



Δημιουργία εφαρμογής εντοπισμού θέσης, με χρήση Gps, πάνω σε λειτουργικό σύστημα Android

Απρίλιος 2013

Σταυρούλα Τσάκωνα Α.Μ : 0484

Επιβλέπων καθηγητής : Χριστοδούλου Σωτήριος

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον προϊστάμενο του Τμήματος Πληροφορικής και Τηλεπικοινωνιών, του Ερευνητικού Κέντρου “Δημόκριτος”, τον κ. Κούρτη Αναστάσιο για την εμπιστοσύνη που μου έδειξε δίνοντας μου το συγκεκριμένο θέμα.

Θα ήθελα επίσης να ευχαριστήσω τον κ. Μαντζουράτο Σπύρο για την βοήθεια του.

Ένα μεγάλο ευχαριστώ σε κάθε ένα από τα άτομα του τμήματος Τηλεπικοινωνιών για την πολύτιμη συμμετοχή τους στην διεξαγωγή του αποτελέσματος της πτυχιακής μου εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή κ. Χριστοδούλου Σωτήρη για την εμπιστοσύνη που μου έδειξε δίνοντας μου τη δυνατότητα να εκπονήσω την πτυχιακή μου εργασία στον τομέα που επιθυμούσα. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του, για την επίλυση διάφορων θεμάτων.

Περίληψη

Η τεχνολογία εξελίσσεται και μαζί με αυτήν και οι ανάγκες των ανθρώπων.

Στα πλαίσια αυτής της πτυχιακής εργασίας σκοπός είναι να εξεταστούν τα στάδια δημιουργίας μιας εφαρμογής θεσιθεσίας, το γνωστό σε όλους μας GPS (GlobalPositioningSystem) μέσω κινητών με λειτουργικό σύστημα Android και να αναλυθεί σε όλους και όλες ένα εργαλείο πλέον χρήσιμο για οποιονδήποτε διαλέξει να ταξιδέψει, να κάνει μία βόλτα σε άγνωστα μέρη ή ακόμα και να σιγουρευτεί για την τοποθεσία στην οποία βρίσκεται.

Θα δημιουργήσουμε μία εφαρμογή GPS με διαφορετικές ρυθμίσεις από όσες έχουν ήδη τα κινητά μας, με την οποία θα καταχωρηθούν σημεία και σημειώσεις, τα οποία θα μπορούμε να χειριστούμε με την χρήση του κινητού μας και του υπολογιστή μας.

Με την χρήση JavaScript θα διαμορφώσουμε γραφιστικά την εφαρμογή και με την HTML θα εξετάσω την αποθήκευση δεδομένων στην συσκευή.

Μετά την δημιουργία του κώδικα θα έχουμε την δυνατότητα να τρέξουμε την εφαρμογή μέσω του εργαλείου Eclipse για την εμφάνιση ενός εικονικού κινητού (Emulator) με τα στοιχεία και τις λεπτομέρειες που θα του έχω δώσει.

Στην συνέχεια θα συνδέσουμε την εφαρμογή μου με την γνωστή, στον διαδικτυακό τόπο, σελίδα <https://maps.Google.com/> για να δούμε τα αποτελέσματα από τον εντοπισμό θέσης και το σχόλιο που έχει αφήσει ο καθένας στην τοποθεσία αυτή.

Για πιο εντυπωσιακό αποτέλεσμα θα προσπαθήσουμε να εφαρμόσουμε στην εφαρμογή και την ιστοσελίδα <https://maps.google.com/?vector=1>, με την οποία μπορούμε να “μπούμε” σε κατάσταση 3D μέσα στον χάρτη σαν να είμαστε πραγματικά εκεί και να δούμε τι ακριβώς αντικρίζουν εκείνη την στιγμή οι χρήστες της εφαρμογής.

Πίνακας Περιεχομένων

Κεφάλαιο 1^ο : Γνωριμία με το Android και το GPS

1.1 Εισαγωγή

1.2 Τι είναι το λειτουργικό σύστημα Android

1.3 Εξέλιξη του Android

1.4 Αρχιτεκτονική του Android

1.4.1 Πυρήνας Linux (Linux Kernel)

1.4.2 Βιβλιοθήκες

1.4.3 Η εικονική μηχανή Dalvik

1.4.4 Χρόνος εκτέλεσης εφαρμογής (AndroidRuntime)

1.4.5 Πλαίσιο εφαρμογής (ApplicationFramework)

1.5 Στο εσωτερικό μιας εφαρμογής του Android

1.5.1 Το αρχείο AndroidManifest.xml

1.5.2 Οι φάκελοι src & res

1.5.3 Οι υπόλοιποι φάκελοι του project

1.5.4 Δομικά μέρη μιας εφαρμογής

1.6 Ασφάλεια στο Android

1.7 Τι είναι το Global Positioning System (GPS)

1.7.1 Το παρελθόν

1.7.2 Λειτουργικά τμήματα

1.7.3 Φορητές συσκευές

Κεφάλαιο 2^ο : Παρόμοιες εφαρμογές χρήσης θέσης και εντοπισμού

2.1 Εισαγωγή

2.2 Τι είναι το GoogleLatitude

2.3 Τι είναι το Foursquare

2.3.1 The social

2.3.2 Το παιχνίδι

2.3.3 Το φλερτ και οι νέες γνωριμίες

2.3.4 Τα τυχερά

2.3.5 Marketing

Κεφάλαιο 3^ο : Ανάλυση απαιτήσεων εφαρμογής

3.1 Κύκλος ανάπτυξης εφαρμογής

3.1.1 Εγκατάσταση λογισμικού

3.1.2 Ανάπτυξη πηγαίου κώδικα εφαρμογής

3.1.3 Αποσφαλμάτωση (Debugging) και δοκιμαστική φάση της εφαρμογής

3.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό

3.4 AndroidSDK

3.4.1 Μερικά από τα χαρακτηριστικά του SDK

3.5 Χρήση του Eclipse IDE με το ADT (Android Development Tools)

3.6 Ο εξομοιωτής (emulator)

3.7 Άλλα εργαλεία του Android

3.8 Μια γρήγορη εφαρμογή για το SDK

Κεφάλαιο 4^ο : Υλοποίηση εφαρμογής

4.1 Εισαγωγή

4.2 Εργαλεία που χρησιμοποιήθηκαν

4.2.1 Android

4.2.2 Eclipse

4.2.3 Java

4.2.4 GoogleMaps

4.2.4.1 Οι χάρτες Google και το Android

4.2.5 GoogleGeocoding

4.3 Προετοιμασία

4.3.1 Εγκατάσταση του EclipseIDE

4.3.2 Εγκατάσταση του ADTPlugin για το Eclipse

4.3.3 Εγκατάσταση του AndroidSDK

4.4 Δημιουργία Android εφαρμογής

4.5 Η εφαρμογή σε εικόνες

4.6 Κώδικας

4.7 Χρήση εντοπισμού θέσης για πλοήγηση

4.8 Χρήση της βιβλιοθήκης MapViewBalloons

4.9 Υλοποίηση και εισαγωγή σημείων ενδιαφέροντος στον χάρτη

4.10 Δήλωση αδειών χρήστη στο AndroidManifest

4.11 Το αποτέλεσμα της εφαρμογής εικονικά

Κεφάλαιο 5^ο : Παραδείγματα και συμπεράσματα

5.1 Παραδείγματα χρήσης

5.2 Συμπεράσματα

Βιβλιογραφία

Κεφάλαιο 1^ο: Γνωριμία με το Android και το GPS

1.1 Εισαγωγή

Στα πλαίσια αυτής της πτυχιακής εργασίας, θα μελετήσουμε τα στάδια ανάπτυξης μίας εφαρμογής ιστού σε κινητά Smartphones ή αλλιώς έξυπνα κινητά με τεχνολογία Android.

Θα εξετάσουμε τα στάδια της δημιουργίας μίας εφαρμογής και θα παρουσιάσουμε τα αποτελέσματα εφαρμογής της.

Ξεκινώντας με το πρώτο κεφάλαιο, στο οποίο θα αναφερθούμε στο τι ακριβώς είναι το λειτουργικό σύστημα Android, ποιο είναι το σύμβολο του το οποίο μας βοηθάει να το ξεχωρίσουμε από τα υπόλοιπα, αλλά θα αναφερθούμε αναλυτικά και σε περαιτέρω λεπτομέρειες για να γνωρίσουμε καλύτερα το λειτουργικό αυτό σύστημα.

Στην συνέχεια, αυτού του κεφαλαίου θα μιλήσουμε αναλυτικά και για τον δεύτερο καταλυτικό παράγοντα αυτής της πτυχιακής εργασίας, το Παγκόσμιο Σύστημα Θεσιθεσίας, γνωστό και ως Gps.

1.2 Τι είναι το λειτουργικό σύστημα Android



Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο Linux, για φορητές συσκευές όπως smartphones και tablets.

Αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις φορητές συσκευές.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance.

Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ.

Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα.

Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Στην Google, η ομάδα με επικεφαλής τον Andy Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος.

Έχει αναφερθεί ότι η Google είχε ήδη συγκεντρώσει μια σειρά από εταιρους hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυξε μια Google-branded συσκευή.

Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητώ τηλεφώνων και τους φορείς δικτύων.

Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android.

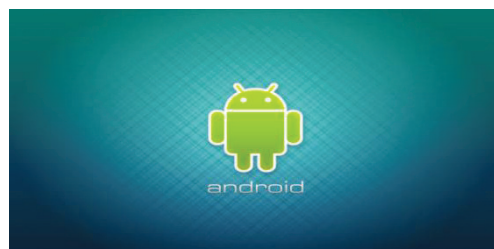
Η συσκευή κατασκευάστηκε από την HTC , και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.



nexus one™
Web meets phone.

Το πρώτο κινητό χρήσης Android

Το Λογότυπο διαφορετικά το λογότυπο της Android είναι το γνωστό σε όλους μας πράσινο ανδροειδές/ανθρωπάκι, όπως πολλοί το αποκαλούν, που εμφανίζεται στην φωτογραφία που ακολουθεί :



Το διάσημο logo των Android

1.3 Εξέλιξη του Android

Όπως αναφέραμε παραπάνω, το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα.

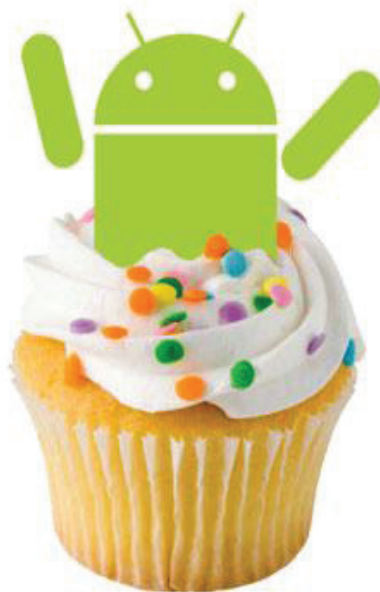
Η εξέλιξη του λόγω της ανοιχτού κώδικα φύσης του είναι ραγδαία και αυτό αντικατοπτρίζεται στο γεγονός ότι οι 7 κύριες εκδόσεις του έχουν κυκλοφορήσει σε διάστημα 2.5 ετών, από τον Απρίλη του 2009 μέχρι τον Νοέμβριο του 2011.

Στην πληροφορική συνηθίζεται τα προϊόντα hardware και software να κυκλοφορούν εκτός από τον αριθμό έκδοσης τους, και με μία κωδική ονομασία.

Η κωδική ονομασία μπορεί να είναι πχ ονόματα πόλεων (Windows Vienna, Chicago), ονόματα ζώων (OSX Leopard, Lion), στην περίπτωση όμως του Android τα κώδικα ονόματα έρχονται στη μορφή επιδόρπιου.

Στην συνέχεια βλέπουμε τις εφαρμογές αναφορικά :

1. Android 1.5 Cupcake



2. Android 1.6 Donut



3. Android 2.0/2.1 Eclair



4. Android 2.2 Froyo



5. Android 2.3 Gingerbread



6. Android 3.0 Honeycomb



7. Android 4.0 Ice Cream Sandwich



1.4 Αρχιτεκτονική του Android

Όπως αναφέραμε πιο πριν, το Android είναι μια στοίβα λογισμικού.

Η λογική πίσω από αυτήν την έκφραση και σε όλη την φιλοσοφία του Android, κρύβεται στο ακόλουθο διάγραμμα με τα βασικά συστατικά του.

Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα, που επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές με πρόσβαση που ελέγχεται από τους περιορισμούς ασφάλειας που έχουν τεθεί.

Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση υπαρχόντων εφαρμογών ή δημιουργίας καινούργιων, χρησιμοποιώντας κώδικα από αυτές.

Οι ακόλουθες υπηρεσίες εφαρμογών είναι οι ακρογωνιαίοι λίθοι της αρχιτεκτονικής όλων των εφαρμογών Android, παρέχοντας το πλαίσιο που θα χρησιμοποιηθεί για τη δημιουργία του λογισμικού :

- **Διαχειριστής Δραστηριοτήτων (Activity Manager)** - Ελέγχει τον κύκλο ζωής των δραστηριοτήτων, συμπεριλαμβανομένης της διαχείρισης του σωρού δραστηριοτήτων.
- **Όψεις (Views)** - Χρησιμοποιούνται για να κατασκευάσουν διεπαφές χρήστη (user interface) για τις δραστηριότητες (activities).
- **Διαχειριστή Ειδοποιήσεων (Notification Manager)** - Παρέχει ένα συνεπή και αποτελεσματικό μηχανισμό για να προειδοποιεί τον χρήστη.
- **Παροχέας Περιεχομένου (Content Providers)** - Επιτρέπουν στις εφαρμογές να μοιραστούν δεδομένα μεταξύ τους.
- **Διαχειριστής Πόρων (Resource Manager)** - Υποστηρίζει πόρους εκτός κώδικα όπως συμβολοσειρές και γραφικά.

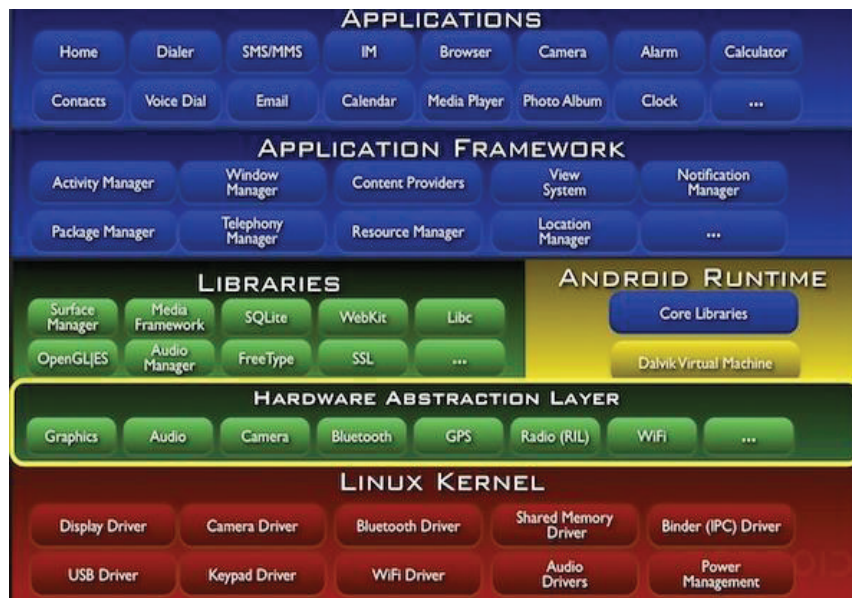
Ακολούθως θα περιγράψουμε συνοπτικά τα βασικά αυτά επίπεδα χωρίς να μπούμε σε λεπτομέρειες για όλα τα περιεχόμενα του κάθε επιπέδου.

Αν ο αναγνώστης επιθυμεί να μάθει περισσότερα, μπορεί να επισκεφθεί την επίσημη ιστοσελίδα του Android για κατασκευαστές

(<http://developer.android.com>).

Κάθε επίπεδο στηναρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο πάνω επίπεδα.

Ας δούμε τώρα αυτά τα επίπεδα ξεκινώντας από το πιο χαμηλό.



Αρχιτεκτονική λειτουργικού συστήματος Android

Από ότι βλέπουμε λοιπόν η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 5 βασικά επίπεδα :

- Τον πυρήνα Linux (**Linux Kernel**)
- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (**Libraries**)
- Την εικονική μηχανή Dalvik (**Dalvik VM**)
- Τον χρόνο εκτέλεσης (**Android Runtime**)
- Το πλαίσιο εφαρμογής (**Application Framework**)

1.4.1 Πυρήνας Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux.

Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος.

Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers).

Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής.

Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης
- Οδηγό Wifi και Bluetooth
- Οδηγό κάμερας
- κλπ

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν.

Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές.

Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη.

Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 4-5 χρόνια ακόμα.

1.4.2 Βιβλιοθήκες

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android.

Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών.

Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές.

Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους.

Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής.

Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό.

Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

- **System C library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
- **Βιβλιοθήκες Πολυμέσων** – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG.
- **Surface Manager** – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- **LibWebCore** – μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- **SGL** – η γνωστή μηχανή δισδιάστατων γραφικών.

- **Βιβλιοθήκες 3D** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.
- **FreeType** – παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.
- **SQLite** – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων.

1.4.3 Η εικονική μηχανή Dalvik

Σχεδόν το σύνολο των APIs του Android βασίζονται στη γλώσσα προγραμματισμού Java.

Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείτε ο κώδικας bytecode των εφαρμογών.

Στο Android υπάρχει κάτι παρόμοιο και δεν είναι άλλο από την εικονική μηχανή Dalvik.

Η Dalvik λοιπόν είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android.

Η κάθε εφαρμογή τρέχει μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα.

Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία .dex ώστε να τρέξουν στην VM.

Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση.

Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο.

Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης.

Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

1.4.4 Χρόνος εκτέλεσης εφαρμογής (Android Runtime)

Ο χρόνος εκτέλεσης των εφαρμογών του Android, βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik.

Εδώ βρίσκουμε το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της εικονικής μηχανής Dalvik τις λειτουργίες τις οποίες, περιγράψαμε παραπάνω.

1.4.5 Πλαίσιο εφαρμογής (Application Framework)

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές.

Οι developers έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα.

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών.

Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα Κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές.

Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android.

Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **Σύστημα προβολών (View System)** – αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ.
- **Πάροχος Περιεχομένου (Content Provider)** – δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- **Διαχειριστής Πόρων (Resource Manager)** – παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.
- **Διαχειριστής Ειδοποιήσεων (Notification Manager)** – δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στην notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ.
- **Διαχειριστής Δραστηριοτήτων (Activity Manager)** – διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών.

1.5 Στο εσωτερικό μιας εφαρμογής του Android

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK και μας δίνουν το αρχείο .apk.

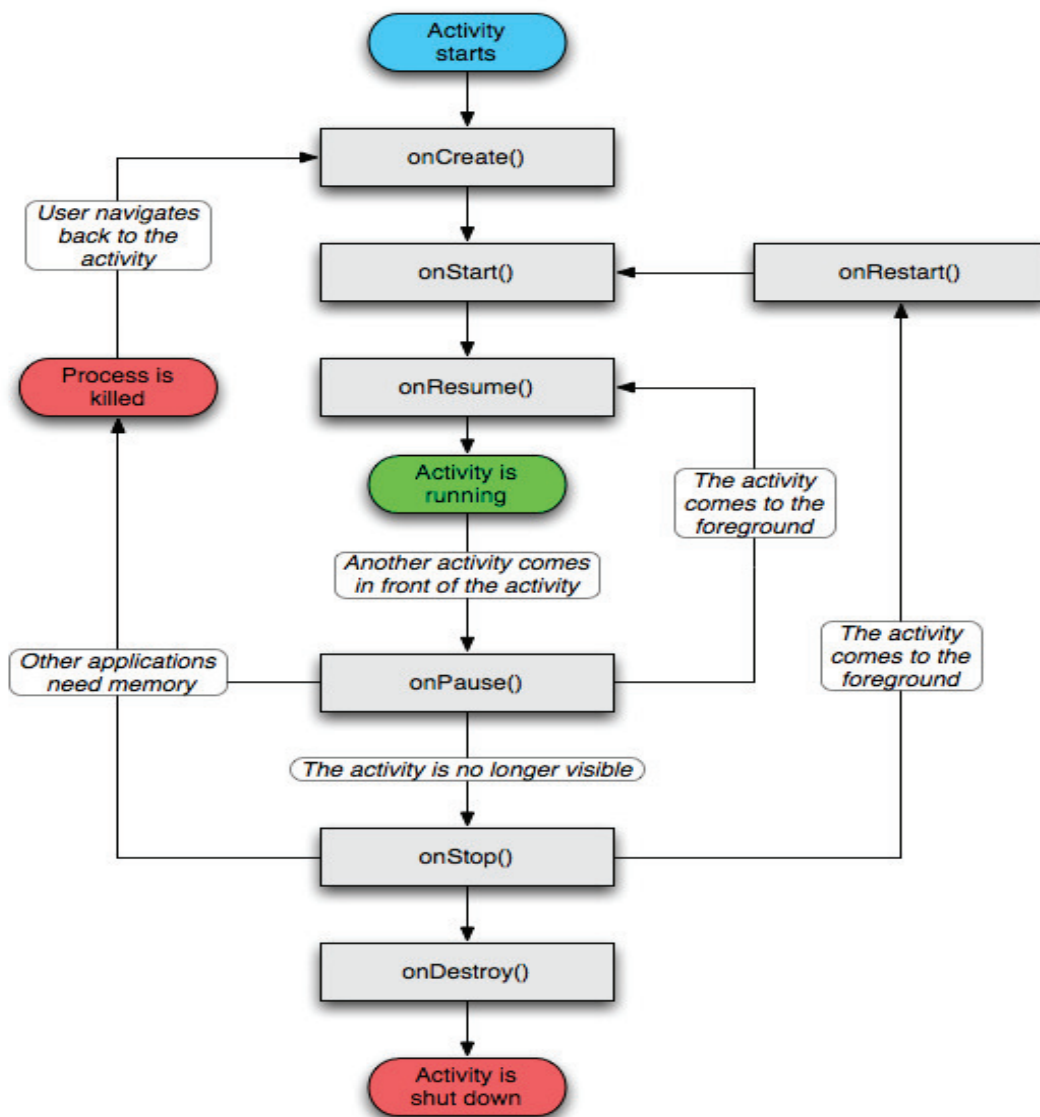
Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να εγκαταστήσουμε στις συσκευές μας.

