



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

με θέμα:



Κατασκευή εφαρμογής περιήγησης στην Ναύπακτο, σε περιβάλλον Android

Κάκκα Ελπίδα Α.Μ. 0698

Φατία Ολιάννα Α.Μ. 0882

Επιβλέπων καθηγητής: Παναγιώτης Αλεφραγκής

Ναύπακτος 2013

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος:

Ημερομηνία:

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.

2.

3.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως σκοπό να παρουσιάσει αναλυτικά το λειτουργικό σύστημα Android, ένα από τα καλύτερα λειτουργικά συστήματα στο χώρο των κινητών συσκευών.

Στα πλαίσια της εργασίας παρουσιάζεται η υπολογιστική πλατφόρμα Android εστιάζοντας στην αρχιτεκτονική, τις εκδόσεις, την ασφάλεια, πλεονεκτήματα και μειονεκτήματα, τα στατιστικά χρήσης της πλατφόρμας καθώς επίσης γίνεται και μια μικρή ανάλυση των βασικών εργαλείων ανάπτυξης εφαρμογών τέτοιου είδους.

Τέλος παρουσιάζεται η εφαρμογή περιήγησης στην πόλη της Ναυπάκτου, με σκοπό την ανάδειξη της πλατφόρμας Android μέσα από τις διεπιφάνειες του χρήστη και τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή της.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	4
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	5
ΚΕΦΑΛΑΙΟ 1 ^ο : ΕΙΣΑΓΩΓΗ.....	7
ΣΚΟΠΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.....	7
ΦΑΣΕΙΣ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.....	7
ΔΟΜΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.....	8
ΚΕΦΑΛΑΙΟ 2 ^ο : ΕΙΣΑΓΩΓΗ ΣΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID ΚΑΙ ΧΑΡΤΕΣ GOOGLE.....	9
ΤΙ ΕΙΝΑΙ ΤΟ ANDROID.....	9
ΙΣΤΟΡΙΑ ΤΟΥ ANDROID.....	9
ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID.....	10
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ANDROID.....	15
GOOGLE PLAY.....	18
ΔΟΜΗ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ.....	18
ΑΣΦΑΛΕΙΑ ΣΤΟ ANDROID.....	20
ANDROID & ANTIVIRUS.....	21
ΠΛΕΟΝΕΚΤΗΜΑΤΑ & ΜΕΙΟΝΕΚΤΗΜΑΤΑ.....	22
ΣΤΑΤΙΣΤΙΚΑ ΧΡΗΣΗΣ.....	23
ΧΑΡΤΕΣ GOOGLE.....	23
ΧΑΡΤΕΣ GOOGLE ΓΙΑ ΚΙΝΗΤΑ.....	25
ΧΑΡΤΕΣ GOOGLE ΚΑΙ ANDROID.....	25
ΚΕΦΑΛΑΙΟ 3 ^ο : ΒΑΣΙΚΑ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ANDROID.....	27
ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID.....	27
ΕΓΚΑΤΑΣΤΑΣΗ JAVA.....	27
ΕΓΚΑΤΑΣΤΑΣΗ ECLIPSE IDE.....	28
ΕΓΚΑΤΑΣΤΑΣΗ ANDROID SDK.....	29
ANDROID SDK MANAGER & AVD MANAGER.....	33
ANDROID SDK MANAGER.....	34
ANDROID AVD MANAGER.....	35
ΔΗΜΙΟΥΡΓΙΑ PROJECT ΣΤΟ ECLIPSE.....	38
ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ACTIVITIES ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	40
ΔΗΛΩΣΗ ΤΩΝ ACTIVITIES ΣΤΟ ΑΡΧΕΙΟ ANDROIDMANIFEST.XML.....	40
ΑΝΑΛΥΣΗ ΣΤΟΙΧΕΙΩΝ ANDROID.....	42
ΚΕΦΑΛΑΙΟ 4 ^ο : ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΣΤΟΧΟΙ ΥΛΟΠΟΙΗΣΗΣ.....	45
ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ ΠΕΡΙΗΓΗΣΗΣ ΣΤΗΝ ΠΟΛΗ ΤΗΣ ΝΑΥΠΑΚΤΟΥ.....	45
ΚΑΘΟΡΙΣΜΟΣ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΓΝΩΡΙΣΜΑΤΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	45

ΚΑΘΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΕΩΝ ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ (ACTIVITIES).....	46
ΚΑΘΟΡΙΣΜΟΣ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΓΝΩΡΙΣΜΑΤΩΝ ΟΘΟΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	46
ΕΦΑΡΜΟΓΗ ΕΝΟΣ ΠΡΩΤΟΤΥΠΟΥ ΕΦΑΡΜΟΓΗΣ.....	51
ΔΗΜΙΟΥΡΓΙΑ ANDROID ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ ECLIPSE.....	51
ΣΤΟΧΟΙ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	52
ΚΕΦΑΛΑΙΟ 5 ^ο : ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΕΦΑΡΜΟΓΗΣ.....	53
ΔΟΜΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	53
ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΚΕΝΤΡΙΚΗΣ ΟΘΟΝΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	53
ΤΟ ΠΡΩΤΟ ΑΡΧΕΙΟ ΠΟΥ ΘΑ ΤΡΕΞΕΙ Η ΕΦΑΡΜΟΓΗ (MAIN.JAVA).....	55
ΕΜΦΑΝΙΣΗ ΒΑΣΙΚΟΥ ΜΕΝΟΥ (OPTIIONSMENU.JAVA ΚΑΙ OPTIIONMENU2.JAVA).....	56
ΕΜΦΑΝΙΣΗ ΟΔΗΓΟΥ (GUIDE.JAVA).....	57
ΕΜΦΑΝΙΣΗ ΥΠΟ-ΟΔΗΓΟΥ (SUBGUIDE.JAVA).....	58
ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ (SHOW.JAVA).....	59
ΕΜΦΑΝΙΣΗ ΧΑΡΤΗ (MAP.JAVA).....	60
ΕΜΦΑΝΙΣΗ ΤΡΕΧΟΥΣΑΣ ΤΟΠΟΘΕΣΙΑΣ (NEARBY.JAVA).....	61
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	62
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	63

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

- Σκοπός της πτυχιακής εργασίας

Σκοπός της πτυχιακής εργασίας είναι να παρουσιάσει το λειτουργικό σύστημα της Google για κινητά Android, αλλά και το περιβάλλον ανάπτυξης εφαρμογών Eclipse IDE. Αρχικά περιγράφεται το Android, οι δυνατότητες που έχει και τα μέσα που μας παρέχει για την ευκολότερη ανάπτυξη εφαρμογών.

Στη συνέχεια περιγράφεται η διαδικασία εγκατάστασης του Eclipse, του Android SDK αλλά και εικονικών συσκευών. Γίνεται περιγραφή του περιβάλλοντός και των δυνατοτήτων του Eclipse, από την δημιουργία ενός Project μέχρι την δημιουργία του apk αρχείου και την δοκιμή του.

Κύριος σκοπός της εργασίας είναι να υλοποιηθεί μια εφαρμογή για την πόλη της Ναυπάκτου, η οποία θα διέπεται από τους κανόνες του συστήματος Android με στόχο να βοηθήσει τους κατοίκους, φοιτητές και ακόμα περισσότερο τουρίστες να γνωρίσουν την πόλη και να περιηγηθούν σε αυτήν όσο το δυνατόν ευκολότερα γίνεται.

- Φάσεις υλοποίησης πτυχιακής εργασίας

Φάση 1: Κατά την διάρκεια της πρώτης φάσης, έγινε μια ενδελεχής μελέτη του λειτουργικού συστήματος Android και της αρχιτεκτονικής του.

Φάση 2: Κατά την διάρκεια της φάσης αυτής, έγινε η επιλογή του Eclipse IDE ως εργαλείο ανάπτυξης της εφαρμογής. Ο πιο βασικός λόγος της επιλογής αυτής είναι ότι το προωθεί η Google.

Φάση 3: Κατά την διάρκεια της φάσης αυτής, μελετήθηκε το περιβάλλον του Eclipse IDE, καθώς και οι δυνατότητες του.

Φάση 4: Κατά την διάρκεια της φάσης αυτής, μελετήθηκαν οι ανάγκες της εφαρμογής περιήγησης στην πόλη της Ναυπάκτου και δημιουργήθηκε ο σκελετός των layout και των Activities που θα χρειαστούν.

