

**Τμήμα  
Μηχανικών  
Πληροφορικής τ.ε.**  
Τεχνολογικό Εκπαιδευτικό Ίδρυμα  
Δυτικής Ελλάδας

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**"Ανάπτυξη Παιχνιδιού Επεκταμένης Πραγματικότητας για  
το Λειτουργικό Σύστημα Android "**

**ΓΚΟΥΝΤΗΣ ΘΟΔΩΡΗΣ**

**ΕΠΙΒΛΕΠΩΝ: ΑΛΕΦΡΑΓΚΗΣ ΠΑΝΑΓΙΩΤΗΣ,**

.....

**ΑΝΤΙΡΡΙΟ 15 / 5/ 2014**



Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, .....

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

.....

.....

.....

## Περίληψη

Το παρόν κείμενο παρουσιάζει την ανάπτυξη εφαρμογής επαυξημένης πραγματικότητας για το λειτουργικό σύστημα Android, με το όνομα TreasureAR, χρησιμοποιώντας την βιβλιοθήκη ανοικτού κώδικα AndAR, τους ενσωματωμένους αισθητήρες και του δέκτη GPS της συσκευής μας. Για την υλοποίηση των χαρτών μας χρησιμοποιήσαμε την ελεύθερη βιβλιοθήκη OSM.

Η επαυξημένη πραγματικότητα (augmented reality) ενισχύει την οπτική του χρήστη για τον πραγματικό κόσμο με δεδομένα, εικόνες και ήχους που παράγονται από υπολογιστές. Τα όποια αλληλεπιδρούν με την κίνηση του χρήστη στον πραγματικό κόσμο, σε πραγματικό χρόνο, προσπαθώντας να κάνουν την χρήση της επαυξημένης πραγματικότητας με τέτοια ταχύτητα και τρόπο που επιτρέπουν στον χρήστη να έχει τα κατάλληλο περιεχόμενο κινούμενος μέσα στον πραγματικό χώρο και με όσο γίνεται πιο ομαλό και φυσικό τρόπο.

Κατά την Διάρκεια υλοποιήσεις της πτυχιακής εργασίας μας μελετήσαμε τόσο την επαυξημένη πραγματικότητα σαν τεχνολογία όσο και την δυνατότητα να αναπτύξουμε μια εφαρμογή που θα συνδυάζει και τους δυο κυρίως τύπους επαυξημένης πραγματικότητας, το σο της Επαυξημένης Πραγματικότητας Με την χρήση σημαδιών τόσο και χωρίς αυτά.

### Λέξεις Κλειδιά

Επαυξημένη Πραγματικότητα, Android, AndAR, OSM

## **Abstracts**

This text presents the development of augmented reality applications for the operating system Android, named TreasureAR, using the open source library AndAR, the embedded sensors of our device and GPS module. The implementation of our maps base to OSM free library.

Augmented reality (AR) enhances the viewing user's real world with data, images and sounds produced by computers. Interacting with the movement of sound in the real world , real time , trying to make use of augmented reality with such a speed and manner that allows the user to have the appropriate content moving into the real space and with as smooth a and natural way.

During the process of our thesis we investigated augmented reality technology as well as the opportunity to develop an application that combines the two main types of augmented reality, in both the Augmented Reality Marker Base and Marker less.

### **Keywords**

Augmented Reality, Android, AndAR, OSM

*Στους Γονείς μου*

## Περιεχόμενα

Περίληψη.....	4
Abstracts .....	5
Κατάλογος Εικόνων.....	11
Κατάλογος Πινάκων .....	12
1.1 Ορισμός .....	14
1.2 Μεικτή Πραγματικότητα .....	15
1.3 Σύγκριση Επαυξημένη Πραγματικότητα και Εικονικής Πραγματικότητας.....	16
1.4 Ιστορική Ανάδρομη.....	16
1.5 Τομείς Χρήσης της Επαυξημένης Πραγματικότητας .....	18
1.5.1 Ιατρική .....	18
1.5.2 Ενημέρωση/Ψυχαγωγία .....	18
1.5.3 Στρατός .....	18
1.5.4 Διαφήμιση /Πληροφόρηση Προϊόντων: .....	19
1.5.5 Δοκιμαστήρια .....	19
1.5.6 Βιομηχανία.....	19
1.5.7 Τουρισμός.....	20
1.5.8 Πλοήγηση σε άγνωστο Χώρο .....	20
1.5.9 Αρχιτεκτονική - Εσωτερική διακόσμηση .....	20
1.5.10 Εκπαίδευση.....	21
1.5.11 Μετάφρασης .....	21
1.5.12 Παιχνίδια.....	21
1.6 Αρχιτεκτονική ενός Συστήματος ΕΠ για κινητές συσκευές.....	22
1.7 Συσκευές Επαυξημένης Πραγματικότητας .....	23
1.7.1 Φορητές Συσκευές Απεικόνισης (HMD).....	23
1.7.2 Οθόνες Φορητών Συσκευών .....	24
1.7.3 Γυαλιά Επαυξημένης Πραγματικότητας.....	24

1.7.3 Φακοί Επαφής.....	25
1.8 Κύριοι Τύποι Εφαρμογών Επαυξημένης Πραγματικότητας για τις κινητές Συσκευές.....	25
1.8.1 Εφαρμογές βασισόμενες Σε Σημάδια (Maker-based AR).....	26
1.8.1.1 Σημάδια - Markers .....	27
1.8.1.2 Αλγόριθμος υπολογιστικής όρασης (Computer vision algorithm).....	28
1.8.1.3 Σύστημα Συντεταγμένων (Coordinate Systems).....	28
1.8.1.4 Σύστημα Συντεταγμένων υπολογιστικής όρασης (Computer Vision) .....	28
1.8.1.5 Σύστημα Συντεταγμένων για την σχεδίαση της σκηνής .....	29
1.8.2 Εφαρμογές Μη βασισόμενες Σε Σημάδια (MakerLess AR).....	30
1.9 Φυλλομέτρησες ΕΠ για κινητές Συσκευές.....	31
1.9.1 Wikitude.....	31
1.9.2 Layar .....	33
1.9.3 Junaio .....	35
2.1 Τι είναι το Android; .....	36
2.2 Η πορεία του Android.....	36
2.3 Χαρακτηριστικά γνωρίσματα του Android SDK .....	38
2.4 Το πλαίσιο ανάπτυξης του Android.....	39
2.5 Πακέτο Ανάπτυξης Λογισμικού (software development kit , SDK) .....	39
2.6 Αρχιτεκτονική της Πλατφορμα του Android.....	41
2.5.1 Η εικονική μηχανή Dalvik .....	43
2.7 Βιβλιοθήκες ΕΠ για το Android .....	44
2.7.1 ARToolkit .....	44
2.7.2 ARToolkitPlus.....	46
2.7.3 NyARToolKit.....	46
2.8. Η Βιβλιοθήκη AndAR .....	47
2.8.1 Δομή.....	47
3.1 Κατασκευή της διαδικτυακής υπηρεσίας μας.....	51
3.2 Δημιουργία της Βάσης Δεδομένων με MySQL και του πίνακα .....	51
3.3 Κατασκευάζοντας τα Αρχεία PHP.....	51



3.3.1 Το αρχείο config.php.....	52
3.3.2 Το αρχείο DB_Connect.php.....	52
3.3.3 Το αρχείο DB_Functions.php .....	53
3.3. 4 Το αρχείο index.php.....	54
3.4 Τύποι Απαντήσεων (Responses) JSON .....	54
3.5 Δομή του Αρχείου Στοιχειά .....	56
3.5 Ξεκινώντας το Πρόγραμμα μας στο Android.....	56
3.5.1 Η κλάση JSON Parser.....	57
3.5.2 Η κλάση SQLite Database Handler .....	57
3.5.3 Η κλάση User Functions .....	58
3.5.4 Η κλάση RegisterActivity .....	60
3.5.5 MainMenuActivity.....	60
3.5.6 LoginActivity.....	61
3.5.7 GameActivity.....	62
Η Εσωτερική κλάση TakeAsyncScreenshot .....	67
Τα Επικαλυπτόμενα Αντικείμενα (CustomObject Overlays) .....	67
3.5. 7 Η Κλάση CustomRenderer .....	69
3.5.8 InstructionsActivity .....	69
3.5.8 AboutActivity .....	70
3.5.9 MapActivity .....	70
3.5.9 Η κλάση StoiceiaActivity .....	71
3.8 Το Αρχείο AndroidManifest.xml.....	72
4.1 Λογισμικό .....	74
4.1.1 Android Studio.....	74
4.1.2 OpenStreetMap .....	75
4.1.3 Το Android NDK (Native Development Kit) .....	76
4.1.4 OpenGL ES.....	77
4.1.5 PHP .....	77
4.1.6 MySQL .....	78

4.1.7 phrmyadmin .....	78
4.1.8 JSON.....	78
4.2 Υλικό.....	79
4.3 Επισκόπηση της Εφαρμογή μας.....	80
LoginActivity.....	80
Register.....	80
MenuActivity.....	81
Game Activity.....	81
Map Activity.....	82
InstunctionActivity .....	82
AboutActivity .....	82
5. Συμπεράσματα.....	83
5.1 Γενικά Συμπεράσματα .....	83
5.2 Μελλοντική Δουλεία .....	84
ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	86
Παράρτημα Β: Βιβλιογραφία.....	122
Δημοσίευσης.....	122
Βιβλία .....	122
Διαδίκτυο .....	122

---

## Κατάλογος Εικόνων

Εικόνα 1-1: HUDs (Head-up Displays) στα μαχητικά αεροσκάφη.....	9
Εικόνα 1-2. Εικονικό Συνεχές κατά τον Mailgram.....	10
Εικόνα 1-3: ARQuake.....	12
Εικόνα 1-4: Παραδείγματα Παιχνιδιών Επαυξημένης Πραγματικότητας.....	15
Εικόνα 1-5 : α) H/Y και HMD b)Ταμπλέτες c)PDA d)έξυπνα κινητά.....	15
Εικόνα 1-6: α) Οπτικές Συσκευές Απεικόνισης β)Βίντεο Συσκευή Απεικόνισης.....	17
Εικόνα 1-7: Google Glass.....	18
Εικόνα 1-12: Φακοί Επαφής Επαυξημένης Πραγματικότητας.....	18
Εικόνα 1-9:Γενική Ροή μιας Εφαρμογής Επαυξημένης Πραγματικότητας.....	19
Εικόνα 1-10. Ένα απλό Παράδειγμα Σημαδιού (marker).....	20
Εικόνα 1-11. Ένα Εργαλείο για την Κατασκευή Σημαδιού (marker).....	20
Εικόνα 1-12: Σύστημα Συντεταγμένων του ARtoolkit.....	21
Εικόνα 1-13: Ο μετασχηματισμός του συστήματος συντεταγμένων.....	21
Εικόνα 1-14: Παράδειγμα Εφαρμογής ΕΠ μη βασιζόμενες σε Σημάδια.....	22
Εικόνα 1-15: Αρχιτεκτονική του Laya.....	25
Εικόνα 1-16: Παράδειγμα χρήσης της εφαρμογής Junaio.....	26
Εικόνα 2-1: Λογότυπο του Android.....	28
Εικόνα 2-2: Αρχιτεκτονική του Λογισμικού Android.....	32
Εικόνα 2-3: Dalvik VM.....	33
Εικόνα 2-4 : Αρχιτεκτονική του ARToolkit.....	35
Εικόνα 2-5 : Αρχιτεκτονική του AndAR.....	36
Εικόνα 2-6: Το Διάγραμμα κλάσεων του AndAR.....	37
Εικόνα 3-1: Σχεδιάγραμμα της Διαδικτυακής μας Υπηρεσίας.....	39
Εικόνα 3-2 Επικαλυπτόμενα Αντικείμενα.....	52
Εικόνα 4-1: Android Studio.....	58
Εικόνα 4-2: Αρχιτεκτονική του Android NDK.....	59
Εικόνα 4-2: HTC Hero.....	61
Εικόνα 4-3 : LoginActivity.....	62
Εικόνα 4-4 : RegisterActivity.....	62
Εικόνα 4-5: Main Activity.....	62
Εικόνα 4-6: GameActivity.....	63
Εικόνα 4-7: MapActivity.....	63
Εικόνα 4-8: InstrunctionActivity.....	63
Εικόνα 4-10: AboutActivity.....	63

## Κατάλογος Πινάκων

Πίνακας 2-1 : Εκδόσεις του λειτουργικού συστήματος Android.....	30
Πίνακας 2-2 : Τα στάδια λειτουργίας του ARToolkit.....	36
Πίνακας 4-1 : Χρέωση Στο Google Map API.....	59

## Εισαγωγή

Η παρούσα πτυχιακή εργασία μελετά την ανάπτυξη Εφαρμογών Επαυξημένη Πραγματικότητα (Augmented Reality) για κινητές συσκευές και ποιο συγκεκριμένα για την πλατφόρμα του Λειτουργικού Συστήματος Android. Μια τεχνολογία που έχει τις ρίζες της στην δεκαετία του 60 με τα πρώτα δειλά βήματα και αναπτύσσονταν σίγα σιγά μέσα σε εργαστήρια και ερευνητικά ιδρύματα

Η τεχνολογία αυτή κατάφερε να περάσει από το στάδιο των ερευνητικών ιδρυμάτων και των εργαστηρίων χάρη στην ανάπτυξη των κινητών τηλεφώνων και ιδιαίτερα της τελευταίας γενιάς τους, των λεγόμενων έξυπνων κινητών. Τα οποία μας παρέχουν τα τρία βασικά στοιχεία που απαιτούνται για τις εφαρμογές επαυξημένης πραγματικότητας.

1. Την απαραίτητη υπολογιστική ισχύ για να τρέχουν οι αλγόριθμοι ,
2. την οθόνη πάνω στην οποία θα εμφανιστούν τα στοιχεία που επαυξάνουν την πραγματικότητα τους και τους ενσωματωμένους αισθητήρες της συσκευής μας (GPS,επιταχυνσιόμετρο, πυξίδα)
3. αλλά και τις δυνατότητες δικτύωσης αυτών των συσκευών

Επίσης τα σύγχρονα κινητά τηλεφωνα τρέχουν λειτουργικά συστήματα που μας παρέχουν όλα τα απαραίτητα προγραμματιστικά εργαλεία για την ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας .

Η συγκεκριμένη εργασία χωρίζεται σε πέντε κεφάλαια:

Στο πρώτο κεφάλαιο περιγράφουμε τι είναι η επαυξημένη πραγματικότητα και όλα τα απαραίτητα στοιχεία που χρειάζεται να γνωρίζουμε για την ανάπτυξη εφαρμογών ΕΠ για τα κινητά τηλεφωνα .

Στο Δεύτερο κεφάλαιο γίνεται μια ανάλυση για το λειτουργικό σύστημα Android και τις διαθέσιμες βιβλιοθήκες για την ανάπτυξη εφαρμογών ΕΠ.

Στο Τρίτο κεφάλαιο περιγράφουμε την ανάπτυξη της εφαρμογής μας και τα κύρια συστατικά που την αποτελούν.

Στο Τέταρτο κεφάλαιο κάνουμε μια περιγραφή των προγραμμάτων που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής αλλά και της συσκευής που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής. Επίσης κάνουμε μια σύντομη περιγραφή της λειτουργίας της εφαρμογής μας.

Τέλος, στο Πέμπτο και τελευταίο κεφάλαιο αναφέρουμε τα συμπεράσματα που αποκομίσαμε κατά την διάρκεια της ανάπτυξης της εφαρμογή μας , αλλά και πιθανές δυνατότητες περαιτέρω ανάπτυξη της εφαρμογή μας.

## 1: Επαυξημένη Πραγματικότητα

### 1.1 Ορισμός

Υπάρχουν αρκετοί ορισμοί σχετικά με το τι είναι η επαυξημένη πραγματικότητα. Το 1997 ο Ronald Azuma [1] έδωσε έναν ορισμό για την επαυξημένη Πραγματικότητα που είναι κοινά αποδεκτός. Ο οποίος ορίζει την Επαυξημένη Πραγματικότητα σαν μια τεχνολογία που συνδυάζει το φυσικό περιβάλλον ενισχυμένη (επαυξημένο) με εικονικά αντικείμενα που δημιουργούνται από υπολογιστή.

Συμφώνα με τον Azuma για να θεωρηθεί ένα σύστημα ή εφαρμογή επαυξημένης πραγματικότητα πρέπει να πλήρη τα τρία παρακάτω χαρακτηριστικά :

1. **Συνδυάζει εικονικών χαρακτήρων με τον πραγματικό κόσμο**, το βασικότερο χαρακτηριστικό της ΕΠ δηλαδή ο συνδυασμός πραγματικού και εικονικού. Η ΕΠ ενισχύει τον τρόπο με τον οποίο ο άνθρωπος αντιλαμβάνεται τον πραγματικό κόσμο μέσω των αισθήσεων του (κυρίως τις όραση μέχρι στιγμής).
2. **Να Αλληλεπιδρά (Interactive)σε πραγματικό χρόνο**, η ΕΠ πρέπει να λειτουργεί σε πραγματικό χρόνο και να αλληλεπιδρά με τα συμβάντα που εκτελούνται εκείνη την στιγμή.
3. **Να λειτουργεί αναφορικά με τον πραγματικό τρισδιάστατο κόσμο**, είναι και αυτό που διαφοροποιεί την επαυξημένη πραγματικότητα από την εικονική πραγματικότητα καθώς απαιτεί το πραγματικό στοιχείο του τρισδιάστατου κόσμου να είναι κυρίαρχο και το «εικονικό» να επικαλύπτεται σε αυτό

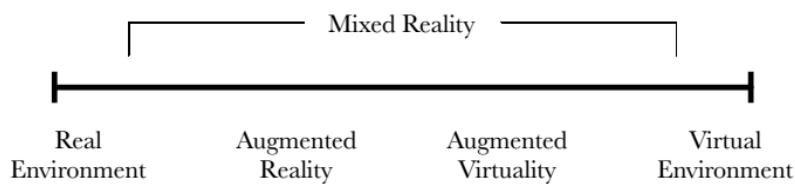
Ένα από τα πιο χαρακτηριστικό παράδειγμα επαυξημένης πραγματικότητας και το πλέον γνωστό σε όλους μας με επιπλέον πληροφορία ήταν τα περίφημα HUDs(Head-up Displays) στα μαχητικά αεροσκάφη, όπου ο πιλότος μπορεί να δει πάνω στο τζαμί του μαχητικού , παραμέτρους της πτήσης, όπως ταχύτητα, υψόμετρο, το στόχο κ.α. χωρίς να χρειάζεται να μετακινήσει το κεφάλι του και χωρίς να χάνει την οπτική επαφή του με το περιβάλλον.



Εικόνα 1-1: HUDs (Head-up Displays) στα μαχητικά αεροσκάφη

## 1.2 Μεικτή Πραγματικότητα

Το 1994 ο Paul Milgram και ο Fumio Kishino πραγματοποίησαν μια ταξινόμηση των συστημάτων μεικτής πραγματικότητας (mixed reality) που συμπεριλαμβάνουν την Εικονική Πραγματικότητα (Virtual Reality, VR) την Επαυξημένη Πραγματικότητα (Augmented Reality, AR) και την Επαυξημένη Εικονικότητα (Augmented Virtuality, AV).



Εικόνα 1-2. Εικονικό Συνεχές κατά τον Mailgram

Ο Milgram το 1994(3) όρισε το «συνεχές πραγματικότητας-εικονικότητας» (reality-virtuality continuum), όπου στο ένα άκρο βρίσκεται το πραγματικό περιβάλλον και στο άλλο άκρο ένα πλήρως εικονικό περιβάλλον. Ανάμεσα στα δυο αυτά άκρα υπάρχουν οι αποχρώσεις της Μίκης πραγματικότητας(mixed reality).

Η επαυξημένη πραγματικότητα, βρίσκεται κοντά στο άκρο της πραγματικότητας, καθώς η κυρίαρχη αντίληψη που μεταφέρεται προς τον χρήστη είναι αυτή του πραγματικού κόσμου, επαυξημένη με δεδομένα από έναν υπολογιστή και η Επαυξημένη Εικονικότητα(augmented virtuality), είναι ένας Όρος που δημιουργήθηκε από τον Milgram, η οποία βρίσκεται πληρύτερα στην εικονική πραγματικότητα και περιγράφει συστήματα τα οποία παρουσιάζουν κυρίως σύνθετες εικόνες με προσθήκη κάποιων στοιχείων από το πραγματικό περιβάλλον, για μεγαλύτερη πιστότητα και ρεαλισμό. Η επαυξημένη εικονικότητα δεν έχει ουσιαστικές εφαρμογές και αναμένεται να εκλείψει ως κατηγορία, όσο η τεχνολογία βελτιώνεται και οι εικονικές σκηνές δεν θα διακρίνονται από τις πραγματικές.

Το σημαντικό στοιχείο στην ταξινόμηση του Milgran είναι το γεγονός ότι οι κατηγορίες της Μίκης πραγματικότητας δεν αντιμετωπίζονται ως διακριτά σημεία, αλλά όλος ο χώρος θεωρείται συνεχές. Αυτό σημαίνει ότι τα όρια δεν είναι απόλυτος σαφή και ότι οι διακρίσεις γίνονται δυσδιάκριτες για το τι ακριβώς αντιλαμβανόμαστε ως πραγματικό και τη ως εικονικό.

### 1.3 Σύγκριση Επαυξημένη Πραγματικότητα και Εικονικής Πραγματικότητας

Η εικονική και η επαυξημένη πραγματικότητα είναι δυο διαφορετικές έννοιες αν και έχουν πολλά κοινά χαρακτηριστικά τόσο ως τεχνολογίες όσο και ως ιδέες. Ο Όρος εικονική πραγματικότητα αποδίδεται στον Jason Lanier, τον ιδρυτή μιας από της πρώτες εταιρίες που ασχολήθηκαν με την κατασκευή τέτοιων συστημάτων, ο οποίος προσδιόρισε τη νέα αυτή τεχνολογία το 1992 ως «ένα τρισδιάστατο και αλληλεπιδραστικό(interactive) περιβάλλον, το οποίο παράγεται εξ ολοκλήρου από έναν υπολογιστή και στο οποίο ο χρήστης μπορεί να εμβολιστεί(immersion)».

Στην εικονική πραγματικότητα ο στόχος είναι η πλήρης αποκοπή του χρηστή από το γύρω του περιβάλλον και η δημιουργία μιας (πιστευτής και απολύτως ρεαλιστικής) ψευδαίσθησης ότι βρίσκεται σε όποιον κόσμο παράγει ο υπολογιστής. Ενώ στην επαυξημένη πραγματικότητα ο στόχος μας δεν είναι η αποκοπή του χρηστή από την πραγματικότητα αλλά η ενίσχυση του πραγματικού κόσμου με νέα στοιχεία που δεν μπορούν να μας παρέχουν οι αισθήσεις μας, ουσιαστικά ενισχύουμε τις αισθήσεις μας (κυρίως την όραση, μέχρι στιγμής) με τις δυνατότητες που μας παρέχει η τεχνολογία της πληροφορίας,

### 1.4 Ιστορική Ανάδρομη

Εδώ θα κάνουμε μια σύντομη ιστορική ανάδρομη στην πορεία εξέλιξης της επαυξημένης πραγματικότητας από την δεκαετία του 60 μέχρι τις μέρες μας.

**1957:** Ο Morton Heilig, ένας κινηματογραφιστής δημιουργεί και πατένταρε έναν προσομοιωτή ονόματι Sensorama με γραφικά, ήχο, δόνηση και οσμές.

**1966:** Ο Ivan Sutherland επινοητή την συσκευή προβολής κεφαλής(head-mounted display) το πρώτο παράθυρο στον εικονικό κόσμο



**1975:** Ο Myron Krueger δημιουργεί το Videoplace που επιτρέπει στους χρήστες να αλληλεπιδράσουν με εικονικά αντικείμενα για πρώτη φορά.

**1989:** Ο Jaron Lanier εφευρίσκει τον όρο εικονική πραγματικότητα(Virtual Reality) και δημιουργεί την πρώτη εμπορική επιχείρηση γύρω από εικονικούς κόσμους.

**1990:** Ο Tom Caudell εισάγει τη φράση επαυξημένη πραγματικότητα(Augmented Reality) ενώ δούλευε στην εταιρία Boeing και βοήθησε εργάτες να τοποθετήσουν τα καλώδια στα αεροσκάφη.

**1992:** Ο L.B. Rosenberg αναπτύσσει ένα από τα πρώτα λειτουργικά συστήματα επαυξημένης πραγματικότητας με όνομα VIRTUAL FIXTURES στο Air Force Research ένα εργαστήριο του αμερικάνικου στρατού.

**1992:** Οι Steven Feiner, Blair MacIntyre και Doree Seligmann παρουσιάζουν το πρώτο σημαντικό paper σε ένα πρωτότυπο σύστημα επαυξημένης πραγματικότητας, KARMA.

**1999:** Ο Hirokazu Kato ( 加藤 博一 ) δημιουργεί το ARToolKit στα εργαστήρια HIT, μια βιβλιοθήκη για την επαυξημένη πραγματικότητα που βασίζεται σε σημάδια (markers) όπου συνεχίστηκε η ανάπτυξη του από άλλους ερευνητές του εργαστηρίου.

**2000:** Ο Bruce H. Thomas αναπτύσσει το ARQuake, το πρώτο παιχνίδι επαυξημένης πραγματικότητας εξωτερικού χώρου και παρουσιάζεται στο διεθνές συνέδριο για Φορητούς Υπολογιστές (International Symposium on Wearable Computers).



Εικόνα 1-3: ARQuake

**2008:** Ο Ταξιδιωτικός Οδηγός του Wikitude (Wikitude AR Travel Guide) που υποστηρίζει επαυξημένη πραγματικότητα ξεκίνησε στις 20 Οκτώβριου του 2008 μαζί με το κινητό τηλέφωνο G1 Android.

**2009:** Το ARToolkit μεταφέρεται στο Adobe Flash, το FLARToolkit από την Saqoosha, φέρνει την επαυξημένη πραγματικότητα στου φυλλομέτρησες.

**2010:** Κάποιες νέες βιβλιοθήκες επαυξημένης πραγματικότητας αναπτύσσονται για το νέο λειτουργικό σύστημα της Google, το Android, όπως η AndAR, που υλοποιείται σε Java.

## 1.5 Τομείς Χρήσης της Επαυξημένης Πραγματικότητας.

Το πεδίο εφαρμογής της επαυξημένης πραγματικότητας μπορεί να βρει εφαρμογή σχεδόν σε κάθε τομέα των ανθρωπινών δραστηριοτήτων αλλά και στην καθημερινή ζωή μας. Ορισμένοι Ερευνητές μάλιστα διερωτώνται μήπως πρόκειται για την εφαρμογή που θα αλλάζει την ζωή μας σαν ένας ενισχυτής νοημοσύνης (intelligence Amplification), όπως έχει ορίσει ο Brooks, σαν εργαλείο υποβοηθήσεις του ανθρώπου στις διαφορές δραστηριότητες του. Κάποιοι γενικοί τομείς που μπορεί να βρει εφαρμογή:

### 1.5.1 Ιατρική

Οι περισσότερες προσπάθειες εφαρμογής στο χώρο της ιατρικής επικεντρώνονται στην υποβοήθηση των εγχειρήσεων. Εικόνες από ακτινογραφίες ή αξονικές/μαγνητικές τομογραφίες μπορούν να προβάλλονται στο σωστό σημείο του ασθενούς καθώς η εγχείρηση βρίσκεται σε εξέλιξη. Επίσης ο γιατρός μπορεί να βλέπει μια τρισδιάστατη απεικόνιση ενός εμβρύου, η οποία έχει δημιουργηθεί με τη χρήση υπερήχων, πάνω στην κοιλία της εγκύου.

### 1.5.2 Ενημέρωση/Ψυχαγωγία

Η Επαυξημένη Πραγματικότητα χρησιμοποιείται εδώ και αρκετά χρονιά από τα τηλεοπτικά κανάλια εισάγοντας άμεσα ψηφιακές πληροφορίες στο εκπεμπόμενο βίντεο σε πραγματικό χρόνο, κλασικό παράδειγμα αυτής της χρήσης είναι η παρουσίαση των δελτίων καιρού αλλά και σε άλλες εφαρμογές κυρίως αθλητικών αγώνων.

