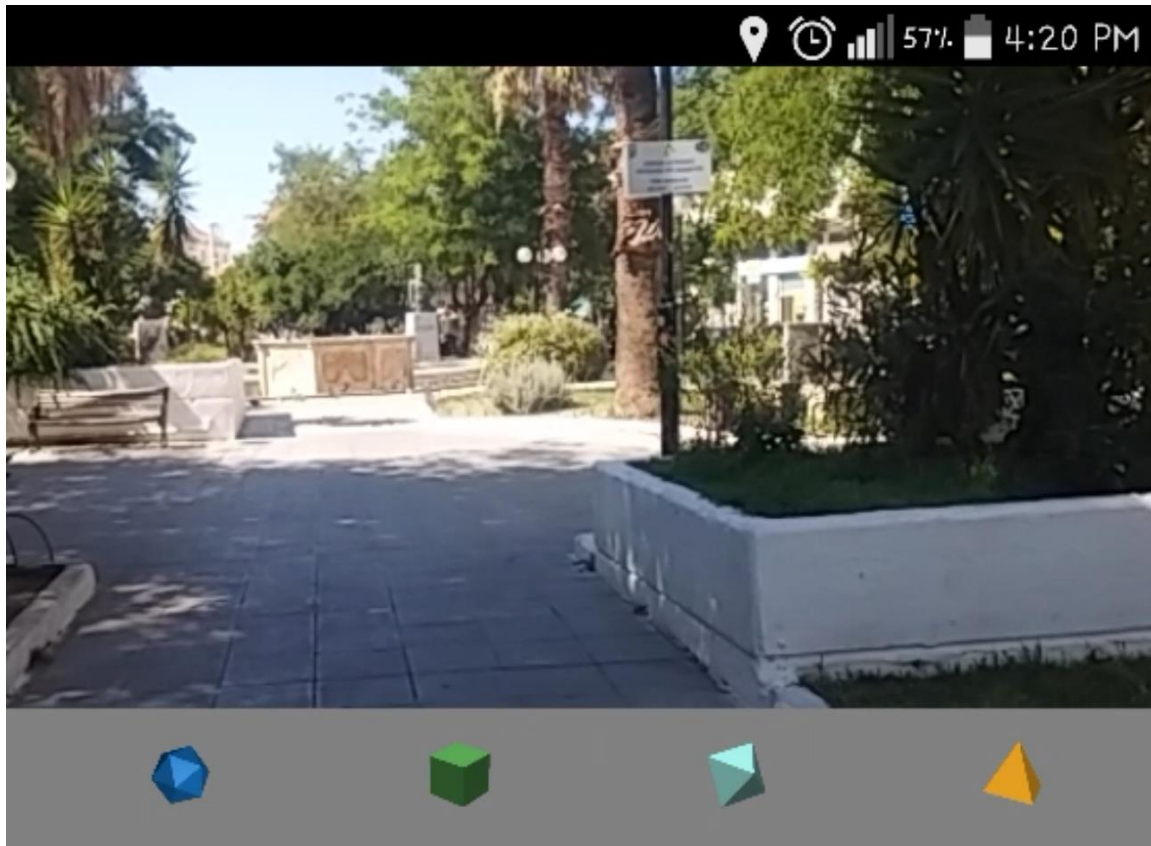
#### 3.6.4 - Η ΟΘΟΝΗ ΟΜΑΔΑΣ

Η οθόνη ομάδας παρέχει πληροφορίες για τη δραστηριότητα της ομάδας του παίκτη αλλά και για τη δραστηριότητα των άλλων ομάδων. Παρουσιάζεται και δραστηριότητα των επί μέρους παικτών κάθε ομάδας.



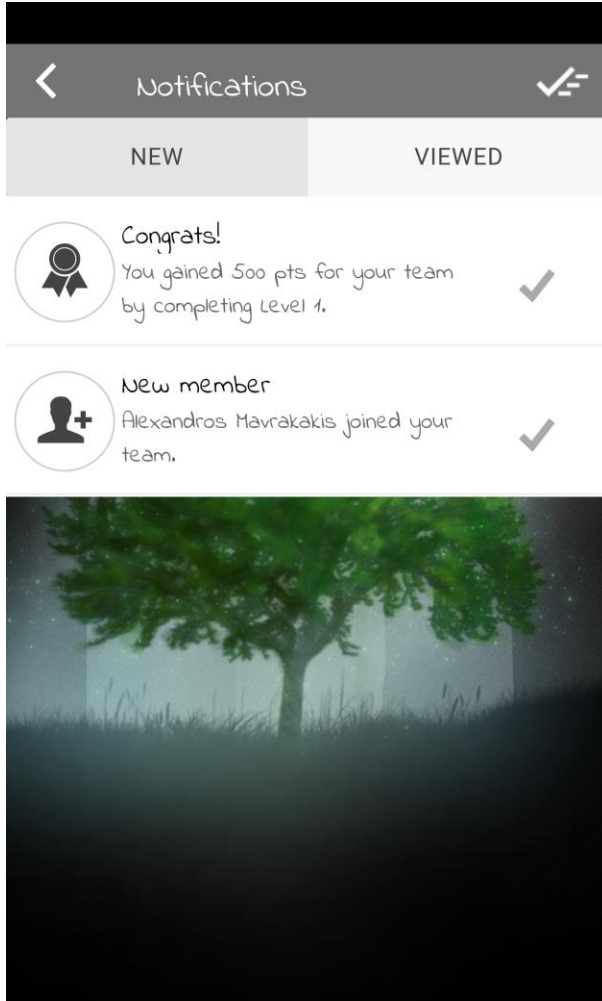
### 3.6.5 - ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΓΡΙΦΟΥ

Όταν ο χρήστης μπει στην ακτίνα του αντικειμένου του γρίφου αρκετά, τότε ανοίγει η κάμερα του κινητού. Λόγω περιορισμών πόρων δεν ήταν ευφικτό να εφαρμοστούν φίλτρα όπως είχε αρχικά σχεδιαστεί. Ο χρήστης έχει στο κάτω μέρος της οθόνης το “inventory” στο οποίο αποθηκεύει τα αντικείμενα που έχει συλλέξει.



### 3.6.6 - Η ΟΘΟΝΗ ΕΙΔΟΠΟΙΗΣΕΩΝ

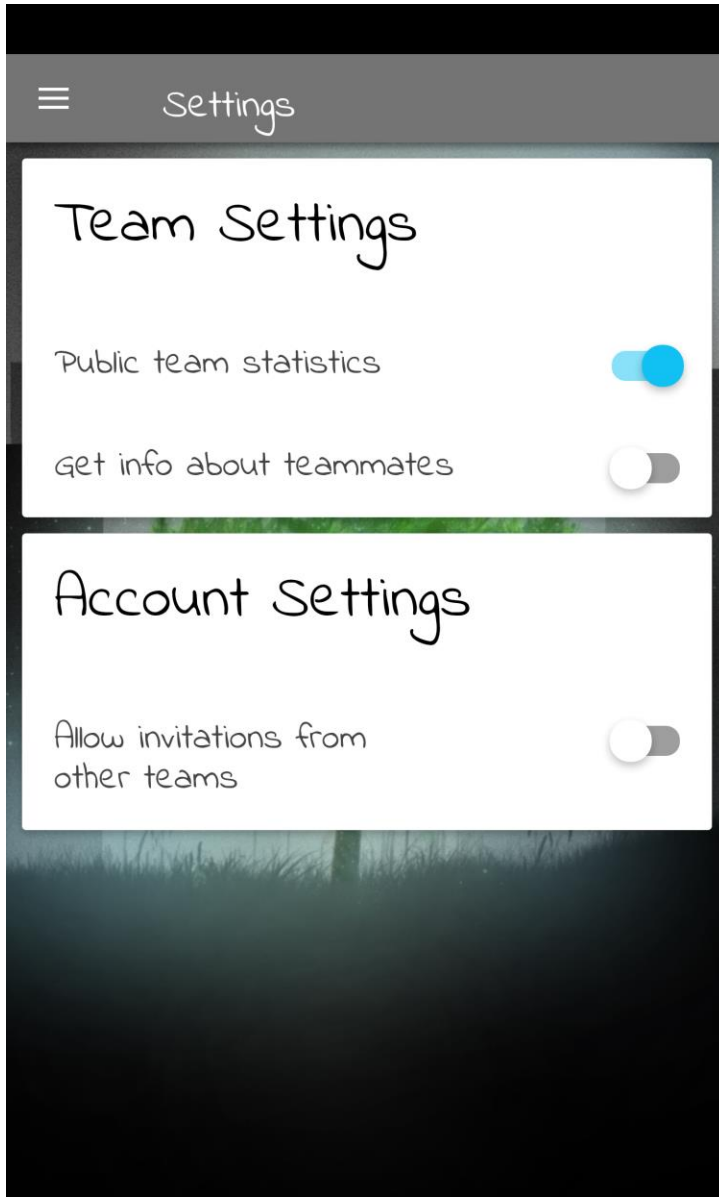
Με βάση την λειτουργικότητα της εφαρμογής κρίθηκε σκόπιμο να δημιουργηθεί μία οθόνη που να ειδοποιεί τον χρήστη για δραστηριότητα που τον αφορά.



Ο χρήστης μπορεί να ενημερωθεί εδώ για τους πόντους που του έχουν πιστωθεί, και για δραστηριότητα που αφορά την ομάδα του.

### 3.6.7 - Η ΟΘΟΝΗ ΡΥΘΜΙΣΕΩΝ

Τέλος ενσωματώθηκε και μία ξεχωριστή οθόνη ρυθμίσεων, για να ελέγχει ο χρήστης κάποια θέματα σχετικά με την ομάδα του και με τον λογαριασμό του στην εφαρμογή.



Παρέχονται επιλογές για απόκρυψη της δραστηριότητας της ομάδας του, για το αν θέλει να ενημερώνεται με ειδοποιήσεις για τη δραστηριότητα των συμπαίκτών του, και για το εάν θέλει να λαμβάνει προσκλήσεις για συμμετοχή σε άλλες ομάδες.

## CHAPTER 4 - DEVELOPING THE APPLICATION

### 4.1 - THE AR LIBRARIES

#### 4.1.1 HITLAB.NZ ANDROID AR LIBRARY

Βασικός πυλώνας της εφαρμογής της παρούσας διπλωματικής είναι η το **mobile AR framework** που αναπτύχθηκε στο HitLab. Το μεγαλύτερο μέρος της υλοποίησης της AR λειτουργικότητας βασίστηκε στις βιβλιοθήκες αυτού του Framework καθώς η υλοποίηση εξ'ολοκλήρου συστήματος Augmented Reality ξεπερνούσε το μέγεθος μιας διπλωματικής εργασίας.

Οι βιβλιοθήκες του συγκεκριμένου Framework βοήθησαν στην τοποθέτηση AR αντικειμένων σε συγκεκριμένες συντεταγμένες, και στην οπτική τους εμφάνιση μέσω της κάμερας του κινητού. Το framework παρείχε native κώδικα android (Java), και απευθυνόταν αποκλειστικά σε συσκευές Android.

Στην πορεία της διπλωματικής ο κώδικας της βιβλιοθήκης εξελίχθηκε, προστέθηκαν στοιχεία επεξεργασίας των αντικειμένων AR, έγιναν βελτιώσεις στην ακρίβεια των μετρήσεων του κινητού.

Επίσης, οι συγκεκριμένες βιβλιοθήκες ενσωματώθηκαν στην εφαρμογή ως plugin, για λόγους που θα αναλυθούν στη συνέχεια.

#### 4.1.2 WIKITUDE PLUGIN

Στην πορεία υλοποίησης της διπλωματικής λόγω προβλημάτων σταθερότητας με την τοποθέτηση αντικειμένων σε συγκεκριμένες γεωγραφικές θέσεις οδηγηθήκαμε στην αναζήτηση εναλλακτικών λύσεων.

Μία βιβλιοθήκη που χρησιμοποιήθηκε τελικά ήταν αυτή της wikitude που προσφέρει image recognition. Έτσι μπορέσαμε να αναγνωρίσουμε σημεία στον χώρο σαν εικόνες και να τοποθετήσουμε εικονικά μοντέλα σε σχέση με αυτά τα σημεία παίρνοντας ένα πολύ πιο ικανοποιητικό και σταθερό αποτέλεσμα.

Και αυτή η βιβλιοθήκη προστέθηκε ως plugin στην τελική εφαρμογή.