Φάση 5: Κατά την διάρκεια της φάσης αυτής, σχεδιάστηκαν τα layout και μελετήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού Java, για την υλοποίηση του κώδικα της εφαρμογής.

Φάση 6: Κατά την διάρκεια της φάσης αυτής, η εφαρμογή δοκιμάστηκε σε πραγματικές συσκευές και έγιναν κάποιες διορθώσεις.

Φάση 7: Στην τελευταία φάση έγινε εξαγωγή συμπερασμάτων για το λειτουργικό σύστημα Android και καταγράφονται οι προοπτικές της εφαρμογής, ύστερα από περαιτέρω εξέλιξη και επεξεργασία αυτής.

- **Δομή πτυχιακής εργασίας**

Αρχικά υπάρχει μια σύντομη περίληψη της εργασίας στην οποία αναφέρονται συνοπτικά τα κύρια σημεία της και στην συνέχεια ακολουθεί ο πίνακας περιεχομένων. Η κύρια εργασία αποτελείται από 5 κεφάλαια των οποίων το περιεχόμενο περιγράφεται παρακάτω.

Κεφάλαιο 1: Εισαγωγή

Πρόκειται για το παρόν κεφάλαιο στο οποίο παρουσιάζεται συνοπτικά το θέμα της εργασίας, οι φάσεις εκπόνησής της, καθώς και το παρόν υποκεφάλαιο που καταδεικνύει την δομή που υιοθετήθηκε.

Κεφάλαιο 2: Εισαγωγή στο λειτουργικό σύστημα Android και χάρτες Google

Το κεφάλαιο αυτό χωρίζεται σε επιμέρους κεφάλαια που παρουσιάζουν την ορολογία του Android, τις εκδόσεις του, την ιστορία του, την αρχιτεκτονική του, τεχνικές ασφαλείας, τον ρόλο του Antivirus σε συσκευές Android καθώς και μερικά από τα πλεονεκτήματα και μειονεκτήματα του λειτουργικού συστήματος. Επίσης αναφερόμαστε στην διαδικτυακή εφαρμογή υπηρεσιών χαρτογράφησης τους χάρτες Google.

Κεφάλαιο 3: Βασικά εργαλεία ανάπτυξης εφαρμογών Android

Αυτό το κεφάλαιο χωρίζεται επίσης σε υποκατηγορίες στις οποίες παρουσιάζεται βήμα προς βήμα η εγκατάσταση όλων των απαραίτητων εργαλείων για την ανάπτυξη εφαρμογών στο περιβάλλον Android, ο κύκλος ανάπτυξης μιας εφαρμογής ξεκινώντας από καθαρό υπολογιστή μέχρι το τελικό .apk αρχείο.

Κεφάλαιο 4: Σχεδιασμός εφαρμογής και στόχοι υλοποίησης

Σε αυτό το κεφάλαιο γίνεται μια παρουσίαση του σχεδιασμού της εφαρμογής που αναλύει τον καθορισμό των χαρακτηριστικών γνωρισμάτων και των απαιτήσεων των δραστηριοτήτων και στο τέλος αναφέρονται οι στόχοι υλοποίησης της εφαρμογής περιήγησης.

Κεφάλαιο 5: Ανάλυση προβλήματος εφαρμογής

Σε αυτό το τελευταίο κεφάλαιο γίνεται μια ανάλυση της εφαρμογής και επεξηγείται ο πηγαίος κώδικας της.

ΚΕΦΑΛΑΙΟ 2

Εισαγωγή στο Λειτουργικό Σύστημα Android

- Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο σε Linux, για κινητά τηλέφωνα και πολλές άλλες χρήσιμες συσκευές που δουλεύουν σε περιβάλλον android όπως tablets, media internet devices (MID), netbooks κτλ. Υλοποιείται σε περιβάλλον Linux, το λογισμικό του είναι γραμμένο σε java και οι βιβλιοθήκες λογισμικού έχουν αναπτυχθεί από την google. Για τις ανάγκες αποθήκευσης κάνει χρήση της SQLite. Η περιήγηση στον ιστό βασίζεται στην ανοιχτή τεχνολογία Web Kit. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινο μήλο και σχεδιάστηκε από τη γραφίστρια Irina Blok.