Ξεκινώντας, η κάθε εφαρμογή αποτελείται όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους.

Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής.

Μια εφαρμογή μπορεί να αποτελείται από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας της εφαρμογής, αλλά μόνο από ένα κύριο.

Στο σχεδιάγραμμα που ακολουθεί μπορούμε να δούμε και την ακολουθία που περιγράψαμε παραπάνω με λόγια.



*Κύκλος ζωής μιας δραστηριότητας
(Activity lifecycle)*

1.5.1 Το αρχείο `AndroidManifest.xml`

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται `AndroidManifest.xml`.

Πρόκειται όπως λέει και το όνομα του για ένα αρχείο `xml` μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα.

Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη
- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.

Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής.

1.5.2 Οι φάκελοι `src` & `res`

Στον φάκελο `src` (εκ του `source`) περιέχονται τα αρχεία κλάσης τις Java όλων των `Activities`, `Services`, `ContentProviders`, βοηθητικά αρχεία, κλπ.

Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας.

Ο φάκελος `res` (εκ του `resources`) περιέχει όλα τα αρχεία εικόνας, κειμένου, `xml layout`, κλπ τα οποία χρησιμοποιούνται από τις `Activities` που βρίσκονται στον φάκελο `src`.

Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους.

Συνηθισμένοι υποφάκελοι του κύριου φακέλου `res`, είναι ο φάκελος `drawable` ο οποίος περιέχει τα αρχεία εικόνας (`.png`, `.jpg`, `.gif`) τα οποία χρησιμοποιεί η εφαρμογή μας, ο φάκελος `layout` ο οποίος περιέχει όλα τα αρχεία `xml` τα οποία ορίζουν τα διάφορα `layouts` που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος `values` στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται στην εφαρμογή.

1.5.3 Οι υπόλοιποι φάκελοι του `project`

Ένα `project` αποτελείται από περισσότερους από τους 3 βασικούς φακέλους, κάποιιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση.

Στο `project` λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο `build path` του `project` μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου `res`, όπως είναι οι φάκελοι `drawable-hdpi`, `drawable-mdpi`, `layout-port`, `menu`, κλπ.

Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίπτωση.

1.5.4 Δομικά μέρη μίας εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο `AndroidManifest.xml`, πια είναι όμως αυτά τα δομικά μέρη και πια η λειτουργία του καθενός;

- **Δραστηριότητες (Activities)** – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες `Activities` όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

- **Προθέσεις (Intents)** – Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.
- **Υπηρεσίες (Services)** – Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.
- **Πάροχος Περιεχόμενου (Content Providers)** - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.
- **Δέκτες Μετάδοσης (Broadcast Receivers)** – Πρόκειται για ένα είδους υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα η τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

1.6 Ασφάλεια στο Android

Τη στιγμή που μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στην δική της εικονική μηχανή, η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής.

Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης
- Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό ID ο οποίος είναι άγνωστος στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής, και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινείται μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείτε πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων.

Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο.

Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να τις παρέχει πρόσβαση στους πόρους που ζητάει.

1.7 Τι είναι το Global Positioning System (GPS)



Το GPS (Global Positioning System), Παγκόσμιο Σύστημα Θεσιθεσίας είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα "πλέγμα" εικοσιτεσσάρων δορυφόρων της Γης, στους οποίους υπάρχουν ειδικές συσκευές, οι οποίες ονομάζονται "δέκτες GPS".

Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές.

Το σύστημα ξεκίνησε από το Υπουργείο Άμυνας των Η.Π.Α και ονομάστηκε "NAVSTAR GPS" (Navigation Signal Timing and Ranging Global Positioning System).

Το δορυφορικό αυτό σύστημα ρυθμίζεται καθημερινά από τη Βάση Πολεμικής Αεροπορίας Σρίβερ (Schriever) με κόστος 400 εκατομμύρια δολάρια το χρόνο.

1.7.1 Το παρελθόν

Τα σημεία του ορίζοντα, ή ακόμη και τα αστέρια, χρησιμοποιούνταν από την αρχαιότητα για τον προσανατολισμό των ανθρώπων.

Ένα σταθερό άστρο στον ουρανό, με γνωστή γεωγραφική θέση ως προς το σημείο παρατήρησης, αποτελούσε σημείο αναφοράς και βοηθούσε τους ανθρώπους στο να βρουν τη σωστή πορεία τους.

Στον προσανατολισμό συνέβαλαν αργότερα και άλλα μέσα, όπως η πυξίδα και ο εξάντας.

Ωστόσο ο εξάντας είναι εύχρηστος μόνο για τον προσδιορισμό του γεωγραφικού πλάτους, ενώ η χρήση του για τον προσδιορισμό του γεωγραφικού μήκους είναι δύσκολη και εξαιρετικά σύνθετη, πράγμα που αποτελεί ένα σημαντικό μειονέκτημα για προσδιορισμό του στίγματος στην θάλασσα.

Ως αποτέλεσμα, τον 17ο αιώνα, το Ηνωμένο Βασίλειο συνέστησε ένα συμβούλιο επιστημόνων, το οποίο θα επιβράβευε χρηματικά όποιον θα μπορούσε να εφεύρει ένα όργανο, το οποίο θα επέτρεπε τον ακριβή υπολογισμό και των δύο γεωγραφικών συντεταγμένων, δηλαδή μήκους και πλάτους.

Το 1761, ο Άγγλος ωρολογοποιός Τζον Χάρισσον (John Harrison), ύστερα από προσπάθειες δώδεκα ετών, κατασκεύασε ένα όργανο, το οποίο δεν ήταν άλλο από το γνωστό σημερινό χρονόμετρο.

Σε συνδυασμό με τον εξάντα, το χρονόμετρο επέτρεπε τον υπολογισμό του στίγματος των πλοίων με εξαιρετική ακρίβεια (για τα δεδομένα της εποχής).

Πέρασαν αρκετά χρόνια μέχρι να δημιουργηθούν τα πρώτα συστήματα εντοπισμού θέσης που βασίζονταν σε ηλεκτρομαγνητικά κύματα (ραντάρ, στα μέσα του 20ού αιώνα).

Τα συστήματα αυτά χρησιμοποιήθηκαν ευρύτατα κατά τη διάρκεια του Δευτέρου Παγκοσμίου Πολέμου (και χρησιμοποιούνται ακόμη).

Τα συστήματα εντοπισμού θέσης της εποχής αποτελούνταν από ένα δίκτυο σταθμών βάσης και κατάλληλους δέκτες.

Ανάλογα με την ισχύ του σήματος που λάμβανε κάθε δέκτης από σταθμούς γνωστής γεωγραφικής θέσης, σχηματίζονταν δύο ή περισσότερες συντεταγμένες, μέσω των οποίων προσδιοριζόταν η θέση των σημείων ενδιαφέροντος επάνω σε ένα χάρτη.

Στην περίπτωση αυτή, όμως, συνέβαιναν υπήρχαν δύο διαφορετικά προβλήματα:

- Στην πρώτη περίπτωση η χρήση σταθμών βάσης, που θα εξέπεμπαν σήμα σε υψηλή συχνότητα, διέθεταν μεν υψηλή ακρίβεια εντοπισμού, αλλά είχαν μικρή εμβέλεια.

- Στη δεύτερη περίπτωση συνέβαινε το ακριβώς αντίθετο, δηλαδή ο σταθμός βάσης χρησιμοποιούσε μεν χαμηλή συχνότητα εκπομπής σήματος, προσφέροντας έτσι υψηλότερη εμβέλεια, αλλά και η ακρίβεια που παρείχε ήταν χαμηλή.

Έστω και με αυτά τα προβλήματα, η αρχή της χρήσης ραδιοκυμάτων για τον εντοπισμό της θέσης ενός σημείου είχε ήδη γίνει.

Το Global Positioning System στη σημερινή του μορφή βασίζεται σε παρεμφερή τεχνολογία.

Συνδυάζει όλες τις μεθόδους που είχαν χρησιμοποιηθεί στον ουρανό, δηλαδή την τεχνολογία των ηλεκτρομαγνητικών κυμάτων καθώς και την παρατήρηση ενός –τεχνητού αυτή τη φορά- ουράνιου σώματος.

Οι σταθμοί βάσης που λαμβάνουν και δέχονται τα απαραίτητα ηλεκτρομαγνητικά κύματα δεν είναι πλέον επίγειοι, αλλά εδρεύουν σε δορυφόρους.

Ένα δίκτυο πολυάριθμων (24 - 32) δορυφόρων που βρίσκεται σε σταθερή θέση γύρω από τον πλανήτη μας, βοηθά τους δέκτες GPS να παρέχουν το ακριβές στίγμα ενός σημείου οπουδήποτε στον κόσμο.

Όταν, το 1957, πραγματοποιήθηκε η εκτόξευση του δορυφόρου Σπούτνικ, οι άνθρωποι είχαν ήδη αντιληφθεί ότι ένα τεχνητό ουράνιο σώμα κοντά στη Γη είναι δυνατό να χρησιμοποιηθεί για να εντοπιστεί η θέση ενός σημείου πάνω στον πλανήτη.

Αμέσως μετά την εκτόξευσή του, οι ερευνητές του Ινστιτούτου Τεχνολογίας της Μασαχουσέτης (MIT) διαπίστωσαν ότι το σήμα που λαμβανόταν από τον δορυφόρο αυξανόταν καθώς αυτός πλησίαζε προς το επίγειο σημείο παρατήρησης και μειωνόταν όταν ο δορυφόρος απομακρυνόταν από αυτό.

Αυτό ήταν και το πρώτο βήμα για την υλοποίηση της τεχνολογίας που σήμερα αποκαλείται Global Positioning System.

Με τον ίδιο τρόπο που η θέση ενός δορυφόρου μπορούσε να εντοπιστεί ανάλογα με την ισχύ του σήματος που λαμβάνεται από αυτόν, υπήρχε και η δυνατότητα να συμβεί το ακριβώς αντίθετο:

- Ο δορυφόρος να εντοπίσει την ενός σημείου θέση με ιδιαίτερη ακρίβεια. Στην πραγματικότητα ένας δορυφόρος δεν είναι αρκετός για να

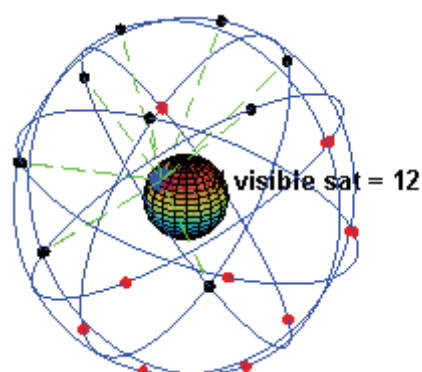
υπάρξουν ακριβή αποτελέσματα, αλλά απαιτούνται τουλάχιστον τρεις, όπως θα δούμε στη συνέχεια.

Το GPS αρχικά δημιουργήθηκε αποκλειστικά για στρατιωτική χρήση και ανήκε στη δικαιοδοσία του αμερικανικού Υπουργείου Εθνικής Άμυνας.

Στα μέσα της δεκαετίας του 1960 το σύστημα δορυφορικής πλοήγησης, γνωστό τότε με την ονομασία Transit System, χρησιμοποιήθηκε ευρέως από το αμερικανικό ναυτικό.

Απαιτήθηκαν αρκετές δεκαετίες, μέχρι δηλαδή τα μέσα της δεκαετίας του 1990, ώστε το σύστημα GPS να εξελιχθεί, να γίνει ιδιαίτερα ακριβές και να αρχίσει να διατίθεται για ελεύθερη χρήση από το ευρύ κοινό.

1.7.2 Λειτουργικά τμήματα



*Αναπαράσταση του αρχικού σχεδίου του συστήματος GPS, με 24 δορυφόρους GPS
(4 δορυφόροι σε καθεμία από τις 6 τροχιές).*

Οι αριθμοί δείχνουν την εξέλιξη του αριθμού των ορατών δορυφόρων από ένα σημείο (45° Βόρεια) που δείχνει το βέλος.

Ο ρυθμός χρόνου της αναπαράστασης, είναι 2.880 φορές ταχύτερος από τον πραγματικό ρυθμό χρόνου (κάθε μισό λεπτό αντιπροσωπεύει 24 ώρες).

Το σύστημα εντοπισμού θέσης GPS σχηματίζει ένα παγκόσμιο δίκτυο, με εμβέλεια που καλύπτει ξηρά, θάλασσα και αέρα.

Εξαιτίας αυτής της έκτασής του, είναι απαραίτητος ο διαχωρισμός του σε επιμέρους τμήματα όπου πραγματοποιούνται όλες οι λειτουργίες του αλλά και ο συντονισμός του.

Αναλυτικά, τα τμήματα αυτά είναι:

- **Διαστημικό τμήμα:** Αποτελείται από το δίκτυο των 24 - 32 δορυφόρων που ήδη αναφέραμε. Οι δορυφόροι αυτοί «σκεπάζουν» ομοιόμορφα με το σήμα τους ολόκληρο τον πλανήτη, γεγονός που αποδεικνύει τη φιλοσοφία που κρύβεται πίσω από τη λειτουργία του συστήματος GPS, δηλαδή τη διαθεσιμότητά του σε κάθε σημείο της Γης, ώστε να μην υπάρχει κίνδυνος να αποπροσανατολιστεί κανείς ποτέ και πουθενά.

Όλοι οι δορυφόροι βρίσκονται σε ύψος 12.552 μιλίων (20.200 χιλιομέτρων) πάνω από την επιφάνεια της θάλασσας και εκτελούν δύο περιστροφές γύρω από τη Γη κάθε 24ωρο.

Η κατασκευάστρια εταιρεία είναι η Rockwell International, η εκτόξευσή τους πραγματοποιήθηκε από το ακρωτήριο Canaveral, ενώ η τροφοδοσία τους με ηλεκτρική ενέργεια πραγματοποιείται μέσω των ηλιακών στοιχείων που διαθέτουν.

- **Επίγειο τμήμα ελέγχου:** Οι δορυφόροι, όπως είναι αναμενόμενο, είναι πολύ πιθανό να αντιμετωπίσουν ανά πάσα στιγμή προβλήματα στη σωστή λειτουργία τους. Οι έλεγχοι που πραγματοποιούνται σε αυτούς αφορούν στη σωστή τους ταχύτητα και υψόμετρο και στην κατάσταση της επάρκειάς τους σε ηλεκτρική ενέργεια. Παράλληλα, εφαρμόζονται όλες οι διορθωτικές ενέργειες που αφορούν στο σύστημα χρονομέτρησης των δορυφόρων, ώστε να αποτρέπεται η παροχή λανθασμένων πληροφοριών στους χρήστες του συστήματος. Το τμήμα επίγειου ελέγχου αποτελείται από ένα επανδρωμένο και τέσσερα μη επανδρωμένα κέντρα, εγκατεστημένα σε ισάριθμες περιοχές του πλανήτη.

Οι περιοχές αυτές είναι οι εξής:

α)Κολοράντο (ΗΠΑ)

β) Χαβάη (Ανατολικός Ειρηνικός Ωκεανός)

γ) Ascension Island (Ατλαντικός Ωκεανός)

δ) Diego Garcia (Ινδικός Ωκεανός)

ε) Kwajalein (Δυτικός Ειρηνικός Ωκεανός)

Ο κυριότερος σταθμός βάσης είναι αυτός του Κολοράντο, ο οποίος είναι μάλιστα και ο μοναδικός που βρίσκεται στην ξηρά.

Αναλαμβάνει τον έλεγχο της σωστής λειτουργίας των εναπομεινάντων τεσσάρων σταθμών, καθώς και τον συντονισμό τους.

Σημειώνοντας τη θέση των σταθμών αυτών πάνω σε έναν παγκόσμιο χάρτη, παρατηρεί κανείς ότι η διάταξή τους δεν είναι τυχαία, αλλά ακολουθούν μια γραμμή παράλληλη με τα γεωγραφικά μήκη της Γης.

- **Το τμήμα τελικού χρήστη:** Απαρτίζεται από τους χιλιάδες χρήστες δεκτών GPS ανά την υφήλιο. Οι δέκτες αυτοί μπορούν να χρησιμοποιηθούν τόσο κατά τη διάρκεια μιας απλής πεζοπορίας, όσο και σε οχήματα ή θαλάσσια σκάφη και κατά κανόνα διαθέτουν αρκετά μικρές διαστάσεις. Για να προσφέρουν όσο το δυνατόν περισσότερες πληροφορίες, οι δέκτες συνδυάζονται με ειδικό λογισμικό, που προβάλλει ένα χάρτη στην οθόνη της συσκευής GPS. Πρόκειται, δηλαδή, για λογισμικό που λαμβάνει από τους δορυφόρους τις πληροφορίες για το στίγμα του σημείου στο οποίο βρίσκεται ο δέκτης και τις μετατρέπει σε κατανοητή «ανθρώπινη» μορφή, πληροφορώντας το χρήστη για την ακριβή γεωγραφική του θέση.

1.7.3 Φορητές συσκευές GPS

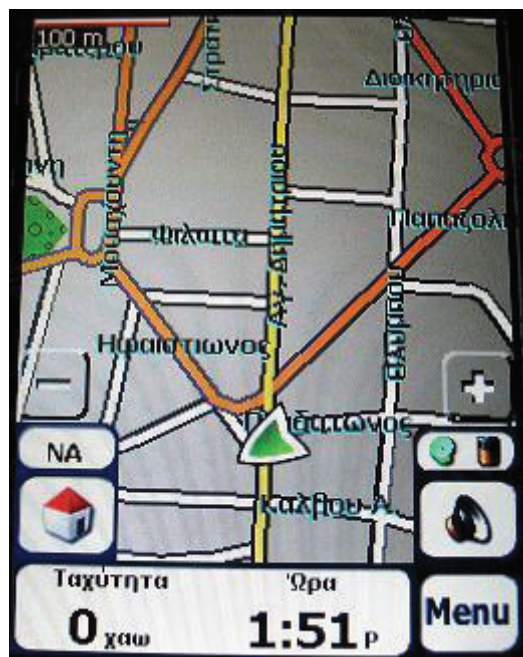


GPS συσκευή πλοήγησης Navigon, σε ταμπλό αυτοκινήτου

Η μεγάλη εξάπλωση της χρήσης του GPS οφείλεται και στη διάδοση των, οικονομικά προσιτών, φορητών δεκτών GPS για πεζούς ή οχήματα και των γενικών υπολογιστικών συσκευών (όπως τα PDA) με ενσωματωμένο δέκτη GPS. Ένας φορητός δέκτης αποτελείται από:

- Την εσωτερική δορυφορική κεραία, η οποία λαμβάνει το σήμα GPS από τους δορυφόρους με τους οποίους έχει οπτική επαφή. Επίσης, λαμβάνει σήμα και από ανακλάσεις, π.χ. σε τοίχους, κάνοντας δυνατή τη λήψη σε δρόμους που περιβάλλονται από πολύ ψηλά κτήρια (στην καθιερωμένη αγγλόφωνη σχετική ορολογία, οι συνθήκες αυτές αποκαλούνται "urban canyon") ή ακόμη και σε κάποιους εσωτερικούς χώρους. Πάντως, αρκετοί δέκτες διαθέτουν υποδοχή για εξωτερική κεραία. Οι εξωτερικές δορυφορικές κεραίες διαθέτουν πάντα προενισχυτή και δίνουν καλύτερη λήψη, λόγω της δυνατότητας τοποθέτησης πάνω από πιθανά εμπόδια (π.χ. στην οροφή του αυτοκινήτου) και της ενίσχυσης που διαθέτουν (στις εσωτερικές κεραίες δεν μπορεί να χρησιμοποιηθεί επιπλέον στάδιο προενίσχυσης, καθώς αυτό θα οδηγούσε σε ανεπιθύμητη ανάδραση, λόγω της γειτνίασης με το αναλογικό τμήμα του δέκτη).
- Τον κυρίως δέκτη GPS ο οποίος χρησιμοποιεί κυκλώματα εξαιρετικά χαμηλού θορύβου και ειδικές τεχνικές επεξεργασίας σήματος ώστε να ξεχωρίζει τα εξαιρετικά ασθενή σήματα από τους δορυφόρους, από τον ισχυρό τηλεπικοινωνιακό θόρυβο ο οποίος

έχει τη μορφή τυχαίου σήματος. Ο κυρίως δέκτης αποτελείται από το αναλογικό τμήμα εισόδου και το ψηφιακό, το οποίο περιέχει σύνθετο ψηφιακό υλικό (hardware), συνήθως κάποιο εξειδικευμένο ολοκληρωμένο κύκλωμα τύπου ASIC και μικροελεγκτή (microcontroller) χαμηλής κατανάλωσης ισχύος. Αυτό το hardware χρησιμοποιεί λογισμικό με πολύ εξελιγμένους αλγορίθμους επεξεργασίας, για να μπορέσει να εξάγει χρήσιμο στίγμα σε συνθήκες urban canyon ή δύσκολης λήψης εν γένει. Σε τέτοιες περιπτώσεις, η ακρίβεια λήψης, λόγω των πολλαπλών σημάτων, τα οποία λαμβάνει η κεραία από τον ίδιο δορυφόρο, με χρονική καθυστέρηση μεταξύ τους (φαινόμενο ηχούς), μπορεί να υποβαθμίσει σημαντικά την ακρίβεια θέσης. Το αποτέλεσμα εξαρτάται έντονα από την ποιότητα των αλγορίθμων και βελτιώνεται σημαντικά από την μία γενιά δεκτών στην επόμενη.