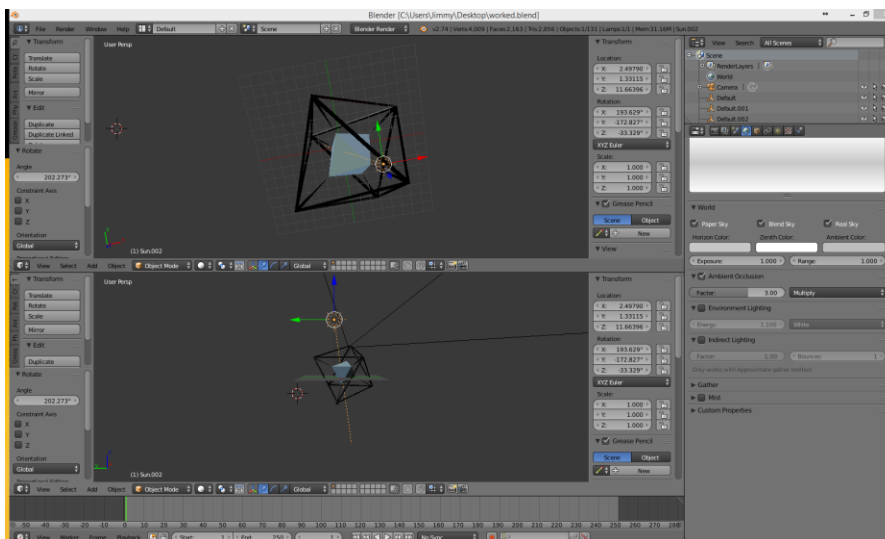
## 4.2 - THE AR RESOURCES

### 4.2.1 - 3D MODELS

Τα μοντέλα που θα χρησιμοποιηθούν για να δημιουργήσουν το περιβάλλον επαυξημένης πραγματικότητας, πρέπει να σχεδιαστούν με συγκεκριμένη παραμετροποίηση για να είναι προσβάσιμα από τις βιβλιοθήκες μας. Το format των αρχείων που υποστηρίζονται είναι τα .obj , .dae , .fbx. Επίσης, πρέπει να χρησιμοποιηθούν οι ακόλουθες ρυθμίσεις (4α):

- ✓ Apply Modifiers
- ✓ Include Edges
- ✓ Include Normals
- ✓ Triangulate Faces
- ✓ Objects as OBJ Objects
- ✓ Material Groups

Το μεγαλύτερο πρόβλημα κατά τη διάρκεια σχεδίασης των μοντέλων που αντιμετωπίστηκε ήταν η εφαρμογή του φωτισμού. Στην σκηνή επαυξημένης πραγματικότητας (camera mode) δεν είναι δυνατόν να εφαρμοστεί καθολικός φωτισμός, οπότε έπρεπε να ενσωματωθεί φωτισμός στα μοντέλα κατά την εξαγωγή τους.



(4α) : Οι ρυθμίσεις εφαρμοσμένες στην πλατφόρμα Blender

Μετά απο πειραματισμούς σε rhino και maya χωρίς κανένα αποτέλεσμα, επανήλθαμε στους πειραματισμούς με το blender όπου και κρυβόταν η λύση του προβήματος. Αρκεί ένα directional light τύπου sun με τις ρυθμίσεις που φαίνονται στην παραπάνω εικόνα. Σημαντική ρύθμιση το Ambient Occlusion στην καρτέλα world που προσδίδει «φωτισμό περιβάλλοντος». Επίσης, σημαντικό, export ως **.dae** και όχι ως **.fbx**.

---

#### 4.2.2 - AR IMAGE EFFECTS

Κατά την περιγραφή των απαιτήσεων της εφαρμογής, αναφερθήκαμε στην προοπτική, το AR View εκτός από τρισδιάστατα αντικείμενα να εμφανίζει και το πραγματικό περιβάλλον παραμορφωμένο με εφαρμογή φίλτρων. Κατά την ανάπτυξη της εφαρμογής έγινε μία νύξη σε βιβλιοθήκες που θα μπορούσαν να βοηθήσουν την υλοποίηση αυτού του χαρακτηριστικού.

Οι πρώτες δοκιμές έγιναν με εφαρμογή απλούστατων φίλτρων στην εικόνα (πχ Seria, μεταβλητό Hue και Saturation). Χρησιμοποιήθηκε η παρακάτω βιβλιοθήκη από το GitHub:

<https://github.com/androidroadies/android-gpuimage-master/tree/master/android-gpuimage-master>

Οι πραγματικές δοκιμές απέδειξαν ότι η εφαρμογή φίλτρων ενώ η κάμερα λειτουργεί και αλλάζει την εικόνα με 30fps (στο κινητό των δοκιμών) ήταν ευφικτή για μία απλή περιήγηση της κάμερας στο πραγματικό περιβάλλον. Όταν όμως δοκιμάστηκε να συνδιαστεί η εικόνα με κάποιο απλό τρισδιάστατο μοντέλο η εφαρμογή αποκτούσε σοβαρά προβλήματα απόκρισης στο κινητό και μετά από λίγα δευτερόλεπτα τερματιζόταν.

Από ότι φάνηκε η εφαρμογή φίλτρων σε λειτουργία κάμερας δέσμευε μεγάλο ποσοστό της μνήμης του κινητού και της επεξεργαστικής του ισχύος, καθώς το φίλτρο, έστω και απλό έπρεπε να εφαρμοστεί σε 30 εικόνες ανά δευτερόλεπτο, χωρίς να λάβουμε υπόψιν μας τις μεταβλητές του περιβάλλοντος (εναλλαγή φωτισμού, χρωμάτων κατά την περιήγηση της κάμερας σε πραγματικό εξωτερικό περιβάλλον). Αυτό σε συνδιασμό με την επεξεργαστική ισχύ και τη μνήμη που απαιτεί η διαχείριση του περιβάλλοντος AR

(τρισδιάστατα μοντέλα, επανασχεδιασμός τους στο χώρο ανάλογα με την κίνηση του χρήστη) κατέστησε τα Image Effects που σκεφτόμασταν αρχικά απαγορευτικά για την παρούσα εφαρμογή.

### 4.3 - ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Περνώντας από την περιγραφή των απαιτήσεων στην υλοποίηση προέκυψαν πολλές δυσκολίες στην εφαρμογή αυτών που είχαν σχεδιαστεί αρχικά. Στο παρόν κεφάλαιο θα παρουσιαστούν τα μεγαλύτερα από αυτά.

#### 4.3.1 - ΠΡΟΒΛΗΜΑ ΣΤΗΝ ΑΚΡΙΒΕΙΑ ΤΩΝ ΜΕΤΡΗΣΕΩΝ ΤΟΥ GPS

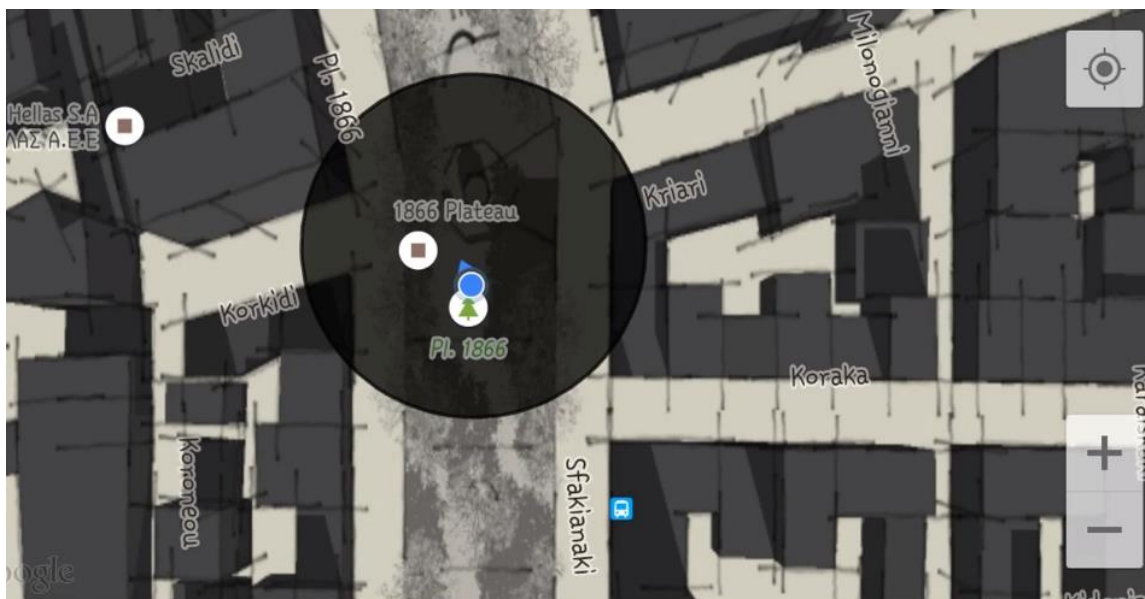
Το περιβάλλον επαυξημένης πραγματικότητας που θέλαμε να δημιουργήσουμε βασιζότανε σε μεγάλο ποσοστό στην ακρίβεια των μετρήσεων που θα συλλεγόntonτουσαν από το GPS του κινητού. Στην πράξη αποδείχθηκε ότι αυτές οι μετρήσεις κάθε άλλο παρά βοηθούσαν αυτόν τον σκοπό.

Ειδικότερα σε διαφορετικές κινητές συσκευές υπήρχε μία απόκλιση 3-5 μέτρα (3 στην πιο σύγχρονη συσκευή, 5 στην λιγότερο σύγχρονη) σε σχέση με την πραγματική θέση της συσκευής και τις συντεταγμένες που επέστρεφε το GPS. Το χειρότερο ήταν ότι η απόκλιση εναλλασσότανε ανά δειγματοληψία μετρήσεων. Δηλαδή στην πρώτη μέτρηση η απόκλιση μπορεί να ήτανε 3 μέτρα Νοτιοανατολικά, ενώ στην δεύτερη 4 μέτρα Βόρεια.

Αυτό για την παρούσα εφαρμογή αποδείχθηκε καταστροφικό καθώς τα τρισδιάστατα μοντέλα είτε εμφανιζόντουσαν σε διαφορετική θέση από αυτήν που θα έπρεπε να φαίνονται, είτε κυρίως, ήταν σε μία διαρκή, κουραστική κίνηση, σε αντίθεση με ότι είχαμε φανταστεί.

Μετά από πλήθος δοκιμών αποφασίστηκε να ακολουθηθεί η τακτική του σημείου-άγκυρας (Mock Location). Εφόσον έχουμε τη βοήθεια του χάρτη οδηγούμε τον χρήστη σε μία περιοχή ακτίνας 2-3 μέτρων και μόνο τότε τον εισάγουμε στο AR View. Τα αντικείμενα εμφανίζονται σε σχέση με την τοποθεσία στην οποία βρίσκεται ο χρήστης, η οποία πλέον είναι πλήρως σταθερή καθώς είναι το κέντρο του κύκλου που αναφέρθηκε προηγουμένως (4B).





(4B) : Το κέντρο του μαύρου κύκλου είναι το «σημείο άγκυρα» περιμετρικά του οποίου εμφανίζονται τα τρισδιάστατα αντικείμενα.

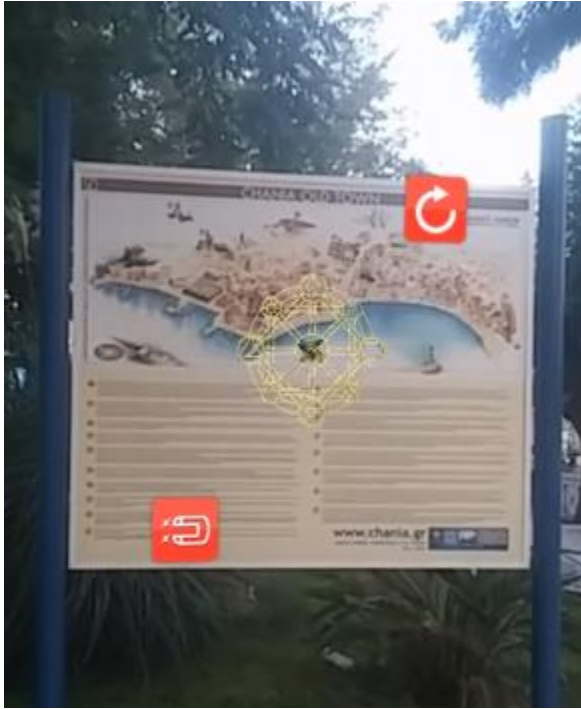
#### 4.3.2 - ΠΡΟΒΛΗΜΑ ΦΥΣΙΚΩΝ ΕΜΠΟΔΙΩΝ

Εκτός από το πρόβλημα που δημιουργήσε η αστάθεια των μετρήσεων του GPS, προέκυψε και το πρόβλημα των φυσικών εμποδίων. Το ότι τα τρισδιάστατα αντικείμενα τοποθετούνται στο χώρο με γεωγραφικές συντεταγμένες προϋποθέτει ότι δε θα εμφανιστεί κάποιο φυσικό εμπόδιο (π.χ. δέντρα, άνθρωποι) μεταξύ της κινητής συσκευής και της γεωγραφικής τοποθεσίας που πρέπει να εμφανιστεί το αντικείμενο.