Εικόνα 2.1: Λογότυπο του Android

- Ιστορία του Android

Όλα αρχίζουν στο Palo Alto της California τον Οκτώβριο 2003 από μια ομάδα προγραμματιστών, Andy Rubin, Rich Miner, Nick Sears και Chris White θέλοντας να αναπτύξουν όπως τότε είχε δηλώσει ο Andy Rubin "...smarter mobile devices that are more aware of its owner's location and preferences". Έπειτα από δύο χρόνια αυτόνομης λειτουργίας και ανάπτυξης η Google τον Αύγουστο του 2005, εξαγοράζει την Android Inc. με σκοπό να εισέλθει στην αγορά κινητής τηλεφωνίας. Προκειμένου να κυκλοφορήσει το λογισμικό στην αγορά το Νοεμβριο του 2007 ιδρύεται η Open handset alliance, μια κοινοπραξία 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Περίπου ένα χρόνο αργότερα (22/10/2008) κυκλοφόρησε το πρώτο εμπορικά διαθέσιμο κινητό στην αγορά, που λειτούργησε σε περιβάλλον android. Πρόκειται για το HTC Dream με τα παρακάτω χαρακτηριστικά:

- Οθόνη: αφής TFT 65 k χρωμάτων
- Διαστάσεις : 117.7 x 55.7 x 17.1 mm (4.63 x 2.19 x 0.67 in)
- Μνήμη : κάρτα μνήμης microSD, συνολική 192 MB RAM, 256 MB ROM
- GPRS Class 10 (4+1/3+2 slots), 32 - 48 kbps
- Bluetooth Yes v2.0 with A2DP, headset support only
- Λειτουργικό σύστημα: Android OS, v1.6 (Donut)

- Chipset: Qualcomm MSM7201A
- CPU: Adreno 130, 528 MHz ARM 11
- GPS
- HTML browser
- Java
- Wi-Fi 802.11 b/g
- USB



Εικόνα 2.2: HTC Dream

• Εκδόσεις του Android

Από τότε και μέχρι σήμερα η Google έχει βγάλει στην αγορά ακόμη 8 εκδόσεις λογισμικού, γεγονός που δείχνει την γρήγορη εξέλιξη της εταιρίας, αν λάβουμε υπ' όψη μας ότι η Microsoft μέσα σε 25 χρόνια έχει διαθέσει στην αγορά μόλις 10 εκδόσεις των Windows. Η Google ακολουθεί μια αλφαβητική συνέπεια στις κώδικες ονομασίες των εκδόσεων του Android οι οποίες και έχουν πάντα σαν θέμα κάποιο γλύκισμα (με εξαίρεση τις δυο πρώτες που ήταν και πρώιμες).

- 1.0 **A**stro
- 1.1 **B**ender
- 1.5 **C**upcake
- 1.6 **D**onut
- 2.0 **E**clair
- 2.1 **F**royo
- 2.3 **G**ingerbread
- 3.0 **H**oneycomb
- 4.0 **I**ce Cream Sandwich
- 4.1 **J**elly Bean
- 5.0 **K**ey Lime Pie

Android 1.5 Cupcake

Η έκδοση 1.5 (CupCake) παρουσιάστηκε στις 30 Απριλίου το 2009 και σας δίνει την δυνατότητα για ανέβασμα βίντεο στο YouTube και εικόνων στο Picasa απευθείας από την κινητή σας συσκευή, για αυτόματη σύνδεση ακουστικών headset σε συγκεκριμένη απόσταση, ενώ περιλαμβάνει εικονικό πληκτρολόγιο με πρόβλεψη λέξεων και νέα για widgets για την αρχική οθόνη.



Εικόνα 2.3: Android 1.5 Cupcake

Android 1.6 Donut

Η έκδοση 1.6 (Donut) παρουσιάστηκε στις 15 Σεπτεμβρίου το 2009 και παρουσιάζει ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση. Επίσης δίνεται έμφαση στη φωνητική αναζήτηση αλλά και στην αναζήτηση από την αρχική οθόνη με bookmarks, ιστορικό, επαφές κ.ά., ενώ υποστηρίζονται και οθόνες αναλύσεων WVGA.