### 1.5.3 Στρατός

Η εφαρμογή των HUD αποτελεί ένα από τα πρώτα παραδείγματα επαυξημένης πραγματικότητας και δείχνουν το ενδιαφέρον των στρατιωτικών βιομηχανιών για της δυνατότητες της χρήσης της επαυξημένης πραγματικότητας στο πεδίο της μάχης και το πώς αυτή μπορεί να βοηθήσει τους στρατιώτες. Στο μέλλον οι στρατιώτες μπορούν να εφοδιαστούν με ειδικά κράνη επαυξημένης πραγματικότητας που μπορούν να υπερθέτουν πληροφορίες για τον πραγματικό κόσμο:

- Για την μορφολογία του εδάφους
- Για τον αντίπαλο σχηματισμο
- Για τη μη ορατές από την συγκεκριμένη οπτική λεπτομέρειες όπως πληροφορίες από δορυφόρους ή από άλλους στρατιώτες.
- Για κρίσιμα σημεία και αδυναμίες του αντίπαλου

#### 1.5.4 Διαφήμιση /Πληροφόρηση Προϊόντων:

Η ΕΠ μπορεί να χρησιμοποιηθεί και για διαφημιστικούς σκοπούς σε περιοδικά εφημερίδες ξεφεύγοντας άτια μόνο από τις δικτυακές διακρατικές εφαρμογές αν και εκεί στην κατηγορία των ηλεκτρονικών καταστημάτων θα ήταν μια παρά πολλή καλή επιλογή. Κυρίως για σκοπούς πληροφόρησης προϊόντων δηλαδή να μας παρέχει την δυνατότητα να δούμε το σε πραγματικό μοντέλο τι υπάρχει μέσα στο κουτί πριν το αγοράσουμε είτε ηλεκτρονικά είτε φυσικά.

#### 1.5.5 Δοκιμαστήρια

Σε πολλά καταστήματα, AR δικάζεται ως μια εικονική αίθουσα δοκιμής . Ο χρήστης μπορεί να σταθεί μπροστά από μια οθόνη με μια κάμερα τοποθετημένη κάπου. Ο χρήστης θα δει ο ίδιος εμφανίζεται στην οθόνη. Ο χρήστης, στη συνέχεια, χρησιμοποιεί μία συσκευή εισόδου, όπως ένα ποντίκι ή το πληκτρολόγιο για να επιλέξετε κάποια από τις διαθέσιμες επιλογές ρούχα. ο υπολογιστής Στη συνέχεια, θα αυξήσει αυτό το στοιχείο επάνω την εικόνα του χρήστη και να το εμφανίσετε στην οθόνη. Ο χρήστης μπορεί να ενεργοποιήσει για να δείτε τον εαυτό του από όλες τις γωνίες.

#### 1.5.6 Βιομηχανία

Η επαυξημένη πραγματικότητα μπορεί να βρει εφαρμογή στον βιομηχανικό σχεδιασμό, στις γραμμές Παράγωγης ,στην συναρμολόγηση , στην συντήρηση και στις κατασκευές, δίνοντας τη δυνατότητα σε μια ομάδα μηχανικών ή εργαζομένων να εργάζονται πάνω στο ίδιο έργο, έχοντας όλοι την δυνατότητα να βλέπουν σε πραγματικό χρόνο τις διορθώσεις και τις βελτιώσεις που γίνονται και να προσαρμόζουν την παράγωγη τους στα νέα δεδομένα.

Ένας συντηρητής θα μπορεί να βλέπει το ελαττωματικό εξάρτημα μιας μηχανής με ένα πιο έντονο χρώμα. Η ΕΠ μπορεί να βοηθήσει βιομηχανικοί σχεδιαστές εμπειρία του σχεδιασμού και της λειτουργίας ενός προϊόντος πριν από την ολοκλήρωση. Ένα τέτοιο παράδειγμα είναι η αυτοκινητοβιομηχανία Volkswagen χρησιμοποιεί AR για τη σύγκριση υπολογισμένη και πραγματικές δοκιμές πρόσκρουσης. εικόνες.<sup>1821</sup> AR μπορεί να χρησιμοποιηθεί για να

απεικονίσει και να τροποποιήσετε μια δομή αμαξώματος του αυτοκινήτου και διάταξη του κινητήρα.

### 1.5.7 Τουρισμός

Εφαρμογές επαυξημένης πραγματικότητας μπορούν να ενισχύσουν την εμπειρία του χρήστη, όταν ταξιδεύουν με την παροχή σε πραγματικό χρόνο ενημερωτική εμφανίζει σχετικά με μια θέση και τα χαρακτηριστικά του, συμπεριλαμβανομένων των παρατηρήσεων που έγιναν από προηγούμενες επισκέπτες.

Οι προσομοιώσεις των ιστορικών γεγονότων, εξερευνώντας και μαθαίνοντας λεπτομέρειες για κάθε σημαντικό τομέα της ιστοσελίδα εκδήλωσης AR εφαρμογές επιτρέπουν στους τουρίστες να γνωρίσουν προσομοιώσεις των ιστορικών γεγονότων, τόπων και αντικειμένων, καθιστώντας τους σε τρέχουσα προβολή τους από ένα τοπίο.

Με την επαυξημένη πραγματικότητα , μπορούμε να δώσουμε μια νέα εμπειρία στους επισκέπτες να ξανά κτίσουμε το κτήριο μέσω ειδικών συσκευών να αναπαραστήσουμε δραστηριότητες , να ζωντανέψουμε προσωπικότητες κ.α

### 1.5.8 Πλοήγηση σε άγνωστο Χώρο

Άλλα δεν περιορίζεται μόνο στους αρχαιολογικούς τόπους. Μπορεί να χρησιμοποιηθεί για να ανακαλύψει ο επισκέπτης, τουρίστας τοποθεσίες, εστιατόρια, ξενοδοχεία, φαρμακεία και ότι άλλο θα ήταν χρήσιμο για αυτό κατά την επίσκεψή του σε μια νέα περιοχή άγνωστη σε αυτόν, η επαυξημένη πραγματικότητα του δίνει την δυνατότητα να ανακαλύψει την περιοχή γρήγορα, εύκολα και με ένα πολλή φυσικό τρόπο (περπατώντας μέσα σε αυτήν).

### 1.5.9 Αρχιτεκτονική - Εσωτερική διακόσμηση

Η ΕΠ μπορεί να βοηθήσει στην αστικοποίηση και στα οικοδομικά έργα. Οι εικόνες μιας δομής μπορεί να τοποθετούνται σε μια πραγματική ζώνη και να έχουμε μια τοπική άποψη της ιδιοκτησίας πριν το κτίριο κατασκευαστεί. Η ΕΠ μπορούν επίσης να χρησιμοποιηθούν μέσα στο χώρο εργασίας ενός αρχιτέκτονα, καθιστώντας την άποψή τους σε κινούμενα 3D απεικονίσεις των 2D σχεδίων τους. Αρχιτεκτονική sight-seeing μπορεί να ενισχυθεί με εφαρμογές AR επιτρέπει στους χρήστες βλέπουν το εξωτερικό ενός κτιρίου για να δείτε σχεδόν μέσα από τα τείχη της, που βλέπουν εσωτερικά αντικείμενα, και τη διάταξή τους. Άλλη μια περιοχή που η δυνατότητα να βλέπουμε το τελικό προϊόν είναι πολύ σημαντική γιατί συνήθως τα λάθη πληρώνονται ακριβά κυρίως βεβαία στην αρχιτεκτονική άλλα

### 1.5.10 Εκπαίδευση

Εφαρμογές επαυξημένης πραγματικότητας μπορούν να συμπληρώσουν ένα τυποποιημένο πρόγραμμα σπουδών. Κείμενο, γραφικά, βίντεο και ήχου μπορούν να τοποθετούνται σε περιβάλλον πραγματικού χρόνου του μαθητή. Διδακτικά βιβλία, flashcards και άλλα εκπαιδευτικό υλικό ανάγνωση μπορεί να περιέχει ενσωματωμένες "δείκτες" ότι, όταν σαρωθεί από μια συσκευή AR, παράγουν συμπληρωματικές πληροφορίες για το μαθητή που παρέχονται σε μορφή πολυμέσων. Οι μαθητές μπορούν να συμμετέχουν διακρατικά με τον υπολογιστή.

Η ΕΠ μπορεί να βοηθήσει τους μαθητές στην κατανόηση της χημείας, επιτρέποντάς τους να απεικονίσει τη χωρική δομή ενός μορίου και να αλληλεπιδρούν με ένα εικονικό μοντέλο αυτό που εμφανίζεται, σε μια εικόνα της κάμερας, τοποθετημένη σε ένα δείκτη που πραγματοποιήθηκε στο χέρι τους. Η τεχνολογία της επαυξημένης πραγματικότητας επιτρέπει επίσης τη μάθηση μέσω της εξ αποστάσεως συνεργασία, στο οποίο οι μαθητές και οι εκπαιδευτές δεν είναι στην ίδια φυσική τοποθεσία μπορεί να μοιράζονται ένα κοινό εικονικό περιβάλλον μάθησης που κατοικείται από εικονικά αντικείμενα και εκπαιδευτικό υλικό και να αλληλεπιδρούν με το άλλο μέσα σε αυτό το περιβάλλον.

### 1.5.11 Μετάφρασης

Τα συστήματα ΕΠ μπορεί να χρησιμοποιούνται για να μεταφράσει το κείμενο από πολλαπλές γλώσσες σε όλο τον κόσμο ερμηνεύσει ξένο κείμενο σχετικά με τα σήματα και τα μενού. Αυτές οι συσκευές διαθέτουν OCR και έχουν είτε ένα ολόκληρο λεξικό (cross language) στη συσκευή ή να μεταφράσει τη γλώσσα μέσω του Διαδικτύου. Οι προφορικές λέξεις μιας ξένης γλώσσας μπορεί να μεταφραστεί και να εμφανίζεται στην προβολή ενός χρήστη ως τυπωμένα υπότιτλους

### 1.5.12 Παιχνίδια

Ένας τομέας στον οποίο η επαυξημένη πραγματικότητα θα μπορεί να προσφέρει ένα άλλο επίπεδο ρεαλισμού, όπου οι πραγματικοί παίκτες και εικονικοί κόσμοι θα συνδυάζονται σε μια σκηνή. Προσφέροντας έτσι μια νέα δυνατότητα για την ανάπτυξη νέων παιχνιδιών πιο διακρατικών και παιχνίδια που να έχουν σχέση με τον φυσικό κόσμο.



Εικόνα 1-4 Παραδείγματα Παιχνιδιών Επαυξημένης Πραγματικότητας.

Παραπάνω είδαμε κάποιες μόνο από τις λίγες επιλογές που έχουμε για την χρήση της Επαυξημένης Πραγματικότητας. Άλλα μπορεί πολλή εύκολα να βρει εφαρμογή σε ένα πολλή μεγάλο μέρος των ανθρωπίνων δραστηριοτήτων και πολλή πιθανόν σε κάθε μια από αυτές.

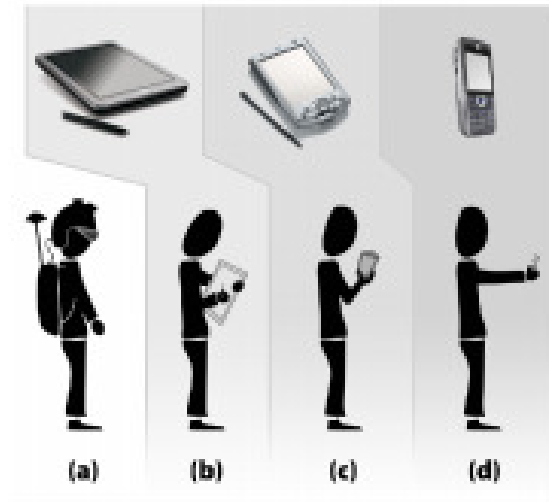
## 1.6 Αρχιτεκτονική ενός Συστήματος ΕΠ για κινητές συσκευές.

Βασικά υπάρχουν μόνο τρία απαραίτητα πράγματα που χρειάζονται για την επαυξημένη πραγματικότητα.

1. Μια κάμερα για να μπορέσουμε να «αιχμαλωτίσουμε» το πραγματικό περιβάλλον
2. Μια επιφάνεια απεικόνισης για να παρουσιάσουμε το τελικό αποτέλεσμα
3. Τέλος μια συσκευής που θα μας παρέχει την απαραίτητη υπολογιστική ισχύ για την δημιουργεί των γραφικών που θα επικαλύψουν την πραγματικότητα.

Οι Σύγχρονες Κινητές Συσκευές, όπως τα έξυπνα κινητά τηλεφώνά και οι ταμπλέτες περιέχουν αυτά τα στοιχεία, τα οποία συχνά περιλαμβάνουν μια φωτογραφική μηχανή και αισθητήρες, όπως επιταχυνσιόμετρο, GPS και πυξίδα στερεάς κατάστασης, αλλά και χάρη στις δυνατότητες δικτύωσης τόσο μέσω ασυρμάτων δικτύων όσο και μέσω των δικτύων κινητών τηλεφώνων τα καθιστά ιδανικές πλατφόρμες για να τρέξουν εφαρμογές επαυξημένης πραγματικότητας.

Στην εικόνα 1-8 βλέπουμε τις συνήθεις συσκευές που χρησιμοποιούνται σε εφαρμογές επαυξημένης Πραγματικότητας και την εξέλιξη τους.



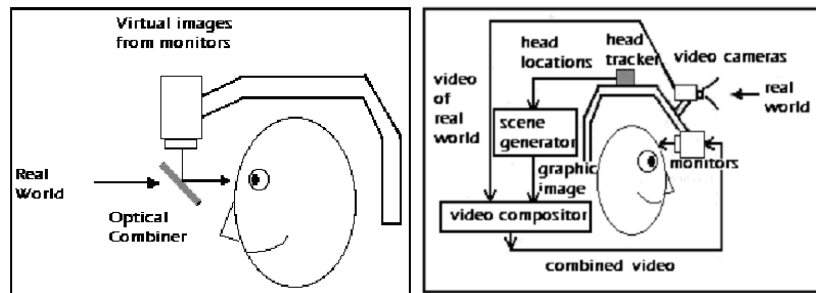
Εικόνα 1-5 : α) Η/Υ και HMD β)Ταμπλέτες γ)PDA δ)έξυπνα κινητά

## 1.7 Συσσκευές Επαυξημένης Πραγματικότητας

### 1.7.1 Φορητές Συσσκευές Απεικόνισης (HMD)

Μια από τις πιο διαδεδομένες συσκευές απεικόνισης. Πρόκειται για συσκευές που φόρουνται στο κεφάλι και προβάλλει τις εικόνες στα μάτια του χρήστη. Υπάρχουν δυο τύποι HMD.

1. **Οπτικές Συσσκευές Απεικόνισης (optical see-through display).** Αποτελούνται από μια διάφανη οθόνη και επιτρέπουν στον χρήστη να δει το πραγματικό περιβάλλον όπως είναι, προβάλλοντας τα εικονικά αντικείμενα πάνω σε αυτή. Συνήθως αποτελούνται από ένα ημιανακλαστικού κάτοπτρου το οποίο μπορεί να ανακλά και να μεταδίδει το φως. Με την σωστή τοποθέτηση μπροστά στο μάτι μπορεί να ανακλά μια εικόνα από τον υπολογιστή στο οπτικό μας πεδίο, χωρίς να μας κρύβει το πραγματικό περιβάλλον.
2. **Βιντεοσυσσκευές απεικόνισης(video see-through displays).** Συνδυάζουν την εικόνα που καταγράφεται από μια κάμερα που φορεί ο χρήστης με γραφικά που παράγονται από υπολογιστή. Το πραγματικό περιβάλλον καταγράφεται από την κάμερα, ψηφιοποιείται και υφίσταται επεξεργασία από τον υπολογιστή με την προσθήκη ή την αφαίρεση του αντικείμενου και παράγεται τελικά μια επαυξημένη εικόνα του περιβάλλοντος. Αποκόπτοντας τον χρήστη από το φυσικό περιβάλλον



Εικόνα 1-6: α) Οπτικές Συσκευές Απεικόνισης β) Βίντεο Συσκευή Απεικόνισης

### 1.7.2 Οθόνες Φορητών Συσκευών.

Συνήθως αφορά φορητές, επίπεδες οθόνες LCD οι οποίες έχουν μια ενσωματωμένη κάμερα μέσα από την αυτή λαμβάνουμε τον πραγματικό κόσμο και τον προβάλλουμε στην οθόνη της συσκευή μας. Πάνω σε αυτή επικαλύπτονται τα πραγματικά αντικείμενα. Δηλαδή ο χρήστης παρατηρεί την σύνθεση του εικονικού και πραγματικού μέσω της οθόνης της φορητής συσκευής του.

Αυτήν την στιγμή είναι η πιο εύκολη και πιο οικονομική για τον χρήστη για να μπορέσει να χρησιμοποιήσει εφαρμογές επαυξημένης πραγματικότητας στην καθημερινή ζωή του. Επίσης αυτές οι συσκευές έρχονται με διάφορους αισθητήρες ενσωματωμένους όπως οι πυξίδες, το επιταχυνσιόμετρο και το GPS αλλά και την δυνατότητα δικτύωσης τους έχουν όλες τις προϋποθέσεις για να έχουν μια καλή εμπορική επιτύχει.

### 1.7.3 Γυαλιά Επαυξημένης Πραγματικότητας

Στο μέλλον η χρήση των γυαλιών το οποίο είναι πολύ πιο εύκολο στην μεταφορά στην χρήση είναι ένα θέμα ακόμα και πιο αποδεκτό ας μην ξεχνάμε τι ποσοστό των ανθρώπων φορούν ήδη γυαλιά βολικό και δεν σε αποσπά και πιθανών άλλων περιφερειακών θα κάνουν την επαυξημένη πραγματικότητα πιο λειτουργική και πιο πρακτική.

Η ιδέα των γυαλιών έχει τεθεί σε πρακτική από την Vuzixs που προσφέρει τα γυαλιά Wrap 920AR για να εμφανίσετε 3D εικόνες στο πραγματικό κόσμο. Παρόμοια με τα Vuzixs, και άλλες εταιρίες ετοιμάζουν τις δίκες τους λύσεις για τα γυαλιά επαυξημένης πραγματικότητας όπως η Nokia , η Microsoft και η Google.

Τον Φεβρουάριο του 2013 η Google παρουσίασε τα Google Glass για τους προγραμματιστές και το 2014 σχεδιάζει να τα διάθεση και στο εύρη κοινό. Αν και είναι ακόμα αρκετά ακριβά 1500 δολάρια, με την πάροδο του χρόνου και των νέων παικτών που



θα μπουν στην αγοράκι τιμή τους θα είναι πιο οικονομική και θα έχουν την δυνατότητα όλο και περισσότεροι χρηστές να τα αποκτήσουν.

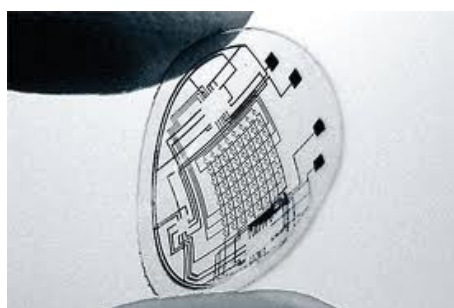
Είναι πολλή πιθανόν στο μέλλον όλοι μας να έχουμε στην διάθεση μας γυαλιά επαυξημένης πραγματικότητας πάντα μαζί μας , όπως συμβαίνει τώρα με τα κινητά τηλεφωνά.



Εικόνα 1-7: Google Glass

### 1.7.3 Φακοί Επαφής

Επίσης οι φακοί επαφής που εμφανίζουν απεικόνιση AR είναι σε εξέλιξη. Αυτά τα βιονικά φακούς επαφής μπορεί να περιέχει τα στοιχεία για την οθόνη ενσωματωμένη στο φακό, συμπεριλαμβανομένων ολοκληρωμένων κυκλωμάτων, LED και μια κεραία για την ασύρματη επικοινωνία. Μια άλλη εκδοχή των φακών επαφής, στην ανάπτυξη για την αμερικανική στρατιωτική , έχει σχεδιαστεί για να λειτουργεί με τα γυαλιά AR, επιτρέποντας στρατιώτες να επικεντρωθεί σε close-to-the-eye εικόνες AR για τα θεάματα και μακρινά αντικείμενα του πραγματικού κόσμου ταυτόχρονα.



Εικόνα 1-8: Φακοί Επαφής Επαυξημένης Πραγματικότητας

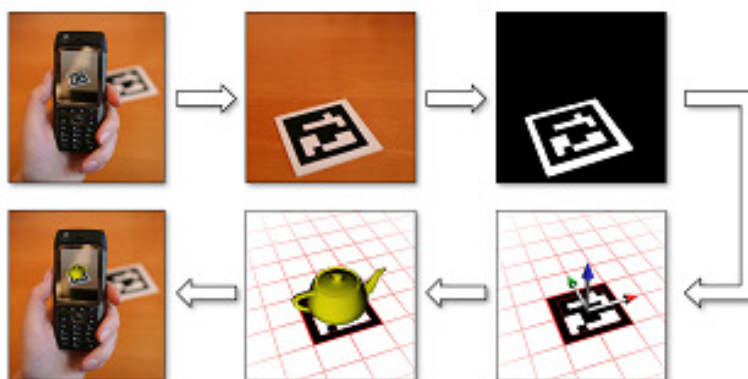
## 1.8 Κύριοι Τύποι Εφαρμογών Επαυξημένης Πραγματικότητας για τις Κινητές Συσκευές.

Τα κινητά τηλεφωνά κατάφεραν σε σχετικά μικρό χρονικό διάστημα να γίνουν αναπόσπαστο κομμάτι της καθημερινότητας μας, κυρίως χάρη στην ταχύτερη βελτίωση των τεχνολογιών των τηλεπικοινωνιών και της μικροηλεκτρονικής .

Τα έξυπνα κινητά τηλεφωνά με την υπολογιστική ισχύ που διαθέτουν άλλα και της δυνατότητας δικτύωσης τους δημιουργούν ένα πολύ σημαντικό οικονομικό κλάδο με πολύ μεγάλα περιθώρια κερδών και ιδιαίτερα ο κλάδος της ανάπτυξης εφαρμογών για τα κινητά τηλεφωνά.

Τα περισσότερα έξυπνα τηλεφωνά είναι εφοδιασμένα με ισχυρούς επεξεργαστές φωτογραφικές μηχανές υψηλής ευκρίνεια, μεγάλη οθόνη , αρκετούς αισθητήρες (GPS, Επιταχυνσιόμετρο κ.α) και δυνατότητες δικτύωσης είτε μέσω ασύρματων δικτύων είτε μέσω 3G και 4G που τα κάνουν ιδανικά για την ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας. Οι εφαρμογές ΕΠ μπορούν να καλύψουν μια μεγάλη ποικιλία από τεχνολογίες και συσκευές άλλα και σκοπού και μπορούν να χωριστού σε διαφορετικές κατηγορίες ανάλογα με το σκοπό που υλοποιούν και τις τεχνολογίες που χρησιμοποιούν. Γενικά μπορούμε να χωρίσουμε της εφαρμογές επαυξημένης πραγματικότητας σε δυο κατηγορίες , τις εφαρμογές που βασίζονται σε σημάδια (marker) και σε εφαρμογές που δεν στηρίζονται στην αναγνώριση σημαδιών άλλα στην αναγνώριση της θέσης του χρηστή και την δυνατότητα να παρέχουν πληροφορίες για αυτήν την τοποθεσία σε πραγματικό χρόνο.

### 1.8.1 Εφαρμογές βασιζόμενες Σε Σημάδια (Marker-based AR)



Εικόνα 1-9:Γενική Ροή μιας Εφαρμογής Επαυξημένης Πραγματικότητας.

Υπάρχει μια βασική λογική που λειτουργούν όλες οι εφαρμογές επαυξημένης πραγματικότητας που βασίζονται σε σημάδια (markers). Αυτή φαίνεται στην Εικόνα 1-13.

- «Σύλληψη» της εικόνας από την κάμερα μας,
- Διαχωρισμός του Σημαδιού από την υπόλοιπη εικόνα.
- Στην συνέχεια εξάγετε το περίγραμμα του σημαδιού.

- Υπολογίσουμε τον πίνακα για το εικονικό μας αντικείμενο από τις τέσσερις ακμές του περιγράμματος του σημαδιού.
- Εφαρμόζουμε τον πίνακα
- Τέλος εμφανίζουμε το αντικείμενο πάνω από την εικόνα του σημαδιού, που έχουμε συλλάβει αρχικά με την κάμερα μας.

### 1.8.1.1 Σημάδια - Markers

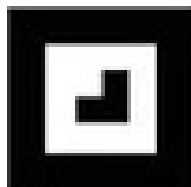
Τα σημάδια (Markers) είναι εικονικά στοιχεία που χρησιμοποιούνται από τις εφαρμογές επαυξημένης πραγματικότητας για να ξέρουν που θα τοποθετηθούν οι υπερκαλύψεις.

Η εφαρμογή μας θα περιέχει τέσσερα σημάδια που θα αναγνωριστούν, για κάθε σημάδι θα παρέχουμε ένα αρχείο .patt που θα χρησιμοποιηθεί το AndAR για να αναγνωριστεί το σημάδι. Αυτά τα αρχεία περιγράφουν το πώς είναι το σημάδι σε ένα τέτοιο τρόπο που μπορεί να αναγνωριστεί από το AndAR μπορούμε αν θέλουμε να κατασκευάσουμε και τα δικά μας σημάδια, εάν το επιθυμούμε.

Αλλά αυτά έχουν κάποιες περιορισμούς όσον αφορά την εμφάνιση αυτών των σημαδιών, που έχει να κάνει με την καλύτερη αναγνώριση τους, πιο συγκεκριμένα θα πρέπει να πληρούν τα παρακάτω κριτήρια:

- Τα σημάδια θα πρέπει να είναι τετράγωνα σε σχήμα.
- Τα όρια του σημαδιού μας πρέπει να έχουν μια καλή αντίθεση.
- Τα όρια του σημαδιού πρέπει να είναι αντίθετα και μονόχρωμα.

Τα σημάδια μπορεί να είναι είτε ασπρόμαυρα ή έγχρωμα. Στην εικόνα βλέπουμε ένα παράδειγμα ενός σημαδιού.



Εικόνα 1-10. Ένα απλό Παράδειγμα Σημαδιού (marker)

Μπορούμε να κατασκευάσουμε τα δικά μας σημάδια μέσα από ένα διαδικτυακό εργαλείο που είναι διαθέσιμο στην παρακάτω διεύθυνση <http://flash.tarotaro.org/blog/2009/07/12/mgo2/>.



Εικόνα 1-11. Ένα Εργαλείο για την Κατασκευή Σημαδιού (marker)

Αυτά τα αρχεία “.pat” πρέπει να αποθηκευτούν μέσα στο αρχείο assets της εφαρμογή μας, όπου θα είναι διαθέσιμα στην εφαρμογή μας.

### 1.8.1.2 Αλγόριθμος υπολογιστικής όρασης (Computer vision algorithm)

Ο αλγόριθμος υπολογιστικής όρασης χρησιμοποιείται για τον εντοπισμό του δείκτη από την εικόνα που έχουμε ως είσοδο από την κάμερα. Αυτό που κάνει είναι να λαμβάνει τα καρέ του βίντεο, έπειτα απομονώνει τα μαύρα τμήματα και με μαθηματικές συναρτήσεις αναγνωρίζει αν τα σχήματα που εμφανίζονται είναι δείκτες ή όχι. Αν υπάρχει δείκτης στην εικόνα τότε ακόλουθη ο μετασχηματισμός συντεταγμένων που περιγράφεται παρακάτω.

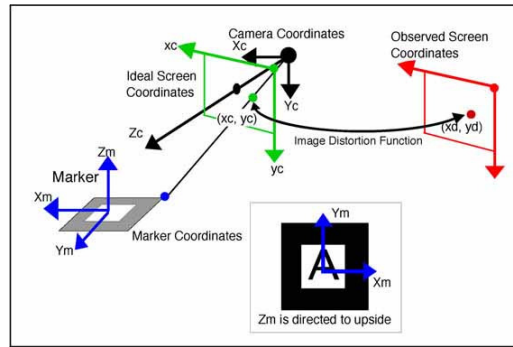
### 1.8.1.3 Σύστημα Συντεταγμένων (Coordinate Systems)

Το ARToolKit υπολογίζει τη θέση του δείκτη στο σύστημα συντεταγμένων της κάμερας και χρησιμοποιεί το σύστημα πινάκων του OpenGL για την θέση του εικονικού αντικείμενου. Έτσι η θέση του δείκτη επιστρέφει σε έναν πίνακα 3X4, με τις ακριβείς συντεταγμένες του.

Το ARToolKit ορίζει διαφορετικά συστήματα συντεταγμένων που χρησιμοποιείται κυρίως από τον αλγόριθμο όραση υπολογιστών και τη σχεδίαση της εικονικής σκηνής.