Αυτό όμως δεν είναι δυνατόν να παραβλεθεί σε πολυσύχναστα μέρη όπως η πλατεία 1866. Στην πράξη, εάν κάποιο φυσικό εμπόδιο βρεθεί ανάμεσα στον θεατή και το τρισδιάστατο αντικείμενο, ενώ θα έπρεπε να κρύψει το εικονικό αντικείμενο από τη γωνία θέασης της κινητής συσκευής το υπερκαλύπτει. Αυτή η υπερκάλυψη δημιούργησε ένα αντιαισθητικό αποτέλεσμα που έπρεπε να αντιμετωπιστεί.

Εδώ αναγκαστήκαμε να αναζητήσουμε λύση αναγνώρισης της θέσης στην οποία πρέπει να εμφανιστεί το εικονικό αντικείμενο με έναν επιπρόσθετο παράγοντα ο οποίος θα επιβεβαιώνει ότι δεν παρεμβάλλεται κάποιο φυσικό εμπόδιο. Η αρχική μας προσέγγιση περιελάμβανε αναγνώριση markers, όμως την εξελίξαμε και με τη βοήθεια της βιβλιοθήκης wikitude περιέλαβε image pattern recognition.

Έτσι ορίζοντας κάποιο σημάδι στο χώρο, για παράδειγμα μια αφίσα ή ένα σήμα οδικής κυκλοφορίας (4γ), δημιουργούμε ένα “pattern” το οποίο μπορεί να αναγνωρίσει το κινητό και να εμφανίσει σε σχετική θέση με αυτό το επιθυμητό εικονικό αντικείμενο.



(4γ) : Η συγκεκριμένη πινακίδα παρέχει ένα πολύ ξεχωριστό “pattern” σε σχέση με το περιβάλλον της(το μπλε του νερού με το άσπρο και το καφέ της πόλης σε αντίθεση με το πράσινο του περιβάλλοντος), το οποίο την καθιστά ιδανική για να χαρτογραφηθεί και να αναγνωριστεί από το κινητό.

### 5.1 CLIENT-SERVER ARCHITECTURE

#### 5.1.1 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ CLIENT-SERVER

Το πληροφοριακό σύστημα της παρούσας διπλωματικής χωρίστηκε σε 2 επίπεδα. Οι σύγχρονες εφαρμογές διαχωρίζουν πλέον τα επίπεδα των λειτουργιών τους. Αυτό επιτρέπει μεγαλύτερη ευελιξία, φορητότητα και ευκολότερη διανομή ρόλων σε ομάδες εργασίας. Τα 2 επίπεδα στα οποία χωρίστηκε το παρών πληροφοριακό σύστημα είναι το Front-end Layer και το Back-end Layer, ορολογίες που θα χρησιμοποιηθούν και στη συνέχεια.

Το Back-end Layer, ένα Web Service, αναλαμβάνει την αποθήκευση και ανάκτηση δεδομένων στη βάση δεδομένων του συστήματος. Το Front-end Layer θα είναι η βασική εφαρμογή, η οποία θα είναι προσβάσιμη από συσκευές Android. Σε αυτήν θα υλοποιείται το κυρίως μέρος της διπλωματικής, το παιχνίδι Augmented Reality δηλαδή. Επίσης, μέσω αυτής οι χρήστες θα έχουν όλη την εμπειρία τους με το πληροφοριακό σύστημα.

Γενικά, το client-server architecture αναφέρεται σε μια βασική αλλαγή στο στυλ λειτουργίας των πληροφοριακών συστημάτων, την αλλαγή από τα συστήματα που βασίζονται στα μηχανήματα στα συστήματα που βασίζονται στον χρήστη.

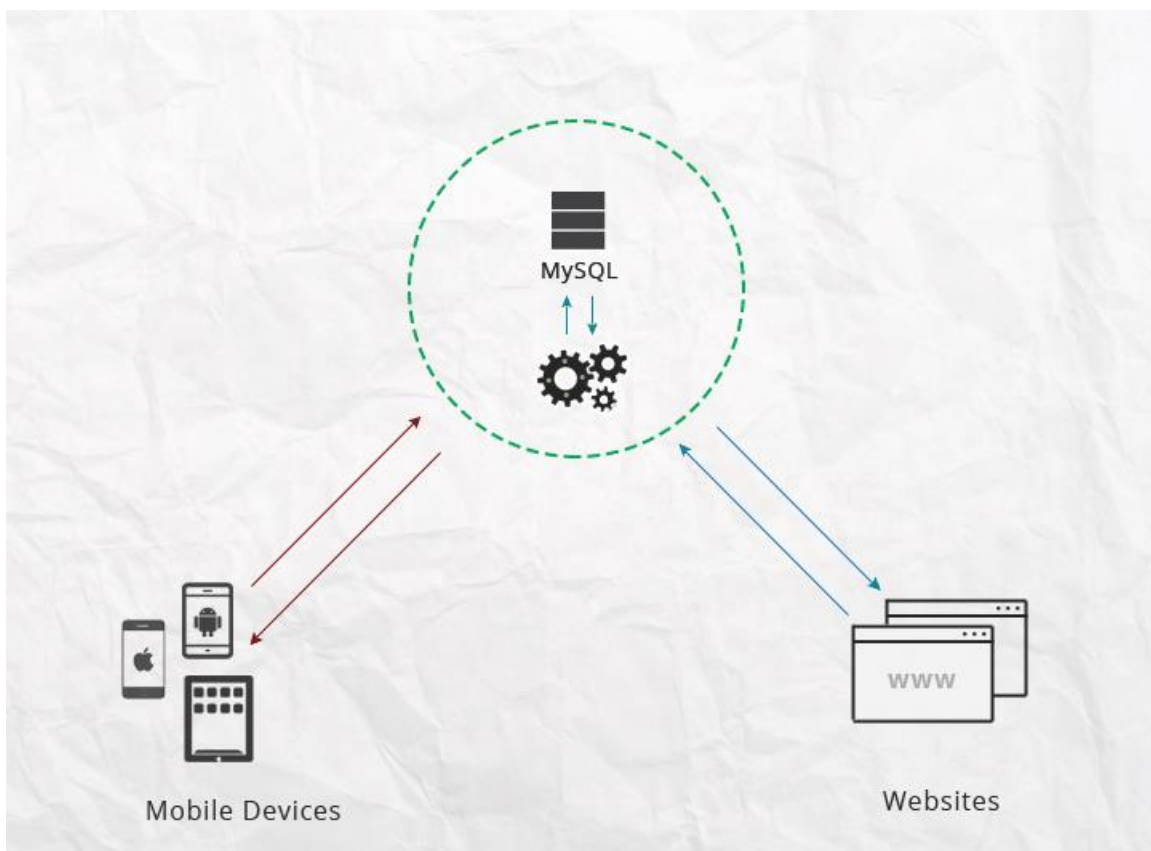
Ειδικότερα, ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients (ή αλλιώς front end) να μπορούν να ζητούν υπηρεσίες από έναν server (ή αλλιώς back end), ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ (5α).

Με άλλα λόγια, στο client-server μοντέλο, ο client(front-end layer) θέτει μια αίτηση και ο server επιστρέφει μια ανταπόκριση ή κάνει μια σειρά από ενέργειες. Ο server(back-end layer) μπορεί να ενεργοποιείται άμεσα για την αίτηση αυτή ή να προσθέτει την αίτηση σε μια ουρά. Η άμεση ενεργοποίηση για την αίτηση μπορεί, για παράδειγμα, να σημαίνει ότι ο server υπολογίζει έναν αριθμό και τον επιστρέφει αμέσως στον client. Η τοποθέτηση της αίτησης σε μια ουρά μπορεί να σημαίνει ότι η αίτηση πρέπει να τεθεί σε αναμονή για να εξυπηρετηθεί. Ένα καλό παράδειγμα για αυτό είναι όταν εκτυπώνουμε

ένα κείμενο σε ένα εκτυπωτή δικτύου. Ο server τοποθετεί την αίτηση σε μια ουρά μαζί με αιτήσεις εκτυπώσεων και από άλλους clients. Μετά επεξεργάζεται την αίτηση με βάση την σειρά προτεραιότητας, η οποία, σε αυτή την περίπτωση, καθορίζεται από τη σειρά με την οποία ο server παρέλαβε την απαίτηση.

Η αρχιτεκτονική client-server είναι πολύ σημαντική, διότι επιτυγχάνει τα εξής:

- Αποτελεσματική χρήση της υπολογιστικής ισχύος.
- Μείωση του κόστους συντήρησης, δημιουργώντας συστήματα client-server που απαιτούν λιγότερη συντήρηση και κοστίζουν λιγότερο στην αναβάθμιση.
- Αύξηση της παραγωγικότητας, προσφέροντας στους χρήστες ξεκάθαρη πρόσβαση στις αναγκαίες πληροφορίες μέσω σταθερών και εύκολων στην χρήση διασυνδέσεων.
- Αύξηση της ευελιξίας και της δυνατότητας δημιουργίας συστημάτων που υποστηρίζουν πολλά περιβάλλοντα.



(5α) : Παράδειγμα επικοινωνίας API με διαφορετικού τύπου Clients.

## 5.2 BACK-END LAYER

Για την ικανοποίηση των απαιτήσεων της εφαρμογής, ήταν αναγκαίο να στήσουμε ένα web service που θα έχει τον ρόλο του μεσολαβητή ανάμεσα στα instances της εφαρμογής και στη βάση δεδομένων.

Στη βάση δεδομένων πρέπει να αποθηκεύονται:

- Τα δεδομένα των λογαριασμών χρήστη
- Η εξέλιξή τους στο παιχνίδι και τα στατιστικά τους στοιχεία
- Δεδομένα άλλων ειδών που υποβοηθούν τη λειτουργία της εφαρμογής

Επομένως, και καθώς θέλουμε να δώσουμε στην εφαρμογή φορητότητα, δηλαδή τα καίρια δεδομένα να μπορούν να είναι προσβάσιμα όχι μόνο από την παρούσα Android εφαρμογή αλλά και από κάποια υλοποίηση που είναι πιθανό να γίνει στο μέλλον, θα διαχωρίσουμε το επίπεδο διακομιστή από το επίπεδο εξυπηρετητή με ένα REST Web Service.

---

### 5.2.1 - WEB SERVICES

Τα web services εμφανίστηκαν το 2001 όταν άρχισε να γίνεται λόγος για αυτο-περιγραφόμενες και αυτο-προσδιοριζόμενες εφαρμογές οι οποίες μπορούν να ανακαλυφθούν και να προσπελαστούν μέσω διαδικτύου από άλλες εφαρμογές.

Οι υπηρεσίες WEB κατάφεραν όχι μόνο να παραμείνουν στο προσκήνιο των εξελίξεων στον χώρο της τεχνολογίας όλη τη δεκαετία, αλλά να αποτελέσουν ένα από τα πιο σημαντικά κεφάλαια της υψηλής τεχνολογίας του Διαδικτύου. Web service είναι οποιοδήποτε τμήμα λογισμικού ή συστατικό εφαρμογής που είναι διαθέσιμο μέσω διαδικτύου και το οποίο μπορεί να ολοκληρώνει εργασίες και να λύνει προβλήματα. Ωστόσο αν θέλουμε να προσδιορίσουμε πιο σωστά την έννοια των web services, θα μπορούσαμε να χρησιμοποιήσουμε τον ορισμό που έρχεται από την IBM :

Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service αποτελεί μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων (XML, JSON). Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα JSON για να περιγράψει μια λειτουργία προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή.

Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services. Με απλά λόγια μια Web service μπορούμε να την αντιληφθούμε σαν ένα λογισμικό που γνωρίζει με ποιόν τρόπο πρέπει να 'μιλήσει' σε διαφορετικού τύπου λογισμικά μέσω δικτύου. Πιο συγκεκριμένα μια υπηρεσία Web πραγματοποιεί τις ακόλουθες λειτουργίες:

1. Παρέχει την δυνατότητα σε άλλες εφαρμογές να κατανοήσουν την λειτουργικότητά της, εκθέτοντας και περιγράφοντας την παρουσία της σε αυτές με τη χρήση ενός πρότυπου δικτυακού πρωτοκόλλου.
2. Μπορεί να γίνει εύκολα αντιληπτή από άλλες εφαρμογές μέσω ενός καταλόγου στον οποίο βρίσκετε καταχωρημένη.
3. Μπορεί να προβληθεί ή να χρησιμοποιηθεί από την εφαρμογή που την χρειάζεται μέσω απλών πρωτοκόλλων.

---

### 5.2.2 - Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ REST

Το REST εμφανίζεται τα τελευταία χρόνια να είναι το ανερχόμενο μοντέλο για web services. Λόγω της απλότητας του, έχει εκτοπίσει σε ένα μεγάλο βαθμό το SOAP και το WSDL, παλαιότερα μοντέλα υλοποίησης web services.

Την αρχιτεκτονική REST, την αποτελούν οι πελάτες και οι εξυπηρετητές. Οι πελάτες ξεκινούν αιτήματα προς τους εξυπηρετητές, οι οποίοι με τη σειρά τους, αφού τα επεξεργαστούν, επιστρέφουν στους πελάτες που τους έστειλαν το αίτημα τις ανάλογες απαντήσεις. Τα αιτήματα και οι απαντήσεις έχουν υλοποιηθεί γύρω από αναπαραστάσεις των πόρων. Πόρος μπορεί να είναι οποιοδήποτε πληροφοριακό αντικείμενο με κάποια

σημασία και η αναπαράστασή του αποτυπώνει ένα στιγμιότυπό του σε μια δεδομένη χρονική στιγμή ή μετά από κάποια ενέργεια που εκτέλεσε.

Ο πρώτος περιορισμός που εφαρμόζουμε χρησιμοποιώντας την αρχιτεκτονική REST είναι ο Client - Server. Ο περιορισμός αυτός, αφορά το διαχωρισμό του user interface (διεπαφή του χρήστη, στην περιπτωσή μας το Android App) από το σύστημα αποθήκευσης των δεδομένων. Με τον τρόπο αυτό βελτιώνουμε τη δυνατότητα επέκτασης ενώ παράλληλα η διεπαφή του χρήστη μπορεί να μεταφερθεί εύκολα σε άλλες πλατφόρμες. Το πιο σημαντικό όμως είναι ότι ο διαχωρισμός αυτός επιτρέπει την αυτόνομη εξέλιξη των συστατικών του (σύστημα αποθήκευσης δεδομένων, Android App).

Στη συνέχεια προσθέτουμε τον περιορισμό ότι η αλληλεπίδραση Client - Server πρέπει να είναι stateless. Με τον όρο stateless εννοούμε ότι κάθε αίτημα από τον Client στον Server πρέπει να περιέχει όλη την απαραίτητη πληροφορία για να γίνει κατανοητό από τον Server, χωρίς να χρησιμοποιεί κάποιο αποθηκευμένο κείμενο στον Server, δηλαδή να μην εξαρτάται από προηγούμενη κατάσταση. Επομένως όλη η κατάσταση - πληροφορία της συνεδρίας (session) αποθηκεύεται στον Client.

Ο συγκεκριμένος περιορισμός επάγει τις ακόλουθες ιδιότητες:

Visibility: βελτιώνεται, γιατί ένα σύστημα παρακολούθησης για να καθορίσει τη φύση του αιτήματος αρκεί να εξετάσει το συγκεκριμένο αίτημα και τίποτα άλλο.

Reliability: βελτιώνεται γιατί είναι ευκολότερη η ανάκαμψη του συστήματος από κάποια αποτυχία Scalability: επειδή δεν είναι απαραίτητη η αποθήκευση της κατάστασης των αιτημάτων, ο Server απελευθερώνει ότι πόρους είχε δεσμεύσει για την εξυπηρέτηση του αιτήματος μετά την ολοκλήρωσή του. Επίσης η υλοποίηση του λογισμικού από την πλευρά του server είναι απλούστερη γιατί δεν απαιτείται διατήρηση πληροφοριών μεταξύ αιτημάτων (Fielding, 2000).

### 5.2.3 - WEB SERVICE RESOURCES (ΠΟΡΟΙ)

Πόρος είναι οτιδήποτε που είναι αρκετά σημαντικό να αναφέρεται σαν ένα πράγμα από μόνο του, να έχει δηλαδή μια ταυτότητα. Είναι συνήθως κάτι που μπορεί να αποθηκευθεί: ένα ηλεκτρονικό αρχείο, μια εγγραφή σε μια βάση δεδομένων, το αποτέλεσμα ενός αλγόριθμου. Όμως πόρος μπορεί να είναι ένας άνθρωπος, ένα χρώμα, μια έννοια κ.λπ.

Ο μόνος περιορισμός που υφίσταται είναι ότι πρέπει να έχει ένα συγκεκριμένο URL.

Τους πόρους μπορούμε να τους διακρίνουμε σε:

- Στατικούς: αν εξεταστεί σε οποιαδήποτε χρονική στιγμή μετά τη δημιουργία του, αντιστοιχεί στην ίδια τιμή που έχει οριστεί.
- Δυναμικούς: αν έχουν υψηλό βαθμό διακύμανσης της τιμής που τους ορίζεται σε σχέση με το χρόνο.

Η διάκριση αυτή μπορεί να γίνει καλύτερα κατανοητή με το ακόλουθο παράδειγμα: Η πρόοδος ενός χρήστη στο παιχνίδι γρίφων θα μπορούσε να υποστεί τακτικές αλλαγές. Αυτό συνιστά έναν δυναμικό πόρο. Αντίθετα τα στοιχεία των επιπέδων και των γρίφων του παιχνιδιού είναι στατικός πόρος διότι δεν μεταβάλλονται στο πέρασμα του χρόνου.

Όπως αναφέραμε στην προηγούμενη παράγραφο, ο άνθρωπος, μια έννοια, μια εγγραφή σε βάση δεδομένων θεωρούνται πόροι αν έχουν URL.

Επομένως όταν ζητάμε, μέσω μιας GET, έναν πόρο, ο Server, πρέπει να μας αποστέλλει μια απάντηση η οποία θα περιέχει, όχι μόνο τα δεδομένα του πόρου αλλά και ένα σύνολο από τα μεταδεδομένα που τα περιγράφουν. Επομένως η αναπαράσταση είναι μια αλληλουχία από bytes με επιπλέον μεταδεδομένα, που περιγράφουν την τρέχουσα ή την αποσκοπούσα κατάσταση ενός πόρου.

Η μορφή των μεταδεδομένων είναι ζευγάρια όνομα - τιμή όπου το όνομα περιγράφει τη δομή και τη σημασιολογία της τιμής (Richardson & Amundsen, 2013)

Όπως αναφέραμε κάθε πόρος έχει ένα μοναδικό αναγνωριστικό, το URI. Για να αποκτήσουμε πρόσβαση στον πόρο πρέπει να χρησιμοποιήσουμε το URI του, με τη



Βοήθεια των HTTP μεθόδων. Για να επιτύχουμε την αλλαγή των οντοτήτων ενός πόρου χρησιμοποιούμε την αρχή CRUD (5β) δηλαδή Create, Retrieve, Update και Delete, ώστε να έχουμε πρόσβαση σε όλες τις βασικές λειτουργίες του πόρου.

HTTP/ REST	CRUD	SQL
POST	Create	INSERT
GET	Retrieve	SELECT
PUT	Update	UPDATE
DELETE	Delete	DELETE

(5β) : Οι εντολές CRUD σε αντιστοίχιση με τις ενέργειες που πραγματοποιούνται στη βάση.

---

#### 5.2.4 - ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Με βάση τις λειτουργίες της εφαρμογής που έχουμε να υλοποιήσουμε ξεκινάμε τον σχεδιασμό του back-end καθορίζοντας τις πηγές(resources) και τα crud routes που θα εξυπηρετούν την κάθε λειτουργικότητα. Ξεκινώντας με την οντότητα των χρηστών υλοποιούμε την παρακάτω σχεδίαση που θα εξυπηρετεί όποια ενέργεια είναι σχετική με τους χρήστες του συστήματος.

## Resource USER, Attributes

---

- Auto increment ID
- String email needed,
- String firstname needed
- String lastname needed
- String password (hashed)
- int team\_id
- int progress
- String facebook\_id
- Deleted\_at date
- Timestamps

## Resource USER, Routes

**/users GET** : Retrieves Collection of users [User, User, User]. If User\_id exists data for specific user are being retrieved.

```
{
```

- User\_id
- String email
- String firstname
- String lastname
- Int team\_id
- Int progress
- Timestamps

```
},{
```

- User\_id
- String email
- String firstname
- String lastname
- Int team\_id
- Int progress
- Timestamps

```
}]
```

**/users POST** : Registers user. Needed attributes:

{

- Int User\_id
- String email needed
- String firstname
- String lastname
- Int team\_id
- Int progress

}

**/users PUT** : Updates user data. Used for team join, progress or user\_info update

{

- Int User\_id needed
- String email
- String firstname
- String lastname
- Int team\_id
- Int progress

}

**/users/auth POST** : Authenticates user, creates a login token, saves it in tokens table

**/users/auth/facebook POST** : Authenticates user via facebook, creates a login token, saves it in tokens table (Token table holds active or last login session for each user)

{

- Int User\_id
- String key (hashed) needed
- String Device\_id needed
- String Device\_OS needed
- String Device\_token (to remember device)
- Timestamps

}

### Resource TEAM, Attributes

- Auto increment team\_ID
- String TeamName
- User leader
- User[] Users
- Progress[] Progress
- Deleted\_at date
- Timestamps

### Resource TEAM, Routes

**/teams GET** : Retrieves Collection of users [User, User, User]. If User\_id exists data for specific user are being retrieved.

```
[[
```

- User\_id
- String key (hashed)
- String firstname
- String lastname
- Timestamps

```
],{
```

- User\_id
- String key (hashed)
- String firstname
- String lastname
- Timestamps

```
]]
```

**/teams POST** : Registers team. Needed attributes:

```
{
```

- String TeamName needed
- User leader needed
- User[] Users

```
}
```

**/teams PUT** : Updates team. Adds user(s) or edits progress. Needed attributes:

```
{  
  • int team_ID needed  
  • String TeamName needed  
  • User leader needed  
  • User[] Users  
  • Progress[] progress  
}
```

**/teams DELETE** : Deletes team. Needed attributes:

```
{  
  • int team_ID needed  
}
```

---

### 5.2.5 - ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η υλοποίηση του back-end θα γίνει χρησιμοποιώντας το PHP framework Laravel.

Για τις ανάγκες της εφαρμογής δεσμεύτηκε ένα VM από την DigitalOcean.com με Ubuntu linux 12.04 για να στεγάσει τον Apache server και την MySQL κομμάτια απαραίτητα για τη λειτουργία της Back-end εφαρμογής

#### Apache Server

Ο **Apache HTTP** γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα url το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού.

Το Laravel είναι ένα μοντέρνο PHP framework το οποίο μπορούμε να χρησιμοποιήσουμε για να κατασκευάσουμε διαδικτυακές εφαρμογές και APIs. Η αρχιτεκτονική του Laravel βασίζεται στο MVC design pattern, ενώ αυτό που το κάνει να ξεχωρίζει από τα υπόλοιπα framework είναι το απλό και διαισθητικό συντακτικό, το οποίο μας επιτρέπει με λίγες γραμμές κώδικα να πετύχουμε πολλά.