Εικόνα 2.4: Android 1.6 Donut

Android 2.0/2.1 Éclair

Οι επόμενες γλυκές εκδόσεις είναι η Éclair 2.0 που παρουσιάστηκε στις 26 Οκτωβρίου το 2009 και τον Ιανουάριο του 2010 η επανέκδοση σε Android 2.1 Éclair, που προχωρούν ακόμα πιο μπροστά, διαθέτοντας νέο browser UI, υποστήριξη HTML5, Google Maps 3.1.2, ενσωματωμένη υποστήριξη για flash στην κάμερα, ψηφιακό ζουμ, δυνατότητα αντίληψης multi-touch, βελτιωμένο εικονικό πληκτρολόγιο, Bluetooth 2.1 και live wallpapers.



Εικόνα 2.5 : Android 2.0/2.1 Éclair

Android 2.2 Froyo

Η έκδοση του Froyo (Froyen Yogurt) παρουσιάστηκε στις 20 Μαΐου το 2010. Η έκδοση αυτήν αναβάθμισε αισθητά την ταχύτητα του OS, αλλά και τη γενικότερη απόδοση, διαθέτει Market με δυνατότητα αυτομάτων updates, επιλογή εγκατάστασης εφαρμογών στην κάρτα μνήμης, υποστήριξη Adobeflash 10.1 ενώ ενσωματώνει τον Chrome V8 JavaScript στα browser applications. Στη FroYo βρίσκουμε και τη δυνατότητα χρήσης της συσκευής για διαμοιρασμό Internet μέσω Wi-Fi σε άλλες συσκευές (tethering).



Εικόνα 2.6: Android 2.2 Froyo

Android 2.3 Gingerbread

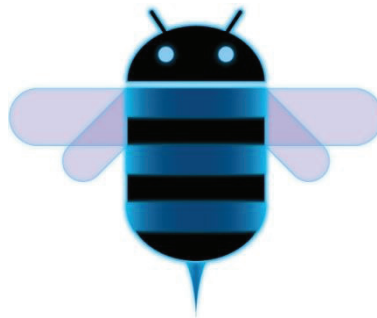
Η έκδοση Gingerbread παρουσιάστηκε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Υποστηρίζει πολύ μεγάλα μεγέθη οθονών και αναλύσεων, διαθέτει προεγκατεστημένη υποστήριξη για τηλεφωνικές κλήσεις μέσω Internet (VoIP), επανασχεδιασμένο multi-touch πληκτρολόγιο, λειτουργίες copy-paste σε όλο το λειτουργικό, download manager για κατέβασμα μεγάλων αρχείων, καθώς και προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες.



Εικόνα 2.7: Android 2.3 Gingerbread

Android 3.0 Honeycomb

Η έκδοση Honeycomb παρουσιάστηκε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets. Διαθέτει ανανεωμένο περιβάλλον με αρκετά 3D στοιχεία, καλύτερο web browser με tabs και ανανεωμένο εικονικό πληκτρολόγιο, βελτιστοποιημένη εφαρμογή για ανάγνωση Google e-books, ειδική έκδοση του Gmail για tablets, δυνατότητα βιντεοκλήσεων μέσω εφαρμογής Google Talk και ανανεωμένη GoogleMaps.



Εικόνα 2.8: Android 3.0 Honeycomb

Android 4.0 Ice Cream Sandwich

Η έκδοση Ice Cream Sandwich παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Η έκδοση αυτή διαθέτει στο User Interface εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν στις συσκευές. Βελτίωση της ασφάλειας του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων.



Εικόνα 2.9: Android 4.0 Ice Cream Sandwich

Android 4.1/4.2 Jelly Bean

Τέλος η έκδοση Jelly Bean παρουσιάστηκε μόλις τον περασμένο Ιούνιο (2012) δίνοντας βάση στο περιβάλλον χρήσης κάνοντας το πιο εύχρηστο. Διαθέτει αξιοσημείωτες αλλαγές στην κάμερα, όπως πανοραμική λήψη, βελτιωμένη φωνητική πληκτρολόγηση, αλλαγές στις ενημερώσεις και μερικές νέες υπηρεσίες όπως η Google Now.