Ενδείξεις συσκευής πλοήγησης GPS.

Η τελική έξοδος του δέκτη είναι το στίγμα (θέση) του και η ακριβής παγκόσμια ώρα UMT.

Αυτά τα δύο δεδομένα, μαζί με άλλες χρήσιμες πληροφορίες όπως ο αριθμός των λαμβανόμενων δορυφορικών σημάτων και η στάθμη τους,

αποστέλλονται σε μια θύρα επικοινωνίας του δέκτη, συνήθως σειριακής μορφής, δηλαδή ασύγχρονη (UART) ή σύγχρονη (π.χ. SPI).

Ο ρυθμός με τον οποίο βγαίνει νέο στίγμα στην έξοδο του δέκτη είναι συνήθως 1 φορά το δευτερόλεπτο (δηλαδή 1 Hz), αν και υπάρχουν δέκτες που μπορούν να δίνουν στίγμα με ταχύτερους ρυθμούς (π.χ. 10 Hz).

Σε συσκευές που λειτουργούν με μπαταρία, ο κυρίως δέκτης GPS διαθέτει και καταστάσεις λειτουργίας όπου ο ρυθμός αποστολής στίγματος μειώνεται σημαντικά, με αποτέλεσμα την εξοικονόμηση ισχύος.

- Τον κυρίως μικροελεγκτή, την οθόνη απεικόνισης (συνήθως υγρών κρυστάλλων) και το υπόλοιπο hardware επικοινωνία με το χρήστη της συσκευής. Ο μικροελεγκτής αυτός, μέσω του ενσωματωμένου λογισμικού του, επεξεργάζεται το στίγμα που λαμβάνει από τον κυρίως δέκτη GPS, μέσω της αντίστοιχης σειριακής του θύρας. Το αποτέλεσμα της επεξεργασίας είναι μια πιο κατανοητή για τον άνθρωπο μορφή του στίγματος, και συνήθως εμφανίζεται σε οθόνη με δυνατότητες γραφικών, πάνω σε ψηφιακό χάρτη, μαζί με άλλες πληροφορίες όπως ώρα, υψόμετρο και ταχύτητα κίνησης. Η ακρίβεια του ενσωματωμένου χάρτη μπορεί να είναι αρκετά μεγάλη, στα ακριβότερα μοντέλα, ενώ συχνά υπάρχει η δυνατότητα αναβάθμισης ή επαύξησής του μέσω σύνδεσης με προσωπικό υπολογιστή (PC).



GPS δορυφορική πλοήγηση, σε smartphone, πάνω σε ποδήλατο

Οι ισχυροί μικροελεγκτές και η μεγάλη μνήμη των σύγχρονων φορητών δεκτών έχουν κάνει δυνατή την ύπαρξη διάφορων απλών και εξελιγμένων βοηθημάτων εύρεσης θέσης και πλοήγησης.

Π.χ. μπορούμε να βλέπουμε τη διαδρομή που έχουμε ήδη κάνει, να κάνουμε μεγέθυνση πάνω στο χάρτη ή να εισάγουμε προορισμό και ο δέκτης να βρίσκει τη βέλτιστη διαδρομή (λειτουργία πλοήγησης).

Σχεδόν πάντα υπάρχει η δυνατότητα ορισμού σημείων στο χάρτη ως προτιμώμενων ή ακόμη και κατάλογος με σημεία ενδιαφέροντος, όπως πρατήρια βενζίνης, καταστήματα και αξιοθέατα.

Στα μοντέλα για αυτοκίνητο συνήθως υπάρχει η δυνατότητα φωνητικών οδηγιών, κατά τη λειτουργία πλοήγησης, ώστε ο οδηγός να μη χρειάζεται να κοιτά την οθόνη.

Επίσης, κυκλώματα δεκτών GPS αρχίζουν να ενσωματώνονται και σε κινητά τηλέφωνα και άλλες συσκευές, όπως ψηφιακές φωτογραφικές μηχανές.

Στο κοντινό μέλλον οι φορητοί δέκτες GPS θα βρίσκουν μεγάλη εφαρμογή και στα άτομα με αναπηρία, όπως οι τυφλοί, οι οποίοι έχουν την δυνατότητα να ζητούν προορισμό και να ακούν φωνητικές οδηγίες από το δέκτη GPS.

Κεφάλαιο 2^ο : Παρόμοιες εφαρμογές χρήσης θέσης και εντοπισμού

2.1 Εισαγωγή

Όπως έχουμε αναφέρει και νωρίτερα, αυτή η εργασία έχει σαν σκοπό τον προσδιορισμό της θέσης ενός ατόμου, την δυνατότητα να στιγματίσει το σημείο σαν σημείο εντοπισμού, αλλά και με το δικό του προσωπικό μήνυμα να βοηθήσει τους υπόλοιπους χρήστες στην μετακίνηση ή στην επιλογή του συγκεκριμένου σημείου.

Πάνω σε αυτήν την σκέψη στηρίζονται πολλές εφαρμογές, τις οποίες όλοι γνωρίζουμε ή έχουμε ακούσει.

Σε αυτήν, όμως την εργασία θα ασχοληθούμε με τις δύο πιο ιδανικές εφαρμογές, κατά την δική μου άποψη, το GoogleLatitude και το Foursquare.

Στην συνέχεια του κεφαλαίου θα αναφέρουμε και θα αναλύσουμε κάθε εφαρμογή ξεχωριστά, με σκοπό να την γνωρίσουμε καλύτερα, αλλά και να κατανοήσουμε ακριβέστατα τον τρόπο λειτουργίας τους, καθώς και το αποτέλεσμα τους.

2.2 Τι είναι το GoogleLatitude



Αρκετοί είναι αυτοί που ξεκινούν μία τηλεφωνική συνδιάλεξη ζητώντας από το συνομιλητή τους να προσδιορίσει τον τόπο στον οποίο βρίσκεται.

Την απάντηση στο κλασικό αυτό ερώτημα, το οποίο απασχολεί τους περισσότερους χρήστες κινητών τηλεφώνων, έρχεται να δώσει το Google Latitude, ένα νέο χαρακτηριστικό της «mobile» έκδοσης της εφαρμογής Google Maps.

Πρακτικά, το Google Latitude θα σας δώσει τη δυνατότητα να ενημερώνεστε για τη γεωγραφική θέση των φίλων και των αγαπημένων σας προσώπων, εφόσον αυτά έχουν επιλέξει να την κοινοποιήσουν προς εσάς.

Με αυτόν τον τρόπο, θα μπορείτε για παράδειγμα να γνωρίζετε αν πράγματι το έτερον σας ήμισυ έχει «κολλήσει» στην κίνηση επιστρέφοντας στο σπίτι από τη δουλειά, αν ένας φίλος σας έχει επισκεφθεί για το Σαββατοκύριακο την πόλη που βρίσκεστε και εσείς ή αν ένα αγαπημένο σας πρόσωπο έφθασε με ασφάλεια στον προορισμό του παρά τις αντίξοες καιρικές συνθήκες.

Φυσικά, το Google Latitude συνεργάζεται άριστα και με τις υπόλοιπες υπηρεσίες της Google και κατά συνέπεια, μπορείτε να επικοινωνήσετε με τους φίλους σας μέσω του Google Talk ή του Gmail.

Επιπρόσθετα, οι χρήστες του Latitude μπορούν να ενημερώσουν το μήνυμα της «τρέχουσας κατάστασής» τους ή να ανεβάσουν μία νέα «εικόνα προφίλ».

Για την ώρα, το Google Latitude παρέχεται σε 27 διαφορετικές χώρες.

Στην επίσημη λίστα δεν περιλαμβάνεται και η Ελλάδα, χωρίς αυτό φυσικά να σημαίνει πως είναι αδύνατη η χρήση του εδώ (άλλωστε, υποστηρίζονται τα ελληνικά).

Φυσικά μία τέτοια δυνατότητα δεν θα μπορούσε να γίνει αποδεκτή με ενθουσιασμό, αν δεν παρείχε όλες τις απαραίτητες επιλογές για την προστασία των «ευαίσθητων» προσωπικών δεδομένων, που σχετίζονται με τη γεωγραφική θέση.

Όπως υπογραμμίζει στην ανακοίνωσή της η Google, καμία πληροφορία δεν αποστέλλεται χωρίς προηγούμενος να υπάρξει η συγκατάθεση του χρήστη.

Ο τελευταίος αποφασίζει για το ποιοι ακριβώς θα είναι αυτοί που θα έχουν τη δυνατότητα να ενημερωθούν για τη γεωγραφική του θέση, ενώ επιπρόσθετα μπορεί να επιλέξει και ποια γεωγραφική θέση θα είναι αυτή.

Έτσι, ακόμη και αν βρίσκεστε στη Ρώμη, μπορείτε προαιρετικά να αποκρύψετε την πραγματική σας θέση και να δηλώσετε οποιαδήποτε άλλη (π.χ. Πεκίνο ή Καταρράκτες του Νιαγάρα).

Εναλλακτικά μπορείτε να αποκρύψετε εντελώς τη θέση σας ώστε να μην μεταδίδεται καμία αληθινή ή ψευδής πληροφορία.

Οι επιλογές αυτές μπορούν να καθοριστούν ξεχωριστά για το καθένα από τα άτομα που βρίσκονται στη λίστα των «επαφών» σας.

Το [Google Latitude](#) διατίθεται ήδη στην νεότερη έκδοση του Google Maps για συσκευές BlackBerry, S60 και Windows Mobile.

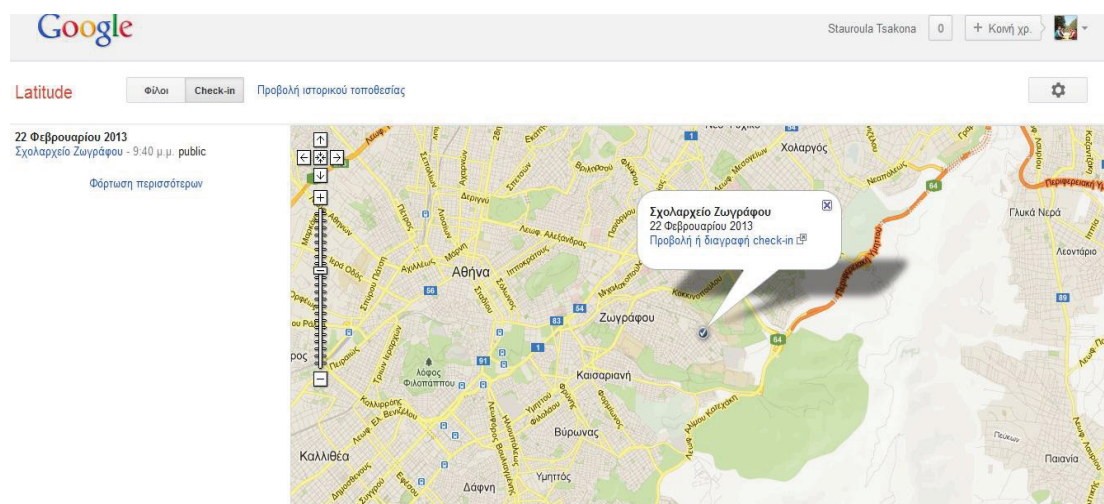
Τις επόμενες ημέρες αναμένεται να κυκλοφορήσει και για το Android (υπάρχει ήδη μία πρώτη ανεπίσημη κυκλοφορία), ενώ στο εγγύς μέλλον θα ακολουθήσει και η έκδοση για το iPhone (μέσω της εφαρμογής [Google Mobile](#)).

Οι χρήστες υπολογιστών τέλος, μπορούν να γευθούν ορισμένες από τις δυνατότητες του Google Latitude μέσω ενός gadget για το iGoogle - το οποίο, σύμφωνα με τη Google, «λειτουργεί» μόνο στις ΗΠΑ.

Για να ξεκινήσετε με το Latitude δεν έχετε παρά να ακολουθήσετε τα παρακάτω βήματα :

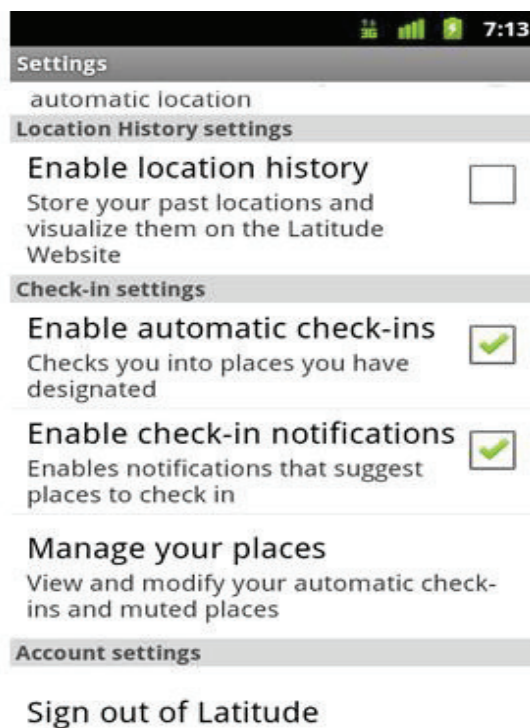
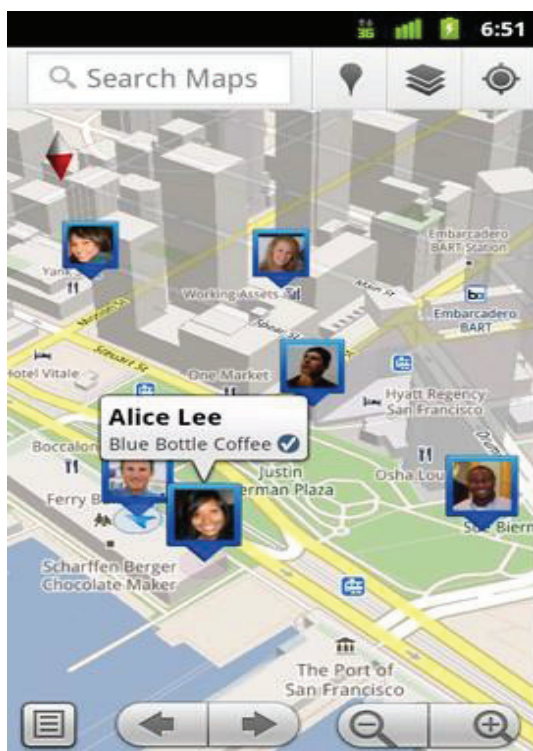
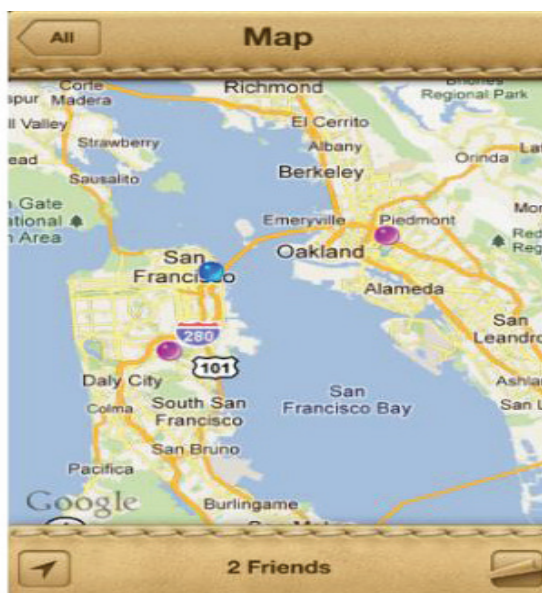
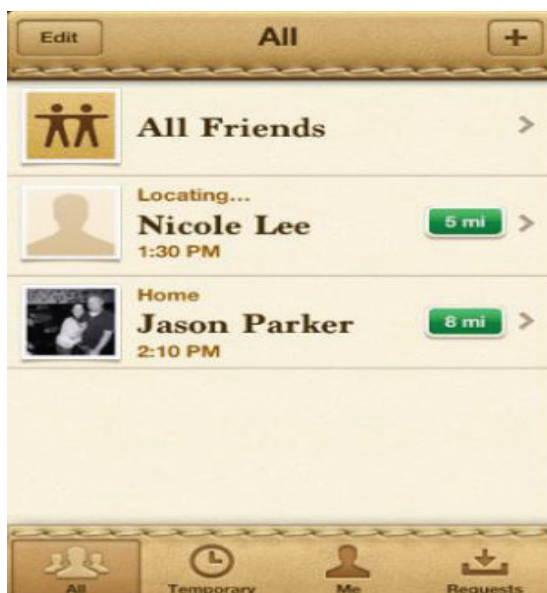
1. Ανοίξτε το Google Maps για κινητά και μεταβείτε στην προβολή χάρτη.
2. Αγγίξτε το κουμπί μενού της συσκευής σας και επιλέξτε Συμμετοχή στο Latitude .
3. Σημειώστε ότι αν θέλετε να κάνετε check in σε περιοχές, θα πρέπει να δημιουργήσετε ένα προφίλ Google+. Αν θέλετε να εγγραφείτε στο Latitude με ένα διαφορετικό λογαριασμό, πατήστε Εναλλαγή λογαριασμών από το μενού Maps.
4. Προσθήκη φίλων από τις Επαφές Google ή χρησιμοποιώντας διευθύνσεις ηλεκτρονικού ταχυδρομείου τους.
5. Επιλέξτε τις ρυθμίσεις απορρήτου σας.
6. Ξεκινήστε την κοινή χρήση και ενημέρωση της τοποθεσίας σας από το τηλέφωνό σας

Ενδεικτικά μία φωτογραφία από τον προσωπικό μου λογαριασμό στο Google Latitude, μέσω του ιστοτόπου της Google.



Στιγμιότυπο χρήσης της εφαρμογής στο διαδικτυακό τόπο

Άλλες δύο φωτογραφίες, από το πώς φαίνεται η συγκεκριμένη εφαρμογή, μέσα από μία κινητή συσκευή χρηστών της Αμερικής :



Παραδείγματα χρήσης της εφαρμογής μέσα από κινητές συσκευές

Για περισσότερες πληροφορίες για την εφαρμογή στο επίσημο site της :

<http://googleblog.blogspot.gr/2009/02/see-where-your-friends-are-with-google.html>

2.3 Τι είναι το Foursquare



Το Foursquare εξελίσσεται με ρυθμούς γεωμετρικής προόδου καθημερινά και σιγά – σιγά εισβάλλει στην καθημερινότητα μας και στο κινητό μας τηλέφωνο.

Ολοένα και περισσότερους fans αρχίζει και κερδίζει το foursquare καθημερινά σε όλο τον κόσμο, το τελευταίο διάστημα και στην χώρα μας.

Τι είδους υπηρεσία είναι όμως το Foursquare?

Πρόκειται για μια απο τις μεγαλύτερες υπηρεσίες κινητών τηλεφώνων και Internet που κυριαρχεί αυτή τη στιγμή στις ΗΠΑ και ειδικά το 2011 έχει αποκτήσει εκατομμύρια θαυμαστές στην Ελλάδα.

Το Foursquare δεν είναι τίποτα άλλο απο ένα location based social network που δίνει την δυνατότητα να κάνετε check in στις τοποθεσίες που επισκέπτεστε (απο καφετέριες και εστιατόρια εως και μουσεία), να αφήνετε τα θετικά ή τα αρνητικά σας σχόλια, να συγκεντρώνεται πόντους, να βλέπετε τι κάνουν οι φίλοι σας, να ανακαλύπτετε νέες τοποθεσίες και νέους τρόπους διασκέδασης.

Ο χρήστης με τα περισσότερα check in σε μια συγκεκριμένη τοποθεσία που επισκέπτεται συνέχεια θα αποτελέσει και τον δήμαρχο της

τοποθεσίας (mayor) και έτσι υπάρχει ένας άτυπος ανταγωνισμός για το ποιος θα γίνει ο ηγέτης του κάθε σημείου.

Η υπηρεσία δημιουργήθηκε το 2009 από τον Dennis Crowley και τον Naveen Selvadurai.

Ο Crowley είχε ιδρύσει παλαιότερα το Dodgeball παρόμοια εφαρμογή, ως εργασία/διατριβή της αποφοίτησης του από το Interactive Telecommunications Program (ITP) στο Πανεπιστήμιο της Νέας Υόρκης.

Η Google αγόρασε την Dodgeball το 2005 και την σταμάτησε το 2009, αντικαθιστώντας την με το Google Latitude.

Οι χρήστες της Dodgeball εφαρμογής βασίζονταν στην τεχνολογία των SMS, παρά στην καθ' εαυτού εφαρμογή.

Το Foursquare είναι η δεύτερη επανάληψη της ίδιας ιδέας, ότι οι άνθρωποι μπορούν να χρησιμοποιήσουν τις κινητές συσκευές για να επικοινωνούν με το περιβάλλον τους.

Από τον Απρίλιο του 2012, η εταιρεία ανέφερε ότι είχε 20 εκατομμύρια εγγεγραμμένους χρήστες.

Η εταιρεία αναμένεται να περάσει τα 750 εκατομμύρια check-in πριν από το τέλος του Ιουνίου του 2011, με μέσο όρο περίπου 3 εκατομμύρια check-in ανά ημέρα.

Άντρες και γυναίκες χρήστες εκπροσωπούν ισότιμα την εφαρμογή και επίσης το 50% των χρηστών είναι εκτός των ΗΠΑ.