Σύμφωνα με τον δημιουργό του, Taylor Otwell, το Laravel θέλει να κάνει το προγραμματισμό διασκεδαστικό και για αυτό δημιουργήθηκε ώστε να είναι :

- **Απλό** - Οι λειτουργίες του Laravel είναι απλές στην κατανόηση και στην υλοποίηση. Με βασικές γνώσης PHP και μικρή εμπειρία σε MVC Patterns μπορεί ο καθένας να υλοποιήσει την εφαρμογή που επιθυμεί.
  - **Κομψό** - Το Laravel βασίζεται στα τρέχοντα standard της βιομηχανίας και σαν αποτέλεσμα δεν απαιτούνται περίπλοκοι κώδικες για να εκτελεστούν λειτουργίες που κανονικά είναι απλές. Σε γενικές γραμμές το Laravel χρειάζεται ελάχιστη παραμετροποίηση για να ρυθμιστεί και να δουλέψει.
  - **Τεκμηριωμένο** - Η τεκμηρίωση του Laravel είναι πλήρης και πάντα ενημερωμένη. Εξίσου σημαντικό για κάποιον που θέλει να μάθει Laravel είναι ότι υπάρχουν πάρα πολλές πηγές με οδηγούς και συμβουλές στο internet και το documentation είναι υπερπλήρες.
-

Πίσω από την Laravel χρησιμοποιήθηκε μία βάση δεδομένων MySQL. Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοικτού κώδικα (relational database management system - RDBMS), που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων.

Επειδή είναι ανοικτού κώδικα (open source), οποιοσδήποτε μπορεί να κατεβάσει τη MySQL και να την διαμορφώσει με βάση τις ανάγκες του, σύμφωνα πάντα με την γενική άδεια χρήσης. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία, και την ευελιξία που παρέχει. Οι περισσότεροι συμφωνούν ωστόσο ότι δουλεύει καλύτερα όταν διαχειρίζεται περιεχόμενο και όχι όταν εκτελεί συναλλαγές. Η MySQL αυτή τη στιγμή μπορεί να λειτουργήσει σε περιβάλλον Linux, Unix, και Windows.

Η σύνδεση μεταξύ τους έγινε χρησιμοποιώντας το configuration της Laravel.

```
'connections' => array(
    'mysql' => array(
        'driver' => 'mysql', 'host' => 'localhost', 'database' => 'natare_db',
        'username' => 'jimmy', 'password' => '1990', 'charset' => 'utf8',
        'collation' => 'utf8_unicode_ci', 'prefix' => "",
    )
);
```

Για τη δημιουργία και την διαχείριση των πινάκων της MySQL χρησιμοποιήθηκε η σύγχρονη τεχνική των **Migrations** που μας προσφέρεται από την Laravel. Σύμφωνα με τα migrations οποιαδήποτε αλλαγή στη δομή της βάσης δεδομένων μας, καταγράφεται σε ένα αρχείο που αποθηκεύεται με το timestamp στο οποίο γίνανε οι αλλαγές. Με αυτόν τον τρόπο μπορούμε ανά πάσα στιγμή να κάνουμε επαναφορά (roll back) στη γλώσσα της Laravel σε μία προηγούμενη κατάσταση της Βάσης μας.

Ένα migration είναι μία PHP κλάση, η οποία αποτελείται από μία συνάρτηση up() η οποία περιέχει της αλλαγές που επιφέρει το αρχείο στη βάση και μία συνάρτηση down() που περιέχει την απαραίτητη εντολή για επαναφορά της βάσης.

```

class CreateUsersTable extends Migration {

    public function up()
    {
        Schema::create('users', function(Blueprint $table)
        {
            $table->increments('_id');
            $table->string('email');
            $table->string('firstname');
            $table->string('lastname');
            $table->string('password');
            $table->string('facebook_id')->nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::drop('users');
    }

}

```

Τα παραπάνω HTTP requests διαχειρίζονται από το routes.php και αντιστοιχούν σε functions στον αντίστοιχο controller για κάθε resource. Για παράδειγμα για το resource USER, υπεύθυνος controller είναι ο UserController.php.



## Routes.php

- `Route::post('users/auth', array('as' => 'v1.users.auth', 'uses' => 'UserController@authenticate') );`
- `Route::post('users/auth/facebook', array('as' => 'v1.users.auth.facebook', 'uses' => 'UserController@authenticateFacebook') );`
- `Route::resource('users', 'UserController', array('only' => array('index', 'store')) );`

Οι 2 πρώτοι handlers είναι τυπικοί HTTP request handlers και ο τρίτος είναι resource handler. Ο resource handler παρέχει τις CRUD functions που αναλύθηκαν παραπάνω.

Ο δεύτερος handler που είναι και ο πιο σημαντικός μιας και χρησιμοποιείται περισσότερο από όλους (η σύνδεση χρήστη μέσω Facebook) αναλύεται παρακάτω.

```
/**
 *      Authenticate a user based on Facebook access token. If the email
address from facebook is already in the database,
 *
 *      the facebook user id will be added.
 *
 *      If not, a new user will be created with a random password and user
info from facebook.
 */
```

```
public function authenticateFacebook() {
    $input = Input::all();

    $validator = Validator::make( $input, User::getAuthFBRules() );
```

Αφού παραλάβουμε τα αντίστοιχα δεδομένα από την αίτηση, τα περνάμε από τη validation συνάρτηση για να βεβαιωθούμε ότι είναι κατάλληλα. Η συνάρτηση θέτει στο `$validator->passes()` αληθή τιμή σε αυτήν την περίπτωση.

```
if ( $validator->passes() ){
    $facebook = new FacebookWrapper();
    $facebook->loginAsUser( $input['access_token'] );
    $profile = $facebook->getMe();
```

```
if ( is_array($profile) && isset($profile['error']) )  
    return json_encode($profile);
```

```
Log::info( json_encode( $profile->toArray() ) );
```

```
$user = User::where('facebook_id', '=', $profile->getId() )->first();
```

```
if ( !($user instanceof User) )
```

```
    $user = User::where('email', '=', $profile->getProperty('email') )->first();
```

Στη συνέχεια ελέγχουμε αν ο χρήστης έχει ήδη εγγραφεί. Αν όχι δημιουργούμε την υπόστασή του και την αποθηκεύουμε στη βάση. Χρησιμοποιούμε τα δεδομένα που έχουμε ανακτήσει από το Facebook.

```
if ( !($user instanceof User) ){  
    // Create an account if none is found  
    $user = new User();  
    $user->firstname = $profile->getFirstName();  
    $user->lastname = $profile->getLastName();  
    $user->email = $profile->getProperty('email');  
    $user->password = Hash::make( uniqid() );  
}
```

```
$user->facebook_id = $profile->getId();
```

```
$user->save();
```

Επίσης, κρατάμε δεδομένα για τον τύπο της συσκευής με την οποία εισέρχεται ο χρήστης. Αυτό θα μας χρησιμεύσει στο να αλλάξουμε τον τύπο της ταυτοποίησης (authentication) εάν χρειαστεί αλλά και για στατιστικούς λόγους.

```

$device_id = Input::has('device_id')? $input['device_id'] : "";
$device_type = Input::has('device_type')? $input['device_type'] : "";
$device_token = Input::has('device_token')? $input['device_token'] : "";

$token = $user->login( $device_id, $device_type, $device_token );

Log::info('<!-- Device Token Received : '. $device_token .' - Device ID
Received : '. $device_id .' for user id: '.$token->user_id);

Log::info('<!-- FACEBOOK Logged : '.$token->user_id.' on '.$token-
>device_os.'['.$token->device_id.'] with token '.$token->token);

$token = $token->toArray();
$token['user'] = $user->toArray();

Log::info( json_encode($token) );

Στο τέλος επιστρέφουμε σε περίπτωση επιτυχίας το κουπόνι που παραλάβαμε κατά τις
συναλλαγές με το Facebook. Σε περίπτωση αποτυχίας η validation συνάρτηση επιστρέφει
λεπτομέρειες για το σφάλμα.

return ApiResponse::json($token);
}

else {

    return ApiResponse::validation($validator);

}

}

```

## 5.3 FRONT-END LAYER

### 5.3.1 - DEVELOPING A MOBILE APPLICATION

Βασικός σκοπός στην υλοποίηση της συγκεκριμένης εφαρμογής ήταν εξ'αρχής η επεκτασιμότητα. Με τις πρώτες δοκιμές διερευνήθηκαν τρόποι η εφαρμογή να μην περιοριστεί μόνο σε μία πλατφόρμα (Android). Έτσι, μελετήθηκαν οι πιθανές υλοποιήσεις. Βασικό δεδομένο της μελέτης ήταν ότι η βιβλιοθήκη AR που χρησιμοποιήσαμε είναι υλοποιημένη σε Java (Native Android).

#### Native Applications - Εγγενείς Εφαρμογές

Με τον όρο εγγενή εφαρμογή (native application) εννοούμε μια εφαρμογή η οποία έχει αναπτυχθεί είτε για συγκεκριμένη έξυπνη συσκευή, είτε (πιο συχνά) για συγκεκριμένο λειτουργικό σύστημα το οποίο προορίζεται για μια ομάδα έξυπνων συσκευών. Η εγγενής εφαρμογή, περιέχει όλα τα απαραίτητα στοιχεία, προκειμένου να εγκατασταθεί στην έξυπνη συσκευή του χρήστη, και να είναι πλήρως λειτουργική για το σκοπό για τον οποίο προορίζεται. Πρόκειται για εφαρμογή η οποία έχει γραφεί σε μία συγκεκριμένη γλώσσα προγραμματισμού, η οποία είναι απόλυτα συμβατή με το εκάστοτε λειτουργικό όπως για παράδειγμα η αντικειμενοστραφής C για το λειτουργικό σύστημα IOS, και η Java για το λειτουργικό σύστημα Android.

Η ανάπτυξη της εφαρμογής σε Native Java είναι η πιο προφανής υλοποίηση, όμως δε θα παρέχει καμία πιθανότητα επεκτασιμότητας. Η εφαρμογή θα λειτουργεί στην πλατφόρμα του Android και μόνο, και σε περίπτωση πιθανής επέκτασης θα πρέπει να ξαναγραφτεί από την αρχή.

#### Web-based mobile Applications - Διαδικτυακές Εφαρμογές

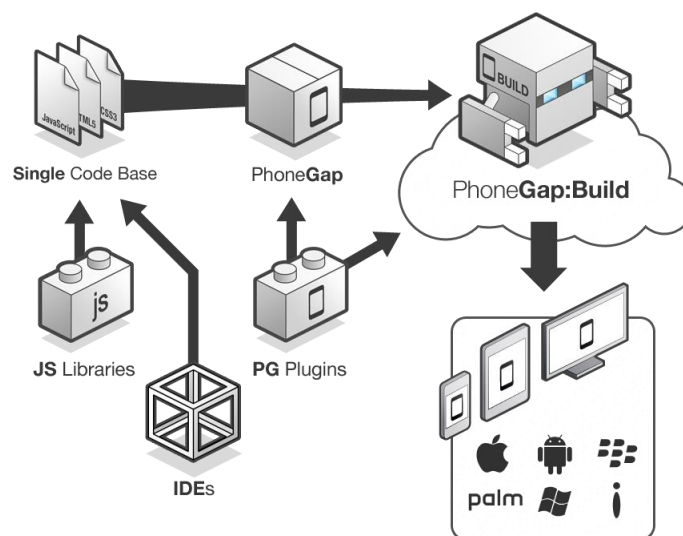
Από την άλλη πλευρά υπάρχουν οι εφαρμογές διαδικτύου (web applications) οι οποίες πρόκειται για εφαρμογές όπου το περιεχόμενο είναι προσβάσιμο από το κινητό, όταν υπάρχει σύνδεση στο διαδίκτυο και πρόκειται για εφαρμογές οι οποίες είναι συνήθως γραμμένες σε HTML με χρήση JavaScript και αναφέρονται κατά κανόνα στο σύνολο των λειτουργικών συστημάτων των διαφορετικών εταιρειών, μέσω συνήθως των περιηγητών που η κάθε συσκευή διαθέτει (Safari, Opera mini, Google περιηγητής)

Ωστόσο δεν παρέχουν εύκολη πρόσβαση στα στοιχεία του κινητού που απαιτεί μια εφαρμογή AR και το framework του HitLab (πρόσβαση σε gps, accelerometer, camera, κώδικας Java) είναι αδύνατο να χρησιμοποιηθεί σε μία τέτοια εφαρμογή.

### Hybrid Applications - Υβριδικές Εφαρμογές

Τα τελευταία χρόνια έχουν εμφανιστεί τρόποι συνένωσης των δύο παραπάνω τύπων εφαρμογών, που μπορούν να χαρακτηριστούν ως «υβρίδια» ή «υβριδικές» εφαρμογές, καθώς φαίνεται να χρησιμοποιούν στοιχεία και των δύο ως προς τον τρόπο ανάπτυξης του πηγαίου κώδικα, με αποτέλεσμα να είναι σχετικά δύσκολη η κατηγοριοποίησή τους.