Εικόνα 2.10: Android 4.1/4.2 Jelly Bean

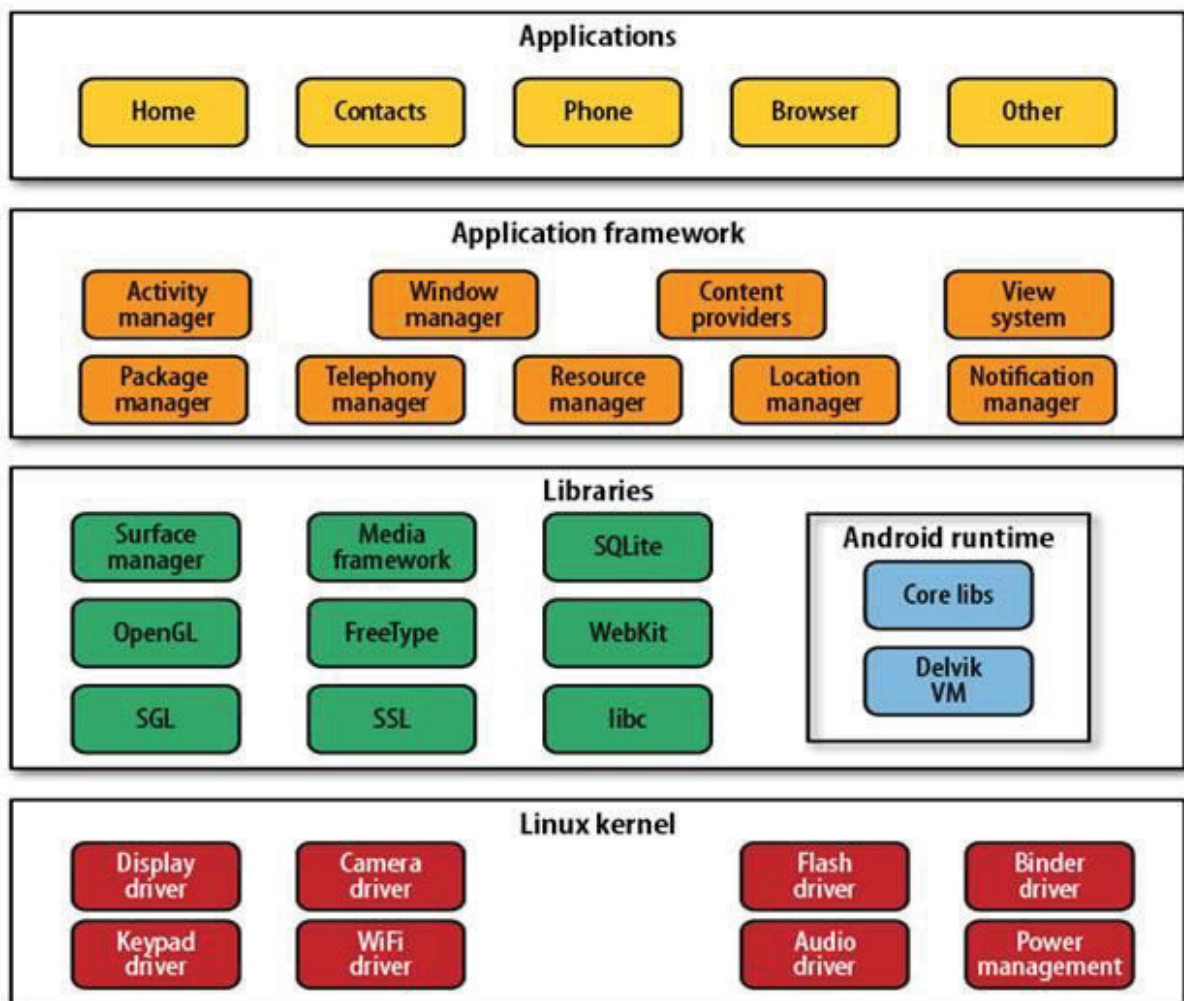
Android 5.0 Key Lime Pie

Όσο για την επόμενη έκδοση του Android φημολογείται ότι θα είναι διαθέσιμη στα τέλη του Οκτώβρη και αναμένεται να τρέχει ακόμα και σε συσκευές με 512MB RAM. Η Google θέλει να βελτιστοποιήσει την τελευταία έκδοση του Android, ώστε να είναι ικανό να τρέξει σε smartphones με μικρή μνήμη, έστω και στα 512MB, χωρίς προβλήματα, όπως συμβαίνει με το iOS και το Windows Phone OS. Για να δούμε τι θα ακολουθήσει λοιπόν..Θα είναι το νέο λειτουργικό καλύτερο από το υπάρχον...;;;