### 1.8.1.4 Σύστημα Συντεταγμένων υπολογιστικής όρασης (Computer Vision)

Στην εικόνα που έχει σχεδιαστεί από τον Kato συνοψίζει το κύριο σύστημα συντεταγμένων που χρησιμοποιείται από το ARToolKit:

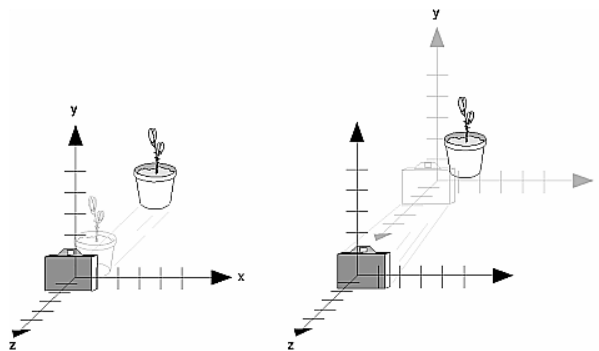


Εικόνα 1-12: Σύστημα Συντεταγμένων του Artoolkit

Πρέπει να σημειωθεί ότι η συνάρτηση `arGetTransMat` μας δίνει την θέση του δείκτη στο σύστημα συντεταγμένων της κάμερας (και όχι το αντίστροφο). Εάν επιθυμούμε να επιστρέψουμε τη θέση της κάμερας στο σύστημα συντεταγμένων του δείκτη θα πρέπει να αντιστρέψουμε τον πίνακα (`arMatrixInverse()`).

### 1.8.1.5 Σύστημα Συντεταγμένων για την σχεδίαση της σκηνής

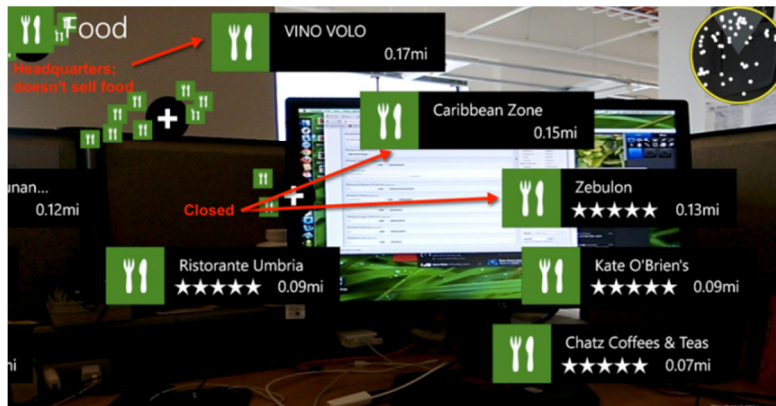
Όταν χρησιμοποιούμε το ARToolKit με το OpenGL, πρέπει να έχουμε στο νου μας ότι το σύστημα συντεταγμένων είναι δεξιόστροφο, με την κάμερα να κοιτά προς την κατεύθυνση  $-Z$  ερχόμενο προς εμάς.



Εικόνα 1-13: Ο μετασχηματισμός του συστήματος συντεταγμένων

Το ARToolKit χρησιμοποιεί μια βαθμονομημένη προοπτική για την κάμερα που συνήθως δίνει ως αποτέλεσμα έναν πίνακα προβολής έκτος άξονα (off-axis projection matrix) για το OpenGL. Μια τέτοια προβολή δεν μπορεί να γίνει από το OpenGL και τις συναρτήσεις που παρέχει αυτό.

## 1.8.2 Εφαρμογές Μη βασιζόμενες Σε Σημάδια (MakerLess AR)



Εικόνα 1-14: Παράδειγμα Εφαρμογής ΕΠ μη βασιζόμενες σε Σημάδια

Οι εφαρμογές μη βασιζόμενες σε σημάδια ή αλλιώς τεχνολογίες πληροφόρηση θέσης (location-aware) και αναγνώρισης περιεχομένου (context-aware) στηρίζονται στο ότι τα σύγχρονα κινητά έρχονται με "ενσωματωμένους αισθητήρες, όπως η φωτογραφική μηχανή, ψηφιακή πυξίδα και το επιταχυνσιόμετρο. Τα όποια μπορούν να χρησιμοποιηθούν για να καθορίσουν την θέση μας και τον προσανατολισμό μας.

Πιο συγκεκριμένα η κάμερα συλλαμβάνει τον κόσμο μέσα από το φακό του και το δείχνει στην οθόνη, ο δέκτης GPS και ο πάροχος δικτύου λάβει την θέση του χρηστή και με την χρηστή της πυξίδας και του επιταχυνσιόμετρο καθορίζουν το οπτικό πεδίο της εφαρμογής μας. Σε συνεργασία με το δεκτή GPS ή Μέσω των Δικτύων Κινητών Τηλεφώνων και μπορούν με την χρήση κάποιας διαδικτυακής υπηρεσίες να μας παρέχει πληροφορίες για την περιοχή που βρισκόμαστε και τα σημεία ενδιαφέροντος που υπάρχουν σε αυτή, μέσω κάποιο κριτήριων που εμείς ορίζουμε μέσα από την συσκευή μας ή μας δίνουν την δυνατότητα να μοιραζόμαστε την τοποθεσία που βρισκόμαστε με τις κινητές συσκευές των φίλων και την οικογένεια μας και με όποιον άλλο επιθυμούμε.

Για παράδειγμα, οι ασύρματοι συνδρομητές μπορούν να παρέχουν σε πραγματικό χρόνο πληροφορίες σχετικά με τους παρατηρητές "περιβάλλον, επικάλυψης και πληροφορίες για τόπους, γύρω κτίρια και τους φίλους που θα μπορούσε να είναι σε κοντινή απόσταση, και αυτές οι πληροφορίες είναι αποθηκεύονται και να μοιραστεί με άλλους μέσω online υπηρεσιών .

Αυτές οι δραστηριότητες οδηγούν σε λεπτομερείς χάρτες και επιτρέπουν επιπρόσθετες πληροφορίες που δεν μπορεί να προσδιοριστεί με την οπτική αντίληψη να εμφανίζεται στο κινητό μας . Τα αντικείμενα που συλλαμβάνεται από κάμερα μπορεί να αναγνωριστούν γρήγορα και να τα παρακολουθούνται με την μετακίνηση της κάμερας ανάλογα με την τοποθεσία του χρήστη και σε πραγματικό χρόνο.

## 1.9 Φυλλομέτρησες ΕΠ για κινητές Συσκευές

Εδώ θα δούμε κάποια από τους γνωστούς φυλλομέτρησες επαυξημένης πραγματικότητας για κινητές συσκευές που χρησιμοποιούν δεδομένα και πληροφορίες από την φυσική τοποθεσία του χρήστη.

### 1.9.1 Wikitude

Το Wikitude (<http://www.Wikitude.com/en/>) είναι εργαλείο ανάπτυξης λογιστικού για εφαρμογές επαυξημένης πραγματικότητας και πρόγραμμα περιήγησης που δίνει την δυνατότητα στους χρήστες του να προβάλλουν στις οθόνες των συσκευών κινητής τηλεφωνίας τους να προβάλλουν πολυφασικό περιεχόμενα (βίντεο, εικόνες, ήχους και διαφορές πληροφορίες) και γραφίστα με πληροφορίες από τα κοινωνικά δίκτυα που είναι γηραλέος ο χρήστης σε ένα επαυξημένο περιβάλλον.

Το Wikitude δημιουργήθηκε από την αυστριακή εταιρεία Wikitude GmbH (πρώην γνωστή ως Mobilizy GmbH) και δημοσιεύτηκε τον Οκτώβριο του 2008 ως δωρεάν λογιστικό. Α Wikitude ήταν το πρώτο διαθέσιμο πρόγραμμα περιήγησης που χρησιμοποιείσαι με τέτοιο προσέγγιση την τεχνολογία Augmented Reality. (Wikipedia, 2010.α)

Αποτελείται από τρία μέρη:

1. Τον Wikitude World Browser
2. Το Wikitude.me
3. Το Wikitude API.

Ο Wikitude World Browser είναι μια εφαρμογή εκτεταμένης πραγματικότητας που διατίθεται για τα λειτουργικά σύστημα Android, iOS και symbian. Επιτρέπει στο χρήστη να αντλεί πληροφορίες για τον χώρο που βρίσκεται και να τις βλέπει ζωντανά μέσα από την κάμερα του κινητού του. Η πληροφορία εμφανίζεται με τη μορφή σημείων ενδιαφέροντος (Points Of Interest- POI) που υπερτίθεται πάνω από την εικόνα της κάμερας.

Ο ρόλος του Wikitude.me είναι ο αντίστροφος αυτού του Wikitude World Browser. Δίνει τη δυνατότητα στους χρήστες του να μοιράζονται μεταξύ τους τοποθεσίες που θεωρούν ενδιαφέρουσες , χρησιμοποιώντας μια διαπροσωπικά βασισμένη σε Google Maps ή ανεβάζοντας τα δικά τους KML ή ARML αρχεία. Επιπλέον παρέχει ένα Webservice σε περίπτωση που ο χρήστης θέλει η πηγή των δεδομένων να βρίσκεται σε δικό του εξυπηρετητή (Server).

Το Wikitude επιτρέπει στους χρήστες της εφαρμογής να ενημερώνουν όλες τις γεωγραφικές πληροφορίες που δημοσιεύονται από τους φίλους τους μέσω του Facebook σε ένα περιβάλλον Augmented Reality τεχνολογίας. Ενημερώσεις Status, ενημερώσεις φωτογραφιών και τοποθεσίας αναγνωρίζονται στο σύνολό τους από το Wikitude, που καθιστά δυνατή την ενημέρωση των κινήσεων ενός φίλου μέσω ενός περιβάλλοντος Augmented Reality. (Wikitude, 2010)

Το περιεχόμενον του προγράμματος Wikitude είναι ως επί το πλείστον παράγεται από τους ίδιους τους χρήστες του προγράμματος. Το περιεχόμενον μπορεί να προστεθεί μέσω ενός δικτυακού περιβάλλοντος (Web interface) με την βοήθεια:

- Αρχείων KML (Keyhole Markup Language). Τα KML είναι μια μορφή αρχείων που χρησιμοποιούνται για προβολή γεωγραφικών πληροφοριών και δεδομένων σε προγράμματα περιήγησης, όπως το Google Earth, Google Maps και Google Maps για κινητά. Στα αρχεία KML μπορούν να οριστούν και να επισημανθούν τοποθεσίες, να προστεθούν εικόνες και άλλα.
- Αρχείων ARML (Augmented Reality Markup Language). Τα ARML είναι μια μορφή αρχείων που επιτρέπει στους προγραμματιστές να δημιουργήσουν περιεχόμενα που εμφανίζονται σε διάφορα προγράμματα περιήγησης για συσκευές κινητής τηλεφωνίας με δυνατότητα χρήσης προγραμμάτων περιήγησης Augmented Reality.

Το Wikitude είναι διαθέσιμο σε λογισμικό iOS και Android για συσκευές κινητής τηλεφωνίας.



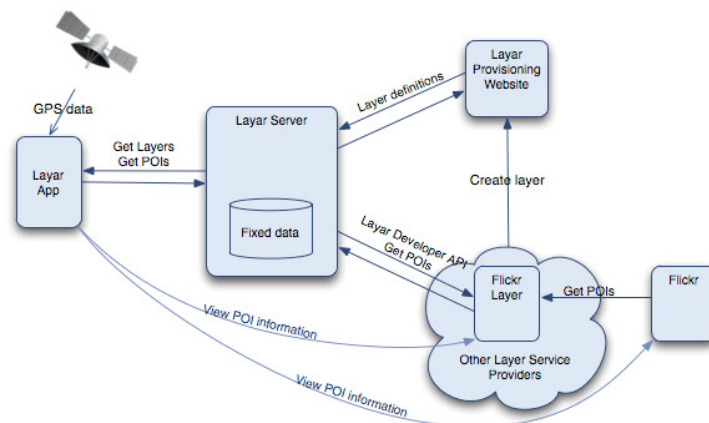
## 1.9.2 Layar

Το Layar (<http://www.layar.com/>), είναι ένα πρόγραμμα περιήγησης αλλά και εργαλείο ανάπτυξης λογισμικού κατασκευασμένο από την ολλανδική εταιρία Layar, επιτρέπει στους χρήστες να εμπλουτίσουν τον φυσικό κόσμο ,όπως αυτό φαίνεται μέσα από την κάμερα του κινητού, με ψηφιακά δεδομένα. Η εφαρμογή είναι διαθέσιμη για τα λειτουργικά συστήματα Android και iPhone .

Το πρόγραμμα περιήγησης Layar κάνει χρήση των ακόλουθων λειτουργιών των συσκευών κινητής τηλεφωνίας για να λειτουργήσει:

- Ενσωματωμένη κάμερα
- Πυξίδα
- GPS
- Επιταχυνσιόμετρο (Accelerometer)

Αυτές οι λειτουργίες χρησιμοποιούνται από κοινού για να προσδιοριστεί η θέση και το οπτικό πεδίο του χρήστη. Από τη γεωγραφική θέση, οι διάφορες μορφές δεδομένων ρίχνονται πάνω που στην κάμερα, σαν να εισάγεται ένα επιπλέον επίπεδο.



Εικόνα 1-25: Αρχιτεκτονική του Layar

**Το Layar αποτελείται από πέντε υποσυστήματα:**

1. Layar App : Είναι η εφαρμογή εκτεταμένης πραγματικότητας που τρέχει στη συσκευή του χρήστη. Χρησιμοποιώντας τα GPS δεδομένα από τον δέκτη της συσκευής στέλνει ερωτήματα προς τον Layar Server για να βρει layers και σημεία ενδιαφέροντος που βρίσκονται κοντά στον χρήστη.

2. Layar Server : Είναι η καρδιά του συστήματος επικοινωνεί με το Layar App για να βρει διαθέσιμα layers και σημεία ενδιαφέροντος. Ταυτόχρονα είναι η γέφυρα επικοινωνίας προς τις βάσεις δεδομένων του Layar αλλά και τρίτων ομάδων.
3. Layar Provisioning Website: Είναι η ιστοσελίδα μέσω της οποίας οποιοσδήποτε μπορεί να δημιουργήσει το δικό του layer. Μαζί του επικοινωνεί ο Layar Server για να μάθει τη δομή κάποιου layer.
4. Layar Service Providers: Αφού βρεθεί η δομή κάποιου layer τότε έρχεται η σειρά των Layar Service Providers για να στείλουν τα περιεχόμενα του. Οι Layer Service Providers μπορεί να ανήκουν στην Layar ή σε οποιονδήποτε χρησιμοποιεί τις υπηρεσίες της.
5. Layer Content Sources : Αποτελούν τις βάσεις δεδομένων όπου αποθηκεύονται τα περιεχόμενα ενός layer. Η αποστολή των δεδομένων τους γίνεται μέσω των Layar ServiceProviders.

Τα δεδομένα και οι πληροφορίες στο πρόγραμμα περιήγησης έρχονται υπό τη μορφή των layers ,τα οποία είναι ότι η ιστοσελίδα για το Internet. Τα layers είναι REST (REpresentational State Transfer) διαδικτυακές υπηρεσίες που εξυπηρετούν γεωγραφικά σημεία ενδιαφέροντος στην περιοχή του χρήστη. Τα Layers αναπτύσσονται και συντηρούνται από τρίτους με τη χρήση δωρεάν προγραμματιστικού περιβάλλοντος (Application Programming Interface, API).

Τα Layers ή αλλιώς εφαρμογές, για την πλατφόρμα Layar, περιλαμβάνουν διάφορους τύπους λειτουργιών με διακριτικά χαρακτηριστικά όπως 3D αντικείμενα και κινούμενα σχέδια.

Τα Layers που είναι βασισμένα στην τοποθεσία του χρηστή βοηθάνε τους χρήστες τους να βρουν τοποθεσίες που τους ενδιαφέρουν, όπως καφετέριες, καταστήματα και άλλες επιχειρήσεις, καθώς και ιστορικές τοποθεσίες και μνημεία. Άλλα Layers δίνουν την δυνατότητα στους χρήστες να παίζουν παιχνίδια μέσα στο περιβάλλον τους, να αναζητήσουν ρούχα σε ένα εικονικό κατάστημα 360 μοιρών ή ακόμα και να προβάλλουν έργα τέχνης που διατίθενται ψηφιακά στον πραγματικό κόσμο (Layar, 2010).

Μια άλλη σημαντική λειτουργία του Layar είναι η αλληλεπίδραση που έχει με τον έντυπο τύπο. Με την βοήθεια των λειτουργιών του Layar οι στατικές και απλές σελίδες ενός περιοδικού, εφημερίδας ή έντυπης διαφήμισης να εμπλουτιστεί με διάφορες διακριτικές και

ψηφιακές λειτουργίες. Το Layar χάρη στην λειτουργία των layers, που περιγράφηκαν παραπάνω, μπορεί να δημιουργήσει και να προβάλει πληροφορίες πάνω από της έντυπες μορφές του τύπου. Ένα σημαντικό πλεονέκτημα αυτής λειτουργίας του Layar είναι ότι η προβολή του παρεχομένου μπορεί να γίνει είτε όταν είμαστε συνδεδεμένοι στο διαδίκτυο είτε όταν δεν είμαστε (Layar, χχ).

### 1.9.3 Junaio

Το Junaio είναι ένας περιηγητής(browsers) επαυξημένης πραγματικότητας και εργαλείο ανάπτυξης λογιστικού σχεδιασμένο για συσκευές κινητής τηλεφωνίας τρίτης γενιάς (3G) και τέταρτης γενιάς (4G).

Αναπτύχθηκε από την εταιρία Metaio GmbH. Το Junaio παρέχει ένα περιβάλλον για την ανάπτυξη Εφαρμογών (Application Programming interface, API) για τους προγραμματιστές και τους παρόχους περιεχομένου να προσφέρουν εφαρμογές επαυξημένης πραγματικότητας για τους τελικούς χρήστες. Το Junaio είναι ο πρώτος περιηγητής επαυξημένης πραγματικότητας που δεν περιορίζεται απλά στη χρήση δεδομένων εντοπισμού θέσης.



Εικόνα 1-16: Παράδειγμα χρήσης της εφαρμογής Junaio

Η πλοήγηση με το Junaio γίνεται μέσω δεικτών γεωγραφικού πλάτους / μήκους / ύψους (latitude/longitude/altitude, LLA). Ο περιηγητής επαυξημένης πραγματικότητας Junaio, το Νοέμβριο του 2009 έγινε διαθέσιμος για συσκευές κινητής τηλεφωνίας με λογισμικό IOS και Android. (Wikipedia, 2010)

# 2: Το Λειτουργικό Σύστημα Android

---

## 2.1 Τι είναι το Android;

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

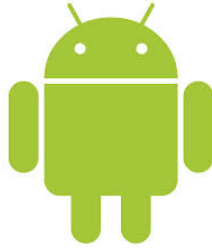
Το Android είναι μια ολοκληρωμένη ανοιχτού κώδικα πλατφόρμα για κινητές συσκευές που περιλαμβάνει το λειτουργικό σύστημα, το απαραίτητο ενδιάμεσο λογισμικό, και τις βασικές εφαρμογές μαζί με ένα σύνολο βιβλιοθηκών API για τη κατασκευή εφαρμογών για κινητές συσκευές.. Εφαρμογές γραμμένες σε C και άλλες γλώσσες μπορούν να μεταγλωττιστούν στον κώδικα ARM και να τρέξουν.

## 2.2 Η πορεία του Android

Τον Ιούλιο του 2005, η Google απέκτησε την Android Inc. μια μικρή νεοσύστατη εταιρεία που έδρευε στο Πάλο Άλτο, της Καλιφόρνια, στις ΗΠΑ

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής Υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας.

Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινο μήλο και σχεδιάστηκε από τη γραφίστρια Irina Blok.



Εικόνα 2-1: Λογότυπο του Android

Στις 12 Νοεμβρίου 2007 κυκλοφόρησε το πρώτο δοκιμαστικό SDK. Οι προγραμματιστές έχουν την δυνατότητα να παρέχουν σχόλια και εντυπώσεις (feedback) και έτσι να διαμορφώσουν την ανάπτυξη του SDK.

Στις 28 Αυγούστου 2008 αναγγέλλεται επίσης η δημιουργία του Android Market. Το Android Market είναι ένα Ανοικτό σύστημα διανομής εφαρμογών Android που συνδέει χρήστες και κατασκευαστές. Το Android Market έγινε στην ουσία διαθέσιμο στους χρήστες στις 22 Οκτωβρίου 2008. και υποστηρίζει δωρεάν εφαρμογές και εφαρμογές επί πληρωμή.

Στις 8 Δεκεμβρίου 2008 κυκλοφορεί η δεύτερη έκδοση του 1.0 Android SDK η οποία λύνει έναν αριθμό από προβλήματα ενώ τον Φεβρουάριο του 2009 κυκλοφορεί η πρώτη έκδοση του Android 1.1 SDK

Τον Σεπτέμβριο του 2009 αναβαθμίστηκε το SDK σε 1.6 . Με τις καινούργιες προσθήκες και αναβαθμίσεις το Android έχει τη δυνατότητα να λειτουργεί σε μεγαλύτερο εύρος συσκευών και δικτύων.

Τον Νοέμβριο του 2009 κυκλοφόρησε το Android SDK 2.0 το οποίο παρέχει καινούρια APIs όσο αφορά στη διαχείριση και συγχρονισμό λογαριασμών και επαφών , Bluetooth και διεπαφές χρήστη. Η πιο επαναστατική εφαρμογή που λειτουργεί μόνο στην έκδοση 2.0 είναι το Google Maps Navigation η οποία μετατρέπει το Android κινητό σε σύστημα Navigation.

Η εξέλιξη του Android σαν λειτουργικό Σύστημα είναι παρά πολλή γρήγορη και αυτό οφείλεται κυρίως στην ανοικτή (open source) διάθεση του κώδικα του λειτουργικού αλλά και στα χαρακτηριστικά του Android SDK που αναλύονται παρακάτω.

Πιο κάτω είναι ένας πίνακας που παρουσιάζει τις εκδόσεις του Android και τι ποσοστό καταλαμβάνει κάθε διανομή του.

Έκδοση	Κωδική ονομασία	Ημερομηνία	API level	Διανομή <sup>[16]</sup>
4.4	Kitkat	31 Οκτωβρίου 2013	19	0%
4.3	Jelly Bean	24 Ιουλίου 2013	18	2,3%
4.2.x	Jelly Bean	13 Νοεμβρίου 2012	17	12,5%
4.1.x	Jelly Bean	9 Ιουλίου 2012	16	37,3%
4.0.x	Ice Cream Sandwich	16 Δεκεμβρίου 2011	15	19,8%
3.2	Honeycomb	15 Ιουλίου 2011	13	0,1%
2.3-2.3.7	Gingerbread	9 Φεβρουαρίου 2011	10	26,3%
2.2	Froyo	20 Μαΐου 2010	8	1,2%
2.0-2.1	Eclair	26 Οκτωβρίου 2009	7	?
1.6	Donut	15 Σεπτεμβρίου 2009	4	?

Πινάκας 2-1 : Εκδόσεις του λειτουργικού συστήματος Android

### 2.3 Χαρακτηριστικά γνωρίσματα του Android SDK

Το πραγματικό πρόσωπο του Android σαν ένα περιβάλλον ανάπτυξης βρίσκεται στα APIs που παρέχει. Σαν μια ουδέτερη όσο αφορά τις εφαρμογές πλατφόρμα, το Android δίνει την ευκαιρία να δημιουργηθούν εφαρμογές που είναι τόσο μέρος του τηλεφώνου όσο και εκτός του πακέτου. Παρακάτω σημειώνονται μερικά από τα πιο αξιοσημείωτα χαρακτηριστικά γνωρίσματα του Android:

- Καμία οικονομική επιβάρυνση όσο αφορά τη χορήγηση αδειών, τη διανομή, ή την αμοιβή ανάπτυξης του λογισμικού.
- WI-IF Πρόσβαση υλικού.
- GEM, EDGE, 3G και 4G δίκτυα για τηλεφωνία ή μεταφορά δεδομένων, που επιτρέπει σε κάποιον να πραγματοποιήσει ή να λάβει κλήσεις ή μηνύματα, ή να στείλει και να ανακτήσει δεδομένα μέσω των δικτύων κινητής τηλεφωνίας.
- Εύκολα και κατανοητά APIs τοποκεντρικών (location-based) υπηρεσιών όπως το GPS.

- Πλήρης έλεγχος του υλικού πολυμέσων συμπεριλαμβανομένου της αναπαραγωγής ήχου και εγγραφής, χρησιμοποιώντας τη φωτογραφική μηχανή και το μικρόφωνο.
- APIs για υλικό επιταχυνσιόμετρο (accelerometer) και πυξίδων (compass).
- Διαμοιραζόμενες βάσεις δεδομένων.
- Έναν ενσωματωμένο ανοικτού κώδικα βασισμένο σε webKit φυλλομετρητή.
- Πλήρης υποστήριξη εφαρμογών που ενσωματώνουν χειρισμό χαρτών ως μέρος της διεπαφής χρήστη.
- Υποστήριξη Peer-to-peer(P2P) χρησιμοποιώντας το Google Talk.
- Βελτιστοποιημένα γραφικά που περιλαμβάνουν path-based 2D βιβλιοθήκη γραφικών και υποστήριξη για τα 3D γραφικά χρησιμοποιώντας OpenGL ES.
- Πολυμεσικές Βιβλιοθήκες για αναπαραγωγή και για εγγραφή εικόνων και video σε ποικίλες μορφές.
- Ένα πλαίσιο εφαρμογών που ενθαρρύνει την επαναχρησιμοποίηση τμημάτων
- εφαρμογών και αντικατάσταση των εγγενών εφαρμογών.

## 2.4 Το πλαίσιο ανάπτυξης του Android

Οι εφαρμογές Android γράφονται χρησιμοποιώντας την Java ως γλώσσα προγραμματισμού αλλά εκτελούνται χρησιμοποιώντας την εικονική μηχανή (virtual machine) Dalvik και όχι μια παραδοσιακή Java VM.

Κάθε εφαρμογή Android τρέχει σε μια ξεχωριστή διαδικασία μέσα στο δικό της στιγμιότυπο της Dalvik VM, εγκαταλείποντας όλη την ευθύνη για μνήμη και διαχείριση διαδικασιών στο χρόνο εκτέλεσης, ο οποίος σταματά και τερματίζει τις διαδικασίες για να κάνει καλύτερη διαχείριση των πόρων του.

Η Dalvik VM και ο χρόνος εκτέλεσης του Android βρίσκονται πάνω από έναν πυρήνα Linux που χειρίζεται τη χαμηλού επιπέδου αλληλεπίδραση υλικού συμπεριλαμβανομένων των οδηγών (drivers) και της διαχείρισης μνήμης, ενώ ένα σύνολο APIs παρέχει πρόσβαση σε όλες τις υπηρεσίες, στα χαρακτηριστικά γνωρίσματα του κινητού, και στο υλικό.