Ένα χαρακτηριστικό παράδειγμα αποτελεί το Phonegap. Με το Phonegap, ο προγραμματιστής έχει τη δυνατότητα, να δημιουργήσει εγγενείς εφαρμογές (native applications) για τα συγκεκριμένα λειτουργικά συστήματα για τα οποία ενδιαφέρεται. Η ιδιαιτερότητα ωστόσο που κάνει το Phonegap, αλλά και άλλες πλατφόρμες που βασίζονται στην ίδια φιλοσοφία, να διαφέρουν, και να χαρακτηρίζονται οι εφαρμογές που δημιουργούνται με τη χρήση αυτών των εργαλείων, ως «υβριδικές» οφείλεται στον τρόπο ανάπτυξης των εφαρμογών (5γ) που προκύπτουν από τη χρήση των συγκεκριμένων εργαλείων.



(5γ) : Η διαδικασία χτισίματος μιας εφαρμογής Phonegap.

Πιο συγκεκριμένα, το Phonegap, «καταφέρνει» να παρέχει εγγενείς εφαρμογές χρησιμοποιώντας ένα χαρακτηριστικό που είναι κατ' εξοχήν γνώρισμα των διαδικτυακών εφαρμογών (web applications). Αντίθετα με τη δημιουργία των κατεχοχών εγγενών εφαρμογών, όπου γίνεται χρήση συγκεκριμένων γλωσσών προγραμματισμού, η δημιουργία μιας εγγενούς εφαρμογής με το Phonegap, δεν απαιτεί τη χρήση συγκεκριμένης γλώσσας προγραμματισμού, «συμβατής» με το εκάστοτε λειτουργικό σύστημα, αλλά μόνο χρήση HTML, JavaScript και CSS, γλώσσες προγραμματισμού δηλαδή οι οποίες προσδιορίζουν κατά κύριο λόγο την ανάπτυξη διαδικτυακών εφαρμογών (web applications).

Άλλες πλατφόρμες «υβριδικής» ανάπτυξης εφαρμογών είναι οι:

- Sencha: ένα πλαίσιο ανάπτυξης (framework), με το οποίο μπορεί ο χρήστης να δημιουργήσει εφαρμογές για έξυπνες συσκευές, χρησιμοποιώντας τη γλώσσα προγραμματισμού HTML5.
- Titanium: το εμπορικό πακέτο της εταιρείας Appcelerator (Appcelerator, 2012) το οποίο επιτρέπει στους προγραμματιστές να δημιουργήσουν εγγενείς εφαρμογές. Το πακέτο ανάπτυξης διατίθεται δωρεάν, ωστόσο απαιτείται εγγραφή του χρήστη για να το «κατεβάσει».

---

### 5.3.2 CHOOSING THE BEST SOLUTION

Τα σημεία στα οποία συγκρίθηκαν οι παραπάνω πιθανές υλοποιήσεις είναι τα εξής:

1. Πρόσβαση στη συσκευή: με τον όρο πρόσβαση στη συσκευή, αναφερόμαστε στη δυνατότητα μιας εφαρμογής, να έχει πρόσβαση σε συγκεκριμένες λειτουργίες που ενσωματώνει η έξυπνη συσκευή, όπως για παράδειγμα η κάμερα το gps, το accelerometer, στοιχεία ζωτικής σημασίας για μία εφαρμογή AR. Τόσο οι εγγενείς εφαρμογές όσο και οι υβριδικές, έχουν πλήρη πρόσβαση σε αυτές τις λειτουργίες, ενώ οι web-based εφαρμογές έχουν περιορισμένη πρόσβαση, οπότε λόγω της κρισιμότητας των μετρήσεων αποκλείονται.
2. Ταχύτητα: αναφέρεται στο πόσο γρήγορα μπορεί να υλοποιήσει η εφαρμογή τα αιτήματα του χρήστη. Η εγγενής εφαρμογή έχει πολύ γρήγορη ταχύτητα, καθώς

εκμεταλλεύεται πλήρως τις δυνατότητες της συσκευής, καθώς είναι γραμμένη σε γλώσσα απόλυτα συμβατή με το λειτουργικό σύστημα της έξυπνης συσκευής. Αρκετά ικανοποιητική ταχύτητα παρουσιάζουν και οι υβριδικές εφαρμογές, ενώ οι διαδικτυακές εφαρμογές, έχουν μια μικρή υστέρηση.

3. Επεκτασιμότητα. Το γεγονός ότι μια καθαρά εγγενή εφαρμογή πρέπει να γραφεί σε συγκεκριμένη γλώσσα προγραμματισμού που προορίζεται για συγκεκριμένη έξυπνη συσκευή, έχει σαν συνέπεια το να γίνεται δύσκολη και ιδιαίτερα χρονοβόρα η πιθανή μεταφορά της σε άλλου είδους πλατφόρμα. Αντίθετα, η δυνατότητα της υβριδικής ανάπτυξης εφαρμογών, με κοινό πηγαίο κώδικα, όπου μπορεί να παραχθεί το κατάλληλο εκτελέσιμο αρχείο για την πλειοψηφία των λειτουργικών συστημάτων, έχει ως αποτέλεσμα το κόστος να διατηρείται σε «λογικά» επίπεδα.
4. Συμβατότητα με τη βιβλιοθήκη AR. Βασικό κριτήριο στην επιλογή του τύπου ανάπτυξης της εφαρμογής, ήταν το ότι τουλάχιστον ένα κομμάτι της εφαρμογής θα πρέπει να υλοποιηθεί αποκλειστικά σε Native Java καθώς σε αυτήν την γλώσσα είναι γραμμένο το Framework που χρησιμοποιούμε στο περιβάλλον γρίφου. Αυτό αυτόματα αποκλείει τη πιθανότητα ανάπτυξης της εφαρμογής σε mobile browser. Μετά από εκτενή μελέτη στις δυνατότητες που προσφέρει το phonegap, ανακαλύφθηκε ότι ο Java-based κώδικας που θα διαχειρίζεται το περιβάλλον γρίφου μπορεί να ενσωματωθεί στην εφαρμογή ως **plugin**.

Στα πλαίσια της παρούσας διπλωματικής, επιλέχθηκε η ανάπτυξη εφαρμογής με χρήση της πλατφόρμας Phonegap (υβριδική εφαρμογή). Πιο συγκεκριμένα, κατέστη πράγματι δυνατή η πρόσβαση στις εγγενείς λειτουργίες της έξυπνης συσκευής, ενώ ο χρόνος εκτέλεσης της εφαρμογής ήταν αρκετά ικανοποιητικός. Το μεγαλύτερο πλεονέκτημα της υβριδικής ανάπτυξης είναι όμως η επεκτασιμότητα.

Στην παρούσα εφαρμογή αποφασίστηκε να υλοποιηθεί το βασικό (AR) μέρος της εφαρμογής σε native Java (στην οποία είναι γραμμένο και το framework που θα χρησιμοποιηθεί) , και η υπόλοιπη λειτουργικότητα θα αναπτυχθεί ως υβριδική εφαρμογή. Στο τέλος, ο εγγενής κώδικας θα ενσωματωθεί στον υβριδικό ως Android plugin. Έτσι θα υπάρχει η δυνατότητα, εάν κάποια στιγμή αναπτυχθεί κάποιο AR

framework σε native γλώσσα άλλης πλατφόρμας, να ενσωματωθεί στην εφαρμογή ως plugin και να την κάνει λειτουργική και εκεί.

---

### 5.3.3 PHONEGAP

Η μεγάλη ανάπτυξη στον τομέα των κινητών τηλεφώνων και συγκεκριμένα των έξυπνων τηλεφώνων (smartphones) που κινείται με ιλιγγιώδεις ταχύτητες τα τελευταία χρόνια, δημιούργησε μια τεράστια ζήτηση για ενσωμάτωση νέων χαρακτηριστικών και νέων δυνατοτήτων στα κινητά τηλέφωνα. Αυτό είχε σαν αποτέλεσμα πολλές εταιρείες να αρχίσουν να αναπτύσσουν εφαρμογές για κινητά τηλέφωνα, προκειμένου να διεκδικήσουν μερίδιο στη νέα αυτή αγορά για να κερδίσουν νέους πελάτες, αλλά και να αυξήσουν την ικανοποίηση ενδεχομένως των πελατών που ήδη είχαν, από την πιο «παραδοσιακή» μέχρι τώρα παρουσία τους.

Τα έξυπνα κινητά τηλέφωνα, όπως και οι υπολογιστές, έχουν ένα λειτουργικό σύστημα, προκειμένου να λειτουργήσουν, και πάνω στο οποίο αναπτύσσονται και εγκαθίστανται οι εκάστοτε εφαρμογές. Σε αντίθεση όμως με τους υπολογιστές, όπου το μεγαλύτερο μέρος των χρηστών χρησιμοποιούν ηλεκτρονικού υπολογιστές με το λειτουργικό σύστημα Windows, και ακολουθούν σε πιο περιορισμένη έκταση η Apple με το MacOS και το Linux, στα κινητά ο ανταγωνισμός προς το παρόν είναι πολύ πιο έντονος, όπου πέρα από τις αρκετές διαφορετικές εταιρείες κατασκευής τηλεφώνων, συναντάμε και ποικίλα λειτουργικά συστήματα, όπως το Android της Google, το iOS της Apple, το BlackBerry, το Windows Mobile της Microsoft, το WebOS της Hewlett Packard, και το Symbian.

Η ύπαρξη τόσων πολλών διαφορετικών λειτουργικών συστημάτων, όπου το κάθε ένα έχει συγκεκριμένες απαιτήσεις αναφορικά με τις εφαρμογές που θα γραφούν για να λειτουργήσουν πάνω σε αυτό, αλλά και την απαίτηση για ειδικά εργαλεία ανάπτυξης, έχει σαν αποτέλεσμα, πολλές φορές να χρειάζεται η ανάπτυξη της ίδιας εγγενούς εφαρμογής (native application) να γίνεται από την αρχή, καθώς υπάρχει ένα χάσμα (gap), ανάμεσα στα λειτουργικά αυτά συστήματα και στις απαιτήσεις τους.



Το χάσμα αυτό που υπάρχει στα λειτουργικά συστήματα που χρησιμοποιούνται στα κινητά τηλέφωνα έρχεται να καλύψει το PhoneGap. Το Phonegap, είναι μια «πλατφόρμα» ανοιχτού κώδικα, όπου ενσωματώνει ανοιχτά πρότυπα και τεχνολογίες διαδικτύου προκειμένου να γεφυρώσει το κενό που υπάρχει ανάμεσα στα λειτουργικά συστήματα των διαφόρων εταιρειών.

Ο προγραμματιστής που χρησιμοποιεί το Phonegap, χρειάζεται μόνο να γράψει τον κώδικά του, με τη χρήση HTML5, CSS και JavaScript. Στη συνέχεια, από το πακέτο ανοιχτού κώδικα του Phonegap, χρησιμοποιεί το κατάλληλο αρχείο που προορίζεται για το συγκεκριμένο λειτουργικό σύστημα το οποίο στοχεύει ο προγραμματιστής, και με τα ειδικά εργαλεία ανάπτυξης, που διαθέτουν οι εταιρείες των αντίστοιχων λειτουργικών συστημάτων γίνεται η δημιουργία (compile) της εκτελέσιμης εφαρμογής. Προκειμένου να δημιουργηθεί ένα εκτελέσιμο αρχείο για λειτουργικό σύστημα άλλης εταιρείας, ο προγραμματιστής το μόνο που έχει να κάνει, είναι να ενσωματώσει το αντίστοιχο αρχείο από το πακέτο του Phonegap, και με το αντίστοιχο επίσης εργαλείο ανάπτυξης, να δημιουργήσει (compile) την εφαρμογή, χωρίς την ανάγκη να ξαναγράψει από την αρχή το πρόγραμμα, εφόσον ο πηγαίος κώδικας (HTML5, CSS, JavaScript) παραμένει ίδιος.

Όσον αφορά κομμάτια τα οποία απαιτούν native υλοποίηση, όπως στην περίπτωση μας το AR περιβάλλον, το phonegap δίνει τη δυνατότητα ενσωμάτωσης native κώδικα στην υβριδική εφαρμογή με την μορφή “επεκτάσεων” (plugins).