Η υποστήριξη για Γαλλικά, Ιταλικά, Γερμανικά, Ισπανικά, Ιαπωνικά και προστέθηκαν τον Φεβρουάριο του 2011.

Υποστήριξη για το Κορεάτικα, Πορτογαλικά, Ρωσικά, και της Ταϊλάνδης προστέθηκαν τον Σεπτέμβριο του 2011.

Η υποστήριξη για την Τουρκική γλώσσα προστέθηκε τον Ιούνιο του 2012.

2.3.1 The social

Το Foursquare είναι κοινωνικό δίκτυο στο οποίο μπορούμε να κάνουμε αίτημα φιλίας σε άλλα άτομα και να ενημερωνόμαστε όχι για τα status τους αλλά για το μέρος που βρίσκονται κάθε φορά σε περίπτωση φυσικά που αποδεχτούν το αίτημα μας και κάνουν check in.



Παρουσίαση αρμοδιοτήτων εφαρμογής

2.3.2 Το παιχνίδι

Το Foursquare έχει και την ψυχαγωγική του πλευρά, εκτός από τις χρήσιμες πληροφορίες που μπορεί να μας δώσει γύρω από το σημείο στο οποίο βρισκόμαστε.

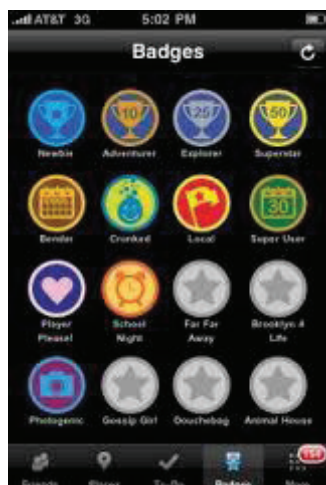
Ανάλογα με τα σημεία που κάνουμε check in κερδίζουμε πόντους και βραβεία.

Όσο περισσότερους πόντους συλλέγουμε τόσο πιο ψηλά φτάνουμε στον βαθμολογικό πίνακα σε έναν άτυπο διαγωνισμό με τους φίλους μας.

Αν σε ένα σημείο βρισκόμαστε πολλές φορές γινόμαστε mayor.

Στο εξωτερικό όταν ένας χρήστης γίνει δήμαρχος κάποιας τοποθεσίας έχει πολλά πλεονεκτήματα όπως για παράδειγμα σε διάφορα μπαρ τα ποτά του είναι δωρεάν.

Ενδεικτικά οι παρακάτω δύο φωτογραφίες απεικονίζουν τα βραβεία, γνωστά ως Badges, τα οποία κερδίζει ο χρήστης κάθε φορά που κάνει και περισσότερα checkin, καθώς και το σύμβολο που αντιστοιχεί σε κάθε mayor μίας τοποθεσίας



Τα “βραβεία” που αποκτά κάθε χρήστης



Το σύμβολο του mayor

2.3.3 Το φλερτ και οι νέες γνωριμίες

Μία άλλη πολύ σημαντική δυνατότητα της συγκεκριμένης εφαρμογής είναι να μπορούμε να βλέπουμε ποια άτομα είναι σε κάποιο σημείο την κάθε χρονική στιγμή και χωρίς να είναι φίλοι μας.

Αν για παράδειγμα βρεθούμε σε ένα μπαρ, το foursquare θα μας ενημερώσει για το ποιά άτομα είναι εκείνη την ώρα γύρω μας.(εφόσον φυσικά είναι χρήστες)

Αφήνοντας τα σχόλια μας κάτω απο την τοποθεσία μπορούμε να επικοινωνήσουμε με αυτούς τους χρήστες γιατί όχι και να γνωριστούμε.

2.3.4 Τα τυχερά

Σίγουρα δεν ήταν λίγες οι φορές που το κινητό σας βρήκε κάποιο κλειδωμένο ασύρματο δίκτυο κάποιας καφετέριας ή κάποιου ξενοδοχείου αλλά εσείς δεν είσασταν εκείνη την στιγμή πελάτης και δεν ξέρατε τον κωδικό.

Το Foursquare είναι ΕΔΩ!

Σχεδόν όλοι οι χρήστες θέλοντας να προσφέρουν στην κοινότητα γράφουν τα κλειδιά από τα ασύρματα δίκτυα των διάφορων σημείων στα οποία βρίσκονται.

Μία γρήγορη αναζήτηση του συγκεκριμένου σημείου είναι αρκετή για να δείτε το wifi password.



Σύμβολο δυνατότητας εύρεσης κωδικού ασυρμάτου

2.3.5 Marketing

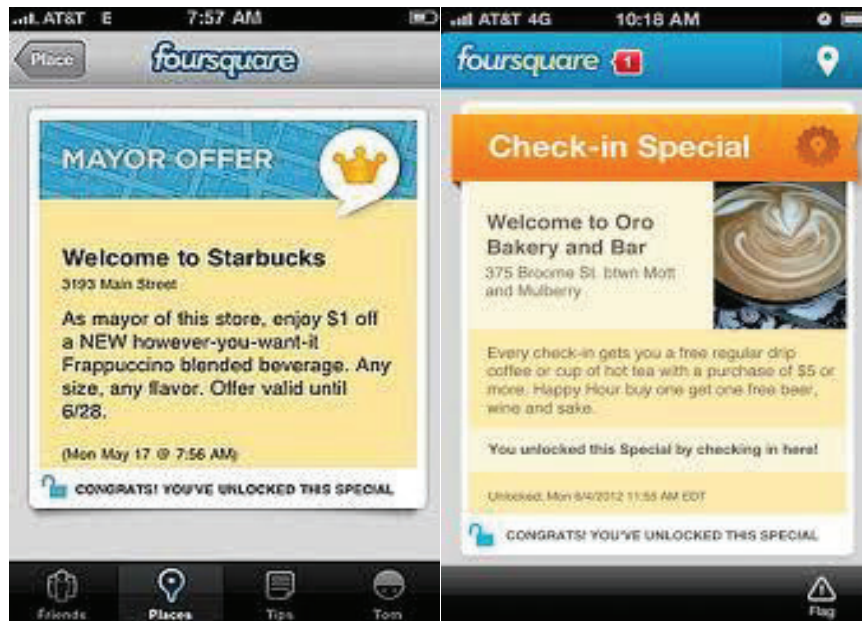
Πολλές είναι οι εταιρίες που επιλέγουν το Foursquare ως εργαλείο marketing.

Στην Ελλάδα ακόμα τα πράγματα είναι σε πρώιμο στάδιο αλλά σίγουρα σε λίγο καιρό πελάτες και πωλητές θα είναι κερδισμένοι από την χρήση της εφαρμογής.

Πολλές επιχειρήσεις λοιπόν(κυρίως στο εξωτερικό) δίνουν την δυνατότητα στους χρήστες που κάνουν check in στην επιχείρησή τους να έχουν κάποια πλεονεκτήματα και κάποιες προσφορές.(αυτές οι τοποθεσίες χαρακτηρίζονται ως special).

Γνωστό μπαρ στην Θεσσαλονίκη κερνάει στο πρώτο check in σφηνάκια σε όλη την παρέα και στο έβδομο check in shaker θέλοντας με αυτόν τον τρόπο να προσελκύσει νέους πελάτες.

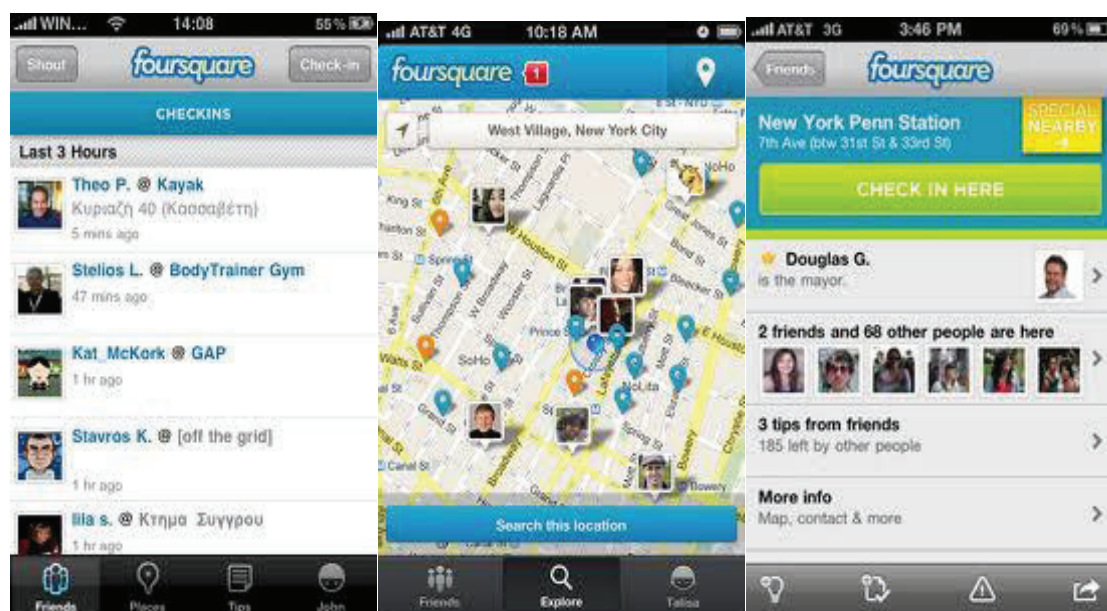
Παραδείγματα προσφορών στις εικόνες που ακολουθούν.



Προσφορά για τον mayorΔώρα-Εκπλήξεις για τους χρήστες

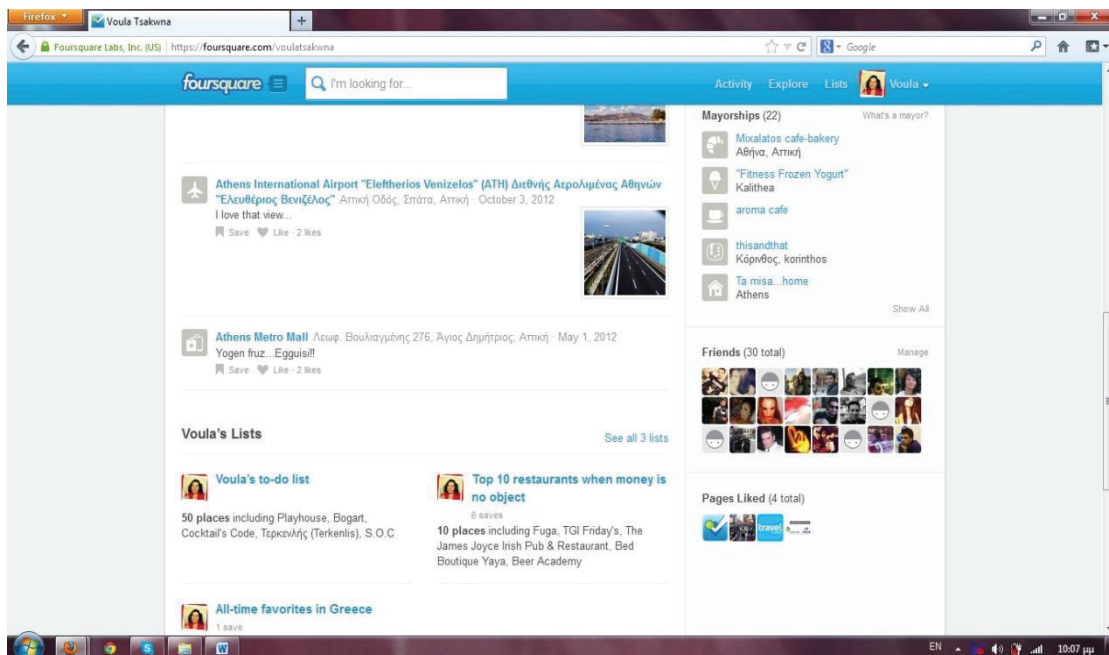
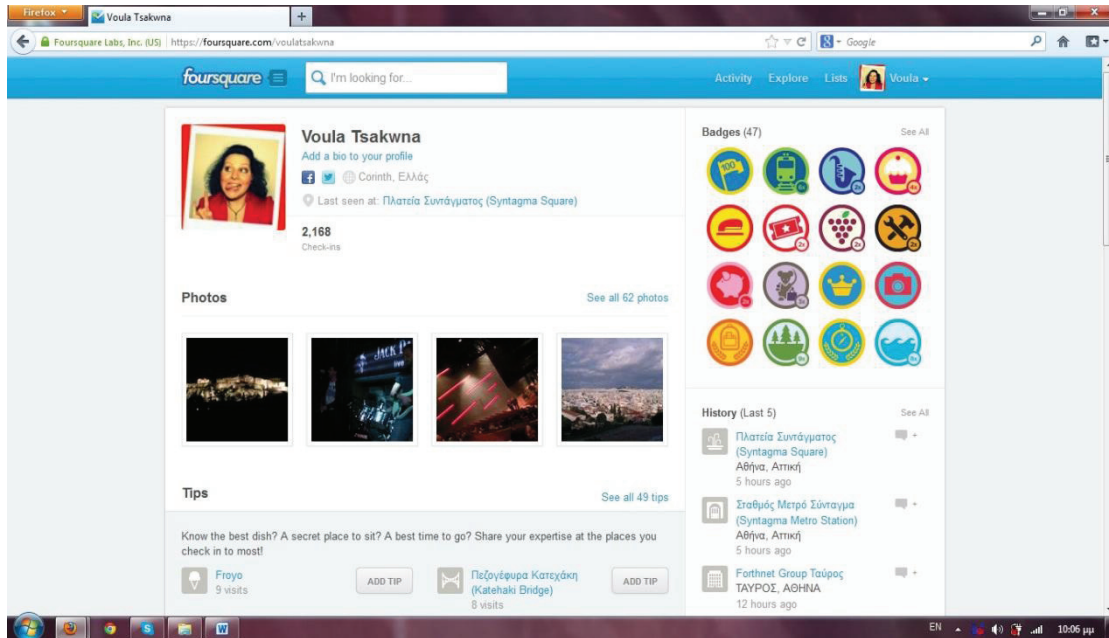
Κάτι τελευταίο που θα δείξουμε μέσω εικόνων είναι το πώς φαίνεται η συγκεκριμένη εφαρμογή μέσα από μία κινητή συσκευή και πώς μέσα από τις σελίδες του διαδικτύου.

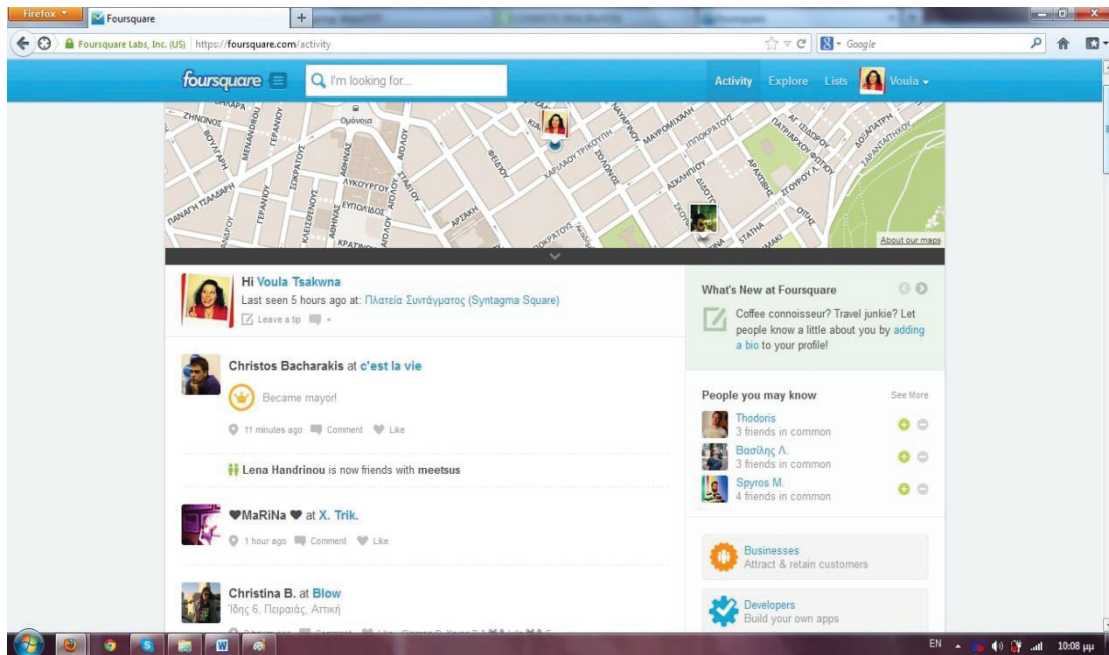
Μέσω μίας κινητής συσκευής, λοιπόν έχουμε σαν εικόνα το εξής :



Παραδείγματα μέσω κινητής συσκευής

Σε αντίθετη περίπτωση, μέσα από τις σελίδες του διαδικτυακού τόπου, παρουσιάζεται με αυτή τη μορφή :





Παραδείγματα χρήσης μέσω διαδικτυακού τόπου

Για περισσότερες πληροφορίες περί της εφαρμογής, επισκεπτόμαστε την επίσημη της σελίδα :

<https://foursquare.com/>

Κεφάλαιο 3^ο : Ανάλυση απαιτήσεων εφαρμογής

3.1 Κύκλος ανάπτυξης εφαρμογής

Η ανάπτυξη εφαρμογών στο Android είναι μια σύνθετη και χρονοβόρα διαδικασία η οποία συνοψίζεται σε 4 βασικά στάδια, αλλά και αρκετά επί μέρους, τα οποία θα σχολιαστούν μεταξύ των βασικών.

3.1.1 Εγκατάσταση λογισμικού

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείτε να στήσει το περιβάλλον εργασίας στο οποίο θα γίνει ο σχεδιασμός, η ανάπτυξη, ο έλεγχος, και η λειτουργία των εφαρμογών.

Μπορεί να επιλέξει όποιο περιβάλλον ανάπτυξης (IDE) τον εξυπηρετεί καλύτερα και να χρησιμοποιήσει όλα τα εργαλεία του Android SDK μηδενός εξαιρουμένου.

Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει την λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας.

Ιδανικά ο developer θα διαθέτει έναν αριθμό διαφορετικών φυσικών συσκευών ώστε να δοκιμάσει ο ίδιος πως συμπεριφέρεται η εφαρμογή του σε κάθε περίπτωση, όμως αυτή η πρακτική μπορεί να αποδειχθεί πολυδάπανη και χρονοβόρα.

Εδώ αναλαμβάνουν δράση η ευελιξία των AVDs.

3.1.2 Ανάπτυξη πηγαίου κώδικα εφαρμογής

Πρόκειται αν μη τι άλλο για τη πιο χρονοβόρα και πολύπλοκη διαδικασία.

Σε αυτό το στάδιο ο προγραμματιστής πρέπει να αποφασίσει για τις δυνατότητες και το περιεχόμενο που θα περιλαμβάνει η εφαρμογή, να εντοπίσει ποιες από αυτές τις δυνατότητες είναι εφικτές και ποιες θέλουν παραπάνω έρευνα για να προστεθούν στο μέλλον, να σχεδιάσει το layout με γνώμονα την λειτουργικότητα και να αποφύγει υπερβολές στο σχεδιασμό, και τέλος να δέσει αρμονικά τον κώδικα με το layout για να φέρει το τελικό αποτέλεσμα.

Η διαδικασία ξεκινάει με ένα νέο Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά ότι χρειάζεται η εφαρμογή για να τρέξει ως οφείλει.

Στο project του ο developer θα πρέπει να φροντίσει ώστε το υλικό του να είναι τακτοποιημένο και ο κώδικας του ευανάγνωστος ώστε να ακολουθήσει η διαδικασία του Debugging.

3.1.3 Αποσφαλμάτωση (Debugging) και δοκιμαστική φάση εφαρμογής

Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσουχρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής.

Αποτελείτε από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω.

Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode.

Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης (IDE) προϋποθέτουν ότι ο κώδικας δεν έχει κανένα συντακτικό λάθος, αλλιώς ειδοποιούν τον χρήστη να τα διορθώσει.

Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB push εντολής.

Στο δεύτερο στάδιο ο προγραμματιστής καλείτε να αντιμετωπίσει τα λειτουργικά και αισθητικά προβλήματα της εφαρμογής του, πρώτα εντοπίζοντας τα στην λειτουργία της συσκευής και μετά διορθώνοντας τα κομμάτια του κώδικα που δημιουργούν τα σφάλματα.

Το κύριο εργαλείο που κάνει αυτή τη διαδικασία εφικτή είναι το “LogCat” το οποίο μας επιστρέφει το stack trace του κώδικα στο σημείο εκείνο που συνέβη το σφάλμα.

Υπάρχουν φυσικά και άλλα εργαλεία που μπορούμε να βρούμε και να χειριστούμε.

Στο τρίτο στάδιο ο προγραμματιστής αφού έχει τελειώσει την αποσφαλμάτωση(debugging) επιστρέφει στο βήμα ένα, δηλαδή στο compile και τη δοκιμή της εφαρμογής σε εικονική η φυσική συσκευή ώστε να διαπιστώσει τα αποτελέσματα του 2 βήματος, της αποσφαλμάτωσης.

Ένα προαιρετικό στάδιο είναι η “Δημόσια δοκιμαστική φάση” της εφαρμογής.

Σε αυτή τη φάση εθελοντές προσφέρονται να δοκιμάσουν τις λειτουργίες της εφαρμογής στις συσκευές τους και να αναφέρουν προβλήματα, παρατηρήσεις, προτάσεις και άλλα σχόλια που μπορεί προκύψουν από τη χρήση της εφαρμογής.