## 2.5 Πακέτο Ανάπτυξης Λογισμικού (software development kit , SDK)

Το Πακέτο Ανάπτυξης Λογισμικού (software development kit SDK) του Android περιλαμβάνει όλα αυτά που χρειάζεται κανείς για να δημιουργήσει, να ελέγξει και να διορθώσει μια εφαρμογή Android. Στο SDK περιλαμβάνονται:

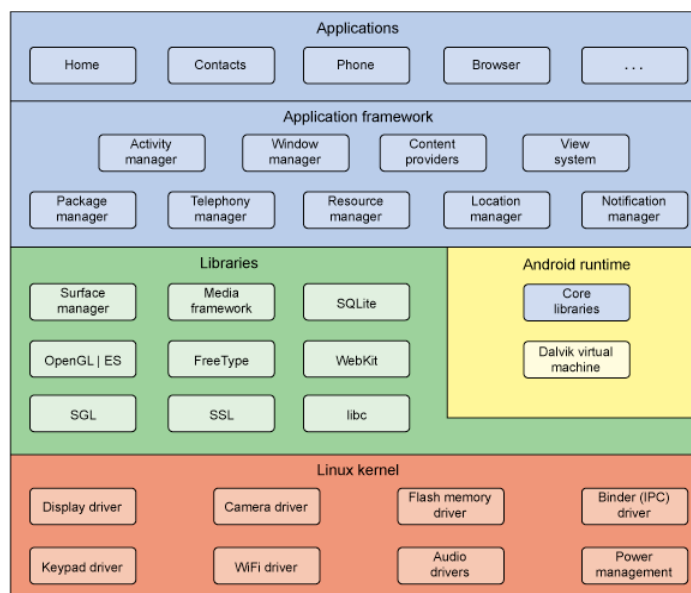
- Τα Android APIs. - Ο πυρήνας του SDK είναι οι Android API βιβλιοθήκες που δίνουν στους κατασκευαστές τη δυνατότητα να έχουν πρόσβαση στη συλλογή λογισμικού του Android (Android stack). Αυτές είναι οι ίδιες βιβλιοθήκες που χρησιμοποιούνται από τη Google για να δημιουργήσουν τις εγγενείς (native) εφαρμογές του Android.
- Τα εργαλεία ανάπτυξης (Development Tools). - Για τη μετατροπή του πηγαίου κώδικα σε εκτελέσιμες εφαρμογές Android, το SDK περιλαμβάνει διάφορα εργαλεία ανάπτυξης που δίνουν τη δυνατότητα να συνταχτούν και να διορθωθούν οι εφαρμογές που περιλαμβάνουν έναν εξομοιωτή εργαλεία για αποσφαλμάτωση (debugging) και εργαλεία για την βελτίωση της χρήσης μνήμης αλλά και την βελτίωση της απόδοσης των εφαρμογών μας
- Τον εξομοιωτή του Android (Android Emulator). - Ο εξομοιωτής του Android είναι ένας πλήρως διαδραστικός εξομοιωτής συσκευών Android που περιέχει διάφορες εναλλακτικές προσόψεις (skins). Μέσω του εξομοιωτή, φαίνεται η όψη και η συμπεριφορά μιας εφαρμογής έτσι όπως θα φαινόταν σε μια πραγματική συσκευή Android. Όλες οι εφαρμογές Android μπορούν να λειτουργήσουν μέσα στη Dalvik VM για το λόγο αυτό ο εξομοιωτής λογισμικού είναι ένα άριστο περιβάλλον - είναι ουδέτερο όσο άφορα την αναπαράσταση κάποιας συσκευής, έτσι παρέχει ένα καλύτερο ανεξάρτητο περιβάλλον δοκιμής από οποιαδήποτε άλλο υλικό δοκιμών εφαρμογών.
- Πλήρης τεκμηρίωση. - Το SDK περιλαμβάνει εκτενείς και λεπτομερείς πληροφορίες για το τι ακριβώς περιλαμβάνεται σε κάθε πακέτο και κλάση και πώς να χρησιμοποιηθεί. Επίσης εκτός από την τεκμηρίωση του κώδικα, δίνονται οδηγίες για το πως μπορεί να ξεκινήσει να δημιουργεί κανείς μια εφαρμογή καθώς επίσης και λεπτομερείς εξηγήσεις των βασικών αρχών πίσω από την ανάπτυξη εφαρμογών Android.
- Δείγματα κώδικα (Sample Code). - Το SDK του Android περιλαμβάνει έναν αριθμό εφαρμογών που επιδεικνύουν μερικές από τη δυνατότητες που γίνονται διαθέσιμες χρησιμοποιώντας το Android, καθώς επίσης και απλών προγραμμάτων που δίνουν έμφαση στο πώς χρησιμοποιούνται μεμονωμένα χαρακτηριστικά γνωρίσματα του API.



- Online υποστήριξη. -Το Android έχει δημιουργήσει μια αναπτυσσόμενη και ενεργή κοινότητα κατασκευαστών λογισμικού. Οι ομάδες του Google στο <http://developer.android.com/index.html> είναι ενεργό φόρουμ κατασκευαστών Android μέσα στους οποίους ανήκουν και μέλη της ομάδας ανάπτυξης Android της Google.

## 2.6 Αρχιτεκτονική της Πλατφορμα του Android

Η Αρχιτεκτονική του λογισμικού του Android φαίνεται στην Εικόνα 2-2 . Ένας πυρήνας Linux και μια συλλογή βιβλιοθηκών C/C++ μέσω ενός πλαισίου εφαρμογών παρέχει υπηρεσίες για το χρόνο εκτέλεσης και τις εφαρμογές καθώς και για τρόπους διαχείρισης τους.



Εικόνα 2-2: Αρχιτεκτονική του Λογισμικού Android

**Πυρήνας Linux (Linux Kernel)** - Υπηρεσίες πυρήνων (συμπεριλαμβανομένων των οδηγών υλικού, διαχείριση διαδικασιών, μνήμης, ασφάλειας, δικτύου, και ενέργειας) χειρίζονται από ένα 2.6 πυρήνα Linux. Ο πυρήνας παρέχει επίσης ένα (abstraction layer) αφαιρετικό στρώμα μεταξύ του υλικού και του υπολοίπου σωρού .

**Βιβλιοθήκες (Libraries)** – Λειτουργούν πάνω από τον πυρήνα. Το Android περιλαμβάνει διάφορες βιβλιοθήκες πυρήνων C/C++ όπως την libc και την SSL, καθώς επίσης και:

- Μια βιβλιοθήκη πολυμέσων για την αναπαραγωγή πολυμέσων.

- Ένα διαχειριστή Επιφάνειας (Surface manager ) που διαχειρίζεται τον τρόπο με τον οποίο φαίνονται τα διάφορα στοιχεία.
- Βιβλιοθήκες γραφικών που περιλαμβάνουν SGL και OpenGL για τα 2D και 3D γραφικά.
- SQLite για την εγγενή υποστήριξη βάσεων δεδομένων
- SSL και WebKit για την μηχανή αναζήτησης Ιστού και για την ασφάλεια Διαδικτύου

**Ο Χρόνος Εκτέλεσης του Android (Android Runtime)** - Αυτό που κάνει ένα Android τηλέφωνο στην ουσία ένα «Android τηλέφωνο» και όχι μια διεπαφή Linux κινητών είναι ο χρόνος εκτέλεσης. Συμπεριλαμβανομένων των βιβλιοθηκών πυρήνα και της Dalvik VM, ο χρόνος εκτέλεσης του Android είναι η κινητήρια δύναμη των εφαρμογών και, μαζί με τις βιβλιοθήκες, αποτελούν τη βάση για το πλαίσιο εφαρμογών.

**Βιβλιοθήκες πυρήνα (Core Libraries).** - Αν και η ανάπτυξη Android εφαρμογών γίνεται σε Java, η Dalvik VM δεν είναι μια Java VM. Οι βιβλιοθήκες πυρήνα παρέχουν μεγαλύτερη χρησιμότητα από βιβλιοθήκες πυρήνα της Java καθώς επίσης και από άλλες βιβλιοθήκες του Android (Android-specific libraries).

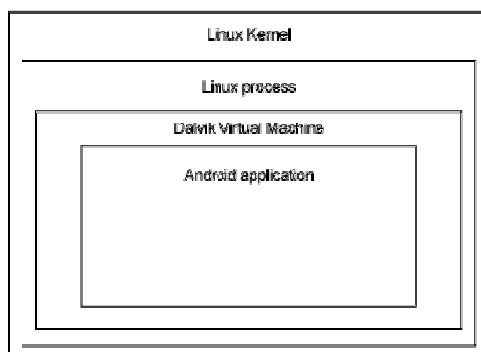
**Η εικονική μηχανή Dalvik (Dalvik Virtual Machine)** - Η Dalvik είναι μια εικονική μηχανή βασισμένη σε μητρώα (register-based) που έχει βελτιστοποιηθεί για να εξασφαλίσει ότι μια συσκευή μπορεί να τρέξει πολλαπλά στιγμιότυπα αποδοτικά. Στηρίζεται στον πυρήνα Linux για να κάνει νηματισμό (threading) και να διαχειριστεί χαμηλά επίπεδα μνήμης.

**Το Πλαίσιο Εφαρμογών (Application Framework)** - Το πλαίσιο εφαρμογών παρέχει τις κλάσεις που χρησιμοποιούνται για να δημιουργηθούν εφαρμογές Android. Παρέχει επίσης μια γενική αφαιρετικότητα (abstraction) για την πρόσβαση υλικού και διαχειρίζεται τους πόρους της διεπαφής χρήστη (user interface) και των εφαρμογών.

**Το στρώμα εφαρμογών (Application Layer)** - Όλες οι εφαρμογές, και οι εγγενείς και αυτές τρίτων, έχουν χτιστεί πάνω στο στρώμα εφαρμογών χρησιμοποιώντας τις ίδιες βιβλιοθήκες API. Το στρώμα εφαρμογών λειτουργεί μέσα στον Android σε χρόνο εκτέλεσης χρησιμοποιώντας τις κλάσεις και τις υπηρεσίες που παρέχονται από το πλαίσιο εφαρμογών.

### 2.5.1 Η εικονική μηχανή Dalvik

Ένα από τα στοιχεία κλειδί του Android είναι η εικονική μηχανή Dalvik. Το Android χρησιμοποιεί την δικιά του εικονική μηχανή, και όχι μια παραδοσιακή με σκοπό να εξασφαλίσει ότι πολλαπλά στιγμιότυπα τρέχουν αποτελεσματικά σε μια ενιαία συσκευή.



Εικόνα 2-3: Dalvik VM

Η Dalvik VM είναι μια τροποποιημένη Εικονική Μηχανή που είναι σχεδιασμένη να τρέχει σε συσκευές που βασίζονται σε συσκευές, με «αργούς» επεξεργαστές και λίγη μνήμη. Βασίζεται στο Apache Harmony και χρησιμοποιεί τον πυρήνα Linux της συσκευής για να χειριστεί τη χαμηλού επιπέδου λειτουργίες συμπεριλαμβανομένου την ασφάλειας, πολυκομματισμού, και διαχείρισης διαδικασιών(process) και μνήμης.

Είναι επίσης δυνατό να γραφτούν εφαρμογές C/C++ που τρέχουν άμεσα στο εσωτερικό του λειτουργικού. Μέσω της Dalvik VM επιτυγχάνεται η ρύθμιση της πρόσβασης στο υλικό και στις υπηρεσίες του συστήματος. Με τη χρησιμοποίηση αυτής της εικονικής μηχανής στην εκτέλεση εφαρμογής, η οποία προσφέρει ένα αφαιρετικό στρώμα, οι κατασκευαστές δεν χρειάζεται να ανησυχήσουν για κάποια υλοποίηση υλικού (hardware implementation).

Η Dalvik VM εκτελεί τα Dalvik εκτελέσιμα αρχεία, μια μορφή βελτιστοποιημένη έτσι ώστε να καταλαμβάνει την ελάχιστη μνήμη. Κανονικά τα αρχεία Java μεταγλωττίζεται σε αρχεία .class. Αλλά η DVM δεν μπορεί να τα εκτελέσει αυτά αλλά χρησιμοποιώντας εργαλεία που παρέχονται μέσα στο SDK μετασχηματίζοντας Java κλάσεις που έχουν σε αρχεία .dex. Ένα αρχείο .dex μπορεί να περιέχει πολλές κλάσεις και το τελικό μέγεθος του αρχείου θα είναι πολύ μικρότερο. Αυτό οφείλεται κυρίως στο ότι μια απλή Java VM είναι μια εικονική μηχανή βασισμένη σε στοίβα (stack-based). Η Dalvik VM από την άλλη είναι μια εικονική μηχανή βασισμένη σε μητρώα (register-based), έτσι αυξάνεται η αποδοτικότητα του επεξεργαστή του κινητού. Επίσης οι εικονικές μηχανές που είναι βασισμένες σε μητρώα επιτρέπουν ταχύτερους χρόνους εκτέλεσης των μεγάλων προγραμμάτων.

## 2.7 Βιβλιοθήκες ΕΠ για το Android

Εδώ θα δούμε κάποιες από τις βιβλιοθήκες που είναι διαθέσιμες για την ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας για το λειτουργικό σύστημα Android.

### 2.7.1 ARToolkit

Το ARToolkit (<http://sourceforge.net/projects/artoolkit/>) είναι μια βιβλιοθήκη λογισμικού ανοικτού κώδικα η οποία βοηθά να δημιουργήσεις εφαρμογές που χρησιμοποιούν επαυξημένη πραγματικότητα που βασίζονται σε σημάδια (marker based AR) που αναπτύχθηκε από το πανεπιστήμιο της Ουάσιγκτον και αποτελεί την βάση για το πλαίσιο AndAR.

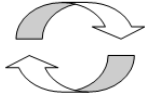
Τα κύρια χαρακτηριστικά του ARToolkit είναι:

- Ενιαία θέση/Προσανατολισμού παρακολούθησης κάμερας. (Single camera position/orientation tracking).
- Ο κώδικας παρακολούθησης που χρησιμοποιεί απλό μαύρα τετράγωνα για δείκτες.
- Η ικανότητα να χρησιμοποιεί κάθε τετράγωνα marker patterns.
- Εύκολος κωδικός βαθμονόμησης (calibration) της κάμερας.
- Αρκετά γρήγορος για πραγματικού χρόνου εφαρμογές AR.
- Διαθέσιμο για τα λειτουργικά συστήματα SGI IRIX, Linux, MacOS και τα Windows OS
- Διανέμεται με πλήρη πηγαίο κώδικα.
- Ελεύθερο για μη εμπορική χρήση σύμφωνα με την αδεία χρήσης GNU

Είναι κάτω από την αδεία General Public Licence(GPL). Αυτός είναι και ο λόγος που αποτελεί την βάση για πολλά έργα και εργασίες που έχουν να κάνουν με την επαυξημένη πραγματικότητα, παρόλα αυτά υπάρχει και διαθέσιμη εμπορική αδεία. Παρόλο που έχει γραφεί σε portable C , έχει σχεδιαστεί για να τρέχει σε σταθερούς υπολογιστές και όχι σε κινητές συσκευές.

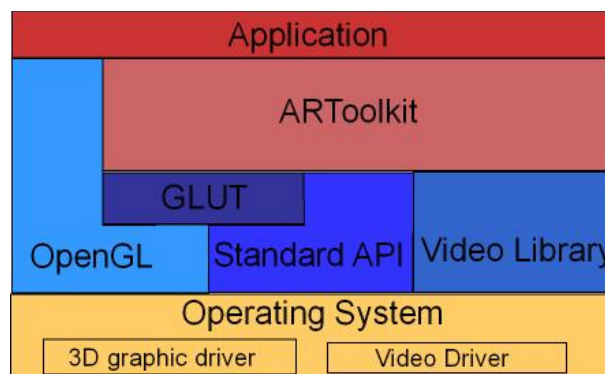
Στον πίνακα 3.1 βλέπουμε τα στάδια λειτουργίας μιας εφαρμογής που κάνει χρήση του ARToolkit. Τα βήματα από το 2 έως το 5 είναι ο κυρίως loop και εκτελείται συνεχώς μέχρι να τερματιστεί η εφαρμογή μας. Κατά την διάρκεια της αρχικοποιήσεως ανοίγει η κάμερα της συσκευής μας, η επιφάνεια OpenGL αρχικοποιείται και διαβάζονται ειδικοί παράμετροι του

ARToolkit που συμπεριλαμβάνουν τα χαρακτηριστικά της κάμερας και τα δεδομένα του pattern . Τα βήματα 2,3, και 4 αντιστοιχούν σε συγκεκριμένες κλήσεις συναρτήσεων . Στο βήμα 5 ο πίνακας μεταφοράς δημιουργείται από το ARToolkit:

<b>Αρχικοποίηση</b>	1. Έναρξη της λειτουργίας της κάμερας, ανάγνωση των αρχείων με τα προαποθηκευμένα marker patterns και των παραμέτρων της κάμερας
<b>Κυρίως loop</b> 	2. Ανάγνωση του τρέχοντος video frame
	3. Εντοπισμός των marker και αναγνώριση των pattern μέσα στο frame
	4. Υπολογισμός της θέσης της κάμερας σε σχέση με τα αναγνωρισμένα markers
	5. Εμφάνιση εικονικών αντικειμένων πάνω στα αναγνωρισμένα markers
<b>Τερματισμός</b>	6. Τερματισμός της λειτουργίας της κάμερας

Πίνακας 2-2 : Τα στάδια λειτουργίας του ARToolkit:

Η βασική αρχιτεκτονική του ARToolkit μπορούμε να την δούμε στην εικόνα . Βασίζεται στο OpenGL για την ερμηνεία (rendering) , το GLUT για την δημιουργεί του παράθυρου OpenGL, μια βιβλιοθήκη για το βίντεο που είναι ανεξάρτητα από την συσκευή υλικού της συσκευής και ένα βασικό API (το οποίο αναπαριστά τα ανεξάρτητα τμήματα της πλατφόρμας). Το ARToolkit αποτελείται από τρία διαφορετικά modules.



Εικόνα 2-4 : Αρχιτεκτονική του ARToolkit

- AR module: το βασικό module με ρουτίνες εντοπισμού δείκτη, βαθμονόμηση και συλλογή παραμέτρων.
- Video module: μια συλλογή από ρουτίνες βίντεο για τη σύλληψη των καρέ του βίντεο εισόδου.
- Gsub module: μια συλλογή από ρουτίνες γραφικών βασισμένες στις βιβλιοθήκες OpenGL και GLUT.

- Gsub Lite: αντικαθιστά το Gsub με μια πιο αποδοτική συλλογή ρουτινών για γραφικά.

### 2.7.2 ARToolkitPlus

Το ARToolkitPlus είναι μια βιβλιοθήκη που βασίζεται στο ARToolkit αναπτύχθηκε στο Τεχνικό Πανεπιστήμιο Graz σαν τμήμα του έργου Handheld AR αλλά έχει σταμάτησε πλέον να αναπτύσσεται από το 2006 [http://studierstube.icg.tugraz.at/handheld\\_ar/artoolkitplus.php](http://studierstube.icg.tugraz.at/handheld_ar/artoolkitplus.php). Αλλά Η βιβλιοθήκη γράφτηκε ξανά σε C++, και απέκτησε έτσι ένα αντικειμενοστραφές API, το οποίο είναι βελτιωμένο για κινητές συσκευές και χρησιμοποιεί έναν νέο αλγόριθμο για τον υπολογισμό της θέσης (pose estimation) ενός αντικείμενου.

### 2.7.3 NyARToolKit

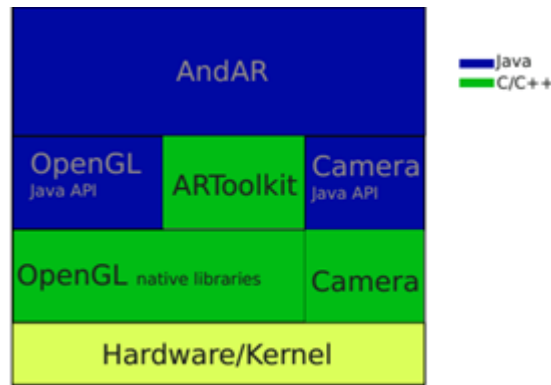
Η NyARToolkit είναι μια μεταφορά της ARToolKit γραμμένη στην γλώσσα Java. Αυτό την καθιστά πιο αργή στην εκτέλεση από την αρχική έκδοση, αλλά έχει το πλεονέκτημα ότι είναι ανεξάρτητη από την αρχιτεκτονική του συστήματος. Μπορεί να τρέξει σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα.

Όπως και το ARToolKit έτσι και το NyARToolKit είναι μια βιβλιοθήκη οπτικών συναρτήσεων και αλληλεπίδρασης δεδομένων εικονικής πραγματικότητας σε φυσικό περιβάλλον, συμπεριλαμβάνοντας τη λειτουργία της κάμερας σε πραγματικό χρόνο παρέχει marker base AR tracking.

Μερικά από τα βασικά χαρακτηριστικά:

- Εντοπισμός ΕΠ με φυσικό δείκτη (Marker based AR tracker)
- Υποστήριξη desktop και φορητές πλατφόρμες
- Βελτιστοποιημένη και την καλύτερη αναγνώριση δείκτη
- Επέκταση για το Unity και διαθέσιμα για επεξεργασία από C# και Java αντίστοιχα

Το NyARToolkit είναι διαθέσιμο με εμπορική αδεία ή μπορεί κάποιος να την κατάβαση ελεύθερα και να την χρησιμοποιήσει υπό την άδεια GPLv2.



Εικόνα 2-5: Αρχιτεκτονική του AndAR

## 2.8. Η Βιβλιοθήκη AndAR

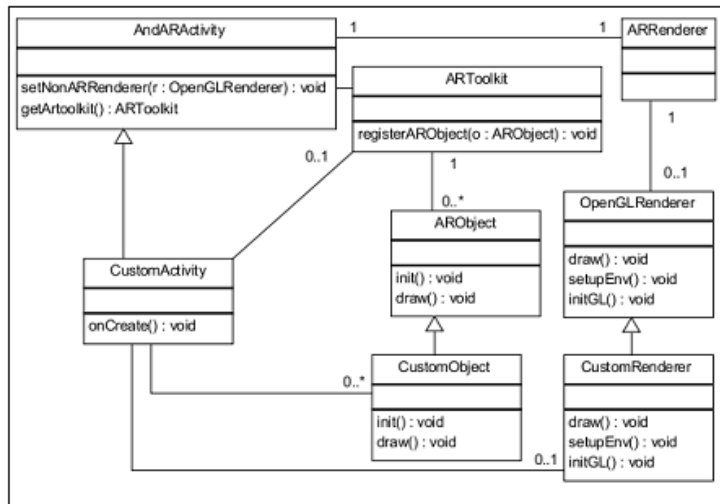
Το AndAR είναι ένα project που μας δίνει την δυνατότητα να έχουμε επαυξημένη πραγματικότητας στην πλατφορμα του Android, είναι κάτω από την αδειά του GNU (General Public Licence). Μας παρέχει ένα καθαρό JAVA API για το ARToolkit κρύβοντας μας όλες τις native κλήσεις βιβλιοθηκών.

Στην εικόνα 2-5 φαίνεται η αρχιτεκτονική του AndAR. Το οποίο χρησιμοποιεί το JAVA API για να έχουμε πρόσβαση στην κάμερα και στην ροη του βίντεο που προέρχεται μέσω αυτής. Η εικόνα στην συνέχεια την χειρίζεται η βιβλιοθήκη ARToolkit. Παράλληλα με αυτό η εικόνα μας μπορεί να διαχωριστεί στα επιμέρους χρώματα που την αποτελούν.

Μέσα αστών βιβλιοθήκη γίνονται μια σειρά από υπολογισμούς για τον πίνακα μετασχηματισμών του τρισδιάστατου αντικείμενο μας. Μέσω του JNI ο πινάκας αυτός παίρνει πίσω στον κώδικα JAVA. Η επεξεργασμένη πλέον εικόνα θα φορτωθεί σαν ένα OpenGL texture , θα εκτελεστεί ο πινάκας μετασχηματισμού και το τρισδιάστατο μας αντικείμενο μας θα σχεδιαστεί.

### 2.8.1 Δομή

Το AndAR είναι ένα πλαίσιο για ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας. Δεν μας παρέβη ένα καθαρό Java API αλλά και μια αντικειμενοστραφή προσέγγιση. Στην εικόνα 12 μπορούμε να δούμε το διάγραμμα κλάσεων του πλαισίου AndAR.



Εικόνα 2-6: Το Διάγραμμα κλάσεων του AndAR

Κάθε εφαρμογή Android αποτελείται από τουλάχιστον μια ή και περισσότερες Activities, δηλαδή μια γραφική διαπάλη χρήστη που υλοποιεί ένα συγκεκριμένη σκοπό. Μονό μια μπορεί να είναι ενεργοποιημένη κάθε φορά. Για να γράψουμε μια εφαρμογή επαυξημένης πραγματικότητας χρησιμοποιώντας το πλαίσιο AndAR, πρέπει να επεκτείνουμε την αφαιρεμένη κλάση AndARActivity.

Αυτή η κλάση χειρίζεται ότι είναι σχετικό με την επαυξημένη πραγματικότητα, όπως το να ανοίγουμε την κάμερα, ανίχνευση του σημαδιού (marker) και να παρουσιάζει το video stream.

Για να το κάνετε αυτό, θα πρέπει να εγγραφείτε(register) ARObjects σε ένα στιγμιότυπο του ARToolkit. Αυτό το στιγμιότυπο μπορεί να ανακτηθεί από την AndARActivity. Η κλάση ARObject είναι αφηρημένη. Αυτό σημαίνει, ότι μπορούμε να την επεκτείνουμε. Αυτό περιμένει το όνομα ενός αρχείου pattern στον κατασκευαστή του. Αυτό το αρχείο πρέπει να βρίσκεται στο φάκελο assets του Προγράμματος μας.

Τα Αρχεία Pattern μπορούν να δημιουργηθούν από ένα αρχείο που ονομάζεται mk\_patt, που βρίσκεται στην ιστοσελίδα του ARToolkit website. Αυτά χρησιμοποιούνται έτσι ώστε να μπορούμε να διακρίνουμε τα διαφορετικά σημάδια μεταξύ τους. Για να σχεδιάσουμε ένα τροποποιημένο αντικείμενο, θα πρέπει να υπερκαλύψουμε (overridden) την μέθοδο σχεδιασμού (draw). Πριν συμπεριλάβουμε αυτή την μέθοδο ένας πίνακας μετασχηματισμού θα πρέπει να έχει χρησιμοποιηθεί.



Αυτό σημαίνει ότι το αντικείμενο θα πρέπει να ευθυγραμμιστεί με το δείκτη, χωρίς να χρειαστούν άλλες ενέργειες. Η μέθοδος δεν θα καλεστεί εάν ο δείκτης(marker) που ανήκουν σε αυτό το αντικείμενο δεν είναι ορατός.

**Η κλάση ARRenderer** είναι υπεύθυνη για ότι σχετίζεται με το OpenGL. Εάν θέλουμε να ενώσουμε αντικείμενα επαυξημένης πραγματικότητας με μη θα πρέπει να παρέχουμε μια υλοποίηση μιας κλάσης που θα υλοποιεί την διαπάλη (interface) OpenGLRenderer . Υπάρχουν τρεις μέθοδοι που ορίζονται από αυτή την διεπαφή .

Η initGL καλείται μονό κατά την αρχικοποίηση της επιφάνειας του OpenGL.

Η setupEnv καλείται μόλις πριν σχεδιαστή το αντικείμενο επαυξημένης πραγματικότητας που θέλουμε να σχεδιάσουμε. Μπορεί να χρησιμοποιηθεί για να εκτελέσει εντολές OpenGL που επηρεάζουν όλα τα ARObjects, όπως αρχικοποίηση του φωτισμού. Στην μέθοδο σχεδιασμού μπορούμε να σχεδιάσουμε οποιοδήποτε μη τρισδιάστατο αντικείμενο επαυξημένης πραγματικότητας. Αυτό θα το καλέσουμε μια φορά για καφέ καρέ (frame).

Το AndARActivity επιπλέον μας παρέχει μια μέθοδο που μας επιτρέπει να περνούμε φωτογραφίες.

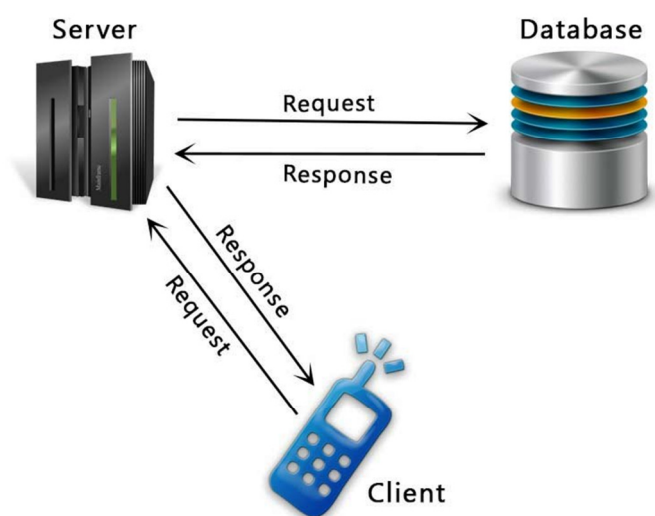
## 3: Ανάπτυξη της εφαρμογής

---

Η εφαρμογή που θα κατασκευάσουμε θα περιέχει μια διαδικτυακή υπηρεσία που θα μπορεί να εγγράφουν νέοι χρήστες ή να συνδεθούν οι ήδη εγγεγραμμένοι χρήστες και να μπορέσουν να έχουν πρόσβαση στην εφαρμογή μας. Επίσης η διαδικτυακή υπηρεσία αυτή θα παρέχει στον χρήστη την δυνατότητα να βλέπει νέα στοιχεία που θα παράγονται κατά την διάρκεια του παιχνιδιού.