## 5.4 - ΔΟΜΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Όπως περιγράφηκε παραπάνω, ο κινητήριος κώδικας του phonegap είναι σε βασικές γλώσσες του Web Development (HTML5, CSS, JavaScript). Σε σημεία διάδρασης με στοιχεία της συσκευής(πχ AR mode) υπάρχει διάδραση με native κώδικα όμως το κινητήριο κομμάτι της εφαρμογής δομείται ως εξής.

Ο φάκελος www είναι ο φάκελος που περιέχει τη ραχοκοκκαλιά της εφαρμογής. Το index.html είναι το αρχείο μέσα στο οποίο τρέχει η javascript και παρουσιάζει το περιεχόμενο στον χρήστη, όμως η δομή του είναι ιδιαίτερα απλοϊκή.

```
<!DOCTYPE html>

<html>

<head>

</head>

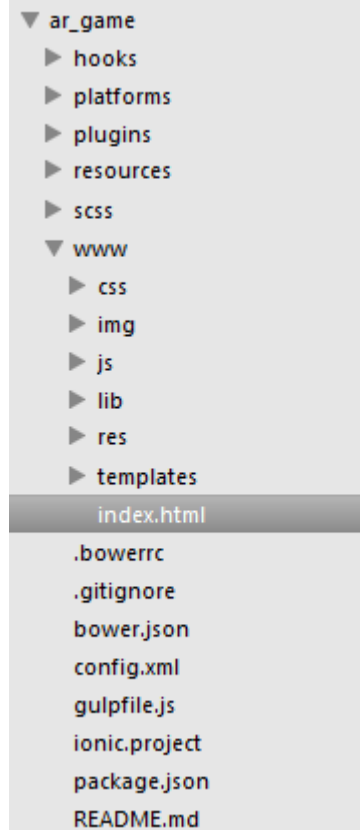
<body ng-app="argame">

  <ion-nav-view></ion-nav-view>

</body>

</html>
```

Η κινητήριος βιβλιοθήκη στο χειρισμό δεδομένων μέσα στην εφαρμογή (προβολή templates, εναλλαγή οθονών, transitions) είναι η **AngularJS**.



---

#### 5.4.1 - ANGULAR JS

Η AngularJS είναι ένα ευρέως διαδεδομένο Javascript framework το οποίο χρησιμοποιεί μια μοναδική φιλοσοφία κώδικα για την ορθή οργάνωση και δόμηση μιας εφαρμογής Web ή Mobile. Η Angular καθιστά ευκολότερο να αναπτυχθεί και να παραμετροποιηθεί, μόνο με JavaScript, μια Front-end εφαρμογή πέρνοντας το βάρος πολύπλοκων λειτουργιών από τη μεριά του Server.

Η AngularJS βελτιώνει δυνατότητες διαχείρισης σε ένα Front-end App, παρέχοντας καλύτερα εργαλεία χειρισμού, όσον αφορά το δυναμικό περιεχόμενο και τις templated διατάξεις.

Δημιουργήθηκε ειδικά για να ασχοληθεί με την HTML 5, και η AngularJS έχει σήμερα καταλήξει να είναι ένα από τα πλέον σύγχρονα και πολυχρησιμοποιημένα πλαίσια JS στον κόσμο, χάρη κυρίως στην ομάδα dev της Google που συνέβαλε τόσο κώδικα, τους πόρους, και την τεκμηρίωση του έργου.

---

#### 5.4.2 - MAIN APPLICATION

Ο ορισμός και η διαχείριση των components της Angular γίνεται μέσα σε ένα κεντρικό αρχείο που ονομάζεται app.js. Το app.js περιλαμβάνει δομικές επιλογές αρχικοποίησης της εφαρμογής (π.χ. αναζήτηση του τύπου της συσκευής) και τις δηλώσεις για κάθε state.

**App.js**

```
var app = {  
  browser: true,  
  
  fbToken:  
'CAAHLISoy91MBAPvp1msBPy7AuT9WZCZAsrZA4GjEZAdPyOaoHYaZAU9F0WbmMTXhWnDjMG  
fSsLITYyGFlyPoNO0Gbj1rtIDtlvsMjLLZCmLJZAqOMgcpOUv3tsJZCwEwhGB35rU8zFvjTtjJ63GO  
IHbbtgP3TCLXOPB8IF3KnCcpZCqPHIW3PVF6q88GsyqcRbYWpaeGTkEHytlqmsZCOX522',  
  
  selectedDate : new Date(),  
  
  apiUrl : "http://188.166.21.108/natare_API/server.php/api/v1",  
  
  deviceType : 'Browser',
```

```
deviceId : 'TEST'
```

```
}
```

Βλέπουμε ότι αρχικοποιούνται τα βασικά στοιχεία της εφαρμογής όπως το fbToken και το apiUrl όπου βρίσκεται ο Apache server μας από τον οποίο τραβάμε δεδομένα.

Παρακάτω έχουμε κάποιο απαραίτητο configuration και τη δήλωση των states.

State ονομάζουμε κάθε διαφορετική κατάσταση της εφαρμογής που συνοδεύεται από προβολή συγκεκριμένης οθόνης (template) σε συνδυασμό με τον αντίστοιχο controller που διαχειρίζεται το template.

```
angular.module('argame', ['ionic', 'argame.services', 'argame.controllers', 'ngCordova'])
```

```
.run(function($ionicPlatform, $cordovaDevice) {
```

```
    $ionicPlatform.ready(function() {
```

```
        // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
```

```
        // for form inputs)
```

```
        if(window.cordova && window.cordova.plugins.Keyboard) {
```

```
            cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
```

```
        }
```

```
        if(window.StatusBar) {
```

```
            StatusBar.styleDefault();
```

```
        }
```

```
        if(!app.browser){
```

```
            app.deviceType = $cordovaDevice.getPlatform() + $cordovaDevice.getModel();
```

```
            app.deviceId = $cordovaDevice.getUUID();
```

```
        }
```

```
});  
})  
  
.config(function($stateProvider, $urlRouterProvider){  
  
    $stateProvider  
  
    .state('app', {  
        url: "/app",  
        abstract: true,  
        cache: false,  
        templateUrl: "templates/menu.html",  
        controller: 'AppCtrl'  
    })  
  
    .state('intro', {  
        url: '/intro',  
        templateUrl: 'templates/intro.html',  
        controller: 'IntroController'  
    })  
  
    .state('app.home', {  
        url: '/home',  
        cache: false,  
        views: {  
            'menuContent': {  
                templateUrl: 'templates/home.html',  
                controller: 'HomeController'  
            }  
        }  
    })  
});
```

```
}  
}  
  
}) ...  
});
```

### 5.4.3 - TEMPLATES - VIEW

Οι οθόνες της εφαρμογής στη δομή της Angular μεταφράζονται ως templates. Οπότε περνάμε στην υλοποίηση των UIs ως templates έχοντας κατά νου τα συμπεράσματα που βγήκανε από το Κεφάλαιο 3.

Templates/

- Intro.html
- Home.html
- Settings.html
- Teams.html
- Map.html
- Camera.html

Παρακάτω παρουσιάζεται μία μορφή template.

**home.html**

```
<ion-view view-title="AR Game" >
  <ion-nav-bar class="bar-assertive-900" ng-class="{expanded: isExpanded, 'has-header-fab-left': hasHeaderFabLeft}" align-title="left">
    <ion-nav-buttons side="left">
      <button class="button button-icon button-clear ion-navicon" menu-
toggle="left">
    </button>
    </ion-nav-buttons>
    <ion-nav-buttons side="right" >
      <button class="button button-icon button-clear ion-android-notifications-none"
ui-sref-active="current" ui-sref="app.notifications"></button>
    </ion-nav-buttons>
  </ion-nav-bar>
```

Η δομή του template είναι κλασσική html. Κάποια tags που περιέχουν το ακρόνυμο ion- είναι ειδικά tags της Angular που παρέχουν προκαθορισμένα CSS rules τα οποία κάνουν το UI πιο εύχρηστο.

```
<ion-content scroll="false" >
  <div class="app" >
    <div class="test" style="font-family: 'Indie Flower';">

    </div>
  <ion-scroll direction="y" style="height:90vh;">
```

```
<div class="list" style="height:auto;">
```

Η AngularJS παρέχει και templating βιβλιοθήκη. Εδώ ετοιμάζουμε στον αντίστοιχο Controller (θα μιλήσουμε παρακάτω για τους Controllers) το array “posts”, το οποίο περιέχει τα post τα οποία ενδιαφέρουν τον χρήστη, και έχουν ανακτηθεί από το API. Η εντολή ng-repeat βάζει το div με κλάση card σε έναν βρόγχο επανάληψης και δημιουργεί τον κώδικα που είναι απαραίτητος για την προβολή των posts στον χρήστη.

```
<div class="card light-bg" ng-repeat="post in posts">  
  <div class="item item-avatar">
```

Τα attributes του εκάστοτε post, μπορούν να χρησιμοποιηθούν με διπλές αγκύλες `{{}}` με τον ακόλουθο τρόπο.

```
      
    <h2>{{post.user.Name}}</h2>  
    <p>{{post.date}}</p>  
  </div>  
  <div class="item item-body" style="padding-top:0px;">  
    <h4>{{post.description}} </h4>  
      
  
    <p>  
      <a class="subdued">{{post.likes}} Likes</a>  
      <a class="subdued">{{post.comments}} Comments</a>  
    </p>  
  </div>  
  <div class="item tabs tabs-secondary tabs-icon-left">  
    <a class="tab-item" data-ink-color="#336699" data-ink-opacity="0.5">  
      <i class="icon ion-thumbsup"></i>  
      Like  
    </a>  
    <a class="tab-item" data-ink-color="#336699" data-ink-opacity="0.5">  
      <i class="icon ion-chatbox"></i>  
      Comment  
    </a>  
    <a class="tab-item" data-ink-color="#336699" data-ink-opacity="0.5">  
      <i class="icon ion-share"></i>  
      Share  
    </a>  
  </div>  
  
</div>  
  
</ion-scroll>
```



```
        </div>

    </ion-content>

</ion-view>
```

Η δομή του template είναι κλασσική html και το styling (fonts, χρώματα κλπ. γίνεται με Cascading Style Sheets CSS)

Πχ για το συγκεκριμένο template μερικοί κανόνες css ακολουθούν παρακάτω.

#### **Style.css**

```
.list > .item {
    font-size: 22px;
    font-family: 'Indie Flower';
}

.item-content > .icon , .item > .icon{
    margin-right: 10px;
}

h2.ng-binding{
    font-family: 'Indie Flower' !important;
    font-size:20px !important;
    margin-top: 20px !important;
    font-weight: bold;
    line-height: 10px;
}
```

```
.item-avatar{  
    padding-bottom:0px;  
    padding-left: 70px !important;  
    min-height:60px !important;  
}
```

```
.item-avatar > h2{  
    font-family: 'Indie Flower';  
}
```

```
.item-avatar > p{  
    font-family: 'Indie Flower';  
}
```

```
.item-body > h4{  
    font-family: 'Indie Flower';  
    font-size:14px;  
}
```

```
.title-left{  
    padding-top: 5px;  
    font-size: 20px !important;  
}
```

#### 5.4.4 - CONTROLLERS

Κάθε ξεχωριστό View έχει τον δικό του controller ο οποίος διαχειρίζεται τις ενέργειες του χρήστη μέσα στο View. Για παράδειγμα εάν πατηθεί κάποιο κουμπί επιλέγεται η αντίστοιχη function του controller.

Ο controller περιέχει και απαραίτητες μεταβλητές για την δημιουργία του View πχ τα δεδομένα που συνθέτουν το Feed για την κεντρική οθόνη.

Παρακάτω παρουσιάζεται το μέρος του Controllers.js που υλοποιεί τον controller της εισαγωγικής οθόνης.

Σαν ορίσματα περνάμε απαραίτητα στοιχεία της Angular που θα χρησιμοποιήσουμε όπως επίσης και εισάγουμε την απαραίτητη για το social login βιβλιοθήκη Facebook SDK.

Σε σημεία που μπορεί να προκύψει σφάλμα ο χρήστης ενημερώνεται με χρήση του `$ionicPopup` το οποίο εμφανίζει ένα `Popup` με το αντίστοιχο μήνυμα.