Εικόνα 2.11: Android 5.0 Key Lime Pie

- Αρχιτεκτονική του Android



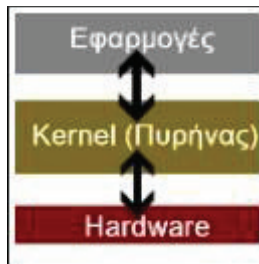
Εικόνα 2.12: Αρχιτεκτονική Android

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από 5 βασικά επίπεδα:

- Τον πυρήνα Linux (Linux Kernel)
- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries)
- Τον χρόνο εκτέλεσης (Android Runtime)
- Το περιβάλλον εφαρμογών (Application Framework)
- Εφαρμογών (Applications)

Πυρήνας Linux (Linux Kernel)

Στην βάση της στοίβας λογισμικού είναι ο τροποποιημένος πυρήνας του συστήματος που βασίζεται στην έκδοση 2.6 του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος που 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.



Εικόνα 2.13: Πυρήνας Kernel

Εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries)

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες οι οποίες ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες δεν αποτελούν από μόνες τους εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία τους. Στο σύνολο τους είναι γραμμένες σε C/C++ και έχουν μεταγλωττιστεί για την χρήση τους από το λειτουργικό. Μερικές από τις βιβλιοθήκες του Android είναι:

- **System C library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
- **Media Libraries** – 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 πλοήγηση στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- **SGI** – η γνωστή μηχανή δισδιάστατων γραφικών
- **3D Libraries** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. 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

Χρόνος εκτέλεσης (Android Runtime)

Η Google επέλεξε την JAVA για την ανάπτυξη Android εφαρμογών. Παρόλα αυτά επέλεξε να μην χρησιμοποιεί την κλασική εικονική μηχανή της Java (JVM – Java Virtual Machine) αλλά να στηριχτεί σε μια νέα εικονική μηχανή κατάλληλη για φορητές υπολογιστικές πλατφόρμες, την Dalvik. Κάθε εφαρμογή Android τρέχει σε ξεχωριστή διεργασία η οποία διεργασία αποτελεί μια αναφορά στην εικονική μηχανή Dalvik που χρησιμοποιεί το Android. Η εικονική αυτή μηχανή έχει γραφτεί έτσι ώστε μια συσκευή να μπορεί να τρέχει

πολλαπλές εικονικές μηχανές αποτελεσματικά. Σε κλασικά περιβάλλοντα Java, ο πηγαίος κώδικας μεταγλωττίζεται σε Java bytecode και αποθηκεύεται σε αρχείο με κατάληξη .class.

Τα αρχεία αυτά διαβάζονται από την εικονική μηχανή της Java κατά την εκτέλεση. Στην πλατφόρμα του Android ο πηγαίος κώδικας μεταγλωττίζεται και προκύπτει αρχείο .class αλλά μέσω του εργαλείου «dx» μετατρέπεται σε αρχείο της μορφής .dex και στην συνέχεια σε εκτελέσιμο αρχείο. Ενώ ένα αρχείο με κατάληξη .class περιέχει μόνο μια κλάση, ένα αρχείο με κατάληξη .dex μπορεί να περιέχει πολλές κλάσεις. Τα αρχεία με κατάληξη .dex εκτελούνται από την εικονική μηχανή Dalvik. Η εικονική μηχανή Dalvik βασίζεται στον πυρήνα του Linux παρέχοντας ισχυρή λειτουργικότητα στον χειρισμό νημάτων και χαμηλή κατανάλωση μνήμης. Πέρα από τη χρήση μιας διαφορετικής εικονικής μηχανής, η Google αποφάσισε να στηριχτεί και σε μια διαφορετική και σε κάποιες πιο εξειδικευμένη υλοποίηση των κλασικών βιβλιοθηκών της Java.

Περιβάλλον εφαρμογών (Application Framework)

Πάνω από τις εγγενής βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής. Αυτό το επίπεδο μας παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών μας. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια. Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής Δραστηριοτήτων - Activity Manager: Υπεύθυνο για τον έλεγχο του χρόνου ζωής (Σχήμα 1.4) των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.
- Διαχειριστής Πόρων - Resource Manager: Οι πόροι, είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.
- Διαχειριστής Κοινοποιήσεων - Notification Manager: Ιδανικός τρόπος για να ενημερώνεις τον χρήστη για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεις την εργασία του.