Η εφαρμογή μας μπορεί να χωριστεί σε δυο τμήματα:

1. Το κομμάτι του πελάτη είναι αυτό που τρέχει στο κινητό μας τηλέφωνο και αλληλεπιδρά με τους αισθητήρες της συσκευής μας
2. Από την άλλη το κομμάτι του εξυπηρετητή είναι το κομμάτι που τρέχει σε ένα διακοσμητή και παρέχει την δυνατότητα του χρήστη να μπορεί να συνδεθεί ή να εγγραφεί στο παιχνίδι μας για να μπορεί να έχει πρόσβαση σε αυτό αλλά πέρα από αυτό, το κομμάτι αυτό μας επιτρέπει να κάνουμε το πρόγραμμα μας πιο δυναμικό παρέχοντας επιπλέον στοιχεία κατά την διάρκεια του παιχνιδιού που μπορούν να αλλάζουν, να τροποποιούνται και τελικά να εμφανίζονται στον τελικό χρήστη. Αυτή την στιγμή η εφαρμογή μας μπορεί να λαμβάνει πληροφορίες για το που βρίσκεται ένα στοιχείο.



Εικόνα 3-1: Σχεδιάγραμμα της Διαδικτυακής μας Υπηρεσίας.

### 3.1 Κατασκευή της διαδικτυακής υπηρεσίας μας.

Για την κατασκευή της διαδικτυακής μας υπηρεσίας θα χρησιμοποιήσουμε την PHP, MySQL, PHPMYADMIN και την SQLite. Εδώ θα δούμε την δημιουργία ενός απλού API που θα επιτρέπει στους χρηστές της εφαρμογής μας να γράφονται ή να συνδέονται στην εφαρμογή μας.

Πιο συγκεκριμένα η διαδικτυακή μας υπηρεσία θα εκτελεί τα παρακάτω:

- Δέχεται αιτήματα μέσω των μεθόδων GET/POST
- Αλληλεπιδρά με τις κλάσεις PHP για να πάρει ή να αποθήκευση δεδομένα από την βάση
- Εμφανίζει την έξοδο σε μορφή JSON
- Τέλος μας παρέχει την δυνατότητα να παράγουμε δυναμικά στοιχεία κατά την διάρκεια της εφαρμογής μας κατά την διάρκεια του παιχνιδιού.

### 3.2 Δημιουργία της Βάσης Δεδομένων με MySQL και του πίνακα

Ανοίγουμε είτε την κονσόλα της MySQL ή phpmyadmin και εκτελούμε το παρακάτω ερώτημα στην βάση για να δημιουργήσουμε την βάση μας αλλά και τον πίνακα που θα περιέχει τους χρηστές μας.

```
create table users(  
uid int(11) primary key auto_increment,  
unique_id varchar(23) not null unique,  
name varchar(50) not null,  
email varchar(100) not null unique,  
encrypted_password varchar(80) not null,  
salt varchar(10) not null,  
created_at datetime,  
updated_at datetime null  
);
```

### 3.3 Κατασκευάζοντας τα Αρχεία PHP

Εδώ θα δούμε την δομή των αρχείων php που θα χρησιμοποιήσουμε στην πτυχιακή.

### 3.3.1 Το αρχείο config.php

Στο αρχείο αυτό δηλώνουμε τις μεταβλητές για την σύνδεση της εφαρμογή μας με την βάση δεδομένων μας. Πιο συγκεκριμένα συμπληρώνουμε την τοποθεσία της βάσης μας, το όνομα χρήστη και των κωδικό του που μπορεί να έχει είσοδος στην υπηρεσία μας.

```
<?php
/**
 * Μεταβλητές Σύνδεσης για την Βάση Δεδομένων μας
 */
define("DB_HOST", "topothesia_basis");
define("DB_USER", "onoma_xristi");
define("DB_PASSWORD", "kodikos");
define("DB_DATABASE", "onoma-basis");
?>
```

### 3.3.2 Το αρχείο DB\_Connect.php

Αυτό το αρχείο χρησιμοποιείται για να συνδεθούμε ή να αποσυνδεθούμε από την βάση. Πιο συγκεκριμένα συμπεριλαμβάνει το αρχείο config και πραγματοποιεί την σύνδεση με την βάση μας και αναλαμβάνει και να κλείσουμε και την σύνδεση μας με αυτήν.

```
public function connect() {
require_once 'config.php';
// connecting to mysql
$con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
// selecting database
mysql_select_db(DB_DATABASE);
// return database handler
return $con;
}

public function close() {
mysql_close();
}
}
```

### 3.3.3 Το αρχείο DB\_Functions.php

Αυτό το αρχείο περιέχει τις λειτουργίες που θα εκτελεί η διαδικτυακή μας υπηρεσία, όπως η αποθήκευση χρηστών στην βάση, να πάρουμε τους χρήστες από την βάση. Επίσης εδώ μπορούμε να προσθέσουμε και άλλες μεθόδους όπως η ενημέρωση και διαγραφή χρηστών.

**user unique id** – Δημιουργούμε μοναδικές ταυτότητες (id) για τους χρήστες στην php, χρησιμοποιώντας την συνάρτηση uniqid("", true). Ένα παράδειγμα για το πώς θα εμφανίζεται η ταυτότητα του χρήστη είναι το παρακάτω : 4f074eca601fb8.88015924

**Κρυπτογραφημένο Κωδικός (Encrypted Password)** – Αυτός ο κωδικός αποθηκεύεται χρησιμοποιώντας την μέθοδο base64\_encode.

Στο αρχείο αυτό αποθηκεύουμε τους νέους χρήστες και επιστροφή των πληροφοριών τους.

```
public function storeUser($name, $email, $password) {  
    $uuid = uniqid("", true);  
    $hash = $this->hashSSHA($password);  
    $encrypted_password = $hash["encrypted"]; // κρυπτογράφηση κωδικού  
    $salt = $hash["salt"]; // salt  
    $result = mysql_query("INSERT INTO users(unique_id, name, email, encrypted_password,  
    salt, created_at) VALUES('$uuid', '$name', '$email', '$encrypted_password', '$salt',  
    NOW())");
```

#### Κρυπτογράφηση του Κωδικού μας

```
public function hashSSHA($password) {  
    $salt = sha1(rand());  
    $salt = substr($salt, 0, 10);  
    $encrypted = base64_encode(sha1($password . $salt, true) . $salt);  
    $hash = array("salt" => $salt, "encrypted" => $encrypted);  
    return $hash;  
}
```

#### Αποκρυπτογράφηση του κωδικού

```
public function checkhashSSHA($salt, $password) {  
    $hash = base64_encode(sha1($password . $salt, true) . $salt);  
    return $hash;
```

```
}
```

### 3.3. 4 Το αρχείο index.php

Αυτό το αρχείο αναλαμβάνει να δέχεται τα αιτήματα και να δημιουργεί απαντήσεις. Πιο συγκεκριμένα δέχεται όλα τα GET και POST αιτήματα για κάθε αίτημα μας θα μιλήσει με την βάση και θα μας δώσει την κατάλληλη απάντηση μέσω της μορφοποίησης JSON.

Πιο συγκεκριμένα, στο αρχείο αυτό παίρνουμε τα tag , συμπεριλαμβάνουμε τον χειρίστη για την βάση μας, κατασκευάζουμε την απάντηση που θα στείλουμε στον πελάτη μας αφού έχουμε προηγουμένως ελέγχει τον τύπο του tag, αλλά και εάν υπάρχει ο χρήστης καταχωρημένος στην βάση μας, εάν υπάρχει επιστρέφει τον κωδικό success = 1 εάν ο χρήστης δεν βρεθεί απάντηση με το μήνυμα error = 1

## 3.4 Τύποι Απαντήσεων (Responses) JSON

Το JSON είναι ένα πολύ ελαφρύ, δομημένο, εύκολο στο να αναλύσει και πολύ εύκολα αναγνώσιμο από τον άνθρωπο. Το JSON είναι η καλύτερη εναλλακτική λύση για την XML, για την εφαρμογή του Android app σας θα πρέπει να εναλλάσσονται δεδομένων με τον διακομιστή σας.

Παρακάτω μπορούμε να δούμε τα διαφορετικά τύπου JSON απαντήσεις που δημιουργείται για την κάθε κατάσταση που μπορεί να δημιουργηθεί.

### Απάντηση για Επιτυχή Εγγραφή– Success Code = 1 (Επιτυχή Αποθήκευση Χρηστη)

```
{  
  "tag": "register",  
  "success": 1,  
  "error": 0,  
  "uid": "432f321e3df49.06340261",  
  "user": {  
    "name": "Ted Gudis",  
    "email": "tedgudis@gmail.com",  
    "created_at": "2014-01-03 01:03:53",  
    "updated_at": null  
  }  
}
```

```
}
```

**Απάντηση για Σφάλμα στην Εγγραφή – Error Code = 1 (Σφάλμα στην Αποθήκευση )**

```
{  
"tag": "register",  
"success": 0,  
"error": 1,  
"error_msg": "Σφάλμα κατά την Εγγραφή "  
}
```

**Απάντηση για Σφάλμα στην Εγγραφή – Error Code = 2 (Ο χρήστης υπάρχει ήδη )**

```
{  
"tag": "register",  
"success": 0,  
"error": 2,  
"error_msg": "Ο Χρήστης Υπάρχει ήδη"  
}
```

**Απάντηση Επιτυχής Εισόδου – Success Code = 1 (Είσοδος του Χρηστη )**

```
{  
"tag": "login",  
"success": 1,  
"error": 0,  
"uid": "4f074eca601fb8.88015924",  
"user": {  
"name": "Ted Gudis",  
"email": "tedgudis@gmail.com",  
"created_at": "2014-01-03 01:03:53",  
"updated_at": null  
}  
}
```

**Απάντηση Σφάλμα Εισόδου – Error Code = 1 (Σφάλμα Εισόδου – Μη Σωστό όνομα  
χρηστη ή κωδικός )**

```
{  
"tag": "login",
```

```
"success": 0,  
"error": 1,  
"error_msg": "Σφάλμα στο email ή το password!"  
}
```

### 3.5 Δομή του Αρχείου Στοιχειά

Εδώ θα δούμε την δομή του αρχείου JSON που θα περιέχει τα στοιχεία που θα μας βοηθήσουν να βρούμε που είναι κρυμμένα τα αντικείμενα. Είναι ένα πολύ απλό αρχείο JSON το οποίο περιλαμβάνει όλες εκείνες τις πληροφορίες που είναι απαραίτητες για την εύρεση των κύβων.

Μπορούμε να δούμε τα δεδομένα μέσα από την [διεύθυνση http://treasurear.seriusb.com/stoixeia/](http://treasurear.seriusb.com/stoixeia/)

```
{  
  "kiboi": [  
    {  
      "id": "001",  
      "name": "Nero",  
      "stoixeio1": "mbortis",  
      "stoixeio1": "5aderfia",  
      "stoixeio1": "boyleytiko",  
    },  
  ]  
}
```

### 3.5 Ξεκινώντας το Πρόγραμμα μας στο Android

Μέχρι τώρα κατασκευάσαμε την διαδικτυακή μας υπηρεσία, την πλευρά του εξυπηρετητή. Τώρα θα κατασκευάσουμε την εφαρμογή που θα τρέχει στο κινητό μας τηλέφωνο. Η εφαρμογή μας θα περιέχει μια Activity για την είσοδο μας στην εφαρμογή, μια για την εγγραφή του χρήστη στην δικτυακή μας υπηρεσία. Στην συνέχεια θα έχουμε μια νέα Activity που θα είναι και το κεντρικό μενού στην εφαρμογή μας. Από εκεί ο χρήστης θα μπορεί είτε να ξεκινήσει το παιχνίδι είτε να δει οδηγίες για το πώς παίζεται είτε να δει οδηγίες για την δημιουργία της εφαρμογής ή τέλος μπορεί να αποσυνδεθεί από το παιχνίδι και να το κλείσει.



### 3.5.1 Η κλάση JSON Parser

Η κλάση αυτή θα επεξεργαστή τις απαντήσεις που θα έρχονται από την δικτυακή υπηρεσία μας. Πιο συγκεκριμένα κατασκευάζει το Αίτημα HTTP request για να λάβει τα δεδομένα και αναλαμβάνει την μετατροπή της Συμβολοσειράς σε αντικείμενο JSON object, για να μπορέσει να το επεξεργαστή στην συνέχεια.

```
try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
    Log.e("JSON", json);
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " + e.toString());
}
```

### 3.5.2 Η κλάση SQLite Database Handler

Στην εφαρμογή μας για την αποθήκευση των πληροφοριών των χρηστών θα χρησιμοποιήσουμε την βάση SQLite. Έτσι δημιουργούμε μια νέα κλάση με το όνομα DatabaseHandler.java. Αυτή η κλάση θα έχει λειτουργίες για τον χειρισμό των λειτουργιών της βάσης μας όπως η αποθήκευση και η ανάκληση των χρηστών.

```
public void addUser(String name, String email, String uid, String created_at) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, name); // Name
    values.put(KEY_EMAIL, email); // Email
```

```

values.put(KEY_UID, uid); // Email
values.put(KEY_CREATED_AT, created_at); // Created At

// Inserting Row
db.insert(TABLE_LOGIN, null, values);
db.close(); // Closing database connection
}

```

### 3.5.3 Η κλάση User Functions

Η κλάση αυτή θα χειριστεί όλες τις λειτουργίες του χρήστη όπως το:

```

loginUser()
registerUser()
getLoginStatus()
logoutUser().

```

Σε αυτή την κλάση όλες οι συναρτήσεις θα αλληλεπιδρούν με τις κλάσεις `JSONParser`, `DatabaseHandler`. Έχουμε την δυνατότητα να δοκιμάσουμε την λειτουργία της εφαρμογής μας τόσο τοπικά όσο και απευθείας στην διαδικτυακή μας τοποθεσία. Συνήθως το `localhost` τρέχει αν γράψουμε στον φυλλομετράτε μας την διεύθυνση <http://127.0.0.1> ή <http://localhost/>. Για να μπορούμε να έχουμε πρόσβαση μέσα από τις εικονικές μας συσκευές μέσω του AVD θα χρησιμοποιήσουμε την διεύθυνση `http://10.0.2.2/` αντί για το `http://localhost/`. Αλλιώς μπορούμε να χρησιμοποιήσουμε κατευθείαν την διαδικτυακή μας διεύθυνση `http://yoursite.com/api/`

Δήλωση μεταβλητών για την δόκιμη της υπηρεσίας μας τοπικά χρησιμοποιώντας το `xampp` μέσα από την διεύθυνση `http://10.0.2.2/` ή το <http://localhost/>

```

private static String loginURL = "http://10.0.2.2/treasurear";
private static String registerURL = "http://10.0.2.2/treasurear";
private static String login_tag = "login";
private static String register_tag = "register";

// constructor
public UserFunctions(){
    jsonParser = new JSONParser();
}

```

Η συνάρτηση που κατασκευάζει το αίτημα για την είσοδο στην υπηρεσία και παίρνει σαν παραμέτρους δυο μεταβλητές το email και το password.

```
public JSONObject loginUser(String email, String password){
// Building Parameters
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("tag", login_tag));
params.add(new BasicNameValuePair("email", email));
params.add(new BasicNameValuePair("password", password));
JSONObject json = jsonParser.getJSONFromUrl(loginURL, params);
// return json
// Log.e("JSON", json.toString());
return json;
}
```

Η Συνάρτηση αυτή καταχωρεί τον χρήστη και παίρνει σαν παράμετρο το όνομα, το ηλεκτρονικό ταχυδρομείο και τον κωδικό του.

```
public JSONObject registerUser(String name, String email, String password){
// Building Parameters
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("tag", register_tag));
params.add(new BasicNameValuePair("name", name));
params.add(new BasicNameValuePair("email", email));
params.add(new BasicNameValuePair("password", password));

// getting JSON Object
JSONObject json = jsonParser.getJSONFromUrl(registerURL, params);
// return json
return json;
}
```

### 3.5.4 Η κλάση RegisterActivity

Η κλάση αυτή είναι υπεύθυνη για την εγγραφή νέων χρηστών στην βάση δεδομένων της εφαρμογή μας. Υλοποιεί την γραφική διεπαφή της φόρμας μας και αφού δημιουργήσει το αρχείο json μας, το στέλνει στην βάση μας.

```
// Register Button Click event  
btnRegister.setOnClickListener(new View.OnClickListener() {  
public void onClick(View view) {  
String name = inputFullName.getText().toString();  
String email = inputEmail.getText().toString();  
String password = inputPassword.getText().toString();  
UserFunctions userFunction = new UserFunctions();  
JSONObject json = userFunction.registerUser(name, email, password);
```

### 3.5.5 MainMenuActivity

Η κεντρική κλάση της εφαρμογή μας, η οποία περιέχει ένα κεντρικό μενού μέσα από το οποίο μπορούμε να επιλέξουμε αν επιθυμούμε να ξεκινήσουμε το παιχνίδι μας, να δούμε τις οδηγίες του παιχνιδιού, να δούμε κάποιες πληροφορίες σχετικά με την δημιουργία του παιχνιδιού αυτού ή να αποσυνδεθούμε και να κλείσουμε την εφαρμογή μας.

```
@Override  
public void onClick(View v) {  
String str = ((TextView)v).getText().toString();  
if(str.equals(getResources().getString(R.string.startgame))) {  
//start the game activity  
((TextView)v).setClickable(false);  
((TextView)v).setText(R.string.starting);  
Intent intent = new Intent(MainMenuActivity.this, GameActivity.class);  
intent.setAction(Intent.ACTION_VIEW);  
startActivity(intent);  
} else if(str.equals(getResources().getString(R.string.instructions))) {  
//show the instructions activity  
startActivity(new Intent(MainMenuActivity.this, InstructionsActivity.class));
```

```

} else if(str.equals(getResources().getString(R.string.about))) {
//start the about activity
startActivity(new Intent(MainMenuActivity.this, AboutActivity.class));
} else if(str.equals(getResources().getString(R.string.exit))){
//go back to login activity
startActivity(new Intent(MainMenuActivity.this, LoginActivity.class));
}
}
}

```

### 3.5.6 LoginActivity

Η κλάση LoginActivity, μέσω της οποίας θα γίνεται η εισαγωγή του χρήστη στην εφαρμογή μας. Ο χρήστης θα εισάγει τον όνομα του και τον κωδικό του και θα γίνεται έλεγχος στην βάση εάν υπάρχει ο χρήστης, τότε μπαίνει κανονικά στην εφαρμογή μας, εάν δεν υπάρχει τον προτρέπει να εγγραφεί μέσω της οθόνης εγγραφής μας ή εάν έχει γράψει λάθος το όνομα του ή τον κωδικό τότε τον πληροφορεί και τον προτρέπει να δοκιμάσει ξανά.

```

// check for login response
try {
if (json.getString(KEY_SUCCESS) != null) {
loginErrorMsg.setText("");
String res = json.getString(KEY_SUCCESS);
if(Integer.parseInt(res) == 1){
// user successfully logged in
// Store user details in SQLite Database
DatabaseHandler db = new DatabaseHandler(getApplicationContext());
JSONObject json_user = json.getJSONObject("user");

// Launch Dashboard Screen
Intent dashboard = new Intent(getApplicationContext(), MainMenuActivity.class);

```

### 3.5.7 GameActivity

Η Game Activity είναι η κεντρική δραστηριότητα της εφαρμογής μας και εκτελεί και όλη την εργασία του παιχνιδιού μας. Πιο συγκεκριμένα αναλαμβάνει να ανοίξει την κάμερα της συσκευής μας, και να παρουσιάσει μια προεπισκόπηση του περιβάλλοντα χώρου μέσα από αυτήν, επίσης λαμβάνει πληροφορίες από τους αισθητήρες της συσκευής μας και εμφανίζει στην αριστερή πλευρά της οθόνης μας υπερκαλύπτοντας την εικόνα που βλέπουμε τις τιμές των αισθητήρων αυτών. Σε αυτήν την έκδοση μπορούμε να πάρουμε τιμές μόνο από τους αισθητήρες του επιταχυνσιόμετρο και τις πυξίδας αλλά μπορούμε να έχουμε και άλλους αισθητήρες π.χ φωτισμού ανάλογα με τις δυνατότητες της συσκευής μας.

Επίσης η κλάση αυτή χρησιμοποιώντας την βιβλιοθήκη AndAR μας δίνει την δυνατότητα να υλοποιήσουμε και χαρακτηριστικά επαυξημένης πραγματικότητας. Πιο συγκεκριμένα η κλάση παρακολουθεί την προεπισκόπηση της κάμερας μας, και αν εντόπιση μοτίβα που έχει καταχωρημένα στην «μνήμη» της, τότε εκεί αναλαμβάνει να εμφανίσει αντικείμενα, που είναι αποθηκευμένα στον φάκελο assets της εφαρμογής μας, επικαλύπτοντας έτσι τον φυσικό κόσμο με αντικείμενα που έχουν κατασκευαστεί από ηλεκτρονικό υπολογιστή.

Η κλάση GameActivity κληρονομεί από την βιβλιοθήκη AndAR την κλάση AndARActivity, η κλάση αυτή χειρίζεται ότι είναι σχετικό με την επαυξημένη πραγματικότητα, όπως το να ανοίγουμε την κάμερα, ανίχνευση του σημαδιού (marker) και υλοποιεί και την μέθοδο SurfaceHolder.Callback.

```
public class GameActivity extends AndARActivity implements SurfaceHolder.Callback  
{
```

Δήλωση μεταβλητών απαραίτητες για την ΕΠ, τα αντικείμενα που θα επικαλύπτονται και δήλωση του ARToolkit

```
CustomObject someObject;  
CustomObject3 someObject3;  
CustomObject2 someObject2;  
CustomObject4 someObject4;  
ARToolkit artoolkit;
```

#### **Η μέθοδος onCreate**

Στην μέθοδο onCreate() , αρχικά περνούμε το savedInstanceState. Μετά από αυτό δημιουργούμε μια αναφορά στην κλάση CustomRenderer που θα την κατασκευάσουμε αργότερα. Ορίζουμε τον non-AR renderer και τώρα ερχόμαστε στο κεντρικό μέρος της κλάσης μας. Δηλώνουμε το artoolkit για να μπορέσουμε να το χρησιμοποιήσουμε. Ορίζουμε τον touch event listener και την call back για την surface view και στην συνέχεια καταχωρούμε όλα τα σημάδια (markers) μας, τα αντικείμενα someObject1-4 ,είναι οι κλάσεις, που ορίζουν το τι θα εμφανίζεται πάνω από κάθε σημάδι. Τέλος η μέθοδο uncaughtException() χρησιμοποιείται για τον τερματισμό της εφαρμογή μας εάν μια κρίσιμη εξαίρεση συμβεί.

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    CustomRenderer renderer = new CustomRenderer();//optional, may be set to null
    super.setNonARRenderer(renderer);//or might be omitted
    prefs = getSharedPreferences("SimpleAR", 0);

    getSurfaceView().setOnTouchListener(new TouchEventHandler());
    getSurfaceView().getHolder().addCallback(this);
    artoolkit = super.getArtoolkit();

    someObject = new CustomObject
    ("earth", "marker_at16.patt", 80.0, new double[] {0,0});
    //("test", "marker_at16.patt", 80.0, new double[] {0,0},
    this.getApplicationContext(), "android");
    artoolkit.registerARObject(someObject);

    // Toast.makeText(CustomActivity.this, getResources().getText(
    R.string.item1),Toast.LENGTH_SHORT ).show();

    someObject2 = new CustomObject2
    ("test", "marker_at17.patt", 80.0, new double[] {0,0});
    artoolkit.registerARObject(someObject2);
    Toast.makeText(GameActivity.this,
    getResources().getText(R.string.item2),Toast.LENGTH_SHORT ).show();

    someObject3 = new CustomObject3
```

```

("test", "new3.patt", 80.0, new double[] {0,0});
artoolkit.registerARObject(someObject3);
Toast.makeText(GameActivity.this,
getResources().getText(R.string.item3),Toast.LENGTH_SHORT ).show();

someObject4 = new CustomObject4
("test", "new6.patt", 80.0, new double[] {0,0});
artoolkit.registerARObject(someObject4);
Toast.makeText(GameActivity.this,
getResources().getText(R.string.item4),Toast.LENGTH_SHORT ).show();

} catch (AndARException ex){ }

startPreview();
createHUD();
sensor();
addItem();
map = (Button) findViewById(R.id.shotButton);
map.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
//To change body of implemented methods use File | Settings | File Templates.
launchMap();
//camera.release();
}
});

@Override
public void uncaughtException(Thread thread, Throwable ex) {
Log.e("AndAR EXCEPTION", ex.getMessage());
finish();
}

```

Κατασκευή του HUD για την εφαρμογή μας, μέσω του οποίου θα εμφανίζονται τα μη επαυξημένα στοιχεία πάνω στην οθόνη της παιχνιδιού μας.

```

public void createHUD() {

```



```

// add layout
LayoutInflater controlInflater = LayoutInflater.from(getBaseContext());
View viewControl = controlInflater.inflate(R.layout.hud, null);
ViewGroup.LayoutParams layoutParamsControl = new ViewGroup.LayoutParams(
ViewGroup.LayoutParams.FILL_PARENT, ViewGroup.LayoutParams.FILL_PARENT);
this.addView(viewControl, layoutParamsControl);
}

```

Σε αυτό το κομμάτι κώδικα ορίζουμε και ρυθμίζουμε την λειτουργία των αισθητήρων της συσκευής μας αλλά και του δέκτη GPS και καταχωρούμε τους listeners για καθένα από αυτούς.

```

void sensor(){
locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 2000, 2,
locationListener);

sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
orientationSensor = Sensor.TYPE_ORIENTATION;
accelerometerSensor = Sensor.TYPE_ACCELEROMETER;
sensorManager.registerListener(sensorEventListener,
sensorManager.getDefaultSensor(orientationSensor),
SensorManager.SENSOR_DELAY_NORMAL);
sensorManager.registerListener(sensorEventListener,
sensorManager.getDefaultSensor(accelerometerSensor),
SensorManager.SENSOR_DELAY_NORMAL);

}

```

Εμφάνιση των τιμών των Αισθητήρων μας που παίρνουμε από το επιταχυνσιόμετρο και τον αισθητήρα προσανατολισμού της συσκευής μας στην οθόνη της κάμερας μας κατά την διάρκεια προεπισκόπησης.

```

final SensorEventListener sensorEventListener = new SensorEventListener() {
public void onSensorChanged(SensorEvent sensorEvent) {
if (sensorEvent.sensor.getType() == Sensor.TYPE_ORIENTATION)
{

```

```

headingAngle = sensorEvent.values[0];
pitchAngle = sensorEvent.values[1];
rollAngle = sensorEvent.values[2];

//Log.d(TAG, "Heading: " + String.valueOf(headingAngle));
// Log.d(TAG, "Pitch: " + String.valueOf(pitchAngle));
// Log.d(TAG, "Roll: " + String.valueOf(rollAngle));

headingValue.setText(String.valueOf(headingAngle));
pitchValue.setText(String.valueOf(pitchAngle));
rollValue.setText(String.valueOf(rollAngle));

```

Έναρξη της Map Activity , από εδώ μπορούμε να ξεκινήσουμε την έναρξη του χάρτη μας. Για να ξεκινήσει ο χάρτης μας, πρέπει η συσκευή μας να τοποθετηθεί σχεδόν παράλληλα στο έδαφος με μια απόκλιση 7%, το οποίο μπορούμε να το υπολογίσουμε μέσα από τις τιμές του αισθητήρα προσανατολισμού μας.

```

if (pitchAngle < 7 && pitchAngle > -7 && rollAngle < 7 && rollAngle > -7)
{
    launchMap();
    //camera.release();
}
}
else if (sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
{
    xAxis = sensorEvent.values[0];
    yAxis = sensorEvent.values[1];
    zAxis = sensorEvent.values[2];

    xAxisValue.setText(String.valueOf(xAxis));
    yAxisValue.setText(String.valueOf(yAxis));
    zAxisValue.setText(String.valueOf(zAxis));
}
}

```

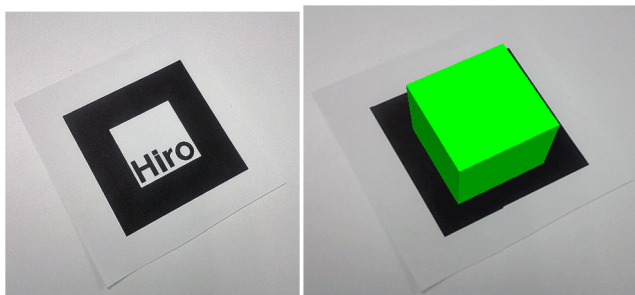
## Η Εσωτερική κλάση TakeAsyncScreenshot

Στην εσωτερική κλάση TakeAsyncScreenshot χρησιμοποιούμε μια κλήση της βιβλιοθήκης AndAR για να μπορούμε να περνούμε φωτογραφίες.