Φυσικά η διαδικασία του debugging είναι σαν ένα βρόγχος (loop) που επαναλαμβάνεται συνέχεια μέχρι να εντοπιστούν και να διορθωθούν όλα τα σφάλματα της εφαρμογής, και για αυτό το λόγο μπορεί να αποδειχθεί εξαιρετικά χρονοβόρα.

3.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό

Στο τέταρτο και τελευταίο στάδιο της ανάπτυξης ο προγραμματιστής έχει να κάνει μερικές τελευταίες κινήσεις.

Πρώτον πρέπει να έχει διορθώσει όλα τα σφάλματα που προέκυψαν από την διαδικασία αποσφαλμάτωσης, να κάνει τις τελευταίες ρυθμίσεις και tweaks της εφαρμογής, και να κάνει το τελικό compile της εφαρμογής σε κανονικήλειτουργία αυτή τη φορά και όχι debug.

Στη συνέχεια ακολουθεί η διάθεση της εφαρμογής με το μέσο της επιλογής του developer, την οποία μπορεί να την διαθέσει στο Google Play, αφού πρώτα κάνει λογαριασμό developer.



Αρχική σελίδα προγράμματος GooglePlay των Android

, ή να την διαθέσει σε κάποιο εναλλακτικό market όπως το marketplace της Amazon



Δημιουργία λογαριασμού για πωλήσεις στο Amazon

Μπορεί κατά τη δημοσίευση σε οποιοδήποτε να ορίσει τιμή πώλησης ή να διαθέσει την εφαρμογή δωρεάν.

Τη στιγμή που γράφονται αυτές οι γραμμές δεν είναι εφικτό ακόμη για τους Έλληνες Developers να διαθέσουν εφαρμογές επί πληρωμή, με την Google να μην έχει ανακοινώσει ακόμη πότε αυτό θα γίνει εφικτό.

Επίσης άλλο ένα μέσο διάθεσης μπορεί να είναι η προσωπική η εταιρική ιστοσελίδα του δημιουργού.

Το μειονέκτημα φυσικά σε αυτή τη περίπτωση είναι η έλλειψη ελέγχου για updates της εφαρμογής από έναν αυτόματο μηχανισμό ελέγχου και λήψης όπως είναι τα διάφορα marketplaces.

3.4 AndroidSDK

Το Android SDK (Software Developers Kit) αποτελεί μια συλλογή εργαλείων και βιβλιοθηκών που καθιστούν εφικτή την ανάπτυξη εφαρμογών στο Android.

Τι στιγμή που γράφονται αυτές οι γραμμές, το SDK έχει φτάσει στην έκδοση r19 η οποία υποστηρίζει το Android 4.0.3.

Το λογισμικό ανάπτυξης λοιπόν περιλαμβάνει μια μεγάλη λίστα με εργαλεία ανάπτυξης.

Σε αυτά περιλαμβάνονται:

- Εργαλεία Debugging των εφαρμογών
- Βιβλιοθήκες
- Εξομοιωτής συσκευών (Android Virtual Machines)
- Documentation
- Δείγματα Κώδικα
- Tutorials

Το Android SDK παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη προγραμμάτων χρησιμοποιώντας την γλώσσα προγραμματισμού Java.

Τα χαρακτηριστικά του παρέχουν τεράστια ευελιξία και δυνατότητα ανάπτυξης έξυπνων εφαρμογών για κινητά τηλέφωνα, παρέχοντας δυνατότητες ανάπτυξης σε επιχειρηματίες όλων των κλάδων, είτε πρόκειται για αυτοματοποίηση πωλήσεων, ψυχαγωγία, παιχνίδια αλλά και κάθε άλλου είδους επιχειρήσεις.

Ακόμα, παρέχει APIs για την χρήση web browser, εμφάνιση διαστάσεων και τρισδιάστατων γραφικών, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού (ήχος, βίντεο, εικόνες), χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο.

Ένα σημαντικό γεγονός που ευνοεί την ανάπτυξη εφαρμογών είναι πως το πακέτο Android SDK συνεργάζεται με το Eclipse και συνεπώς ο προγραμματιστής μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, χωρίς να χρειάζεται να εξάγει κάθε φορά την εφαρμογή και να την εγκαθιστά σε κινητό.

Επίσης, ο emulator είναι πολύ αξιόπιστος, καθώς έχει ακριβώς την ίδια συμπεριφορά με αυτή που θα είχε η εφαρμογή εάν είχε εγκατασταθεί σε ένα κινητό τηλέφωνο Android.

Ένα άλλο σημαντικό πλεονέκτημα είναι το γεγονός πως σε αναβαθμίσεις του λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη, hardware, του κινητού τηλεφώνου.

Παρακάτω, γίνεται ανάπτυξη του εξομοιωτή (emulator).

3.4.1 Μερικά από τα χαρακτηριστικά του SDK

Το Android SDK είναι ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση των «ενδοτέρων» του Android.

Παρακάτω, φαίνονται τα βασικότερα χαρακτηριστικά του.

- Εφαρμογή πλαισίου που επιτρέπει την επαναχρησιμοποίηση και την αντικατάσταση στοιχείων
- Βελτιστοποιημένη Dalvik εικονική μηχανή για κινητές συσκευές.
- Ολοκληρωμένο πρόγραμμα περιήγησης, το οποίο βασίζεται στο open source WebKit.
- Βελτιστοποιημένα γραφικά, τα οποία τροφοδοτούνται από μια προσαρμοσμένη βιβλιοθήκη 2D και 3D γραφικών με βάση τις προδιαγραφές του OpenGL ES 1.0 (επιτάχυνση hardware προαιρετικά).
- SQLite για δομημένη αποθήκευση δεδομένων.
- Υποστήριξη πολυμέσων για αρχεία ήχου, βίντεο, ακόμα και εικόνων (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- GSM Τηλεφωνία (εξαρτώμενη από το hardware)
- Bluetooth, EDGE, 3G, WiFi (εξαρτώμενα από το hardware).
- Φωτογραφική μηχανή, GPS, πυξίδα, και επιταχυνσιόμετρο (εξαρτώμενα από το hardware)
- Πλούσιο περιβάλλον ανάπτυξης, συμπεριλαμβανομένου ενός εξομοιωτή συσκευής, εργαλεία για τον εντοπισμό σφαλμάτων, μνήμη και προφίλ απόδοσης όπως επίσης και ένα plugin για το Eclipse IDE.

3.5 Χρήση του Eclipse IDE μαζί με το ADT (Android Development Tools)

Ο προγραμματισμός στο Android βασίζεται στην γλώσσα Java και ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει κώδικα για να επεξεργαστεί τα αρχεία *.Java και *.XML και μετέπειτα να τα κάνει compile μέσω γραμμής εντολών χρησιμοποιώντας το JDK (Java Development Kit).

Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός στον χρήστη γιατί συνίσταται η χρήση ενός IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse ή το Netbeans.

Η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό το ADT plugin, το οποίο παρέχει σύνδεση με το Android SDK με όλες τις δυνατότητες που περιλαμβάνει αυτό

Επίσης το plugin παρέχει σύνδεση με τον AVD Manager, για διαχείριση και εκκίνηση από το GUI του, εικονικών συσκευών Android για δοκιμές και debugging των εφαρμογών.

Φυσικά όπως είπαμε και παραπάνω, ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει τον TextEditor ή IDE της επιλογής του για τη δημιουργία του κώδικα και μετέπειτα να χρησιμοποιήσει τα εργαλεία JDK και Apache Ant μέσω γραμμής εντολών για να κάνει compile την εφαρμογή του ώστε να την τεστάρει με όλες τις δυνατότητες που το παρέχει το Android SDK.

Η επιλογή ενός IDE που κάνει όλη την πολύπλοκη δουλειά για μας είναι προφανής λοιπόν.

Επίσης τα περισσότερα παραδείγματα και άρθρα για το Android στηρίζονται στο γεγονός ότι η πλειονότητα των developers χρησιμοποιεί το Eclipse μαζί με το ADT plugin, οπότε ξεκινάμε με αυτό σαν δεδομένο.

3.6 Ο εξομοιωτής (emulator)

Προκειμένου να γίνει ευκολότερη η διαδικασία της ανάπτυξης και αποσφαλμάτωσης μιας εφαρμογής, το Android SDK περιλαμβάνει έναν εξομοιωτή μιας εικονικής κινητής συσκευής, η οποία τρέχει το λειτουργικό του Android.

Έτσι δεν είναι η αναγκαία η ύπαρξη πραγματικής κινητής συσκευής για την εκτέλεση και δοκιμή των εφαρμογών.

Ο εξομοιωτής προσομοιώνει ένα μεγάλο πλήθος λειτουργιών μιας τυπικής συσκευής, η οποία τρέχει το Android:

- 1) Παρέχει μια ποικιλία πλήκτρων πλοήγησης και ελέγχου.
- 2) Παρέχει μια οθόνη για την προβολή των εφαρμογών που τρέχουν στον εξομοιωτή.
- 3) Επιτρέπει στις εφαρμογές την χρήση των υπηρεσιών που προσφέρει η πλατφόρμα του Android, δηλαδή την κλήση άλλων εφαρμογών, την πρόσβαση στο δίκτυο, την αναπαραγωγή ήχου και

βίντεο, την αποθήκευση και επαναφορά δεδομένων, τηνειδοποίηση χρήστη, τογραφικό περιβάλλον του Android.

Επίσης παρέχει ένα πλήθος λειτουργιών για την ευκολότερηαποσφαλμάτωση:

- Κονσόλα για την καταγραφή της εξόδου του πυρήνα
- Προσομοίωση διακοπών (όπως η άφιξη SMS μηνύματος ή τηλεφωνικής κλήσης)
- Προσομοίωση καθυστέρησης και απώλειας στο κανάλι δεδομένων
- Προσομοίωση λήψης δεδομένων θέσης από την συσκευή GPS.

3.7 Άλλα εργαλεία του Android

Το Android SDK περιλαμβάνει μερικά ακόμη εργαλεία για την ανάπτυξη εφαρμογών:

- 1) Το Dalvik Debug Monitor Service (DDMS) το οποίο επιτρέπει την διαχείριση των διεργασιώνστον εξομοιωτή ή στην συσκευή. Συγκεκριμένα δίνεται η δυνατότητα port-forwardingυπηρεσιών, λήψη screenshots, εμφάνιση πληροφοριών για τον σωρό και τα νήματα, logcat,εμφάνιση πληροφοριών ράδιο και πληροφοριώνδιεργασιών, προσομοίωση εισερχόμενωνκλήσεων και μηνυμάτων, προσομοίωση δεδομένων θέσης κ.α.
- 2) Την Android Debug Bridge (ADB) η οποία επιτρέπει την διαχείριση της κατάστασης τουεξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, ηδιαχείριση της προώθησης θυρών και η αντιγραφή από και προς την συσκευή ή τονεξομοιωτή.
- 3) Το Android Asset Packaging Tool (AAPT) το οποίο δίνει την δυνατότητα δημιουργίας .apkαρχείων τα οποία περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.
- 4) Την Android Interface Description Language (AIDL) η οποία επιτρέπει την δημιουργίακώδικα που επιτρέπει σε δύο διεργασίες σε μια συσκευή βασισμένη στο Android νασυνομιλούν χρησιμοποιώντας διαδιεργασιακή επικοινωνία.

- 5) Το `sqlite3` το οποίο επιτρέπει την πρόσβαση στα δεδομένα της `SQLite` που δημιουργούνται από τις διάφορες εφαρμογές.
- 6) Το `Traceview` που επιτρέπει την γραφική προβολή της ανάλυσης των `trace log data` που δημιουργούν οι διάφορες εφαρμογές.
- 7) Το `mksdcard` το οποίο βοηθά στην δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (όπως η `SD card`).
- 8) Το `dx tool` το οποίο μετατρέπει τα αρχεία `.class` από `java bytecode` σε `Android bytecode`.
- 9) Το `UI/Application Exerciser Monkey` το οποίο είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδο-τυχαίες σειρές από συμβάντα χρήστη όπως `clicks`, `touches`, `gestures` καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.
- 10) Το `activitycreator` το οποίο είναι ένα `script` που δημιουργεί `Ant build` αρχεία τα οποία μπορούν να χρησιμοποιηθούν για την μεταγλώττιση των εφαρμογών.

3.8 Μια γρήγορη εφαρμογή για το SDK

Το μόνο που μένει για να ολοκληρωθεί το κεφάλαιο του `Android SDK` είναι η ανάπτυξη μιας εφαρμογής.

Για να επιτευχθεί αυτό, το πρώτο βήμα που πρέπει να γίνει είναι η εγκατάσταση του απαραίτητου λογισμικού, και ακολούθως η ανάπτυξη μιας απλής εφαρμογής.

Τα προγράμματα που θα είναι απαραίτητα, είναι το `Android SDK` και στην συνέχεια το περιβάλλον ανάπτυξης λογισμικού `Eclipse`.

Το αρχικό βήμα είναι η μετάβαση στο developer.android.com/sdk, από το οποίο γίνεται κατέβαση του αρχείου `Android SDK`, εφόσον η εργασία πραγματοποιείται σε `Windows`.

Δοκιμαστικά θα δούμε τον αρχικό κώδικα της Android, το “HelloWorld”, για να γνωρίσουμε το Eclipse και τον Emulator.

Ο κώδικας είναι ο εξής :

```
package com.example.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello World!");
        setContentView(tv);
    }
}
```

Το επόμενο βήμα είναι να τρέξετε την εφαρμογή σας στον Android προσομοιωτή.

Μέσα από τον Eclipse, πρέπει να επιλέξετε Run > Run, στη συνέχεια το ADT plugin αυτόματα δημιουργεί το αρχείο ρυθμίσεων για την εκτέλεση της εφαρμογής και ξεκινάει τον Android προσομοιωτή.

Για να ξεκινήσει ο Android προσομοιωτής χρειάζονται να περάσουν μερικά λεπτά.

Όταν ξεκινήσει ο προσομοιωτής, τότε το ADT plugin κάνει εγκατάσταση της εφαρμογής και στη συνέχεια τρέχει την εφαρμογή καλώντας το Default Activity.

Μετά την εκτέλεση της εφαρμογής θα πρέπει να δείτε κάτι σαν την παρακάτω εικόνα, Hello World.jpg.

Το αποτέλεσμα που θα δείτε μετά την εκτέλεση της εφαρμογής στον προσομοιωτή του Android :



Εξομοιωτής

Κεφάλαιο 4^ο : Υλοποίηση εφαρμογής

4.1 Εισαγωγή

Βασικό και κύριο μέρος της εργασίας, που παρουσιάζω είναι η αναφορά στα εργαλεία, που χρησιμοποιήθηκαν για την ολοκλήρωση της εφαρμογής, η εγκατάσταση των λειτουργικών προγραμμάτων, η δημιουργία του κώδικα βήμα-βήμα και το τελικό αποτέλεσμα.

Στην συνέχεια γράφω αναλυτικά, για κάθε κομμάτι που ανέφερε παραπάνω.

4.2 Εργαλεία που χρησιμοποιήθηκαν

4.2.1 Android



Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει στον πυρήνα του λειτουργικού Linux.

Αρχικά αναπτύχθηκε από την Google και αργότερα από την OpenHandseAlliance.

Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OpenHandseAlliance, μίας κοινοπραξίας 48 τηλεπικοινωνιακών εταιρειών, εταιρειών λογισμικού καθώς και κατασκευής hardware, 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 ApacheLicense, μίας ελεύθερης άδειας λογισμικού.

4.2.2 Eclipse



Το Eclipse είναι ένα περιβάλλον ανάπτυξης λογισμικού για διάφορες γλώσσες προγραμματισμού, συνδυάζοντας ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IntegratedDevelopmentEnvironment – IDE) και ένα σύστημα πρόσθετων plug-ins.

Ενδεικτικά, το Eclipse μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε γλώσσες όπως η Java, η C, η C++, η PHP, η Perl, η Python, κ.α.

Προέρχεται από την γενιά IDE της IBM με την ονομασία VisualAge, και η πιο συχνή αφορά την ανάπτυξη εφαρμογών Java με την χρήση των JavaDevelopmentTools (JDT).

Οι δυνατότητες είναι επεκτάσιμες τοποθετώντας κατάλληλα πρόσθετα για ένα μεγάλο εύρος θεμάτων. Με την εξαίρεση μάλιστα ενός μικρού run-timekernel, οτιδήποτε στο Eclipse είναι στην πραγματικότητα ένα plug-in.

Πρόκειται για λογισμικό ανοιχτού κώδικα, και το επίσημο site είναι στην διεύθυνση :

www.eclipse.or

4.2.3 Java



Πρόκειται για ένα σύνολο προϊόντων λογισμικού και προδιαγραφών από την SunMicrosystems, η οποία είναι πλέον θυγατρική εταιρεία της Oracle.

Επιτρέπει την ανάπτυξη εφαρμογών οι οποίες μπορούν να λειτουργήσουν ανεξάρτητα πλατφόρμας, σε ένα πολύ μεγάλο εύρος συσκευών, όπως personalcomputers, servers, κινητά τηλέφωνα και μικροσυσκευές.

Αυτό είναι δυνατόν λόγω του ότι η εκτέλεση του κώδικα δεν πραγματοποιείται άμεσα από τον υπολογιστή, αλλά μέσω του JavaVirtualMachine (JVM).

Οcompiler της Java, ο οποίος περιέχεται στο JavaDevelopmentKit (JDK), παράγει έναν ενδιάμεσο κώδικα, τον JavaBytecode, ο οποίος είναι ανεξάρτητος της πλατφόρμας εκτέλεσης, και ο οποίος μετασχηματίζεται σε εκτελέσιμη μορφή με τον JI (just-in-time) compiler του JavaRuntimeEnvironment (JRE).

Ο τελευταίος μετασχηματισμός γίνεται “ onthefly “, δηλαδή την στιγμή της εκτέλεσης.

Ο κύριος λόγος μάλιστα ανάπτυξης της Java στην αρχή ήταν αυτή ακριβώς η δυνατότητα.

Το να μαθαίνει κάποιος μόνο μία γλώσσα, και να γράφει έναν κώδικα που να μπορεί να εκτελείται από ένα μεγάλο εύρος συσκευών, ιδίως από αυτές χωρίς μεγάλη επεξεργαστική ισχύ και απαιτήσεις, όπως πχ είναι οι οικιακές συσκευές, ήταν ένα ζητούμενο της βιομηχανίας, αφού περιόριζε σημαντικά το κόστος.

Σήμερα η Java, λόγω της σύνδεσης της με τον webserver από τα μέσα της δεκαετίας του 1990, χρησιμοποιείται στην πλειοψηφία των εφαρμογών που βρίσκονται στην πλευρά του server.

Λόγω της ευελιξίας που παρέχει το web, οι επιχειρήσεις και οι οργανισμοί έχουν μετατοπίσει τις εσωτερικές τους εφαρμογές σε περιβάλλον client-server, με αποτέλεσμα να έχει ωφεληθεί σημαντικότερα η Java, χρησιμοποιούμενη πλέον ως η γλώσσα ανάπτυξης εταιρικών εφαρμογών.

Βέβαια, η ωφέλεια προέρχεται και από την ανάπτυξη του internet ως μέσο για πλήθος νέων εφαρμογών προς πελάτες, όπως είναι πχ τα online καταστήματα, οι αγορές εισιτηρίων, κ.α., για τις οποίες εφαρμογές, και πάλι στην πλευρά του server χρησιμοποιείται κατά κανόνα η Java.

Λόγω αυτού του τρόπου λειτουργίας, η Java δεν διαθέτει αρκετά στοιχεία low-level, όπως είναι οι pointers, κι έχει ένα πολύ καλό μοντέλο μνήμης, στο οποίο κάθε αντικείμενο τοποθετείται στο heap και όλες οι μεταβλητές τύπου object είναι αναφορές.

Η διαχείριση μνήμης γίνεται από το JVM με χρήση ενός ολοκληρωμένου ενσωματωμένου μηχανισμού.

Η Java παρέχεται σε διάφορες εκδοχές όπως :

- Javacard, μία τεχνολογία που επιτρέπει μικρές Java εφαρμογές (applets) να τρέχουν με ασφάλεια σε έξυπνες κάρτες (smartcards) και αντίστοιχες συσκευές.
- JavaME (MicroEdition), η οποία έχει κάποια διαφορετικά σετ βιβλιοθηκών, έτσι ώστε να εξασφαλίζουν ότι δεν θα προκύψει χρήση υπερβολικά μεγάλων ποσοτήτων μνήμης.
- JavaSE (StandardEdition), η οποία αποτελεί το πακέτο γενικής χρήσης, όπως πχ σε desktop υπολογιστές.
- JavaEE (EnterpriseEdition), η οποία αποτελεί ένα υπερσύνολο της SE, με επιπλέον διάφορα χρήσιμα APIs για την ανάπτυξη client-server εφαρμογών.

Περισσότερα στοιχεία για την Java μπορεί κανείς να βρει στην διεύθυνση:

www.java.com

4.2.4 Google Maps



Το Google Maps είναι μία υπηρεσία της Google γνωστή σχεδόν σε όλους σήμερα, η οποία παρέχει χάρτες και δορυφόρους ή αεροφωτοχάρτες για το σύνολο του πλανήτη.

Επιπλέον των χαρτών όμως, η υπηρεσία συμπεριλαμβάνει την τεχνολογία για την χαρτογραφική υποστήριξη διάφορων εφαρμογών, μέσα από την Google Maps API.

Η υπηρεσία παρέχεται δωρεάν για μη εμπορική χρήση από κάποιο συγκεκριμένο domain, για το οποίο αφού δηλωθεί, παραχωρείται ένα αλφαριθμητικό κλειδί-ταυτότητα.