Έχει υλοποιηθεί ο απαραίτητος έλεγχος για πρόσβαση στο διαδίκτυο με χρήση του σχετικού component της Angular `cordovaNetwork`.

```
.controller('IntroController', function($scope, $state, $ionicPopup, $cordovaFacebook, $ionicPlatform,
```

```
    $cordovaNetwork, $ionicHistory, userService){
```

```
    $ionicPlatform.ready(function() {
```

```
        if(!app.browser){
```

```
            if($cordovaNetwork.isOffline()){
```

Χρησιμοποιώντας το directive `$ionicPopup` μπορούμε να εμφανίσουμε μηνύματα πληροφοριών στον χρήστη.

```
        $ionicPopup.alert({
```

```
            title: 'AR Game',
```

```
            template: 'You have no Internet connection.'
```

```
        });
```

```
    }else{
```

```
        $cordovaFacebook.getLoginStatus()
```

```

        .then(function(response) {

            if(response.status==="connected"){

                getFBtoken();

            }

        }, function (error) {

        });

    }

}

});

```

```

$scope.fbLogin = function(){

```

Ο κώδικας έχει φτιαχτεί προκειμένου να μπορεί να τίθεται υπό δοκιμές και σε περιβάλλον desktop browser. Για να γίνει αυτό έχει φτιαχτεί μία μεταβλητή (app.browser) η οποία όταν τίθεται αληθής παρακάμπτει τα σημεία του κώδικα που απαιτούν πρόσβαση σε μέρη μιας κινητής συσκευής. Εδώ ελέγχουμε αν η κινητή συσκευή έχει δυνατότητα πρόσβασης στο διαδίκτυο είτε μέσω περιαγωγής δεδομένων είτε μέσω Wifi (\$cordovaNetwork). Ο έλεγχος παρακάμπτεται αν η μεταβλητή app.browser είναι αληθής.

```

if(!app.browser && $cordovaNetwork.isOffline()){

    $ionicPopup.alert({

        title: 'AR Game',

        template: 'You have no Internet connection.'

    });

}else{

    if(app.browser){

        userService.loginUser().then(function(user) {

```

```

        //Hotfix for edge threshold

        ARuser.name = user.firstname + " " + user.lastname;

        ARuser.fbid = user.facebook_id;

        $ionicHistory.nextViewOptions({
            historyRoot: true
        });
        $state.go('app.home');

    });

} else {
    FBlogin();

}

}

}

```

Η συνάρτηση FBlogin() και getFBtoken() χρησιμοποιούν συναρτήσεις του [Facebook SDK](#), το οποίο έχει συμπεριληφθεί στην εφαρμογή.

```

function FBlogin(){
    $cordovaFacebook.login(["public_profile", "email", "user_friends"])
        .then(function(success) {
            getFBtoken();

        }, function (error) {

            // Facebook returns error message due to which login was cancelled.

            // Depending on your platform show the message inside the appropriate UI widget

            // For example, show the error message inside a toast notification on Android

```

```

        $ionicPopup.alert({
            title: 'AR Game',
            template: 'Error.'
        });

    });
}

function getFBtoken(){
    $cordovaFacebook.getAccessToken()
        .then(function(success) {
            app.fbToken = success;
            userService.loginUser().then(function(user) {
                //Hotfix for edge threshold
                ARuser.name = user.firstname + " " + user.lastname;
                ARuser.fbid = user.facebook_id;

                $ionicHistory.nextViewOptions({
                    historyRoot: true
                });
                $state.go('app.home');

            });
        }, function (error) {
            alert('There was an error logging you in.')
        });
}

```

```
})
```

---

#### 5.4.5 - SERVICES - MODEL

Για το layer που αφορά επικοινωνία με το back-end υπάρχουν τα “Services”. Κάθε service αναλαμβάνει την επικοινωνία με ένα route του API.

Παρουσιάζεται κομμάτι του κώδικα του userService που συμμετέχει στο social login από το Services.js .

```
.factory('userService', function($http, API_URL, $ionicLoading,$ionicPopup) {  
  
    var users = [];  
  
    return {  
  
        ..... ,  
  
        loginUser: function(){
```

Χρησιμοποιώντας το \$ionicLoading εμφανίζουμε ένα loading animation που είναι απαραίτητο για να μην επεμβαίνει ο χρήστης κατά τη διάρκεια της συνδιαλλαγής με το API.

```
        $ionicLoading.show({  
            content: 'Loading',  
            animation: 'fade-in',  
            showBackdrop: true,  
            maxWidth: 200,  
            showDelay: 0  
        });  
  
        var postUrl = API_URL + "/users/auth/facebook";
```

Τα απαραίτητα δεδομένα στέλνονται ως παράμετροι, ενσωματωμένοι στο array `paramsVal`.

```
var paramsVal = {
    access_token : app.fbToken,
    device_id: app.deviceID,
    device_type: app.deviceType
};

return $http({
    method: 'POST',
    url: postUrl,
    params: paramsVal,
    headers: {'Content-Type': 'application/x-www-form-urlencoded;charset=utf-8'}
}).then(function(response){
    $ionicLoading.hide();
    return response.data.user;

}).catch(function(response) {
    $ionicLoading.hide();
    $ionicPopup.alert({
        title: "AR Game",
        template: 'Something went wrong.'
    });
});

},
.....
}
});
```



### Use Cases - Περιπτώσεις Χρήσης

#### **Είσοδος Χρήστη**

Στην αρχική οθόνη ο χρήστης εισέρχεται χρησιμοποιώντας τον υπάρχων λογαριασμό του. Εναλλακτικά κάνει την εγγραφή του χρησιμοποιώντας το λογαριασμό του στο facebook.

Τα στοιχεία του χρήστη αποθηκεύονται στη βάση της εφαρμογής που βρίσκεται σε έναν κοινόχρηστο web server, ο οποίος χρησιμεύει για τη φύλαξη των κοινόχρηστων δεδομένων. Ο server παρέχει ένα RESTful API για την επικοινωνία των instances της εφαρμογής με τη βάση και την ανταλλαγή δεδομένων.

#### **Κεντρικό SideMenu**

Ο εισελθών χρήστης, έχει τις εξής επιλογές στο κεντρικό μενού της εφαρμογής.

#### **HOME**

Εδώ βλέπει την εξέλιξη του παιχνιδιού σε τίτλους, για παράδειγμα: “Ο Χ συμπαίκτης επισκέφτηκε το εξής σημείο γρίφου σήμερα.”, “Ο Υ αντίπαλος επισκέφτηκε το εξής σημείο σήμερα και κοινοποίησε την εξής φωτογραφία”, Ο Ζ συμπαίκτης επισκέφτηκε το εξής σημείο γρίφου και πέρασε τη δοκιμασία δίνοντας 200 βαθμούς στην ομάδα του.”

Τα δεδομένα παίρνονται από τη βάση δεδομένων του web server με κλήσεις συγχρονισμού στο ήδη υπάρχον RESTful API. Εξάλλου κάθε πρόοδος οποιουδήποτε παίκτη/ομάδας καταγράφεται στη βάση δεδομένων.

## TEAM

Εδώ ο χρήστης βλέπει τα μέλη της ομάδας του και έναν χάρτη που δείχνει την πρόσφατη δραστηριότητα των συμπαίκτών του. Ο χρήστης μπορεί να δει τους βαθμούς της ομάδας του και των αντίπαλων ομάδων.

## MAP

Εδώ διαδραματίζεται το κυρίως μέρος του παιχνιδιού. Ο χρήστης βλέποντας έναν Google Maps χάρτη, μπορεί να δει τα πιθανά σημεία γρίφων, και να σχεδιάσει τη διαδρομή του προκειμένου να ξεκινήσει το παιχνίδι. Σκοπός είναι να περάσει από όλα τα οριοθετημένα μέρη στα Χανιά και να επιλύσει τους υπάρχοντες γρίφους. Όσο είναι στο χάρτη και ενώ πλησιάζει στην “ζώνη γρίφου” το κινητό κάνει κάποιον προειδοποιητικό ήχο.

Εφόσον μπει στη “ζώνη γρίφου” ο χάρτης αλλάζει μορφή και γίνεται πιο κεντραρισμένος στη συγκεκριμένη ζώνη που βρίσκεται ο χρήστης. Εδώ φαίνονται λίγο πιο ξεκάθαρα οι τοποθεσίες των 3οδιάστατων αντικειμένων προκειμένου να μπορέσει να τα πλησιάσει ο χρήστης.

Σε ακτίνα 5 μέτρων από το εκάστοτε αντικείμενο του γρίφου αυτόματα ανοίγει η λειτουργία κάμερας, στην οποία ο χρήστης βλέπει τον περιβάλλοντα χώρο με την προσθήκη των AR μοντέλων. Στο κάτω μέρος της οθόνης βρίσκεται ο χώρος που ο χρήστης “αποθηκεύει” αντικείμενα προκειμένου να τα χρησιμοποιήσει σε μεταγενέστερο γρίφο. Επίσης, υπάρχει η δυνατότητα παροχής βοήθειας - “hint” εάν ο χρήστης δυσκολευτεί πολύ και περάσει χρόνος περισσότερος του συνηθισμένου. Οι πόντοι που παίρνει ο παίκτης έχουν να κάνουν με την ταχύτητα επίλυσης του γρίφου και με την χρησιμοποίηση ή όχι του hint.

### *Level 1 - Πλατεία 1866*

Ο παίκτης φτάνοντας στην πλατεία 1866 βλέπει το χάρτη της συγκεκριμένης ζώνης γρίφου. Η συγκεκριμένη ζώνη γρίφου είναι διαχωρισμένη σε 2 χώρους. Ο χρήστης πρέπει αρχικά να συλλέξει τα στοιχεία της φύσης που είναι διασκορπισμένα στην πλατεία. Σε δεύτερο στάδιο πρέπει να βρει το αντικείμενο «συλλέκτη» στο οποίο πρέπει να τοποθετήσει με τη σωστή σειρά τα αντικείμενα για να λύσει τον γρίφο και να πάρει το μήνυμα που θα τον οδηγήσει στο επόμενο επίπεδο.

Η εφαρμογή αποδείχθηκε εξαιρετικά πολύπλοκη και χρονοβόρα. Ο τομέας του Augmented Reality ειδικά σε κινητές συσκευές, δεν παρέχει επαρκές υλικό εξοικείωσης στο διαδίκτυο σε σχέση με άλλους τομείς. Χαρακτηριστικά που οραματιζόμασταν στο αρχικό στάδιο ήταν αδύνατο να υλοποιηθούν λόγω έλλειψης υπολογιστικής ισχύος. Όπως αναλύθηκε στο 4<sup>ο</sup> κεφάλαιο υπήρξαν ανυπέρβλητα προβλήματα τα οποία οδήγησαν στον επανειλημμένο επαναπροσδιορισμό των απαιτήσεων τις εφαρμογής.

Επίσης η πολυπλοκότητα του κάθε επιπέδου, κατέστησε αδύνατη την υλοποίηση των 7 επιπέδων που σχεδιάστηκαν αρχικά σε λογικά χρονικά πλαίσια. Περιοριστήκαμε στην υλοποίηση του ενός επιπέδου και του πιο πολύπλοκου.

Στο μέλλον και με τον κατάλληλο χρόνο θα μπορούσαν να υλοποιηθούν και τα υπόλοιπα επίπεδα.

Τέλος, με γνώμονα και την τεχνολογική εξέλιξη και την ενίσχυση των πόρων των κινητών συσκευών, θα μπορούσαν να ενσωματωθούν και τα υπόλοιπα features του αρχικού σχεδιασμού χωρίς τα προβλήματα απόδοσης που αντιμετωπίστηκαν κατά τη διάρκεια αυτής της υλοποίησης.

## BIBLIOGRAPHY

Fielding, R. T. «Architectural Styles and the Design of Network-based Software Architectures», Doctoral dissertation, University of California, Irvine, 2000

Richardson L. & Amundsen M. (2013), «RESTful WEB APIs», O' REILY

Heuristic evaluation of user interfaces, Nielsen, J., and Molich, R. (1990)

Mobile Usability [Raluca Budi](#) and [Jakob Nielsen](#), (2012)

Augmented Reality: Linking real and virtual worlds A new paradigm for interacting with computers, Wendy E. Mackay

## REFERENCES

[1] HitLab NZ Mobile AR Framework Documentation

Available at [www.hitlabnz.org/](http://www.hitlabnz.org/)

[2] Ionic Framework Documentation

Available at [www.ionicframework.com/](http://www.ionicframework.com/)

[3] AngularJS Documentation

Available at [www.angularjs.org/](http://www.angularjs.org/)

[4] Laravel Documentation

Available at [www.laravel.com/](http://www.laravel.com/)