```
@Override
protected Void doInBackground(Void... params) {
    Bitmap bm = takeScreenshot();
    FileOutputStream fos;
    try {
        fos = new FileOutputStream("/sdcard/AndARScreenshot"+new Date().getTime()+".png");
        bm.compress(Bitmap.CompressFormat.PNG, 100, fos);
        fos.flush();
        fos.close();
    } catch (FileNotFoundException e) {
        errorMsg = e.getMessage();
        e.printStackTrace();
    } catch (IOException e) {
        errorMsg = e.getMessage();
        e.printStackTrace();
    }
    return null;
}
}
```

## Τα Επικαλυπτόμενα Αντικείμενα (CustomObject Overlays)

Τα αντικείμενα που θα εμφανίσουμε στην εφαρμογή μας είναι ουσιαστικά τρισδιάστατα αντικείμενα, στην προκειμένη περίπτωση κουτιά σε τέσσερα διαφορετικά χρώματα. Που έχουν την δυνατότητα να περιστρέφονται ανάλογα με την γωνιά που την βλέπουμε μέσα από το κινητό μας.



### Εικόνα 3-2 Επικαλυπτόμενα Αντικείμενα.

Αρχίζουμε ρυθμίζοντας τις διαφορετικούς τύπου φωτισμού για το κουτί και δημιουργώντας FloatBuffers για αυτούς μέσω του κατασκευαστή. Στην συνέχεια ένα απλό κουτί κατευθείαν από το AndAR. Στην μέθοδο draw() σχεδιάζουμε ότι άλλο έχουμε για να σχεδιάσουμε το οποίο θα εμφανιστεί κατευθείαν πάνω στο σημάδι (marker) μας.

```
public CustomObject1(String name, String patternName,
double markerWidth, double[] markerCenter, float[] customColor) {
super(name, patternName, markerWidth, markerCenter);
float mat_flash_shinyf[] = {50.0f};
mat_ambient = GraphicsUtil.makeFloatBuffer(customColor);
mat_flash = GraphicsUtil.makeFloatBuffer(customColor);
mat_flash_shiny =
GraphicsUtil.makeFloatBuffer(mat_flash_shinyf);
mat_diffuse = GraphicsUtil.makeFloatBuffer(customColor);
}

@Override
public final void draw(GL10 gl) {
super.draw(gl);
gl.glMaterialfv(GL10.GL_FRONT_AND_BACK,
GL10.GL_SPECULAR,mat_flash);
gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_SHININESS,
mat_flash_shiny);
gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_DIFFUSE,
mat_diffuse);
gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_AMBIENT,
mat_ambient);
gl.glColor4f(0, 1.0f, 0, 1.0f);
gl.glTranslatef( 0.0f, 0.0f, 12.5f );
box.draw(gl);
}
```

Οι άλλες τρεις κλάσεις CustomObject είναι ακριβώς ίδιες με την CustomObject1, Αλλά αλλάζουμε μόνο λίγο το χρώμα των κουτιών που θα εμφανίζονται.

### 3.5.7 Η Κλάση CustomRenderer

Η Κλάση CustomRenderer μας επιτρέπει να εργαζόμαστε με μη-επαυξημένα στοιχεία , όπως την ρύθμιση του περιβάλλοντος OpenGL. Στην δήλωση μεταβλητών, δηλώνουμε τους διαφορετικούς τύπους φωτισμού και δημιουργούμε αντικείμενα FloatBuffers από αυτά. Η setupEnv() καλείται πριν εμφανίσουμε κάποια από τα κουτιά. Εγκαθιστά τον φωτισμό και τα αλλά στοιχεία που έχουν να κάνουν με το OpenGL. Η μέθοδο initGL() καλείται μια φορά κατά την δημιουργία του Surface.

```
public class CustomRenderer implements OpenGLRenderer {  
    public final void setupEnv(GL10 gl) {  
        gl.glEnable(GL10.GL_LIGHTING);  
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_AMBIENT, ambientLightBuffer1);  
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_DIFFUSE, diffuseLightBuffer1);  
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_SPECULAR, specularLightBuffer1);  
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_POSITION, lightPositionBuffer1);  
        gl.glEnable(GL10.GL_LIGHT1);  
        gl.glDisableClientState(GL10.GL_TEXTURE_COORD_ARRAY);  
        gl.glDisable(GL10.GL_TEXTURE_2D);  
        initGL(gl);  
    }  
}
```

### 3.5.8 InstructionsActivity

Εδώ είναι μια πολύ απλή κλάση. Η κλάση αυτή περιέχει μια Activity που εμφανίζει τις οδηγίες που εμφανίζονται ένα αρχείο HTML που βρίσκεται στον κατάλογο /assets/help σε ένα WebView.

```
WebSettings webSettings = mWebView.getSettings();  
webSettings.setDefaultTextEncodingName("utf-8");  
webSettings.setSupportZoom(true);  
webSettings.setBuiltInZoomControls(true);  
WebChromeClient client = new WebChromeClient();  
mWebView.setWebChromeClient(client);  
mWebView.loadUrl("file:///android_asset/help/"+getResources().getString(R.string.help_file)  
);
```

```
}  
}
```

### 3.5.8 AboutActivity

Είναι η πιο μικρή Activity της εφαρμογή μας, που εμφανίζει ένα απλός μήνυμα για τον σκοπό κατασκευή της εφαρμογή και τον δημιουργό της.

### 3.5.9 MapActivity

Η κλάση MapActivity μας επιτρέπει να εμφανίζεται ένας χάρτης στην εφαρμογή μας και εμφανίζει σημεία ενδιαφέροντος στην περιοχή που βρισκόμαστε.

```
ArrayList<OverlayItem> anotherOverlayItemArray;  
Context context;  
MyLocationOverlay myLocationOverlay = null;  
  
/** Called when the activity is first created. */  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.map);  
    prefs = getSharedPreferences("MapView", 0);  
    mapView = (MapView)findViewById(R.id.mapview);  
    mapView.setBuiltInZoomControls(true);  
    mapController = mapView.getController();  
    mapController.setZoom(10);
```

Δημιουργία σταθερών σημείων πάνω στον χαρτί μας που θα μας βοηθήσουν στον εντοπισμό των κύβων και στην ουσία θα λειτουργήσουν σαν φάροι για τον παίκτη.

```
contactList = new ArrayList<HashMap<String, String>>();  
//--- Create Another Overlay for multi marker  
anotherOverlayItemArray = new ArrayList<OverlayItem>();  
// anotherOverlayItemArray.add(new OverlayItem( "0, 0", "0, 0", new GeoPoint(0, 0)));  
anotherOverlayItemArray.add(new OverlayItem(  
    "Stoixeio1", "Roloi", new GeoPoint(37.564621, 22.796373)));
```

```

anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio2", "5Aderfia", new GeoPoint(37.565339, 22.793171)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio6", "Bouli", new GeoPoint(37.565624, 22.796288)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio7", "Nisi", new GeoPoint(37.569698, 22.79054)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio9", "Filaki", new GeoPoint(37.562324, 22.803175)));

```

```

ItemizedIconOverlay<OverlayItem> anotherItemizedIconOverlay
= new ItemizedIconOverlay<OverlayItem>(
    this, anotherOverlayItemArray, myOnItemGestureListener);
mapView.getOverlays().add(anotherItemizedIconOverlay);

```

Προσθήκη της Scale Bar στον Χάρτη μας

```

ScaleBarOverlay myScaleBarOverlay = new ScaleBarOverlay(this);
mapView.getOverlays().add(myScaleBarOverlay);

```

Προσθήκη της τοποθεσίας του χρηστη (MyLocationOverlay) πάνω στον Χάρτη μας.

```

myLocationOverlay = new MyLocationOverlay(this, mapView);
mapView.getOverlays().add(myLocationOverlay);
mapView.postInvalidate();

```

```

myLocationOverlay.runOnFirstFix(new Runnable() {
    @Override
    public void run() {
        mapView.getController().animateTo(myLocationOverlay.getMyLocation());
    }
});

```

### 3.5.9 Η κλάση StoixeiaActivity

Η κλάση StoixeiaActivity αναλαμβάνει να εμφάνιση τα στοιχεία που θα είναι διαθέσιμα στον χρηστη κατά την διάρκεια του παιχνιδιού. Τα αρχεία αυτά είναι αποθηκευμένα σε μορφή json και εμφανίζονται στην εφαρμογή μας μέσα από μια ListActivity.

```

JSONObject jsonObj = new JSONObject(jsonStr);

// Getting JSON Array node
= jsonObj.getJSONArray(TAG_STOIXEIA);
// looping through All Stoixeia
for (int i = 0; i < stoixeia.length(); i++) {
    JSONObject c = stoixeia.getJSONObject(i);
    String id = c.getString(TAG_ID);
    String name = c.getString(TAG_NAME);
    String lat = c.getString(TAG_LAT);
    String lon = c.getString(TAG_LON);
    String alt = c.getString(TAG_ALT);

    // Stoixeia node is JSON Object
    JSONObject phone = c.getJSONObject(TAG_STOIXEIO);
    String stoixeio1 = phone.getString(TAG_STOIXEIO_1);
    String stoixeio2 = phone.getString(TAG_STOIXEIO_2);
    String stoixeio3 = phone.getString(TAG_STOIXEIO_3);

    // tmp hashmap for single contact
    HashMap<String, String> stoixeia = new HashMap<String, String>();

    // adding each child node to HashMap key => value
    stoixeia.put(TAG_ID, id);
    stoixeia.put(TAG_NAME, name);
    stoixeia.put(TAG_STOIXEIO_1, stoixeio1);
    stoixeia.put(TAG_STOIXEIO_2, stoixeio2);
    stoixeia.put(TAG_STOIXEIO_3, stoixeio3);
    // adding contact to contact list
    stoixeiaList.add(stoixeia);
}

```

### 3.8 Το Αρχείο AndroidManifest.xml

Τέλος το αρχείο AndroidManifest.xml της εφαρμογή μας στο οποίο θα εισάγουμε τις απαραίτητες άδειες για την εφαρμογή μας και εκεί που θα δηλώσουμε όλες τις Δραστηριότητες (Activity) της εφαρμογή μας.



```
<!-- Allow to connect with internet -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<supports-screens android:smallScreens="true"
android:normalScreens="true"
android:largeScreens="true"
android:anyDensity="true" />
</manifest>
```

# 4: Εργαλεία για την Ανάπτυξη της Εφαρμογής

---

Σε αυτό το κεφάλαιο θα δούμε το λογισμικό και τις συσκευές που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας και στο τέλος περιγράφουμε τα βασικά στοιχεία της εφαρμογής μας. Το πρόγραμμα μας θα κάνει χρήση και των δυο «κύριων» τεχνικών της επαυξημένης πραγματικότητας τόσο μέσω της χρήση σημαδιών (marker) όσο και χωρίς αυτά, τα οποία περιγράψαμε σε προηγούμενο κεφάλαιο.

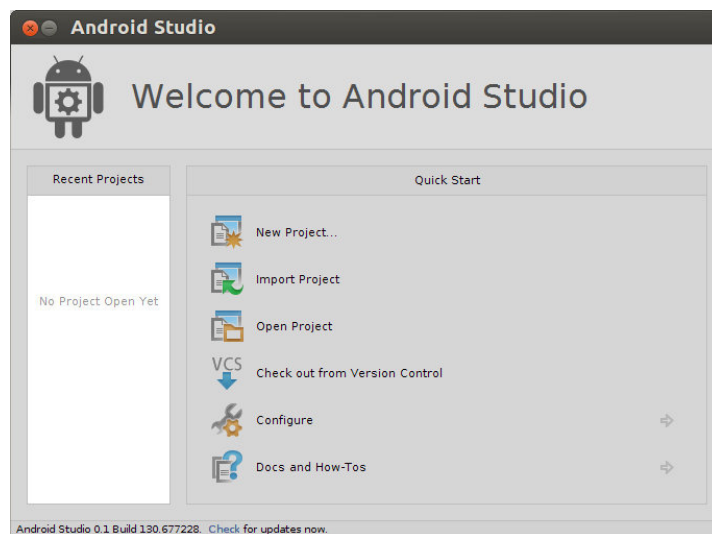
Το παιχνίδι που θα κατασκευάσουμε είναι ένα παιχνίδι κρυμμένου θησαυρού που θα κάνει χρήση των αισθητήρων, του δέκτη GPS και τις δυνατότητες δικτύωσης της συσκευής μας, της βιβλιοθήκης OSM για την υλοποίηση του χάρτη μας, που εμφανίζει τα σημεία που υπάρχουν στοιχεία για το θησαυρό μας και τέλος την βιβλιοθήκη AndAR για την υλοποίηση του τμήματος της επαυξημένης πραγματικότητας, η όποια είναι αυτή που θα μας βοηθήσει να διαβάσουμε τα σημάδια που είναι κρυμμένα έτσι ώστε να μπορέσουμε να βρούμε τους κύβους μας και να μπορέσουμε να τα «αιχμαλωτίσουμε».

## 4.1 Λογισμικό

Η εφαρμογή μας μπορεί να παρουσιάσει μοντέλα που είναι αποθηκευμένα στην εφαρμογή μας και μας δίνει την δυνατότητα να αλλάζουμε το μέγεθος, να τα περιστρέψουμε και να αλλάζουν ανάλογα με την κίνηση της συσκευής μας. Επίσης χρησιμοποιήσαμε τους Χάρτες της OpenStreetMap για να υλοποιήσουμε ένα χάρτη που μας βοηθού στο να μας εμφανίζει τα σημεία που βρίσκονται τα στοιχεία μας.

### 4.1.1 Android Studio

Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε το Android Studio. Το οποίο είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών για την πλατφόρμα του Android. Ανακοινώθηκε στις 16 Μαΐου το 2013 και από τον Ιούλιο του 2013 είναι διαθέσιμο για τους χρήστες, δωρεάν. Βασίζεται σε ένα άλλο πολύ δημοφιλή λογισμικό το IntelliJ IDEA, το έχει σχεδιαστεί και αναπτυχτεί αποκλειστικά για την ανάπτυξη εφαρμογών για το Android και είναι διαθέσιμο τόσο για τα Windows, A OS X και Linux.



Εικόνα 4-1: Android Studio.

### 4.1.2 OpenStreetMap

Το έργο OpenStreetMap (OSM) φιλοδοξεί να φτιάξει ένα ελεύθερο χάρτη του πλανήτη. Όπως η Wikipedia είναι η μεγαλύτερη συμμετοχική εγκυκλοπαίδεια, έτσι το OpenStreetMap θέλει να γίνει ο μεγαλύτερος συμμετοχικός χάρτης. Τα δεδομένα που εισάγονται στο OpenStreetMap διατίθενται με την άδεια Creative Commons Attribution-Share Alike.

Γιατί χρειαζόμαστε το OpenStreetMap αφού το Google Maps είναι «δωρεάν» και σχετικά ολοκληρωμένο;

- Το ζήτημα είναι ότι το Google Maps, για το περισσότερο κόσμο και την Ελλάδα, χρησιμοποιεί δορυφορικές εικόνες και δεδομένα χαρτογράφησης που δεν είναι διαθέσιμα με ελεύθερες άδειες. Δείτε τους περιορισμούς χρήσης των χαρτών του Google Maps, ενότητα Restrictions on Use.
- Επίσης η Google Αρχίζει να χρεώνει την χρήση της υπηρεσίας Google Maps API από την 1 Ιανουαρίου του 2012 όταν ξεπεράσουμε τα 25.000 χτυπήματα στον χάρτη μας.

Daily Map Loads	Cost for JavaScript v3, Staticmaps, Streetview (per day)	Cost for JavaScript v2 (per day)
5,000	\$0	\$0
15,000	\$0	\$0
25,000	\$0	\$0
35,000	\$5	\$10
45,000	\$10	\$20
75,000	\$25	\$50
100,000	\$37.50	\$75

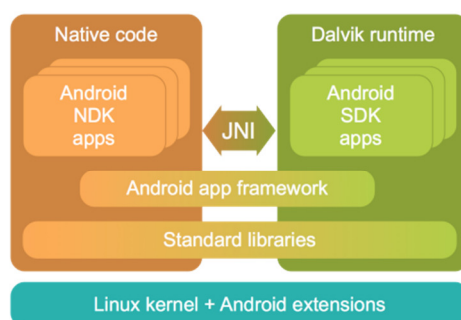
Πίνακας 4-1: Χρέωση Στο Google Map API

Η `OpenStreetMapView` είναι (σχεδόν) πλήρη / ελεύθερη αντικατάσταση της κλάσης `MapView` του Android.

Κάποιοι Περιορισμοί που έχει η χρήση του `OpenStreetMapView` η εφαρμογή μας θα πρέπει να υποστηρίζει τουλάχιστον το Android 3 ή επίπεδο API level 1.5A και να συμπεριλάβουμε τα αρχεία `osmdroid-android-x.xx.jar` και `slf4j-android-1.5.8.jar` στην εφαρμογή μας.

### 4.1.3 Το Android NDK (Native Development Kit)

Παρόλο που η ανάπτυξη των εφαρμογών για το λειτουργικό Android γίνεται σε Java. Ωστόσο υπάρχουν περιπτώσεις κατά τις οποίες χρειάζεται να γράψουμε κώδικα σε C/C++ και να εκτελεστή μέσω του JNI. Όταν μια εφαρμογή χρησιμοποιεί το Android NDK, το ένα τμήμα (κυρίως η διεπαφή χρηστή και ένα μέρος της λογικής της εφαρμογής) συνεχίζει να εκτελείται από τον Dalvin Machine και το άλλο τμήμα της εκτελείτε απευθείας στον επεξεργαστή της συσκευής μας.



Εικόνα 4-2: Αρχιτεκτονική του Android NDK

Τα τμήματα του κώδικα που εκτελούνται κατευθείαν στον επεξεργαστή μεταφράζονται σε βιβλιοθήκες και καλούνται από τα τμήματα του κώδικα που διερμηνεύονται από τον Dalvik

μέσω του JNI (Java Native Interface) (Εικόνα 10). Η βιβλιοθήκη μετά την μετάφρασή της πρέπει να συμπεριληφθεί στο .apk αρχείο της εφαρμογής και να φορτωθεί ρητά πριν χρησιμοποιηθεί. Εδώ αξίζει να παρατηρήσει κανείς ένα πιθανό πρόβλημα ασφάλειας. Ο κώδικας που εκτελείται στον επεξεργαστή δεν περνά από κανέναν από τους περιορισμούς για το ποιες λειτουργικότητες του συστήματος μπορεί να χρησιμοποιήσει.

Ο κυριότερος λόγος για να χρησιμοποιήσει κανείς το Android NDK είναι για να επιτύχει καλύτερες επιδόσεις σε εφαρμογές που απαιτούν μεγάλο όγκο υπολογισμών (εξομοιώσεις κ.λπ) ή εφαρμογές που κάνουν βαριά χρήση γραφικών (παιχνίδια κ.λπ).

#### 4.1.4 OpenGL ES

OpenGL είναι μια πιο ελαφριά (lightweight) έκδοση της OpenGL. Είναι μια low-level Διασύνδεση Προγραμματισμού Εφαρμογών API (application programming interface) που χρησιμοποιεί υποσύνολο των εφαρμογών της OpenGL. Παρέχει διασύνδεση μεταξύ λογισμικού και υλικού (software and hardware).

Αυτό το API είναι το πιο διαδεδομένο στα κινητά λόγω του ότι βασίζεται στο διαδεδομένο OpenGL και δεν χρειάζεται μεγάλη προσπάθεια για την μετάβαση σε αυτή την επιφάνεια εργασίας, το μικρό του μέγεθος βοηθά στην χαμηλή κατανάλωση ενέργειας και λιγότερος χώρος αποθήκευσης.

Το μεγαλύτερο πλεονέκτημα που προσφέρει για τους προγραμματιστές είναι ότι οποιοσδήποτε μπορεί να κατεβάσει και να εφαρμόσει αφού είναι δωρεάν και μπορεί να χρησιμοποιηθεί σε όλες τις πλατφόρμες. Ο προγραμματιστής μπορεί να συγκεντρωθεί στην εφαρμογή καθ' αυτή παρά στις λεπτομέρειες του κώδικα για να δουλέψει για κάθε πλατφόρμα. Επίσης λόγω τις δημοτικότητας έχει αρκετά βιβλία, παραδείγματα και εγχειρίδια.

#### 4.1.5 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό

περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

#### 4.1.6 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr[1], το YouTube[2], η Wikipedia[3], το Google[4], το Facebook[5] και το Twitter[6].

#### 4.1.7 phpmyadmin

phpMyAdmin είναι ένα δωρεάν λογισμικό εργαλείο γραμμένο σε PHP, που προορίζεται για να χειριστεί τη διοίκηση της MySQL στο Web. phpMyAdmin υποστηρίζει ένα ευρύ φάσμα δραστηριοτήτων στο MySQL. Συχνά χρησιμοποιούμενες λειτουργίες (διαχείριση βάσεων δεδομένων, πίνακες, στήλες, τις σχέσεις, ευρετήρια, χρήστες, δικαιώματα, κ.λπ.) μπορεί να πραγματοποιηθεί μέσω της διεπαφής χρήστη, ενώ εξακολουθείτε να έχετε τη δυνατότητα να εκτελέσει άμεσα κάποια δήλωση SQL.

#### 4.1.8 JSON

Το **JSON** (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική μορφή για την ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

## 4.2 Υλικό

Η εφαρμογή είναι συμβατή με συσκευές που τρέχουν Android 2.1 και όλες τις υπόλοιπες εκδόσεις που υλοποιήθηκαν μετά. Οι περισσότερες εφαρμογές επαυξημένης πραγματικότητας απαιτούν την διαδικασία της επεξεργασίας της εικόνας κάτι που είναι πολύ ακριβό σε υπολογιστικού πόρους. Επιπλέον υπάρχει ένας περιορισμός στην Μίμη που μπορεί να χρησιμοποιήσει κάθε διεργασία(Process), το οποίο είναι 24MB, άλλα σε κάποιες παλιότερες συσκευές, το όριο ήταν ακόμα μικρότερο στα 16MB. Η μνήμη που χρησιμοποιείται από τις εικόνες (BITMAPS) συμπεριλαμβάνεται σε αυτή την ποσότητα.

Κατά την ανάπτυξη μιας εφαρμογής επαυξημένης πραγματικότητας είναι πολύ εύκολο να υπερβείς αυτό το όριο και να έχεις μια εξαίρεση (exception) OutOfMemoryError. Αυτό το πρόβλημα τελικά λύνεται εάν δεσμεύσουμε μνήμη μέσω native κώδικα χρησιμοποιώντας το NDK. Ωστόσο, αυτά τα προβλήματα σιγά θα ξαλεθούν με τις νέες εκδόσεις του λειτουργικού συστήματος Android.

Το κινητό τηλέφωνο που χρησιμοποιήσαμε για την ανάπτυξη αυτής της εφαρμογής ήταν το HTC Hero. Στην εικόνα μπορούμε να το δούμε. Το κινητό Αυτό έχει ένα επεξεργαστή 528 MHz ARM 11, μνήμη RAM 288MB 512MB ROM, μια κάμερα 5MP, έναν δεκτή GPS, Wi-Fi 802.11 b/g, μια ψηφιακή πυξίδα και έναν επιταχυνσιόμετρο.



Εικόνα 4-3: HTC Hero

[http://www.gsmarena.com/htc\\_hero-review-382.php](http://www.gsmarena.com/htc_hero-review-382.php)

### 4.3 Επισκόπηση της Εφαρμογής μας

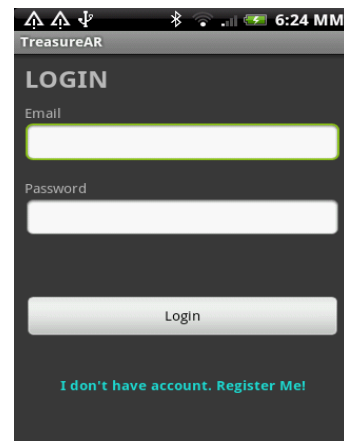
Εδώ θα κάνουμε μια επισκόπηση στην εφαρμογή μας δούμε την κατασκευή της εφαρμογής μας

#### LoginActivity

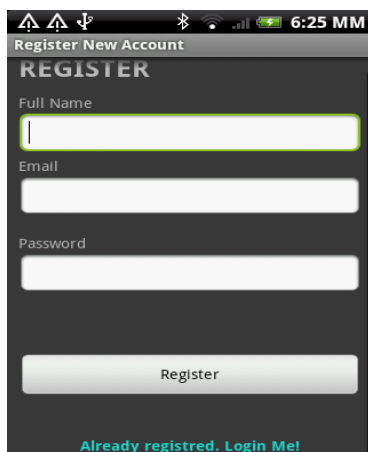
Αυτή είναι η πρώτη οθόνη που εμφανίζεται στον χρήστη όταν ξεκινήσει η εφαρμογή μας. Εμφανίζει ένα μενού που ζητάμε από τον χρήστη να εισάγει το email και τον κωδικό του για να μπορέσει να μπει στην εφαρμογή μας.

Εάν δεν είναι ήδη γραμμένος στο κάτω μέρος υπάρχει η επιλογή να κάνει εγγραφή.

Η LoginActivity στέλνει ένα αίτημα στην διαδικτυακή υπηρεσία μας και πιο συγκεκριμένα στην βάση μας και ελέγχει εάν υπάρχει χρήστης με αυτά τα στοιχεία εάν υπάρχει, του εμφανίζει την κεντρική οθόνη της εφαρμογής μας αλλιώς εμφανίζει μήνυμα ότι υπάρχει λάθος στα στοιχεία.



Εικόνα 4-4 : LoginActivity



#### Register

Αυτή η οθόνη ενεργοποιείται για να γράψει νέους χρηστές στην εφαρμογή μας. Και αφού γίνει η εγγραφή επιστρέφει πίσω στην οθόνη εισόδου από την οποία μπορεί ο χρήστης να μπει στην εφαρμογή μας.



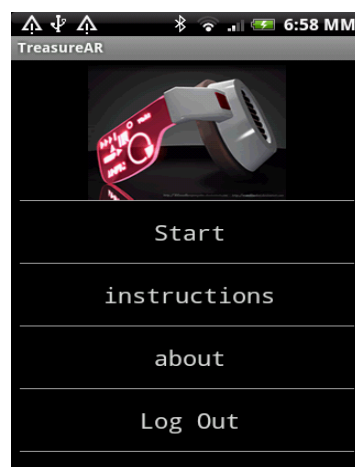
Εικόνα 4-5 : RegisterActivity

## MenuActivity

Είναι η πρώτη Οθόνη που εμφανίζεται με την έναρξη της Εφαρμογή μας, που περιέχει μια φωτογραφία και ένα μενού με τρεις Επιλογές. Όπως φαίνεται και στην εικόνα :

Ο σκοπός αυτής της Activity είναι η δυνατότητα επιλογή του παίκτη ανάμεσα στις παρακάτω επιλογές

- Start
- Instructions
- About
- Logout

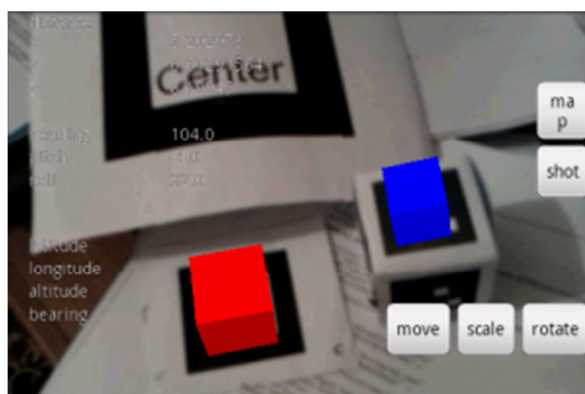


Εικόνα 4-6: Main Activity

## Game Activity

Η Επομένη Δραστηριότητα είναι η Κεντρική Δραστηριότητα της εφαρμογή μας. Η οποία ανοίγει την κάμερα της συσκευής, Επαυξάνει την εικόνα με επιπλέον αντικείμενα εάν ανιχνεύσει σημάδια μέσω της κάμερας, και τα εμφανίζει στην προεπισκόπηση της εικόνας που περνούμε από την κάμερα. Επίσης χρησιμοποιεί τους ενσωματωμένων αισθητήρων που διαθέτει η συσκευή μας για να εμφανίσει πάνω

στην οθόνη μας πληροφορίες σχετικά με αυτούς και εάν η συσκευή μας βρεθεί παράλληλα με το έδαφος, ενεργοποιεί την επομένη δραστηριότητα που θα δούμε η οποία ανοίγει τον χάρτη μας.

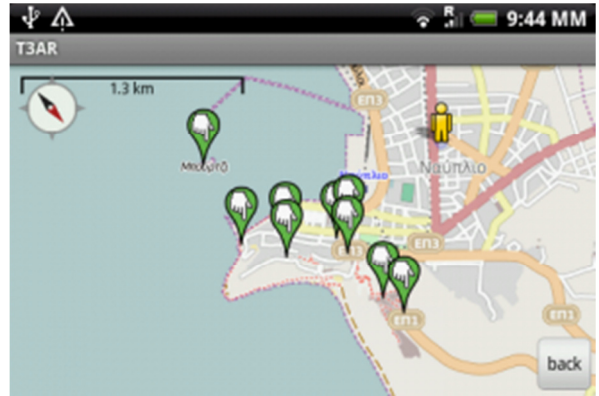


Εικόνα 4-7: GameActivity

## Map Activity

Ενεργοποιείται η Δραστηριότητα που ανοίγει τον χάρτη μας. Ο Χάρτης μας εκτελεί δυο σημαντικές εργασίες

- Εντοπίζει την θέση μας μέσω του δεκτή GPS και την εμφανίζει στον χάρτη μας.
- Εμφανίζει τις τοποθεσίες που υπάρχουν σημεία ενδιαφέροντος, και με την χρήση των στοιχείων που λαμβάνουμε από την διαδικτυακή μας υπηρεσία μπορούμε να οδηγηθούμε στους κρυμμένους κύβους.



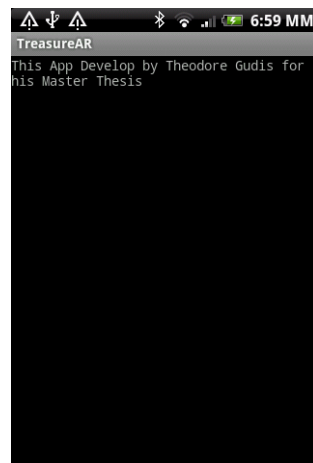
Εικόνα 4-8: MapActivity

## InstunctionActivity

Αυτή η Activity ανοίγει μια σελίδα html, μέσω του ενσωματωμένου φυλλομετράτε του android που περιέχει οδηγίες για τον τρόπο που παίζεται το παιχνίδι.

## AboutActivity

Είναι η πιο απλή Activity της εφαρμογή μας και δεν κάνει κάτι ιδιαίτερο περάν από την εμφάνιση ενός κειμένου με πληροφορίες σχετικά με τον σκοπό δημιουργεί της εφαρμογή μας και τον δημιουργό της.



Εικόνα 4-10: AboutActivity

# 5. Συμπεράσματα

## 5.1 Γενικά Συμπεράσματα

Σε αυτή την πτυχιακή εργασία αναλύσαμε την επαυξημένη πραγματικότητα στα κινητά τηλεφωνα που τρέχουν Android. Πιο συγκεκριμένα είδαμε και τα δυο ήδη επαυξημένης πραγματικότητας, τόσο των εφαρμογών που βασίζονται σε σημάδια (marker) όσο και σε εφαρμογές που δεν βασίζονται στα σημάδια και αναπτύξαμε μια εφαρμογή που κάνει χρήση και των δυο, ένα παιχνίδι κρυμμένου θησαυρού που λαμβάνει χώρα στην πόλη του Ναύπλιου.

Το παιχνίδι επιτρέπει στους χρήστες αφού συνδεθούν σε αυτό, μέσα από μια διαδικτυακή υπηρεσία, που έχει κατασκευαστεί για αυτήν την εφαρμογή, μπορούν να γράφουν πολλή απλά εάν δεν είναι ήδη γραμμένη, να λάβουν πληροφορίες για το που βρίσκονται στοιχειά μέσα στην πόλη του Ναύπλιου, και μέσω του AndAR να μπορέσουν να βρουν τους κρυμμένους κύβους, που είναι «παγιδευμένα» μέσα σε εικόνες και να τα ελευθερώσουν.

Η Επαυξημένη Πραγματικότητα είναι ένα «νέο» πεδίο που έχει ακόμη πολλά εμπόδια να υπερνικήσει, όπως και κάθε νέα τεχνολογία

- Οι περισσότερες από αυτές τις εφαρμογές που βασίζονται σε εφαρμογές που στηρίζονται στην τοποθεσία ( Location based applications ) και κυρίως λειτουργούν με τους δέκτες του GPS της κινητής συσκευής μας. Άλλα οι συσκευές GPS από μόνες τους έχουν αρκετούς περιορισμούς και κυρίως οι υπάρξει πολλών κατασκευαστών με διαφορετική ακρίβεια στις μετρήσεις τους, συνήθως μεταξύ 10 μέχρι 30 μέτρων, και την δυνατότητα να δουλεύουν μόνο έξω είναι τα σημαντικότερα.
- Άλλα περάν των περιορισμών του GPS, οι απαιτήσεις των κινητών δικτύων, η περιορισμένη διάρκεια ισχύ της μπαταρίας, όπως και ζήτημα που έχουν να κάνουν με την χρήση της κάμερας της συσκευής, η οποία δεσμεύει αρκετή μνήμη μπορούν να επηρεάσουν παρά πολλή την ευχρηστία των εφαρμογών αυτού του είδους

- Οι μονάδες απεικόνισης και οι συσκευές εντοπισμού πρέπει να γίνουν πιο ελαφριές, περισσότερες ακριβείς και ταχύτερες με λιγότερο υπολογιστικό κόστος και κατανάλωση ενέργειας.
- Άλλη μια σημαντική παράμετρο που θα πρέπει να σκεφτούμε είναι η δυνατότητα να συνεχίσει να δουλεύει η συσκευή μας ακόμα και εάν χάσουμε την επικοινωνία προσωρινά ή μόνιμα με το σύστημα παρακολουθήσεις μας ή την σύνδεση μας στο διαδίκτυο είτε στο δίκτυο της κινητής τηλεφωνίας.
- Αλληλεπίδραση με τον Χρηστή: Απαιτούνται καλύτεροι τρόποι παρουσίασης των δεδομένων και αλληλεπίδρασης του χρήστη με αυτά. Η χρήση του συστήματος θα πρέπει να είναι «διάφανης» για τον χρήστη, να μην προβληματίζεται για το πώς θα εκτελεστεί μια εργασία, αλλά το ίδιο το σύστημα να τον βοηθά να επικεντρωθεί στην εργασία.
- Άλλο ένα σημαντικό ζήτημα είναι αυτό της ιδιωτικότητας, καθώς αυτές οι εφαρμογές αποκαλύπτουν την θέση του χρήστη χωρίς πολλές φορές ο ίδιος να έχει δώσει την άδεια για κάτι τέτοιο και μπορεί να μην το γνωρίζει.

Όσο αφορά τα τεχνικά θέματα η διαρκής εξέλιξη της τεχνολογίας θα τα καλύψει πολλή πιο σύντομα από ότι περιμένουμε. Άλλα πέραν από αυτά προκύπτουν άλλα θέματα που προκύπτουν με την χρήση της τεχνολογία της επαυξημένης πραγματικότητας και ανακύπτουν θέματα που έχουν σχέση με την κοινωνική αποδοχή και τις επιπτώσεις που θα έχει η επαυξημένη πραγματικότητα στην καθημερινότητα μας.

## 5.2 Μελλοντική Δουλεία

Σε μια μελλοντική ενημέρωση της εφαρμογής μας μπορούμε να κάνουμε την εφαρμογή πιο ενδιαφέρουσα αξιοποιώντας την ήδη υπάρχουσα διαδικτυακή μας υπηρεσία προσθέτοντας σε αυτήν και άλλες δυνατότητες όπως για παράδειγμα να μπορούν οι άλλοι συμμετέχοντες στην εφαρμογή μας να δουν που είναι οι αντίπαλοι τους. Επίσης θα μπορούσαμε να εισάγουμε και πιο πολύπλοκα τρισδιάστατα αντικείμενα αλλά και χαρακτήρες που θα μπορούσαμε μέσω αλγορίθμων ανίχνευσης ακμών και ανίχνευση συγκρούσεων (collision detection) να δημιουργήσουμε μια νέα εμπειρία στον χρήστη.

Άλλη μια πολλή ενδιαφέρουσα προσθήκη που θα μπορούσαμε να κάνουμε θα ήταν να μπορούμε να αναπτύξουμε ένα σύστημα για την ανταλλαγή άμεσων μηνυμάτων μέσα από την εφαρμογή μας για τους συμμετέχοντες στο παιχνίδι μας κατά την διάρκεια του παιχνιδιού πάνω στην οθόνη της εφαρμογή μας.

Γενικά η επαυξημένη πραγματικότητα μπορεί να μας δώσει πολλές ιδέες για την ανάπτυξη πολλών ειδών εφαρμογών σε παρά πολλούς τομείς των ανθρωπίνων δραστηριοτήτων και ειδικά σε συσκευές έξυπνων κινητών που είναι πιο εύχρηστες, πιο βολικές για τον τελικό χρήστη και είναι πάντα μαζί του.

## ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

### Η κλάση LoginActivity