Επίσης, μέχρι αυτήν την στιγμή δεν υπάρχουν διαφημίσεις, αλλά υπάρχει η επιφύλαξη για την εισαγωγή τους.

Λόγω της μεγάλης πληρότητας και ακρίβειας των δεδομένων που περιέχουν, οι χάρτες της Google έχουν γίνει αυτή τη στιγμή οι ευρύτερα διαδεδομένοι διαδικτυακοί χάρτες.

Σε αυτό έχουν βοηθήσει και παράγοντες βέβαια όπως η αξιοπιστία της υπηρεσίας, η παροχή των πληροφοριών σε πάρα πολλές γλώσσες, αλλά και η προσθήκη συνεχώς νέων θεμάτων, όπως είναι πχ χάρτες κυκλοφοριακών συνθηκών ή και ακόμα χάρτες προβολής σε 3D.

Ιδιαίτερα έχει βοηθήσει η ύπαρξη της GoogleMapsAPI, το οποίο έχει επιτρέψει την δημιουργία πολυάριθμων εφαρμογών ανά τον κόσμο, με χρήση των GoogleMaps ως υπόβαθρο.

Με τα εργαλεία που παρέχονται μπορεί κανείς να δημιουργήσει σχετικά εύκολα μία εφαρμογή, της οποίας ο χάρτης και οι λειτουργίες του να είναι ήδη οικεία στο μεγαλύτερο ποσοστό των χρηστών.

Το GoogleMapsAPI παρέχεται σε έκδοση JavaScript, αλλά και για AdobeFlash, όπου μάλιστα μπορεί να γίνει μίξη με τα στοιχεία του AdobeFlash.

-----Κύρια χαρακτηριστικά του API

Το κύριο αντικείμενο του GoogleMapsAPI είναι το map, πάνω στο οποίο τοποθετούνται όλα τα υπόλοιπα αντικείμενα-παιδιά.

Μια βελτιωμένη έκδοση του map είναι το map 3D, το οποίο παρέχει επιπλέον την δυνατότητα για χάρτες με τρισδιάστατη οπτική.

Το αντικείμενο map έχει ιδιότητες που καθορίζουν το μέγεθος, το κεντράρισμα, την κλίμακα, την γλώσσα απεικόνισης, όπως επίσης τα είδη χάρτη και ποια από αυτά θα είναι επιλέξιμα από τον χρήστη (πχ χάρτης, δορυφορικός, ορθοφωτοχάρτης, κ.α.).

Το map δέχεται επάνω του διάφορα χειριστήρια, όπως για τον προσανατολισμό, την κλίμακα ή την απεικόνιση ενός ευρύτερου χάρτη μικρής κλίμακας.

Το αντικείμενο map μπορεί να αντιδρά σε διάφορα συμβάντα (πχ click του χρήστη σε κάποιο σημείο), κι ανάλογα να ενεργοποιούνται οι σχετικές ρουτίνες της εφαρμογής.

Αντικείμενα-παιδιά του map μπορεί να είναι διάφορα επιθέματα, όπως Markers για σημεία, Polylines για γραμμικά στοιχεία, και Polygons επιφανειακά στοιχεία, ή ακόμα και εικόνες (groundoverlays).

Κάθε ένα από αυτά έχει ένα πλήθος από ιδιότητες (σχήμα, μέγεθος, χρώμα κ.τ.λ.), και μπορεί να υπάρχει χειρισμός των συμβάντων, όπως πχ το click του χρήστη πάνω τους.

Ένα ιδιαίτερα χρήσιμο στοιχείο, το οποίο μπορεί να ανήκει σε κάποιο από τα υπόλοιπα (Map, Markers, Polylines, Polygons) είναι το InfoWindow.

Πρόκειται για ένα συννεφάκι στην πιο συνηθισμένη του μορφή, το οποίο μπορεί να περιέχει μέσα του περιεχόμενο από απλό κείμενο και HTML, μέχρι άλλα αντικείμενα, ακόμη και Flash.

Περισσότερες και αναλυτικές πληροφορίες μπορεί κανείς να βρει στην διεύθυνση :

<http://code.google.com/intl/en/apis/maps/documentation/flash>

4.2.4.1 Οι χάρτες Google και το Android

Το Android προσφέρει στους προγραμματιστές την δυνατότητα να ενσωματώσουν κάποιο χάρτη στην εφαρμογή τους.

Αυτό μπορεί να γίνει με δύο τρόπους :

Σύμφωνα με τον πρώτο τρόπο, που είναι και ο πιο απλός, ο προγραμματιστής μπορεί να ενσωματώσει στην οθόνη ένα στοιχείο το οποίο καλείται 'WebView'.

Στο 'WebView', ο προγραμματιστής μπορεί να απεικονίσει οποιαδήποτε διαδικτυακή εφαρμογή με αντίστοιχο τρόπο που αυτή απεικονίζεται σε έναν φυλλομετρητή και κατ' επέκταση τους Χάρτες Google.

Η Google όμως θέλοντας να δώσει περισσότερη ελευθερία στους προγραμματιστές, δημιούργησε ένα άλλο στοιχείο διεπιφάνειας χρήστη, το 'MapView' γεγονός που μας οδηγεί στην δεύτερη εναλλακτική.

Το MapView σε συνδυασμό με την κλάση MapActivity και τις διεπαφές mapping APIs, είναι ένα ισχυρό εργαλείο στα χέρια οποιουδήποτε προγραμματιστή που θέλει να δημιουργήσει καινοτόμες και πρωτοποριακές εφαρμογές με την χρήση των χαρτών.

Με αυτό τον τρόπο ο κατασκευαστής μπορεί να χρησιμοποιήσει υπαρχών χάρτες και να τους τροποποιήσει όπως αυτός θέλει, με την δυνατότητα να

χειριστεί τις αλληλεπιδράσεις του χρήστη με τον χάρτη, να τοποθετήσει δικά του δεδομένα πάνω σε αυτόν κ.τ.λ.

Το πακέτο για την χρησιμοποίηση των mapping APIs δεν ανήκει στο πλαίσιο του Android, αλλά σε αυτό της Google (com.google.android.maps), γεγονός που πρέπει να δηλωθεί ξεχωριστά.

Επιπρόσθετα για την χρησιμοποίηση των χαρτών η Google απαιτεί την απόκτηση από τον εκάστοτε προγραμματιστή ενός map-api key, κλειδί το οποίο προσφέρει η ίδια

[\(<http://code.google.com/android/maps-api-signup.html>\)](http://code.google.com/android/maps-api-signup.html).

Για την απόκτηση του κλειδιού ο ενδιαφερόμενος πρέπει να δώσει το αποτύπωμα MD5 του ψηφιακού πιστοποιητικού το οποίο θα χρησιμοποιηθεί για την υπογραφή της εφαρμογής.

Το MD5 fingerprint του αρχείου keytool το οποίο βρίσκεται στον φάκελο που έχουμε εγκαταστήσει το JAVA SDK στο λειτουργικό μας σύστημα.

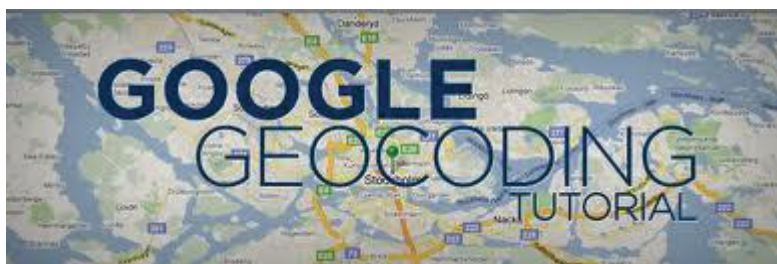
Αφού εντοπίσουμε το αρχείο, χρησιμοποιούμε την εντολή “keytool -list -keystore .keystore” στην γραμμή εντολών, για να πάρουμε το MD5 fingerprint του αρχείου.

Στη συνέχεια πηγαίνουμε στην ιστοσελίδα του παραπάνω συνδέσμου, και παρέχοντας το στο σχετικό πεδίο, μας παρέχεται το κλειδί του API.

Το κλειδί είναι ένα αλφαριθμητικό 40 χαρακτήρων το οποίο είναι καλό να αποθηκεύσουμε σε ένα String resource και να του δώσουμε ένα όνομα αναφοράς, πχ “maps_api_key” ώστε να το έχουμε διαθέσιμο προς χρήση στο layout των χαρτών.

Αν το κλειδί μας είναι λανθασμένο τότε οι χάρτες δεν θα εμφανίζονται στην MapView.

4.2.5 Google Geocoding



Το Google Geocoding είναι μία διαδικτυακή υπηρεσία που παρέχεται από την Google και η οποία μπορεί να μετατρέπει διευθύνσεις σε γεωγραφικές συντεταγμένες, αλλά και το αντίστροφο.

Το API δίνει την δυνατότητα πρόσβασης σε έναν Geocoder μέσω ενός HTTP request.

Η χρήση της υπηρεσίας είναι δωρεάν και ελεύθερη από διαφημίσεις μέχρι αυτήν την στιγμή, αρκεί τα ερωτήματα (queries) να μην ξεπεράσουν τα 2500/ημέρα για κάποιον χρήστη ή να μην γίνονται σε πολύ γρήγορο ρυθμό.

Απαραίτητη προϋπόθεση της άδειας χρήσης της υπηρεσίας είναι τα αποτελέσματα να σχετίζονται με την εμφάνιση τους σε έναν χάρτη της Google, ο λεγόμενος GoogleMap.

Περισσότερες πληροφορίες μπορεί κανείς να βρει στην διεύθυνση :

<http://code.google.com/apis/maps/documentation/geocoding/#Geocoding>

4.3 Προετοιμασία

Για την προετοιμασία της εφαρμογής απαιτείται η εγκατάσταση μερικών λειτουργικών προγραμμάτων, που αναφέρονται λεπτομερώς στην συνέχεια.

Ότι άλλη έκδοση χρησιμοποιηθεί σε Linux ή Windows, δεν θα έχει κάποια αλλαγή στον τρόπο λειτουργίας

- Εγκατάσταση του Eclipse IDE

- Εγκατάσταση ADTpluginγια το Eclipse
- Εγκατάσταση JDK και JRE
- Εγκατάσταση του Android SDK

4.3.1 Εγκατάσταση του Eclipse IDE

Αρχικά κατεβάζουμε το πρόγραμμα από την επίσημη σελίδα του :

<http://www.eclipse.org/downloads/>

και το εγκαθιστούμε στον υπολογιστή μας.

Αποσυμπιέζουμε το αρχείο, κατά προτίμηση στον φάκελο C.

Σε περίπτωση που δεν έχουμε εγκαταστήσει το JavaRunTimeEnvironment (JRE) και το JavaDevelopmentKit (JDK), δεν έχουμε παρά να τα κατεβάσουμε από την σελίδα της Oracle:

www.oracle.com/technetwork/java/javase/downloads/index.html

επιλέγοντας “ Λήψη και εγκατάσταση JDKκαι JRE”.

Java SE 6 Update 43

This release brings in security features and bug fixes. Oracle strongly recommends that all Java SE 6 users upgrade to this release. [Learn more](#) ▶

JDK

DOWNLOAD ↓

JRE

DOWNLOAD ↓

- [Installation Instructions](#)
- [ReadMe](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Products](#)
- [Third Party Licenses](#)
- [Certified System Configurations](#)

JRE 6 Docs

- [Installation Instructions](#)
- [ReadMe](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Products](#)
- [Third Party Licenses](#)
- [Certified System Configurations](#)

4.3.2 Εγκατάσταση του ADTPlugin για το Eclipse

Τρέχουμε το Eclipse.

Αρχικά θα μας ζητηθεί να επιλέξουμε τον φάκελο που θα αποθηκεύονται τα projects μας.

Για την παρούσα εργασία επιλέχθηκε το C:\Eclipse\Projects.

Τώρα πρέπει να εγκαταστήσουμε το plugin, επιλέγοντας στο Help->Install New Software.

Στο πεδίο “WorkWith” γράφουμε :

<https://dl-ssl.google.com/android/eclipse/>

και πατάμε add.

Στο παράθυρο που θα ανοίξει γράφουμε ένα όνομα, ώστε αργότερα να μπορούμε να τα διαχωρίζουμε.

Επιλέγουμε το “Developer tools” και πατάμε next.

Αφού ολοκληρωθεί η εγκατάσταση κάνουμε επανεκκίνηση του Eclipse.

4.3.3 Εγκατάσταση του Android SDK

Κατεβάζουμε το SDK από το επίσημο site του Android :

<http://developer.android.com/sdk/index.html>

Για τις ανάγκες της εργασίας, επιλέχθηκε το zip αρχείο και έγινε unzip στο C:\Android.

Τρέχουμε το αρχείο SDK Manager.exe που βρίσκεται στον φάκελο Android και στο παράθυρο που θα ανοίξει, επιλέγουμε τις εκδόσεις του Android και διάφορα άλλα στοιχεία που θέλουμε να χρησιμοποιήσουμε και πατάμε το install.

Εγκατάσταση του SDK.
Όταν ολοκληρωθεί η εγκατάσταση, πατάμε Restart και κλείνουμε το παράθυρο.

4.4 Δημιουργία Android εφαρμογής

Μια καλή συμβουλή πριν από τη δημιουργία της πρώτης εφαρμογής, είναι η δημιουργία μιας εικονικής Android συσκευής για τον ταχύτερο έλεγχο του λογισμικού.

Το Android Virtual Device (AVD) είναι ένα τέτοιο εικονικό smartphone που τρέχει Android, στο οποίο μπορούμε εύκολα να εκτελέσουμε οποιοδήποτε πρόγραμμα έχουμε δημιουργήσει.

Το προφανές πλεονέκτημα του AVD είναι ότι μπορούμε να δούμε πώς το λογισμικό μας θα προβάλλεται σε διαφορετικά smartphones τα οποία υποστηρίζουν Android, αντί να τα αγοράσουμε (κάτι που είναι πρακτικά και κυρίως, οικονομικά αδύνατον) και να κάνουμε τις απαραίτητες δοκιμές στο καθένα ξεχωριστά.

Πριν φτιάξουμε το AVD, πρώτα θα συνδέσουμε το Android SDK, επιλέγοντας στο Window->Preferences.

Στο παράθυρο που ανοίγει, πάμε στο Android και εκεί επιλέγουμε το φάκελο που έχουμε κάνει Unzip το SDK, πατάμε Apply και τέλος OK.

Τώρα πάμε στο Window -> Android SDK and AVD Manager, πατάμε New και διαλέγουμε τα χαρακτηριστικά που θέλουμε να έχει το εικονικό smartphone μας.

Εδώ επιλέχθηκε το Android 4.0.3, επειδή αργότερα ήταν η τελευταία έκδοση που υπήρχε.

Κλείνουμε το παράθυρο και πάμε στο File -> New -> Other, όπου επιλέγουμε το Android -> Android Project.

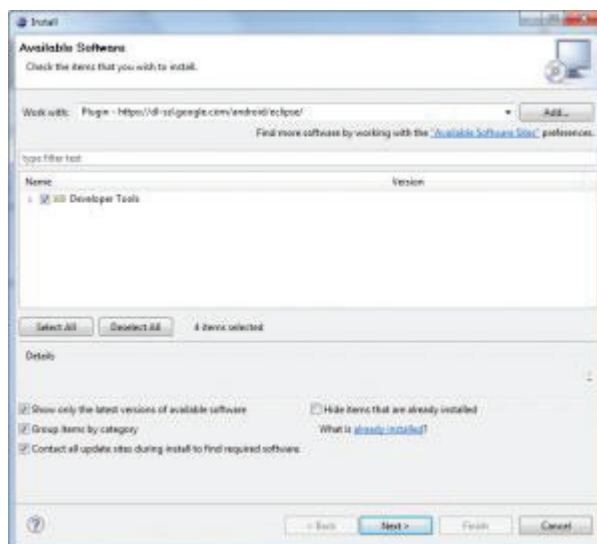
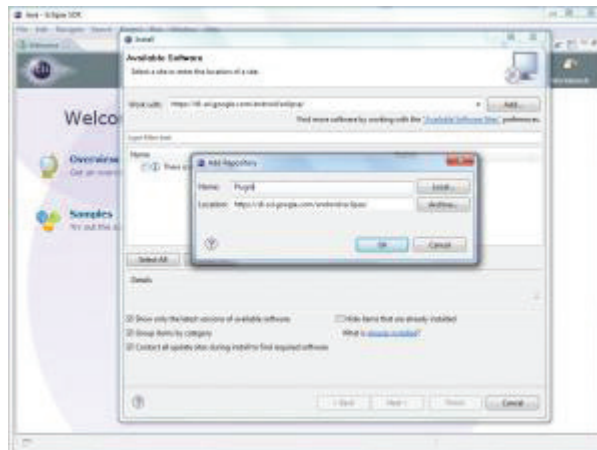
Πατάμε Finish.

Μετά την δημιουργία του project, στα αριστερά θα εμφανιστεί ένα δέντρο φακέλων.

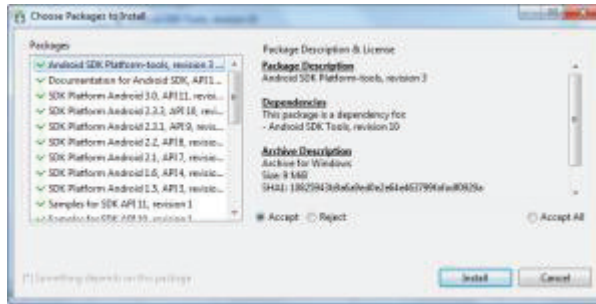
Αρχικά μπορούμε να επιλέξουμε την εικόνα του app μας.

Αυτό πρέπει να περιγραφτεί στο αρχείο main.xml (res->layout).

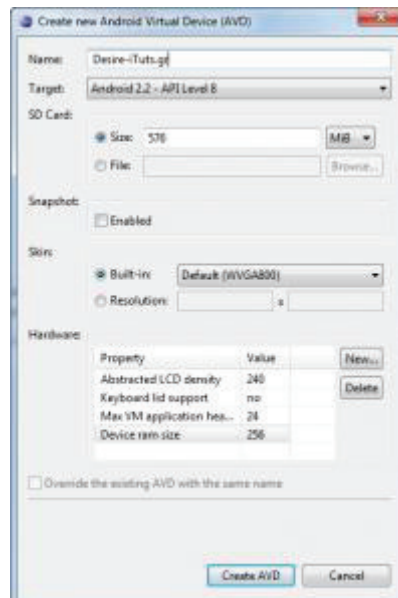
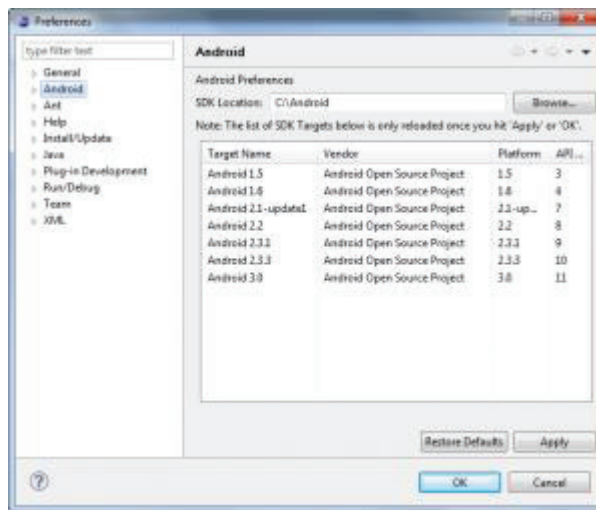
4.5 Η εφαρμογή σε εικόνες

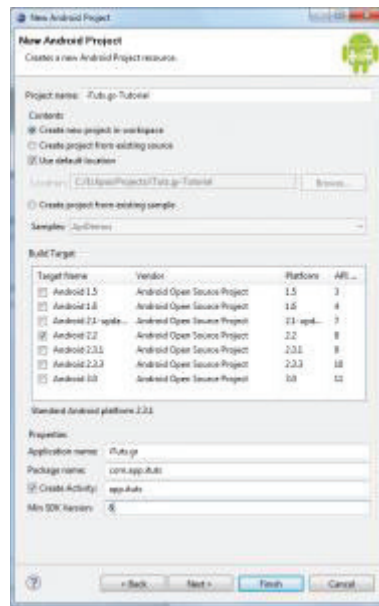


Εγκατάσταση του ADT plugin για το Eclipse



Εγκατάσταση του Android SDK





Δημιουργία Android-εφαρμογής

Ξεκινάμε λοιπόν με τον κώδικα.

4.6 Κώδικας

Αφού αναφερθήκαμε με κάποιες λεπτομέρειες στα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής, θα ασχοληθούμε με το βασικό σκέλος της εργασίας, τον κώδικα.

Παρακάτω καταγράφω κομμάτι-κομμάτι τον κώδικα και αναλύω τι κάνει κάθε εντολή.