```
public class LoginActivity extends Activity {
    Button btnLogin;
    Button btnLinkToRegister;
    EditText inputEmail;
    EditText inputPassword;
    TextView loginErrorMsg;

    // JSON Response node names
    private static String KEY_SUCCESS = "success";
    private static String KEY_ERROR = "error";
    private static String KEY_ERROR_MSG = "error_msg";
    private static String KEY_UID = "uid";
    private static String KEY_NAME = "name";
    private static String KEY_EMAIL = "email";
    private static String KEY_CREATED_AT = "created_at";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);

        // Importing all assets like buttons, text fields
        inputEmail = (EditText) findViewById(R.id.loginEmail);
        inputPassword = (EditText) findViewById(R.id.loginPassword);
        btnLogin = (Button) findViewById(R.id.btnLogin);
        btnLinkToRegister = (Button) findViewById(R.id.btnLinkToRegisterScreen);
        loginErrorMsg = (TextView) findViewById(R.id.login_error);

        // Login button Click Event
        btnLogin.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {
                String email = inputEmail.getText().toString();
                String password = inputPassword.getText().toString();
                UserFunctions userFunction = new UserFunctions();
                Log.d("Button", "Login");
                JSONObject json = userFunction.loginUser(email, password);

                // check for login response
                try {
                    if (json.getString(KEY_SUCCESS) != null) {
                        loginErrorMsg.setText("");
                        String res = json.getString(KEY_SUCCESS);
```

```

        if(Integer.parseInt(res) == 1){
            // user successfully logged in
            // Store user details in SQLite Database
            DatabaseHandler db = new DatabaseHandler(getApplicationContext());
            JSONObject json_user = json.getJSONObject("user");

            // Clear all previous data in database
            userFunction.logoutUser(getApplicationContext());
            db.addUser(json_user.getString(KEY_NAME),
json_user.getString(KEY_EMAIL), json.getString(KEY_UID),
json_user.getString(KEY_CREATED_AT));

            // Launch Dashboard Screen
            Intent dashboard = new Intent(getApplicationContext(),
MainMenuActivity.class);

            // Close all views before launching Dashboard
            dashboard.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(dashboard);

            // Close Login Screen
            finish();
        }else{
            // Error in login
            loginErrorMsg.setText("Incorrect username/password");
        }
    }
} catch (JSONException e) {
    e.printStackTrace();
}
}
});

// Link to Register Screen
btnLinkToRegister.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {
        Intent i = new Intent(getApplicationContext(),
RegisterActivity.class);
        startActivity(i);
        finish();
    }
});
}
}
}

```

### **RegisterActivity**

```

public class RegisterActivity extends Activity{

```

```

Button btnRegister;
Button btnLinkToLogin;
EditText inputFullName;
EditText inputEmail;
EditText inputPassword;
TextView registerErrorMsg;

// JSON Response node names
private static String KEY_SUCCESS = "success";
private static String KEY_ERROR = "error";
private static String KEY_ERROR_MSG = "error_msg";
private static String KEY_UID = "uid";
private static String KEY_NAME = "name";
private static String KEY_EMAIL = "email";
private static String KEY_CREATED_AT = "created_at";

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.register);

    // Importing all assets like buttons, text fields
    inputFullName = (EditText) findViewById(R.id.registerName);
    inputEmail = (EditText) findViewById(R.id.registerEmail);
    inputPassword = (EditText) findViewById(R.id.registerPassword);
    btnRegister = (Button) findViewById(R.id.btnRegister);
    btnLinkToLogin = (Button) findViewById(R.id.btnLinkToLoginScreen);
    registerErrorMsg = (TextView) findViewById(R.id.register_error);

    // Register Button Click event
    btnRegister.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            String name = inputFullName.getText().toString();
            String email = inputEmail.getText().toString();
            String password = inputPassword.getText().toString();
            UserFunctions userFunction = new UserFunctions();
            JSONObject json = userFunction.registerUser(name, email, password);

            // check for login response
            try {
                if (json.getString(KEY_SUCCESS) != null) {
                    registerErrorMsg.setText("");
                    String res = json.getString(KEY_SUCCESS);
                    if (Integer.parseInt(res) == 1){
                        // user successfully registred
                        // Store user details in SQLite Database
                        DatabaseHandler db = new DatabaseHandler(getApplicationContext());
                        JSONObject json_user = json.getJSONObject("user");

```



```

        // Clear all previous data in database
        userFunction.logoutUser(getApplicationContext());
        db.addUser(json_user.getString(KEY_NAME),
json_user.getString(KEY_EMAIL), json_user.getString(KEY_UID),
json_user.getString(KEY_CREATED_AT));
        // Launch Dashboard Screen
        Intent dashboard = new Intent(getApplicationContext(),
MainMenuActivity.class);
        // Close all views before launching Dashboard
        dashboard.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(dashboard);
        // Close Registration Screen
        finish();
    }else{
        // Error in registration
        registerErrorMsg.setText("Error occurred in registration");
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
}
});

// Link to Login Screen
btnLinkToLogin.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {
        Intent i = new Intent(getApplicationContext(),
            LoginActivity.class);
        startActivity(i);
        // Close Registration View
        finish();
    }
});
}
}
}

```

### **Η κλάση MainMenuActivity**

```

public class MainMenuActivity extends Activity implements View.OnClickListener {
    UserFunctions userFunctions;
    Button btnLogout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

setContentView(R.layout.mainmenu);

TextView t = ((TextView)findViewById(R.id.main_menu_startgame));
t.setOnClickListener(this);t.setText(R.string.startgame);
t = ((TextView)findViewById(R.id.main_menu_intructions));
t.setOnClickListener(this);t.setText(R.string.instructions);
t = ((TextView)findViewById(R.id.main_menu_about));
t.setOnClickListener(this);t.setText(R.string.about);
t=((TextView)findViewById(R.id.main_menu_exit));
t.setOnClickListener(this);t.setText(R.string.exit);
}

@Override
protected void onResume() {
    super.onResume();
    TextView t = ((TextView)findViewById(R.id.main_menu_startgame));
    t.setOnClickListener(this);t.setText(R.string.startgame);
}

@Override
public void onClick(View v) {
    String str = ((TextView)v).getText().toString();
    if(str.equals(getResources().getString(R.string.startgame))) {
        //start the game activity
        ((TextView)v).setClickable(false);
        ((TextView)v).setText(R.string.starting);
        Intent intent = new Intent(MainMenuActivity.this, GameActivity.class);
        intent.setAction(Intent.ACTION_VIEW);
        startActivity(intent);
    } else if(str.equals(getResources().getString(R.string.instructions))) {
        //show the instructions activity
        startActivity(new Intent(MainMenuActivity.this, InstructionsActivity.class));
    } else if(str.equals(getResources().getString(R.string.about))) {
        //start the about activity
        startActivity(new Intent(MainMenuActivity.this, AboutActivity.class));
    } else if(str.equals(getResources().getString(R.string.exit))){
        //go back to login activity
        startActivity(new Intent(MainMenuActivity.this, LoginActivity.class));
    }
}

}

GameActivity
public class GameActivity extends AndARActivity implements SurfaceHolder.Callback
{

```

```
public static final boolean DEBUG = false;
```

```
SensorManager sensorManager;  
private Camera camera;  
int orientationSensor;  
float headingAngle;  
float pitchAngle;  
float rollAngle;  
int accelerometerSensor;  
float xAxis;  
float yAxis;  
float zAxis;  
LocationManager locationManager;  
double latitude;  
double longitude;  
double altitude;  
double bearing;  
double distance;
```

```
float lat;  
float lon;  
Location setLoc;  
Location locationInUse;
```

```
SharedPreferences prefs;  
TextView itemValue;  
TextView xAxisValue;  
TextView yAxisValue;  
TextView zAxisValue;  
TextView headingValue;  
TextView pitchValue;  
TextView rollValue;  
TextView altitudeValue;  
TextView latitudeValue;  
TextView longitudeValue;  
TextView bearingValue;  
TextView distanceValue;
```

```
CustomObject someObject;  
CustomObject3 someObject3;  
CustomObject2 someObject2;  
CustomObject4 someObject4;
```

```
ARToolkit artoolkit;  
Button shot;
```

```

Button stoiceia;
private Resources res;
int ic;

@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    CustomRenderer renderer = new CustomRenderer();//optional, may be set to null
    super.setNonARRenderer(renderer);//or might be omitted

    prefs = getSharedPreferences("SimpleAR", 0);

    // getSurfaceView().setOnTouchListener(new TouchEventHandler());
    getSurfaceView().getHolder().addCallback(this);

    //internal url for model
    //fileUtil =new AssetsFileUtil(getResources());
    // fileUtil.setBaseFolder("/models/");
    //read model
    // ObjParser parser = new ObjParser(getResources().getAssets(),"/models/android.obj");
    //String modelFileName2 = "chair.obj";
    // BufferedReader fileReader = fileUtil.getReaderFromName(modelFileName2);
    try {
        //register a object for each marker type
        artoolkit = super.getArtoolkit();

        someObject = new CustomObject
            ("earth", "marker_at16.patt", 80.0, new double[]{0,0});
        //("test", "marker_at16.patt", 80.0, new double[]{0,0},
this.getContext(),"android");
        artoolkit.registerARObject(someObject);

        // Toast.makeText(CustomActivity.this, getResources().getText(
R.string.item1),Toast.LENGTH_SHORT ).show();

        someObject2 = new CustomObject2
            ("test", "marker_at17.patt", 80.0, new double[]{0,0});
        artoolkit.registerARObject(someObject2);
        Toast.makeText(GameActivity.this,
getResources().getText(R.string.item2),Toast.LENGTH_SHORT ).show();
        someObject3 = new CustomObject3
            ("test", "new3.patt", 80.0, new double[]{0,0});
        artoolkit.registerARObject(someObject3);
        Toast.makeText(GameActivity.this,
getResources().getText(R.string.item3),Toast.LENGTH_SHORT ).show();

```

```

        someObject4 = new CustomObject4
            ("test", "new6.patt", 80.0, new double[]{0,0});

        artoolkit.registerARObject(someObject4);
        Toast.makeText(GameActivity.this,
getResources().getText(R.string.item4),Toast.LENGTH_SHORT ).show();

    } catch (AndARException ex){ }

startPreview();
createHUD();
sensor();

stoixeia = (Button) findViewById(R.id.stoixeiaButton);
stoixeia.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //To change body of implemented methods use File | Settings | File Templates.
        launchStoixeia();
    }
});

shot = (Button) findViewById(R.id.shotButton);
shot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //To change body of implemented methods use File | Settings | File Templates.
        new TakeAsyncScreenshot().execute();
    }
});
}

@Override
public void uncaughtException(Thread thread, Throwable ex) {
    Log.e("AndAR EXCEPTION", ex.getMessage());
    finish();
}

/* create the menu
* @see android.app.Activity#onCreateOptionsMenu(android.view.Menu)

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0, MENU_TRANSLATE, 0, res.getText(R.string.translate))
        .setIcon(R.drawable.translate);
    menu.add(0, MENU_ROTATE, 0, res.getText(R.string.rotate))
        .setIcon(R.drawable.rotate);
    menu.add(0, MENU_SCALE, 0, res.getText(R.string.scale))
        .setIcon(R.drawable.scale);
    menu.add(0, MENU_SCREENSHOT, 0, res.getText(R.string.take_screenshot))
        .setIcon(R.drawable.screenshoticon);
    return true;
}

public void createHUD() {

    // add layout
    LayoutInflater controlInflater = LayoutInflater.from(getBaseContext());
    View viewControl = controlInflater.inflate(R.layout.hud, null);
    ViewGroup.LayoutParams layoutParamsControl = new ViewGroup.LayoutParams(
        ViewGroup.LayoutParams.FILL_PARENT,
ViewGroup.LayoutParams.FILL_PARENT);
    this.addView(viewControl, layoutParamsControl);

    RatingBar ratingBar = (RatingBar)findViewById(R.id.ratingBar1);
    ratingBar.setRating(0);
    ratingBar.setFocusable(false);

}

void sensor(){
    locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 2000, 2,
locationListener);

    sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    orientationSensor = Sensor.TYPE_ORIENTATION;
    accelerometerSensor = Sensor.TYPE_ACCELEROMETER;
    sensorManager.registerListener(sensorEventListener,
sensorManager.getDefaultSensor(orientationSensor),
        SensorManager.SENSOR_DELAY_NORMAL);
    sensorManager.registerListener(sensorEventListener,
sensorManager.getDefaultSensor(accelerometerSensor),
        SensorManager.SENSOR_DELAY_NORMAL);

    xAxisValue = (TextView) findViewById(R.id.xAxisValue);
    yAxisValue = (TextView) findViewById(R.id.yAxisValue);
    zAxisValue = (TextView) findViewById(R.id.zAxisValue);
}

```

```

    headingValue = (TextView) findViewById(R.id.headingValue);
    pitchValue = (TextView) findViewById(R.id.pitchValue);
    rollValue = (TextView) findViewById(R.id.rollValue);
    altitudeValue = (TextView) findViewById(R.id.altitudeValue);
    longitudeValue = (TextView) findViewById(R.id.longitudeValue);
    latitudeValue = (TextView) findViewById(R.id.latitudeValue);
    bearingValue = (TextView) findViewById(R.id.bearingValue);
    distanceValue = (TextView) findViewById(R.id.itemValue);

}

final SensorEventListener sensorEventListener = new SensorEventListener() {
    public void onSensorChanged(SensorEvent sensorEvent) {
        if (sensorEvent.sensor.getType() == Sensor.TYPE_ORIENTATION)
        {
            headingAngle = sensorEvent.values[0];
            pitchAngle = sensorEvent.values[1];
            rollAngle = sensorEvent.values[2];

            //Log.d(TAG, "Heading: " + String.valueOf(headingAngle));
            // Log.d(TAG, "Pitch: " + String.valueOf(pitchAngle));
            // Log.d(TAG, "Roll: " + String.valueOf(rollAngle));

            headingValue.setText(String.valueOf(headingAngle));
            pitchValue.setText(String.valueOf(pitchAngle));
            rollValue.setText(String.valueOf(rollAngle));

            if (pitchAngle < 7 && pitchAngle > -7 && rollAngle < 7 && rollAngle > -7)
            {
                launchMap();
                // camera.release();
            }
        }
    }

    else if (sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
    {
        xAxis = sensorEvent.values[0];
        yAxis = sensorEvent.values[1];
        zAxis = sensorEvent.values[2];

        //Log.d(TAG, "X Axis: " + String.valueOf(xAxis));
        // Log.d(TAG, "Y Axis: " + String.valueOf(yAxis));
        // Log.d(TAG, "Z Axis: " + String.valueOf(zAxis));

        xAxisValue.setText(String.valueOf(xAxis));
        yAxisValue.setText(String.valueOf(yAxis));
        zAxisValue.setText(String.valueOf(zAxis));
    }
}

```

```

    }

    public void onAccuracyChanged (Sensor sensor, int accuracy) {
        //Not used
    }
};

LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        locationInUse = location;
        latitude = location.getLatitude();
        longitude = location.getLongitude();
        altitude = location.getAltitude();

        //Log.d(TAG, "Latitude: " + String.valueOf(latitude));
        // Log.d(TAG, "Longitude: " + String.valueOf(longitude));
        //Log.d(TAG, "Altitude: " + String.valueOf(altitude));

        latitudeValue.setText(String.valueOf(latitude));
        longitudeValue.setText(String.valueOf(longitude));
        altitudeValue.setText(String.valueOf(altitude));

        lat = prefs.getFloat("SetLatitude", 0.0f);
        lon = prefs.getFloat("SetLongitude", 0.0f);

        setLoc.setLatitude(lat);
        setLoc.setLongitude(lon);
        if(locationInUse != null)
        {
            bearing = locationInUse.bearingTo(setLoc);
            distance = locationInUse.distanceTo(setLoc);
            bearingValue.setText(String.valueOf(bearing));
            distanceValue.setText(String.valueOf(distance));
        }
    }
};

public void onProviderDisabled(String arg0) {

}

public void onProviderEnabled(String arg0) {

}

public void onStatusChanged(String arg0, int arg1, Bundle arg2) {

}

```



```

};

//launchMap Activity
public void launchMap() {
    Intent showMapIntent = new Intent(GameActivity.this, MapActivity.class);
    startActivity(showMapIntent);
}

//launch StoixeiaActivity
public void launchStoixeia(){
    Intent showStoixeiaActivity =new Intent(GameActivity.this, StoixeiaActivity.class);
    startActivity(showStoixeiaActivity);
    finish();
}

//TakeAsyncScreenshot
class TakeAsyncScreenshot extends AsyncTask<Void, Void, Void> {

    private String errorMsg = null;

    @Override
    protected Void doInBackground(Void... params) {
        Bitmap bm = takeScreenshot();
        FileOutputStream fos;
        try {
            fos = new FileOutputStream("/sdcard/AndARScreenshot"+new
Date().getTime()+".png");
            bm.compress(Bitmap.CompressFormat.PNG, 100, fos);
            fos.flush();
            fos.close();
        } catch (FileNotFoundException e) {
            errorMsg = e.getMessage();
            e.printStackTrace();
        } catch (IOException e) {
            errorMsg = e.getMessage();
            e.printStackTrace();
        }
        return null;
    }

    protected void onPostExecute(Void result) {
        if(errorMsg == null)
            Toast.makeText(GameActivity.this,
getResources().getText(R.string.screenshotsaved), Toast.LENGTH_SHORT ).show();
        else
            Toast.makeText(GameActivity.this,
getResources().getText(R.string.screenshotfailed)+errorMsg, Toast.LENGTH_SHORT
).show();
    }
}

```

```

    };

}

public void onResume(){
    super.onResume();

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,2000,2,locationListener);

sensorManager.registerListener(sensorEventListener,sensorManager.getDefaultSensor(orientationSensor), SensorManager.SENSOR_DELAY_NORMAL);

sensorManager.registerListener(sensorEventListener,sensorManager.getDefaultSensor(accelerometerSensor),SensorManager.SENSOR_DELAY_NORMAL);
    lat=prefs.getFloat("SetLatitude", 0.0f);
    lon=prefs.getFloat("SetLatitude", 0.0f);

    setLoc.setLatitude(lat);
    setLoc.setLongitude(lon);

    if (locationInUse !=null){
        bearing=locationInUse.bearingTo(setLoc);
        distance=locationInUse.distanceTo(setLoc);
        bearingValue.setText(String.valueOf(bearing));
        distanceValue.setText(String.valueOf(distance));
    }
    else
    {
        bearingValue.setText("no data");
        distanceValue.setText("no data");
    }
}

public void onPause(){
    super.onPause();
    locationManager.removeUpdates(locationListener);
    //sensorManager.registerListener(sensorEventListener);
    camera.release();
    camera=null;
}

public void onDestroy(){
    camera.release();
    camera=null;
} */
}

```

## CustomRenderer

```
public class CustomRenderer implements OpenGLRenderer {

    /**
     * Light definitions
     */

    private float[] ambientlight1 = { .3f, .3f, .3f, 1f };
    private float[] diffuselight1 = { .7f, .7f, .7f, 1f };
    private float[] specularlight1 = { 0.6f, 0.6f, 0.6f, 1f };
    private float[] lightposition1 = { 20.0f, -40.0f, 100.0f, 1f };

    private FloatBuffer lightPositionBuffer1 =
GraphicsUtil.makeFloatBuffer(lightposition1);
    private FloatBuffer specularLightBuffer1 =
GraphicsUtil.makeFloatBuffer(specularlight1);
    private FloatBuffer diffuseLightBuffer1 =
GraphicsUtil.makeFloatBuffer(diffuselight1);
    private FloatBuffer ambientLightBuffer1 =
GraphicsUtil.makeFloatBuffer(ambientlight1);

    /**
     * Do non Augmented Reality stuff here. Will be called once after all AR objects
have
     * been drawn. The transformation matrices may have to be reset.
     */
    public final void draw(GL10 gl) {
    }

    /**
     * Directly called before each object is drawn. Used to setup lighting and
     * other OpenGL specific things.
     */
    public final void setupEnv(GL10 gl) {
        gl.glEnable(GL10.GL_LIGHTING);
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_AMBIENT,
ambientLightBuffer1);
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_DIFFUSE, diffuseLightBuffer1);
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_SPECULAR,
specularLightBuffer1);
        gl.glLightfv(GL10.GL_LIGHT1, GL10.GL_POSITION,
lightPositionBuffer1);
        gl.glEnable(GL10.GL_LIGHT1);
        gl.glDisableClientState(GL10.GL_TEXTURE_COORD_ARRAY);
        gl.glDisable(GL10.GL_TEXTURE_2D);
    }
}
```

```

        initGL(gl);
    }

    /**
     * Called once when the OpenGL Surface was created.
     */
    public final void initGL(GL10 gl) {
        gl.glDisable(GL10.GL_COLOR_MATERIAL);
        gl.glEnable(GL10.GL_CULL_FACE);
        gl.glShadeModel(GL10.GL_SMOOTH);
        gl.glDisable(GL10.GL_COLOR_MATERIAL);
        gl.glEnable(GL10.GL_LIGHTING);
        gl.glEnable(GL10.GL_CULL_FACE);
        gl.glEnable(GL10.GL_DEPTH_TEST);
        gl.glEnable(GL10.GL_NORMALIZE);
    }
}

```

### MapActivity

```

public class MapActivity extends Activity {
    final static String TAG = "MapView";
    SensorManager sensorManager;
    SharedPreferences prefs;
    int orientationSensor;
    float headingAngle;
    float pitchAngle;
    float rollAngle;
    private MapView mapView;
    private MapController mapController;
    Button back;

    ArrayList<OverlayItem> anotherOverlayItemArray;
    Context context;
    MyLocationOverlay myLocationOverlay = null;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map);

        prefs = getSharedPreferences("MapView", 0);

        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        orientationSensor = Sensor.TYPE_ORIENTATION;
        //
        sensorManager.registerListener(sensorEventListener, sensorManager.getDefaultSensor(orientationSensor), SensorManager.SENSOR_DELAY_NORMAL);
    }
}

```

```

mapView = (MapView)findViewById(R.id.mapview);
mapView.setBuiltInZoomControls(true);
mapController = mapView.getController();
mapController.setZoom(10);

/*contactList = new ArrayList<HashMap<String, String>>();
//--- Create Another Overlay for multi marker
anotherOverlayItemArray = new ArrayList<OverlayItem>();
// anotherOverlayItemArray.add(new OverlayItem( "0, 0", "0, 0", new GeoPoint(0, 0)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio1", "Roi", new GeoPoint(37.564621, 22.796373)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio2", "5Aderfia", new GeoPoint(37.565339, 22.793171)));

anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio6", "Bouli", new GeoPoint(37.565624, 22.796288)));
anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio7", "Nisi", new GeoPoint(37.569698, 22.79054)));

anotherOverlayItemArray.add(new OverlayItem(
    "Stoixeio9", "Filaki", new GeoPoint(37.562324, 22.803175)));

ItemizedIconOverlay<OverlayItem> anotherItemizedIconOverlay
    = new ItemizedIconOverlay<OverlayItem>(
        this, anotherOverlayItemArray, myOnItemGestureListener);
mapView.getOverlays().add(anotherItemizedIconOverlay);
//---
*/
//Add Scale Bar
ScaleBarOverlay myScaleBarOverlay = new ScaleBarOverlay(this);
mapView.getOverlays().add(myScaleBarOverlay);

//Add MyLocationOverlay
myLocationOverlay = new MyLocationOverlay(this, mapView);
mapView.getOverlays().add(myLocationOverlay);
mapView.postInvalidate();

myLocationOverlay.runOnFirstFix(new Runnable() {
    @Override
    public void run() {
        mapView.getController().animateTo(myLocationOverlay.getMyLocation());
    }
});

back = (Button) findViewById(R.id.backButton);
back.setOnClickListener(new View.OnClickListener() {

```

```

        public void onClick(View v) {
            launchMain();
        }

    });

}

OnItemClickListener<OverlayItem> myOnItemClickListener
    = new.OnItemClickListener<OverlayItem>(){

    @Override
    public boolean onItemClick(int arg0, OverlayItem arg1) {
        // TODO Auto-generated method stub
        return false;
    }

    // @Override
    public boolean onItemClickSingleTapUp(int index, OverlayItem item) {
        String modelToShow = "android";
        Toast.makeText(MapActivity.this,
            item.mDescription + "\n"
                + item.mTitle + "\n"
                + item.mGeoPoint.getLatitudeE6() + " : " +
item.mGeoPoint.getLongitudeE6(),
            Toast.LENGTH_LONG).show();

        //create new Activity
        Intent intent=new Intent(getApplicationContext(), StoixeiaActivity.class);
        intent.putExtra("name", modelToShow) ;
        //intent.putExtra("title", anotherOverlayItemArray.get(index).getTitle());
        //intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        // context.startActivity(intent);
        startActivity(intent);
        finish();
        return true;
    }

};

private void launchMain() {

    Intent showMainIntent = new Intent(MapActivity.this, GameActivity.class);
    startActivity(showMainIntent);
    finish();

}

```

```

// @Override
protected boolean onTap(int index){

    Toast.makeText(context, "Touch on marker:\n"+
anotherOverlayItemArray.get(index).getTitle(), Toast.LENGTH_LONG).show();
    //create new Activity
    Intent intent=new Intent(getBaseContext(), StoixeiaActivity.class);
    intent.putExtra("title", anotherOverlayItemArray.get(index).getTitle());
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(intent);
    return true;
}

final SensorEventListener sensorEventListener = new SensorEventListener() {
    public void onSensorChanged(SensorEvent sensorEvent) {
        if (sensorEvent.sensor.getType() == Sensor.TYPE_ORIENTATION)
        {
            headingAngle = sensorEvent.values[0];
            pitchAngle = sensorEvent.values[1];
            rollAngle = sensorEvent.values[2];

            //Log.d(TAG, "Heading: " + String.valueOf(headingAngle));
            //Log.d(TAG, "Pitch: " + String.valueOf(pitchAngle));
            //Log.d(TAG, "Roll: " + String.valueOf(rollAngle));

            if (pitchAngle > 7 || pitchAngle < -7 || rollAngle > 7 || rollAngle < -7)
            {
                // launchMain();
            }
        }
    }

    public void onAccuracyChanged(Sensor arg0, int arg1) {

    }
};

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    myLocationOverlay.enableMyLocation();
    myLocationOverlay.enableCompass();
    myLocationOverlay.enableFollowLocation();
}

```

```

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    myLocationOverlay.disableMyLocation();
    myLocationOverlay.disableCompass();
    myLocationOverlay.disableFollowLocation();
}
}

```

### **StoixeiaActivity**

```

public class StoixeiaActivity extends ListActivity {
    private ProgressDialog pDialog;

    // URL to get contacts JSON
    private static String url = "http://treasurear/seriusb.com/stoixeia/";

    // JSON Node names
    private static final String TAG_STOIXEIA = "stoixeia";
    private static final String TAG_ID = "id";
    private static final String TAG_NAME = "name";
    private static final String TAG_LAT = "lat";
    private static final String TAG_LON = "lon";
    private static final String TAG_ALT = "alt";
    private static final String TAG_STOIXEIO = "stoixeio";
    private static final String TAG_STOIXEIO_1 = "stoixeio1";
    private static final String TAG_STOIXEIO_2 = "stoixeio2";
    private static final String TAG_STOIXEIO_3 = "stoixeio3";

    // contacts JSONArray
    JSONArray stoixeia = null;

    // Hashmap for ListView
    ArrayList<HashMap<String, String>> stoixeiaList;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.service_main);

        stoixeiaList = new ArrayList<HashMap<String, String>>();

        ListView lv = getListView();

        // Listview on item click listener
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {

```



```

@Override
public void onItemClick(AdapterView<?> parent, View view,
                        int position, long id) {
    // getting values from selected ListItem
    String name = ((TextView) view.findViewById(R.id.name))
        .getText().toString();
    String cost = ((TextView) view.findViewById(R.id.lan))
        .getText().toString();
    String description = ((TextView) view.findViewById(R.id.lon))
        .getText().toString();

    // Starting single contact activity
    Intent in = new Intent(getApplicationContext(),
        SingleStoixeiaActivity.class);
    in.putExtra(TAG_NAME, name);
    in.putExtra(TAG_STOIXEIO_1, stoixeio1);
    in.putExtra(TAG_STOIXEIO_2, stoixeio2);
    in.putExtra(TAG_STOIXEIO_3, stoixeio3);
    startActivity(in);

    }
});

// Calling async task to get json
new GetContacts().execute();
}

/**
 * Async task class to get json by making HTTP call
 * */
private class GetContacts extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Showing progress dialog
        pDialog = new ProgressDialog(StoixeiaActivity.this);
        pDialog.setMessage("Please wait...");
        pDialog.setCancelable(false);
        pDialog.show();

    }

    @Override
    protected Void doInBackground(Void... arg0) {
        // Creating service handler class instance
        ServiceHandler sh = new ServiceHandler();

```

```

// Making a request to url and getting response
String jsonStr = sh.makeServiceCall(url, ServiceHandler.GET);

Log.d("Response: ", "> " + jsonStr);

if (jsonStr != null) {
    try {
        JSONObject jsonObj = new JSONObject(jsonStr);

        // Getting JSON Array node
        stoixeia = jsonObj.getJSONArray(TAG_STOIXEIA);

        // looping through All Contacts
        for (int i = 0; i < stoixeia.length(); i++) {
            JSONObject c = stoixeia.getJSONObject(i);

            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String lat = c.getString(TAG_LAT);
            String lon = c.getString(TAG_LON);
            String alt = c.getString(TAG_ALT);

            // Stoixeia node is JSON Object
            JSONObject phone = c.getJSONObject(TAG_STOIXEIO);
            String stoixeio1 = phone.getString(TAG_STOIXEIO_1);
            String stoixeio2 = phone.getString(TAG_STOIXEIO_2);
            String stoixeio3 = phone.getString(TAG_STOIXEIO_3);

            // tmp hashmap for single contact
            HashMap<String, String> contact = new HashMap<String, String>();

            // adding each child node to HashMap key => value
            contact.put(TAG_ID, id);
            contact.put(TAG_NAME, name);
            contact.put(TAG_STOIXEIO_1, stoixeio1);
            contact.put(TAG_STOIXEIO_2, stoixeio2);
            contact.put(TAG_STOIXEIO_3, stoixeio3);
            // adding contact to contact list
            stoixeiaList.add(contact);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
} else {
    Log.e("ServiceHandler", "Couldn't get any data from the url");
}

```

```

        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        super.onPostExecute(result);
        // Dismiss the progress dialog
        if (pDialog.isShowing())
            pDialog.dismiss();
        /**
         * Updating parsed JSON data into ListView
         * */
        ListAdapter adapter = new SimpleAdapter(
            StoixeiaActivity.this, stoixeiaList,
            R.layout.list_item, new String[] { TAG_NAME, TAG_STOIXEIO_1,
TAG_STOIXEIO_2,
            TAG_STOIXEIO_3 }, new int[] { R.id.name,
            R.id.stoixeio1_label, R.id.stoixeio2_label ,R.id.stoixeio3_label });

        setListAdapter(adapter);
    }
}
}
}

```

### **SingleStoixeiaActivity**

```

public class SingleStoixeiaActivity extends Activity {

    // JSON node keys
    private static final String TAG_NAME = "name";
    private static final String lblStoixeio1 = "stoixeio1";
    private static final String lblStoixeio2 = "stoixeio2";
    private static final String lblStoixeio3 = "stoixeio3";
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.single_stoixeia);

        // getting intent data
        Intent in = getIntent();

        // Get JSON values from previous intent
        String name = in.getStringExtra(TAG_NAME);
        String stoixeio1 = in.getStringExtra(lblStoixeio1);
        String stoixeio2 = in.getStringExtra(lblStoixeio2);
        String stoixeio3 = in.getStringExtra(lblStoixeio3);
    }
}

```

```

// Displaying all values on the screen
TextView lblName = (TextView) findViewById(R.id.name_label);
TextView lblStoixeio1 = (TextView) findViewById(R.id.stoixeio1_label);
TextView lblStoixeio2 = (TextView) findViewById(R.id.stoixeio2_label);
TextView lblStoixeio3 = (TextView) findViewById(R.id.stoixeio3_label);
lblName.setText(name);
lblStoixeio1.setText(stoixeio1);
lblStoixeio2.setText(stoixeio2);
lblStoixeio3.setText(stoixeio3);
}
}

```

### **DatabaseHandler**

```

public class DatabaseHandler extends SQLiteOpenHelper {

// All Static variables
// Database Version
private static final int DATABASE_VERSION = 1;

// Database Name
private static final String DATABASE_NAME = "a8809512_trear";

// Login table name
private static final String TABLE_LOGIN = "users";

// Login Table Columns names
private static final String KEY_ID = "id";
private static final String KEY_NAME = "name";
private static final String KEY_EMAIL = "email";
private static final String KEY_UID = "uid";
private static final String KEY_CREATED_AT = "created_at";

public DatabaseHandler(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

// Creating Tables
@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_LOGIN + "("
        + KEY_ID + " INTEGER PRIMARY KEY,"
        + KEY_NAME + " TEXT,"
        + KEY_EMAIL + " TEXT UNIQUE,"
        + KEY_UID + " TEXT,"
        + KEY_CREATED_AT + " TEXT" + ")";
    db.execSQL(CREATE_LOGIN_TABLE);
}
}

```

```

// Upgrading database
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // Drop older table if existed
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_LOGIN);

    // Create tables again
    onCreate(db);
}

/**
 * Storing user details in database
 */
public void addUser(String name, String email, String uid, String created_at) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_NAME, name); // Name
    values.put(KEY_EMAIL, email); // Email
    values.put(KEY_UID, uid); // Email
    values.put(KEY_CREATED_AT, created_at); // Created At

    // Inserting Row
    db.insert(TABLE_LOGIN, null, values);
    db.close(); // Closing database connection
}

/**
 * Getting user data from database
 */
public HashMap<String, String> getUserDetails(){
    HashMap<String,String> user = new HashMap<String,String>();
    String selectQuery = "SELECT * FROM " + TABLE_LOGIN;

    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    // Move to first row
    cursor.moveToFirst();
    if(cursor.getCount() > 0){
        user.put("name", cursor.getString(1));
        user.put("email", cursor.getString(2));
        user.put("uid", cursor.getString(3));
        user.put("created_at", cursor.getString(4));
    }
    cursor.close();
    db.close();
    // return user
}

```

```

        return user;
    }

    /**
     * Getting user login status
     * return true if rows are there in table
     */
    public int getRowCount() {
        String countQuery = "SELECT * FROM " + TABLE_LOGIN;
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(countQuery, null);
        int rowCount = cursor.getCount();
        db.close();
        cursor.close();

        // return row count
        return rowCount;
    }

    /**
     * Re crate database
     * Delete all tables and create them again
     */
    public void resetTables(){
        SQLiteDatabase db = this.getWritableDatabase();
        // Delete All Rows
        db.delete(TABLE_LOGIN, null, null);
        db.close();
    }
}

```

## JSONParser

```

public class JSONParser {
    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    public JSONObject getJSONFromUrl(String url, List<NameValuePair> params) {

        // Making HTTP request
    }
}

```

```

try {
    // defaultHttpClient
    DefaultHttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(url);
    httpPost.setEntity(new UrlEncodedFormEntity(params));

    HttpResponse httpResponse = httpClient.execute(httpPost);
    HttpEntity httpEntity = httpResponse.getEntity();
    is = httpEntity.getContent();

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
    Log.e("JSON", json);
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " + e.toString());
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}

// return JSON String
return jsonObj;

}
}

```

### **UserFunctions**

```
public class UserFunctions {
```

```

private JSONParser jsonParser;

private static String loginURL = "http://treasurear.seriusb.com/";
private static String registerURL = "http://treasurear.seriusb.com/";

private static String login_tag = "login";
private static String register_tag = "register";

// constructor
public UserFunctions(){
    jsonParser = new JSONParser();
}

/**
 * function make Login Request
 * @param email
 * @param password
 * */
public JSONObject loginUser(String email, String password){
    // Building Parameters
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("tag", login_tag));
    params.add(new BasicNameValuePair("email", email));
    params.add(new BasicNameValuePair("password", password));
    JSONObject json = jsonParser.getJSONFromUrl(loginURL, params);
    // return json
    // Log.e("JSON", json.toString());
    return json;
}

/**
 * function make Login Request
 * @param name
 * @param email
 * @param password
 * */
public JSONObject registerUser(String name, String email, String password){
    // Building Parameters
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("tag", register_tag));
    params.add(new BasicNameValuePair("name", name));
    params.add(new BasicNameValuePair("email", email));
    params.add(new BasicNameValuePair("password", password));

    // getting JSON Object
    JSONObject json = jsonParser.getJSONFromUrl(registerURL, params);
    // return json
    return json;
}

```



```

}

/**
 * Function get Login status
 * */
public boolean isUserLoggedIn(Context context){
    DatabaseHandler db = new DatabaseHandler(context);
    int count = db.getRowCount();
    if(count > 0){
        // user logged in
        return true;
    }
    return false;
}

/**
 * Function to logout user
 * Reset Database
 * */
public boolean logoutUser(Context context){
    DatabaseHandler db = new DatabaseHandler(context);
    db.resetTables();
    return true;
}
}

```

### **InstructionsActivity**

```

public class InstructionsActivity extends Activity {

    private WebView mWebView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.instructions);
        mWebView = (WebView) findViewById(R.id.instructions_webview);

        WebSettings webSettings = mWebView.getSettings();
        webSettings.setDefaultTextEncodingName("utf-8");
        webSettings.setSupportZoom(true);
        webSettings.setBuiltInZoomControls(true);

        WebChromeClient client = new WebChromeClient();
        mWebView.setWebChromeClient(client);
    }
}

```

```

mWebView.loadUrl("file:///android_asset/help/"+getResources().getString(R.string.help_file)
);
}
}

```

### **AboutActivity**

```

public class AboutActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about);
        TextView t = (TextView)findViewById(R.id.about_text);
        t.setText("This App Develop by Theodore Gudis for his Master Thesis");
        // finish();
    }
}

```

### **CustomObject**

```

public class CustomObject extends ARObject {

    public CustomObject(String name, String patternName,
        double markerWidth, double[] markerCenter) {
        super(name, patternName, markerWidth, markerCenter);
        float mat_ambientf[] = {0f, 1.0f, 0f, 1.0f};
        float mat_flashf[] = {0f, 1.0f, 0f, 1.0f};
        float mat_diffusef[] = {0f, 1.0f, 0f, 1.0f};
        float mat_flash_shinyf[] = {50.0f};

        mat_ambient = GraphicsUtil.makeFloatBuffer(mat_ambientf);
        mat_flash = GraphicsUtil.makeFloatBuffer(mat_flashf);
        mat_flash_shiny = GraphicsUtil.makeFloatBuffer(mat_flash_shinyf);
        mat_diffuse = GraphicsUtil.makeFloatBuffer(mat_diffusef);

    }
    public CustomObject(String name, String patternName,
        double markerWidth, double[] markerCenter, float[] customColor) {
        super(name, patternName, markerWidth, markerCenter);
        float mat_flash_shinyf[] = {50.0f};

        mat_ambient = GraphicsUtil.makeFloatBuffer(customColor);
        mat_flash = GraphicsUtil.makeFloatBuffer(customColor);
        mat_flash_shiny = GraphicsUtil.makeFloatBuffer(mat_flash_shinyf);
        mat_diffuse = GraphicsUtil.makeFloatBuffer(customColor);

    }

    /**

```

```

    * Just a box, imported from the AndAR project.
    */
    private SimpleBox box = new SimpleBox();
    private FloatBuffer mat_flash;
    private FloatBuffer mat_ambient;
    private FloatBuffer mat_flash_shiny;
    private FloatBuffer mat_diffuse;

    /**
     * Everything drawn here will be drawn directly onto the marker,
     * as the corresponding translation matrix will already be applied.
     */
    @Override
    public final void draw(GL10 gl) {
        super.draw(gl);

        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK,
GL10.GL_SPECULAR,mat_flash);
        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_SHININESS,
mat_flash_shiny);
        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_DIFFUSE,
mat_diffuse);
        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_AMBIENT,
mat_ambient);

        //draw cube
        gl.glColor4f(1.0f, 1.0f, 1.0f, 1.0f);
        gl.glTranslatef( 0.0f, 0.0f, 12.5f );

        //draw the box
        box.draw(gl);
    }
    @Override
    public void init(GL10 gl) {

    }
}

```

## Κατασκευή Διαδικτυακή Υπηρεσία

### Config.php

```

<?php

/**
 * Database config variables
 */
define("DB_HOST", "mysql5.000webhost.com");

```

```
define("DB_USER", "a8809512_trear");
define("DB_PASSWORD", "xxxxxxxx");
define("DB_DATABASE", "a8809512_trear");//"android_api");
?>
```

### **DB\_Connect.php**

```
<?php
class DB_Connect {

    // constructor
    function __construct() {

    }

    // destructor
    function __destruct() {
        // $this->close();
    }

    // Connecting to database
    public function connect() {
        require_once 'include/config.php';
        // connecting to mysql
        $con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
        // selecting database
        mysql_select_db(DB_DATABASE);

        // return database handler
        return $con;
    }

    // Closing database connection
    public function close() {
        mysql_close();
    }

}

?>

?php
```

### **DB\_Functions**

```
class DB_Functions {

    private $db;

    //put your code here
```

```

// constructor
function __construct() {
    require_once 'DB_Connect.php';
    // connecting to database
    $this->db = new DB_Connect();
    $this->db->connect();
}

// destructor
function __destruct() {

}

/**
 * Storing new user
 * returns user details
 */
public function storeUser($name, $email, $password) {
    $uuid = uniqid("", true);
    $hash = $this->hashSHA($password);
    $encrypted_password = $hash["encrypted"]; // encrypted password
    $salt = $hash["salt"]; // salt
    $result = mysql_query("INSERT INTO users(unique_id, name, email,
encrypted_password, salt, created_at) VALUES('$uuid', '$name', '$email',
'$encrypted_password', '$salt', NOW())");
    // check for successful store
    if ($result) {
        // get user details
        $uid = mysql_insert_id(); // last inserted id
        $result = mysql_query("SELECT * FROM users WHERE uid = $uid");
        // return user details
        return mysql_fetch_array($result);
    } else {
        return false;
    }
}

/**
 * Get user by email and password
 */
public function getUserByEmailAndPassword($email, $password) {
    $result = mysql_query("SELECT * FROM users WHERE email = '$email'") or
die(mysql_error());
    // check for result
    $no_of_rows = mysql_num_rows($result);
    if ($no_of_rows > 0) {
        $result = mysql_fetch_array($result);
        $salt = $result['salt'];
    }
}

```

```

    $encrypted_password = $result['encrypted_password'];
    $hash = $this->checkhashSSHA($salt, $password);
    // check for password equality
    if ($encrypted_password == $hash) {
        // user authentication details are correct
        return $result;
    }
} else {
    // user not found
    return false;
}
}

/**
 * Check user is existed or not
 */
public function isUserExisted($email) {
    $result = mysql_query("SELECT email from users WHERE email = '$email'");
    $no_of_rows = mysql_num_rows($result);
    if ($no_of_rows > 0) {
        // user existed
        return true;
    } else {
        // user not existed
        return false;
    }
}

/**
 * Encrypting password
 * @param password
 * returns salt and encrypted password
 */
public function hashSSHA($password) {

    $salt = sha1(rand());
    $salt = substr($salt, 0, 10);
    $encrypted = base64_encode(sha1($password . $salt, true) . $salt);
    $hash = array("salt" => $salt, "encrypted" => $encrypted);
    return $hash;
}

/**
 * Decrypting password
 * @param salt, password
 * returns hash string
 */
public function checkhashSSHA($salt, $password) {

```

```

        $hash = base64_encode(sha1($password . $salt, true) . $salt);

        return $hash;
    }

}

?>

```

### Index.php

```

<?php

/**
 * File to handle all API requests
 * Accepts GET and POST
 *
 * Each request will be identified by TAG
 * Response will be JSON data
 */

/**
 * check for POST request
 */
if (isset($_POST['tag']) && $_POST['tag'] != "") {
    // get tag
    $tag = $_POST['tag'];

    // include db handler
    require_once 'include/DB_Functions.php';
    $db = new DB_Functions();

    // response Array
    $response = array("tag" => $tag, "success" => 0, "error" => 0);

    // check for tag type
    if ($tag == 'login') {
        // Request type is check Login
        $email = $_POST['email'];
        $password = $_POST['password'];

        // check for user
        $user = $db->getUserByEmailAndPassword($email, $password);
        if ($user != false) {
            // user found
            // echo json with success = 1
            $response["success"] = 1;
            $response["uid"] = $user["unique_id"];
        }
    }
}

```

```

$response["user"]["name"] = $user["name"];
$response["user"]["email"] = $user["email"];
$response["user"]["created_at"] = $user["created_at"];
$response["user"]["updated_at"] = $user["updated_at"];
echo json_encode($response);
} else {
    // user not found
    // echo json with error = 1
    $response["error"] = 1;
    $response["error_msg"] = "Incorrect email or password!";
    echo json_encode($response);
}
} else if ($tag == 'register') {
    // Request type is Register new user
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    // check if user is already existed
    if ($db->isUserExisted($email)) {
        // user is already existed - error response
        $response["error"] = 2;
        $response["error_msg"] = "User already existed";
        echo json_encode($response);
    } else {
        // store user
        $user = $db->storeUser($name, $email, $password);
        if ($user) {
            // user stored successfully
            $response["success"] = 1;
            $response["uid"] = $user["unique_id"];
            $response["user"]["name"] = $user["name"];
            $response["user"]["email"] = $user["email"];
            $response["user"]["created_at"] = $user["created_at"];
            $response["user"]["updated_at"] = $user["updated_at"];
            echo json_encode($response);
        } else {
            // user failed to store
            $response["error"] = 1;
            $response["error_msg"] = "Error occurred in Registration";
            echo json_encode($response);
        }
    }
} else {
    echo "Invalid Request";
}
} else {
    echo "Access Denied";
}

```



```
}  
?>
```

### Stoixeia JSON

```
{  
  "kiboi": [  
    {  
      "id": "001",  
      "name": "Nero",  
      "stoixeio1": "mbortis",  
      "stoixeio1": "5aderfia",  
      "stoixeio1": "boyleytiko",  
    },  
    {  
      "id": "002",  
      "name": "Photia",  
      "stoixeio1": "anaktora",  
      "stoixeio1": "bouli",  
      "stoixeio1": "tsami",  
    },  
    {  
      "id": "003",  
      "name": "hi",  
      "stoixeio1": "pyli",  
      "stoixeio1": "arvanitia",  
      "stoixeio1": "roloi",  
    },  
    {  
      "id": "004",  
      "name": "Aeras",  
      "stoixeio1": "faros",  
      "stoixeio2": "nayarxoi",  
      "stoixeio3": "kaposistrias",  
    },  
  ]  
}
```

# Παράρτημα Β: Βιβλιογραφία

## Δημοσίευση

Paul Milgram and Fumio Kishino, "Taxonomy of Mixed Reality Visual Displays",  
IEICE Transactions on Information Systems, pp. 1321-1329, December 1994.

Ronald T. Azuma, "A Survey of Augmented Reality", Presence: Teleoperators and  
Virtual Environments, pp. 355-385, August 1997.

[http://softwareforschung.de/fileadmin/\\_softwareforschung/downloads/WIS  
TA/Tobias\\_Domhan\\_Studienarbeit.pdf](http://softwareforschung.de/fileadmin/_softwareforschung/downloads/WIS<br/>TA/Tobias_Domhan_Studienarbeit.pdf)

## Βιβλία

Professional Android 2 Application Development, Reto Meier

Pro Android Augmented Reality, Raghav Sood

Professional Augmented Reality Browsers for Smartphones, Lester Madden

Learning Android Game Programming, Richard A. Rogers

## Διαδίκτυο

<http://en.wikipedia.org/wiki/ARToolKit>

<http://www.hitl.washington.edu/artoolkit/documentation/vision.htm>

<http://www.hitl.washington.edu/artoolkit/documentation/hardware.htm>

<http://www.hitl.washington.edu/artoolkit/documentation/cs.htm>

<http://www.hitl.washington.edu/artoolkit/>

<http://www.hitl.washington.edu/artoolkit/documentation/cs.htm>

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

<http://el.wikipedia.org/wiki/PHP>

<http://en.wikipedia.org/wiki/JSON>

<http://el.wikipedia.org/wiki/MySQL>

<http://en.wikipedia.org/wiki/PhpMyAdmin>

<http://www.json.org/json-el.html>

<http://www.php.net/manual/en/refs.webservice.php>

[http://en.wikipedia.org/wiki/Augmented\\_reality#cite\\_note-118](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-118)

<https://launchpad.net/artoolkitplus>