Android Manifest (XML File - AndroidManifest.xml)

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="demokritos.medianetlab.MediaNetLabGPS" android:versionCode="
1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8"/>

```

```

name"
    <applicationandroid:icon="@drawable/icon"android:label="@string/app_
        android:debuggable="true">
me">
    <activityandroid:name=".MediaNetLabGPS"android:label="@string/app_na
        <intent-filter>
    <actionandroid:name="android.intent.action.MAIN"/>
    <categoryandroid:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activityandroid:name=".CreateXML"android:label="CreateXML">
    </activity>

    </application>
    <uses-
permissionandroid:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-
permissionandroid:name="android.permission.ACCESS_MOCK_LOCATION"/>
    <uses-
permissionandroid:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-
permissionandroid:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permissionandroid:name="android.permission.INTERNET"/>

</manifest>

```

Main Layout (XML File - main.xml)

```

<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"android:layout_width="fill_parent"
    android:layout_height="fill_parent"android:background="#FFFFFF"
    android:weightSum="1">
iew1"
    <ImageViewandroid:layout_width="wrap_content"android:id="@+id/imageV
        android:src="@drawable/mnl_logo"android:layout_height="wrap_content"
        android:layout_gravity="center"/>
    <TextViewandroid:id="@+id/textView1"android:textAppearance="?android
:attr/textAppearanceLarge"
        android:layout_gravity="center"android:textColor="#000000"
        android:layout_width="wrap_content"android:layout_height="wrap_conte
nt"/>
    <Buttonandroid:id="@+id/retrieve_location_button"android:text="Retri
eve Location"

```

```
        android:layout_width="match_parent"android:layout_height="wrap_conte  
nt"/>  
</LinearLayout>
```

MediaNetLabGPS Class (Java File – MediaNetLabGPS.java)

```
package demokritos.medianetlab.MediaNetLabGPS;  
  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.location.Location;  
import android.location.LocationListener;  
import android.location.LocationManager;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.Toast;  
  
publicclass MediaNetLabGPS extends Activity {  
  
    privatestaticfinallongMINIMUM_DISTANCE_CHANGE_FOR_UPDATES = 1; // in  
Meters  
    privatestaticfinallongMINIMUM_TIME_BETWEEN_UPDATES = 1000; // in  
Milliseconds  
    public String latitude;  
    public String longitude;  
  
    protected LocationManager locationManager;  
  
    protected Button retrieveLocationButton;  
  
    @Override  
    publicvoid onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        retrieveLocationButton = (Button)  
findViewById(R.id.retrieve_location_button);  
  
        locationManager = (LocationManager)  
getSystemService(Context.LOCATION_SERVICE);
```

```

        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
            MINIMUM_TIME_BETWEEN_UPDATES,
            MINIMUM_DISTANCE_CHANGE_FOR_UPDATES,
newMyLocationListener());

        retrieveLocationButton.setOnClickListener(new
OnClickListener() {
            publicvoid onClick(View v) {
                showCurrentLocation();
            }
        });
    }

    protectedvoid showCurrentLocation() {
        Location location = locationManager
            .getLastKnownLocation(LocationManager.GPS_PROVIDER);

        if (location != null) {
            String message = String.format(
                "Current Location \n Longitude: %1$s \n
Latitude: %2$s",
                location.getLongitude(),
                location.getLatitude());
            Toast.makeText(MediaNetLabGPS.this, message,
                Toast.LENGTH_LONG)
                .show();
        }

        privateclass MyLocationListener implements LocationListener {

            publicvoid onLocationChanged(Location location) {
                String message = String.format(
                    "New Location \n Longitude: %1$s \n
Latitude: %2$s",
                    location.getLongitude(),
                    location.getLatitude());
                Toast.makeText(MediaNetLabGPS.this, message,
                    Toast.LENGTH_LONG)
                    .show();
                double lat = location.getLatitude();
                double longi = location.getLongitude();
                latitude = Double.toString(lat);
                longitude = Double.toString(longi);
            }

            publicvoid onStatusChanged(String s, int i, Bundle b) {
                Toast.makeText(MediaNetLabGPS.this, "Provider status
changed",
                    Toast.LENGTH_LONG).show();
            }

            publicvoid onProviderDisabled(String s) {
                Toast.makeText(MediaNetLabGPS.this,

```

```

off",
        "Provider disabled by the user. GPS turned
        Toast.LENGTH_LONG).show();
    }

    public void onProviderEnabled(String s) {
        Toast.makeText(MediaNetLabGPS.this,
        "Provider enabled by the user. GPS turned
on",
        Toast.LENGTH_LONG).show();
    }
}

public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.createXML:
            // Pass Latitude and Longitude to CreateXML activity
            Bundle bundle = new Bundle();
            bundle.putString("latitude", latitude);
            bundle.putString("longitude", longitude);
            Intent newIntent =
new Intent(this, getApplicationContext(),
                CreateXML.class);
            newIntent.putExtras(bundle);
            startActivityForResult(newIntent, 0);
            break;
        case R.id.about:
            onClickAbout();
            break;
        case R.id.exit:
            int pid = android.os.Process.myPid();
            android.os.Process.killProcess(pid);
            super.onDestroy();
            this.finish();
            break;
    }
    return true;
}

private void onClickAbout() {
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setTitle("About");
    alert.setMessage("Voula Tsakona \nNCSR Demokritos
\nMediaNetLab \nwww.medianetlab.gr");
    alert.setNegativeButton("Close", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int
whichButton) {
            // Canceled.
        }
    });
    alert.show();
}

public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
}

```



```

        return true;
    }
}

```

Create XML Layout (XML File – createxml.xml)

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:background="#FFFFFF">
    <TextView android:id="@+id/result" android:textColor="#000000"
        android:layout_height="wrap_content" android:layout_width="match_pare
nt"
        android:layout_gravity="center"/>
    <TextView android:id="@+id/date" android:textColor="#000000"
        android:layout_height="wrap_content" android:layout_width="match_pare
nt"
        android:layout_gravity="center"/>
</LinearLayout>

```

Create XML Class (Java File – CreateXML.java)

```

package demokritos.medianetlab.MediaNetLabGPS;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.Locale;
import org.xmlpull.v1.XmlSerializer;
import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.util.Xml;
import android.widget.TextView;

public class CreateXML extends Activity {

    public String latitudexml;
    public String longitudexml;
    public String dateTime;
    public TextView tv;
}

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.createxml);
    tv = (TextView) this.findViewById(R.id.date);

    // Using SimpleDateFormat class
    SimpleDateFormat sdfDateTime = new SimpleDateFormat("dd-MM-
yyyy_HH-mm-ss", Locale.ENGLISH);
    dateTime = sdfDateTime.format(new
Date(System.currentTimeMillis()));
    tv.append(dateTime);

    // Retrieve Latitude and Longitude From MediaNetLabGPS
activity
    Bundle bundle = this.getIntent().getExtras();
    latitudexml = bundle.getString("latitude");
    longitudexml = bundle.getString("longitude");

    // Create a new file called "position.xml" in the SD card
    File newxmlfile =
newFile(Environment.getExternalStorageDirectory() + "/pos_" + dateTime +
.xml");
    try {
        newxmlfile.createNewFile();
    } catch (IOException e) {
        Log.e("IOException", "exception in createNewFile()
method");
    }
    // we have to bind the new file with a FileOutputStream
    FileOutputStream fileos = null;
    try {
        fileos = new FileOutputStream(newxmlfile);
    } catch (FileNotFoundException e) {
        Log.e("FileNotFoundException", "can't create
FileOutputStream");
    }
    // we create a XmlSerializer in order to write xml data
    XmlSerializer serializer = Xml.newSerializer();
    try {
        // we set the FileOutputStream as output for the
serializer, using
        // UTF-8 encoding
        serializer.setOutput(fileos, "UTF-8");
        // Write <?xml declaration with encoding (if encoding
not null) and
        // standalone flag (if standalone not null)
        serializer.startDocument(null, Boolean.valueOf(true));
        // set indentation option

        serializer.setFeature("http://xmlpull.org/v1/doc/features.html#inden
t-output", true);
        // start a tag called "position"

        serializer.startTag(null, "position");

        // i indent code just to have a view similar to xml-
tree

```

```

        serializer.startTag(null, "latitude");
        // set an attribute called "attribute" with a "value"
for <latitude>
        serializer.attribute(null, "attribute", longitudexml);
        serializer.endTag(null, "latitude");

        serializer.startTag(null, "longitude");

for
        // set an attribute called "attribute" with a "value"
        // <longitude>
        serializer.attribute(null, "attribute", longitudexml);
        serializer.endTag(null, "longitude");

        serializer.endTag(null, "position");
        serializer.endDocument();

        // write xml data into the FileOutputStream
        serializer.flush();

        // finally we close the file stream
        fileos.close();

        TextView tv = (TextView)
this.findViewById(R.id.result);
        tv.setText("XML file has been created on SD card.");
        } catch (Exception e) {
            Log.e("Exception", "Error occurred while creating xml
file.");
        }
    }
}
}

```

Menu Layout (XML File – menu.xml)

```

<?xmlversion="1.0"encoding="utf-8"?>
<menuxmlns:android="http://schemas.android.com/apk/res/android">
    <itemandroid:id="@+id/createXML"android:title="@string/createXML"/>
    <itemandroid:id="@+id/about"android:title="@string/about"/>
    <itemandroid:id="@+id/exit"android:title="@string/exit"/>
</menu>

```

String Values (XML File –strings.xml)

```

<?xmlversion="1.0"encoding="utf-8"?>
<resources>

```

```
<stringname="app_name">MediaNetLabGPS</string>
<stringname="createXML">Create XML</string>
<stringname="about">About</string>
<stringname="exit">Exit</string>
</resources>
```

4.7 Χρήση εντοπισμού θέσης για πλοήγηση

Η τελευταία δυνατότητα που θα προσθέσουμε στην Activity “CSMapView”, είναι ο εντοπισμός της θέσης του χρήστη και η δυνατότητα λήψης οδηγιών πλοήγησης, εφόσον ο εντοπισμός θέσης είναι επιτυχής.

Θα ξεκινήσουμε αλλάζοντας λίγο την δήλωση της κλάσης μας.

```
PublicclassCSMapView extendsSherlockMapActivity
implementsLocationListener
```

Πλέον η Activity μας θα μπορεί να ανταποκρίνεται στην αλλαγή θέσης της συσκευής.

Συνεχίζουμε με την δημιουργία των τριών μεθόδων η οποίες θα μας επιτρέψουν να εντοπίσουμε τη θέση της συσκευής.

Ξεκινάμε με την μέθοδο “getBestProvider” η οποία θα ελέγξει ποιες μέθοδοι είναι διαθέσιμες και αναλόγως θα θέσει τα κριτήρια εντοπισμού.

Εαν είναι ενεργό το GPS, ή ο εντοπισμός μέσω δικτύων WiFi, θα επιλέξει το GPS, αλλιώς όποιο από τα δύο είναι διαθέσιμα.

Σε περίπτωση που κανέναν δεν είναι διαθέσιμο θα παραπέμψουμε τον χρήστη να ενεργοποιήσει τις μεθόδους εντοπισμού της συσκευής του.

```
Public String getBestProvider() {  
String bestProvider = null;  
locationManager =  
(LocationManager) getSystemService(Context.LOCATION_SERVIC  
E);
```

```

gps_enabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
network_enabled =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
Criteria criteria = newCriteria();
criteria.setPowerRequirement(Criteria.POWER_MEDIUM);
if(!network_enabled && !gps_enabled) {
Toast.makeText(getApplicationContext(),"Δεν έχει ενεργοποιηθεί καμία υπηρεσία εντοπισμού.",Toast.LENGTH_SHORT).show();
}
Elseif(network_enabled && !gps_enabled) {
criteria.setAccuracy(Criteria.ACCURACY_COARSE);
bestProvider = locationManager.getBestProvider(criteria, true);
}
Elseif((gps_enabled && !network_enabled)|| (network_enabled && gps_enabled)) {
criteria.setAccuracy(Criteria.ACCURACY_FINE);
bestProvider = locationManager.getBestProvider(criteria, true);
}
Return bestProvider;
}

```

Η επόμενη μέθοδος είναι η “setCurrentLocation” η οποία όταν καλείτε ενημερώνει την μεταβλητή τρέχουσας θέσης.

```

Public void setCurrentLocation(Location location) {
Int currLatitude = (int) (location.getLatitude() * 1E6);
Int currLongitude = (int) (location.getLongitude() * 1E6);
currentPoint = new GeoPoint(currLatitude, currLongitude);
currentLocation = new Location("");
currentLocation.setLatitude(currentPoint.getLatitudeE6() / 1e6);
currentLocation.setLongitude(currentPoint.getLongitudeE6() / 1e6);
}

```

Τέλος η “getLastLocation()” χρησιμοποιώντας τις δύο παραπάνω μεθόδους, ενημερώνει τη θέση της συσκευής ώστε να είναι εφικτή η παροχή οδηγιών πλοήγησης.

```
PublicvoidgetLastLocation() {  
provider= getBestProvider();  
currentLocation=  
locationManager.getLastKnownLocation(provider);  
}  
}
```

4.8 Χρήση της βιβλιοθήκης MapViewBalloons

Έχοντας σημειώσει στον χάρτη μας όλα τα σημεία ενδιαφέροντος, θα μπορούσαμε να προσθέσουμε μια βιβλιοθήκη η οποία μας δίνει την δυνατότητα να εμφανίσουμε ένα μπαλονάκι με το όνομα του σημείου ενδιαφέροντος μόλις ο χρήστης κάνει “tap” με το δάχτυλο του σε αυτό.

Θα μπορούσαμε να χρησιμοποιήσουμε την βιβλιοθήκη “MapView Ballons” η οποία έχει αναπτυχθεί από τον Jeff Gilfelt και είναι διαθέσιμη για χρήση στο github, υπό την Apache License 2.0.

Η ενσωμάτωση της βιβλιοθήκης γίνεται με τον ίδιο τρόπο που ενσωματώσαμε και τις δύο προηγούμενες βιβλιοθήκες (ActionBarSherlock & ViewPagerIndicator) δημιουργώντας ένα νέο project βάση του υπάρχοντα κώδικα της βιβλιοθήκης, και εισάγοντας την βιβλιοθήκη στην λίστα με τις βιβλιοθήκες αναφοράς του project μας.

Στην παρούσα περίπτωση προτιμήθηκε να μείνει ακριβώς ο τρόπος σημείωσης που μας δίνει η Google Maps.

4.9 Υλοποίηση και εισαγωγή σημείων ενδιαφέροντος στον χάρτη

Είμαστε έτοιμοι λοιπόν να χρησιμοποιήσουμε το API των χαρτών.

Θα ξεκινήσουμε με μια μικρή τροποποίηση στην SuperClass της Activity μας.

Οι Activities που χρησιμοποιούν MapView στο layout τους πρέπει να είναι τύπου “MapActivity”.

Στην περίπτωση μας επειδή θέλουμε να χρησιμοποιήσουμε και την ActionBar μαζί με τον χάρτη η MapActivity θα γίνει “SherlockMapActivity”.

Public class CSMapView extends SherlockMapActivity

Στη συνέχεια θα χρησιμοποιήσουμε το layout της MapView, και θα ρυθμίσουμε τον χάρτη μας να χρησιμοποιεί την απεικόνιση δορυφόρου με ενεργοποιημένα τα χειριστήρια zoom.

```
mapView= (MapView) findViewById(R.id.mapview);  
mapView.setBuiltInZoomControls(true);  
mapView.setSatellite(true);
```

Μέσω του mapcontroller και ρυθμίζοντας το αρχικό zoom, θα εστιάσουμε τον χάρτη μας ακριβώς πάνω από τον χώρο που θέλουμε, π.χ την σχολή.

```
mapController= mapView.getController();  
mapController.setCenter(new GeoPoint((int) (..... * 1e6),(int) (...  
* 1e6)));  
mapController.setZoom(18);
```

Τα σημεία ενδιαφέροντος δεν είναι δυνατόν να τοποθετηθούν απευθείας πάνω στον χάρτη, γιατί θα χρησιμοποιήσουμε μια κλάση τύπου overlay, που θα τα εμφανίζει.

```
CSOverlay csoverlay = new CSOverlay();  
List<Overlay> overlayList = mapView.getOverlays();  
overlayList.add(csoverlay);
```

Τέλος θα δηλώσουμε όλα τα αντικείμενα με τις συντεταγμένες τους στον χάρτη, ώστε να τα εμφανίσουμε στον χάρτη χρησιμοποιώντας ένα γραφικό το οποίο έχουμε εισάγει στον φάκελο drawables.

```
GeoPoint CST = new GeoPoint((int) (.... * 1e6),(int) (.... * 1e6));
```

```

OverlayItem overlayItem_CST = newOverlayItem(CST,"Τεσυδ",
""");
mapOverlays= mapView.getOverlays();
drawable=
this.getResources().getDrawable(R.drawable.map_pin_mini);
itemizedOverlay= newMapItems(drawable, mapView);
itemizedOverlay.addOverlay(overlayItem_CST);

```

Επαναλαμβάνουμε τον παραπάνω κώδικα για όλα τα σημεία ενδιαφέροντος, δηλώνοντας κάθε φορά και νέα μεταβλητή τύπου “GeoPoint” και νέα τύπου “OverlayItem” με τις αντίστοιχες συντεταγμένες.

Η τελική μορφή της Activity μας θα φανεί στις φωτογραφίες που θα δούμε παρακάτω.

4.10 Δήλωση αδειών χρήστη στο AndroidManifest

Η υλοποίηση της Activity έχει ολοκληρωθεί και πλέον έχουμε έναν πλήρη μηχανισμό λήψης, αποθήκευσης και προβολής της θέσης, πχ του τμήματος.

Για να λειτουργήσουν όλα τα αντικείμενα που δημιουργήσαμε όμως, χρειάζονται κάποιες άδειες χρήσης του συστήματος, οι οποίες θα μας δώσουν την δυνατότητα να:

- έχουμε πρόσβαση στο Ίντερνετ
- να ελέγχουμε την κατάσταση του δικτύου
- να αποθηκεύσουμε αρχεία στην κάρτα SD της συσκευής

Θα δηλώσουμε λοιπόν τις τρεις παραπάνω άδειες στο αρχείο AndroidManifest.xml:

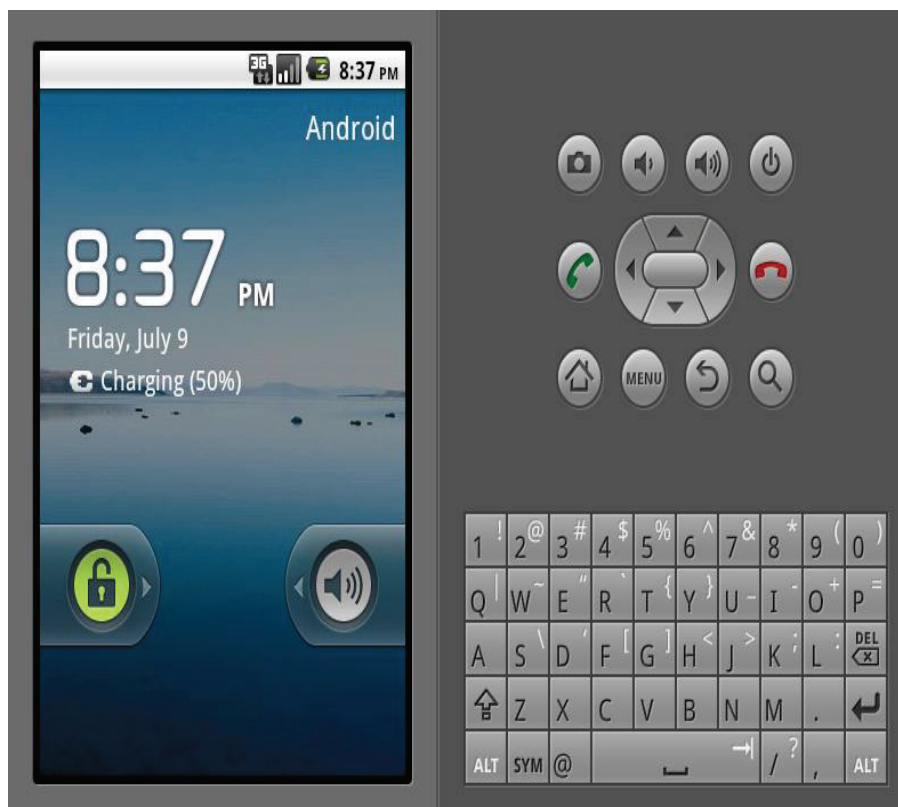
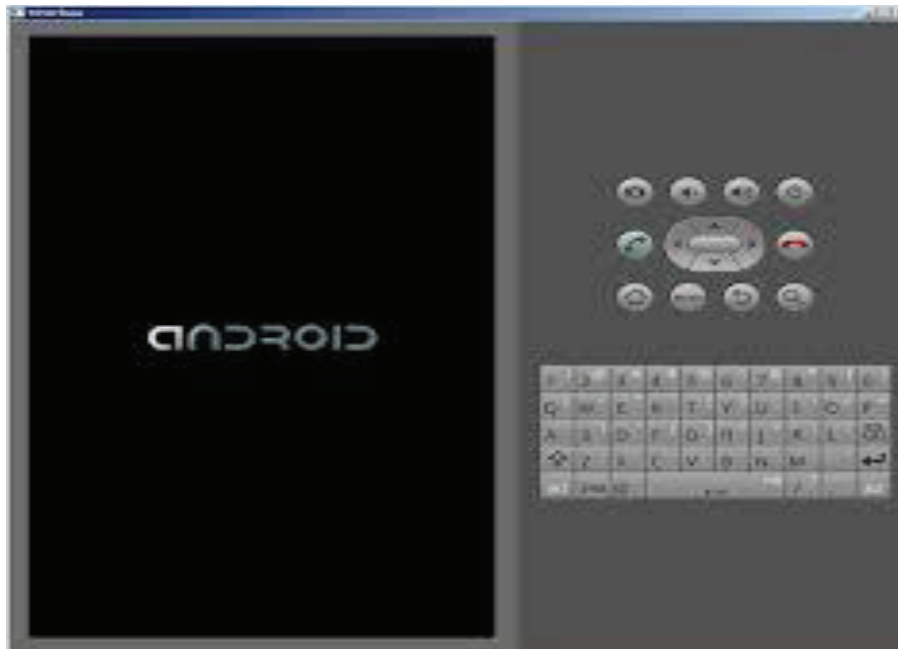
```

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

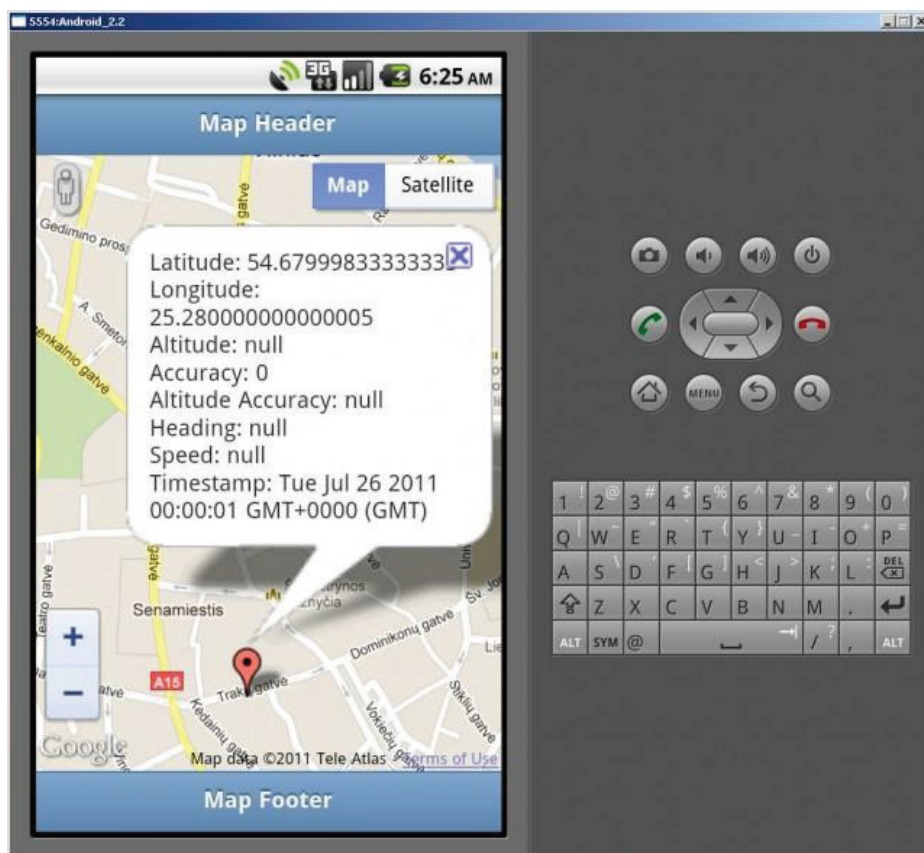
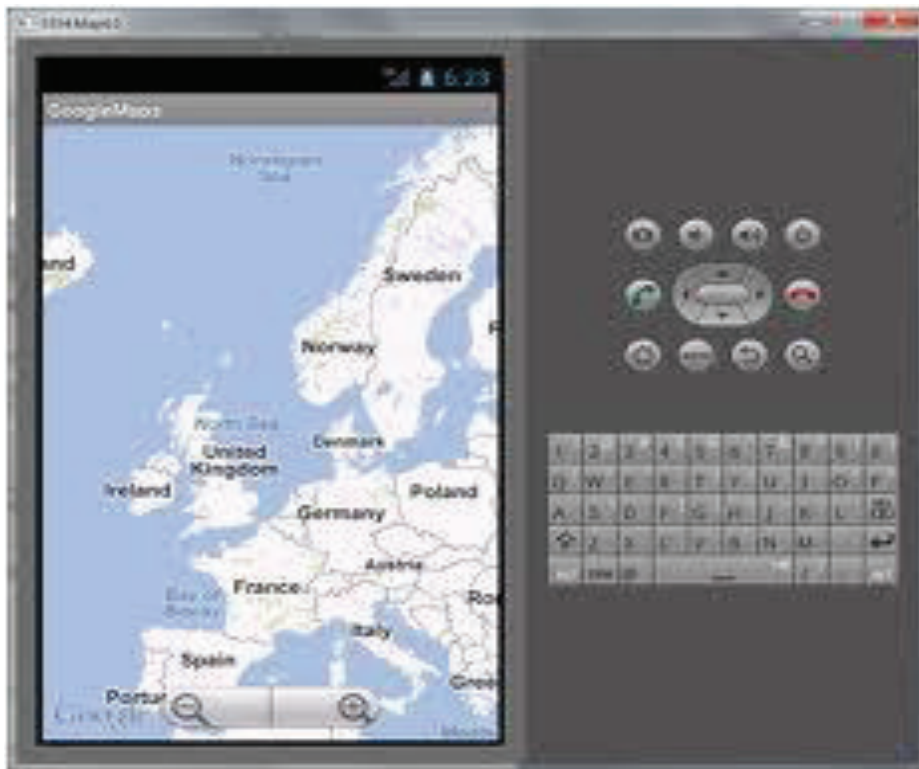
```

Αυτές οι τρεις άδειες θα ζητηθούν από το σύστημα, και ο χρήστης θα ενημερωθεί για αυτές κατά την εγκατάσταση της εφαρμογής.

4.11 Το αποτέλεσμα της εφαρμογής εικονικά







Κεφάλαιο 5^ο : Παραδείγματα και συμπεράσματα

5.1 Παραδείγματα χρήσης

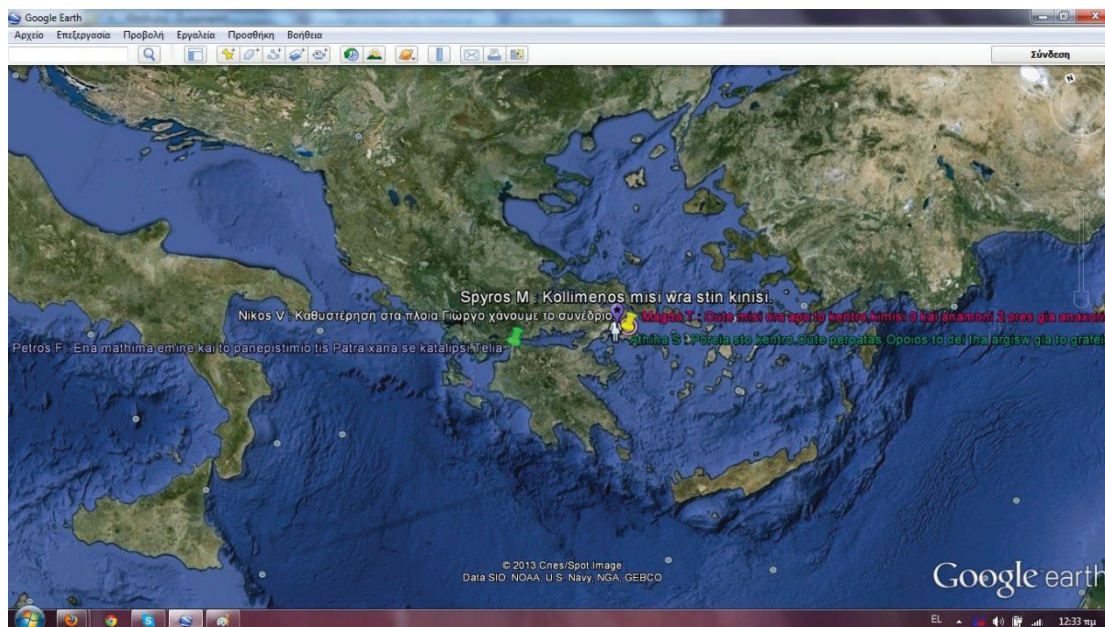
Πριν κλείσει αυτή την εργασία, θα παρουσιάσουμε 3-4 εικόνες, που με την βοήθεια των συνεργατών μου, από τον Δημόκριτο, θα μας δείξουν πως λειτουργεί η εφαρμογή που αναφέρθηκε σε αυτά τα κεφάλαια.

Στην αρχή, ελάχιστοι από τα παιδιά δέχτηκαν να περάσουν την εφαρμογή στις κινητές τους συσκευές και να την χρησιμοποιήσουν, όπως τους εξηγήθηκε, στην συνέχεια όμως όλοι συμμετείχαν για ένα όμορφο αποτέλεσμα.

Στην πρώτη φωτογραφία που ακολουθεί, τα checkinήταν λίγα, αλλά βοηθητικά.

Τα παιδιά κάνανε checkin στο μέρος που βρίσκονταν, προσθέτοντας το προσωπικό τους σχόλιο.

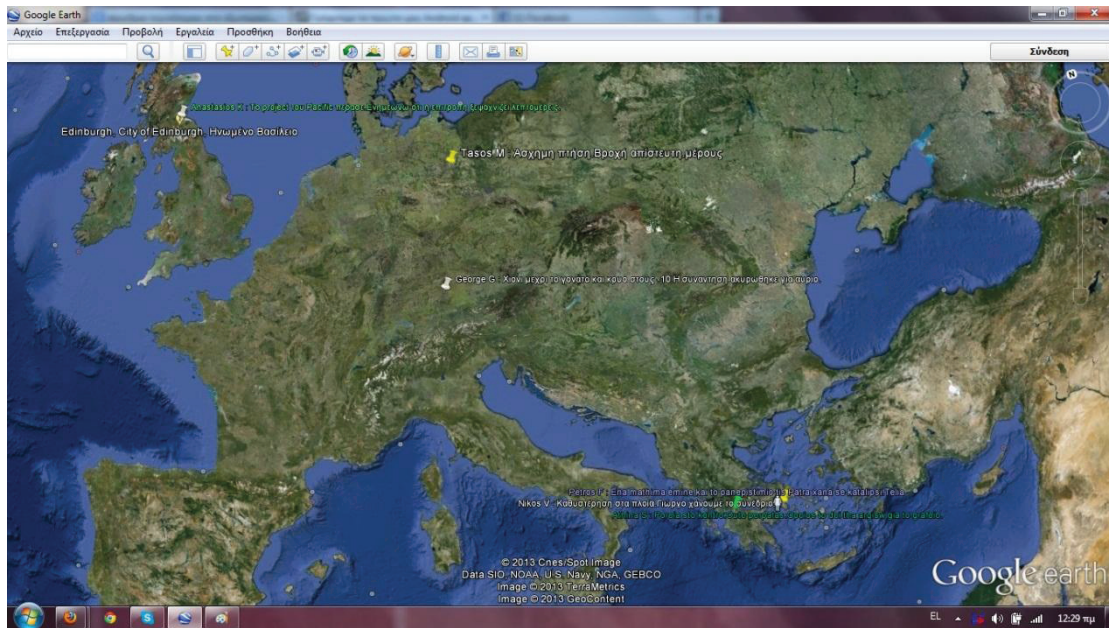
Τα checkinήταν μόνο εντός Ελλάδος.



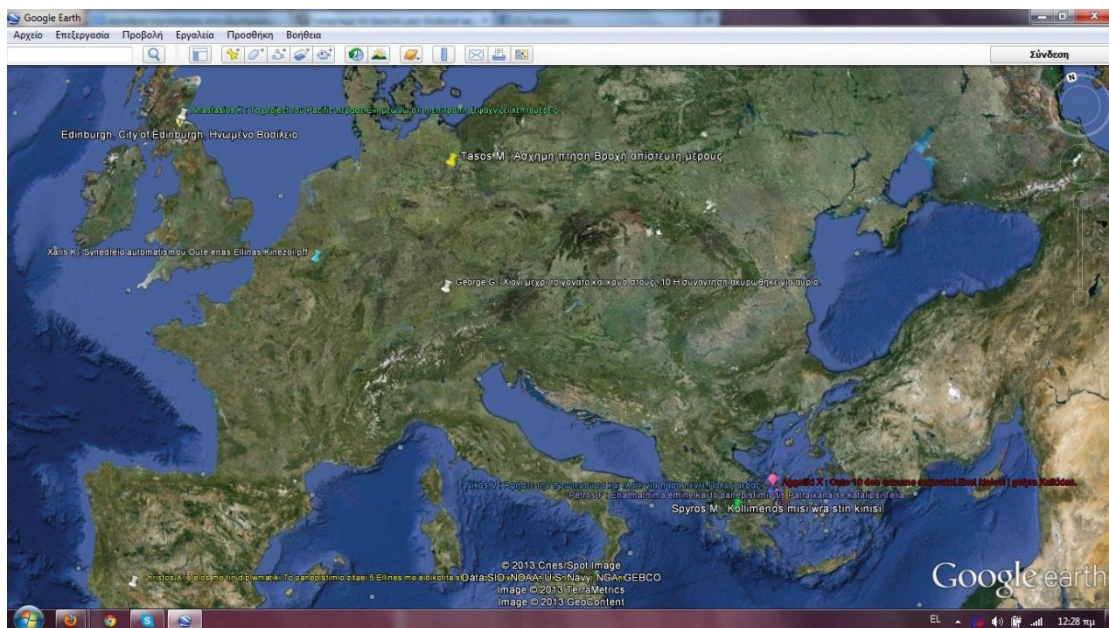
*Στίγμα και σχολιασμός στην θέση εντοπισμού
(εντός Ελλάδας)*

Σε μία συνάντηση, που ζητήθηκε να γίνει μέσα στο τμήμα Τηλεπικοινωνιών στο Ερευνητικό Κέντρο Δημόκριτος, αναλύθηκε στους συνεργάτες μου με πιο απλά λόγια η εφαρμογή και η χρήση της και τους παρουσιάστηκε σαν ένα παιχνίδι, στο οποίο μπορούσε ο καθένας τους να συμμετέχει, σε όποιο μέρος του κόσμου και να βρισκόταν

Έτσι και το αποτέλεσμα, που βλέπουμε στην φωτογραφία που ακολουθεί, έχει πλέον βρεθεί και εκτός συνόρων.



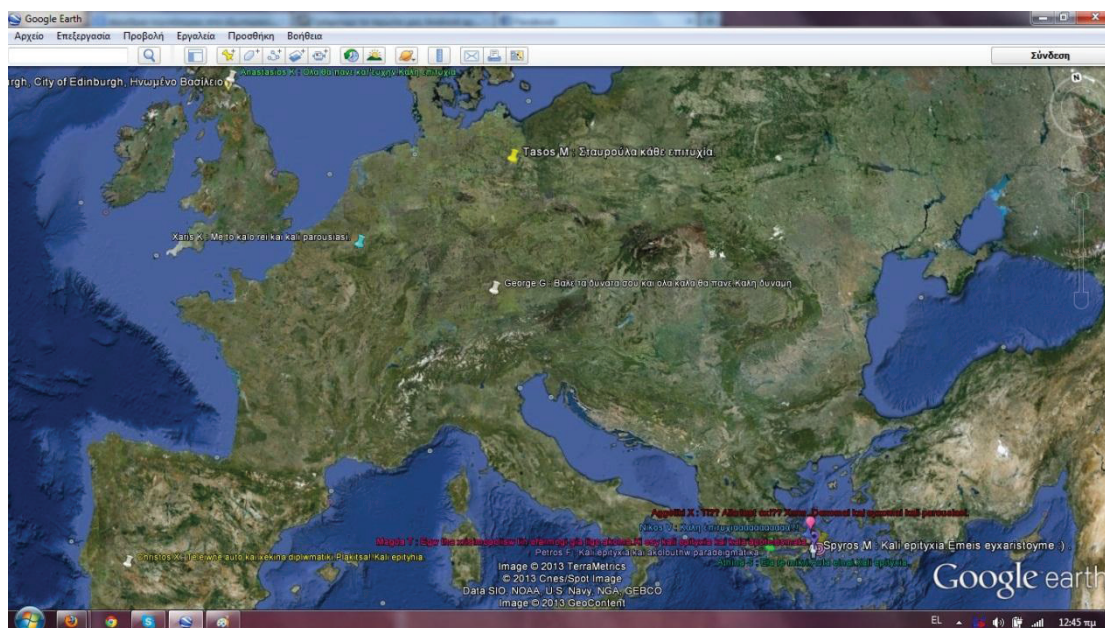
Και συνεχίστηκε για λίγο ακόμα.



Επέκταση χρήσης εφαρμογής εκτός συνόρων

Μέχρι την στιγμή που είχαμε καταφέρει να συλλέξουμε αυτά τα οποία χρειαζόνταν και τους ανακοινώθηκε ότι τους ευχαριστώ για την βοήθεια τους και ότι η δοκιμή της εφαρμογής έφτασε στο τέλος της.

Κάθε ένας, από όποιο μέρος κι αν ήταν έδωσε τις ευχές του για ένα όμορφο αποτέλεσμα που σας παρουσιάζετε τώρα.



Τελικό στίγμα για ολοκλήρωση εργασίας

Η μόνη παράλειψη, που θέλω να δηλωθεί, είναι ότι όλοι οι συνάδελφοι, των οποίων τα ονόματα αναγράφονται στις παραπάνω εικόνες, είχαν την δυνατότητα να περάσουν το checkin τους από το GoogleMaps στο GoogleEarth.

Αυτή η δυνατότητα τους δίνεται από ένα άλλο project, το οποίο αναλύεται στο εργαστήριο και έχει παρόμοιες λειτουργίες με την εφαρμογή που παρουσιάζεται σε αυτήν την εργασία.

Λόγω αυτού λοιπόν, οι παραπάνω φωτογραφίες εμφανίζονται μέσω του GoogleEarth.

5.2 Συπεράσματα

Η ανάπτυξη εφαρμογής στο Android είναι μια ιδιαίτερα απαιτητική εργασία η οποία απαιτεί σημαντικές γνώσεις Java και ταλέντο στον προγραμματισμό, πάνω από όλα όμως χρειάζεται το μεράκι του προγραμματιστή να δημιουργήσει κάτι πραγματικά χρήσιμο.

Η υλοποίηση της εφαρμογής απαιτήσε να αφιερωθεί αρκετός χρόνος στην εκμάθηση του λειτουργικού συστήματος Android, μιάς και την γλώσσα Java, ως προερχόμενη από τον κύκλο σπουδών “Δικτύων και Τηλεπικοινωνιών”, την είχα διδαχτεί.

Για να ξεκινήσει το project , λοιπόν ασχοληθήκα με την υλοποίηση της εφαρμογής, την διαδικασία του Debugging και την φάση του Beta Testing.

Συνάντησα αρκετές δυσκολίες καθότι αρκετές από τις δομές της εφαρμογής ήταν αρκετά απαιτητικές για αρχάριους προγραμματιστές, όπως η λήψη και προβολή των αρχείων xml, των RSS, και η υλοποίηση της Βάσης Δεδομένων.

Μέσω στενής συνεργασίας, με τα παιδιά στο Ερευνητικό Κέντρο Δημόκριτος και ομαδικής δουλειάς, όμως κατάφερα να φέρω εις πέρας την εργασία και να αποκτήσω στην πορεία πολλά εφόδια τα οποία αδιαμφισβήτητα θα αξιοποιήσω στο μέλλον.

Στο κεφάλαιο αυτό λοιπόν, συνοψίζονται σύντομα κάποια χρήσιμα συμπεράσματα της εφαρμογής που αναπτύχθηκε

Πλέον το Android έχει καταφέρει να εισβάλλει για τα καλά στην παγκόσμια αγορά.

Κυριαρχεί σχεδόν σε όλες τις κατασκευάστριες εταιρείες κινητών τηλεφώνων και όχι μόνο, που το επιλέγουν ως κύριο λειτουργικό σύστημα για τις φορητές συσκευές τους.

Υπάρχουν χιλιάδες δωρεάν και επίπληρωμή εφαρμογές διαθέσιμες για όλους τους χρήστες του Android.

Όπως είδαμε ο σχεδιασμός μίας Android εφαρμογής εντοπισμού θέσης, είναι μια σχετικά εύκολη και ευχάριστη διαδικασία.

Σ' αυτό έχουν συνεισφέρει πολύ τα έτοιμα εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google και μπορεί να τα βρει κανείς εντελώς δωρεάν στο Διαδίκτυο.

Σε συνεργασία με το προγραμματιστικό περιβάλλον Eclipse, το οποίο είναι ένα εύχρηστο πρόγραμμα ανοιχτού κώδικα, αποτελούν ένα πολύ ισχυρό εργαλείο για όλους τους προγραμματιστές.

Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε τη Java, η οποία είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, που χρησιμοποιείται κατά κόρον στην ανάπτυξη εφαρμογών.

Γενικά η φιλοσοφία αυτής της πτυχιακής στηρίζεται στην ιδέα του λογισμικού ανοιχτού κώδικα, το οποίο βρίσκει όλο και πιο πολλούς φανατικούς θαυμαστές τελευταία.

Το λογισμικό ανοιχτού κώδικα είναι ένα λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται ελεύθερα σε αυτούς που θέλουν να τον εξετάσουν, να τον τροποποιήσουν ή να τον χρησιμοποιήσουν σε άλλες εφαρμογές.

Αυτό σημαίνει ότι για προγράμματα ανοιχτού λογισμικού, όπως είναι οι Android εφαρμογές, υπάρχουν άπειρα παραδείγματα στο Διαδίκτυο.

Αυτά ήταν και η βάση μου για την επιλογή και την ολοκλήρωση του θέματος μου.

Βιβλιογραφία

Βήματα δημιουργίας σωστής πτυχιακής εργασίας
Ένας πρακτικός οδηγός για την συγγραφή τεχνικών
κειμένων (εκδόσεις ΙΩΝ)

Ζ. Γ. Αγιουταντης & Σ. Π. Μερτικας

Ιστορική αναδρομή

Estia.hua.gr

Σκοπός – Στόχος – Δομή

Dspace.lib.ntua.gr

Λειτουργίες χρήστη – Τεχνική περιγραφή

Eureka.lib.teithe.gr

Τεχνικά χαρακτηριστικά

www.navimania.gr/archive/index.php/t-24073.html

Θεωρητικό υπόβαθρο

el.wikipedia.orgen.wikipedia.org

Κώδικες – Βήματα προγράμματος εφαρμογής - Εργαλεία

developer.android.com

Βασικές πληροφορίες

Isag.meng.ath.gr

Γενικό

Users.auth.gr/kvek/GPS.appl.html