Εφαρμογές (Applications)

Στο υψηλότερο επίπεδο της στοίβας Android, φιγουράρουν οι εφαρμογές και τα widgets. Αυτό είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την όλη από κάτω διαδικασία. Αυτές είναι εφαρμογές που γράφουν οι κατασκευαστές λογισμικού, εφαρμογές που ήδη είναι εγκατεστημένες στο τηλέφωνο ή που ο χρήστης παίρνει από το Android Market. Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Από την άλλη τα widget λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή.

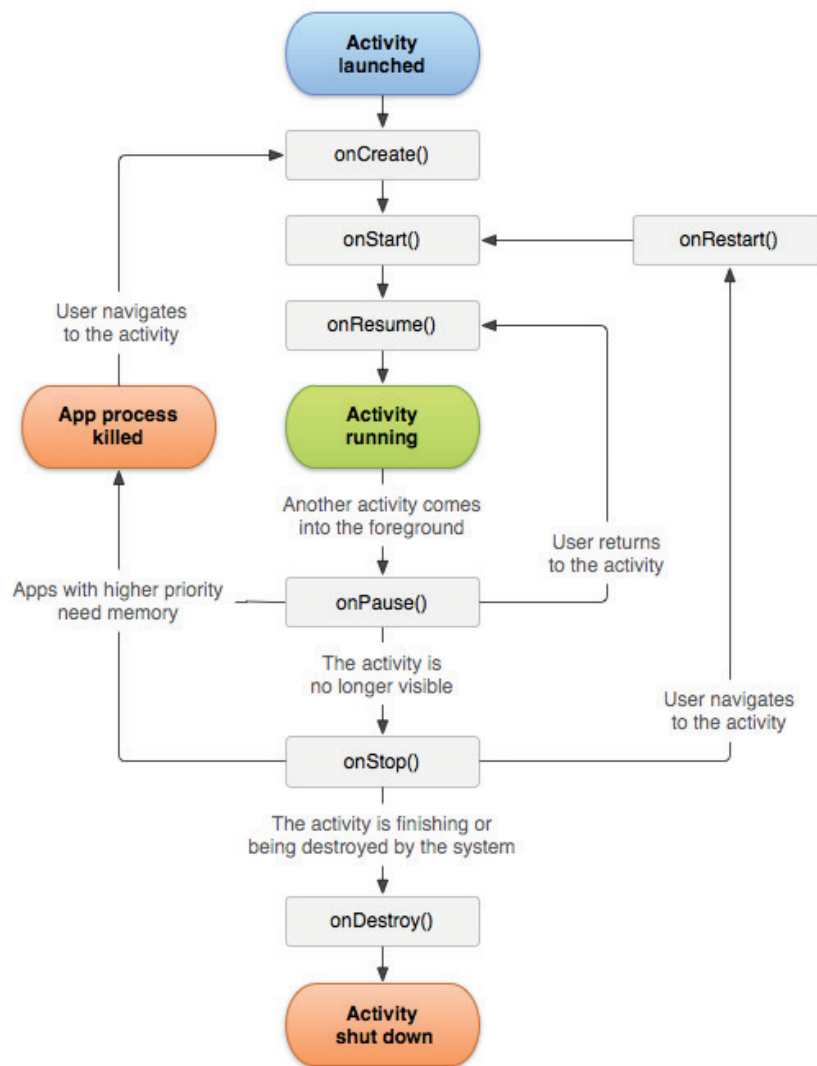
- **Google Play**

Το Google Play, αντικαταστάτης του Android Market, είναι ένα online κατάστημα της Google το οποίο προσφέρει σε κάθε χρήστη εφαρμογές επί πληρωμή ή και ακόμα περισσότερες δωρεάν . Πρόκειται ουσιαστικά για μια υπηρεσία που συγκεντρώνει όλη τη μουσική, τα videos/ταινίες, τις εφαρμογές και τα eBooks σε ένα μέρος το οποίο είναι προσβάσιμο από οποιονδήποτε υπολογιστή, Android smartphone ή Android tablet. Η αποθήκευση των αρχείων γίνεται σε cloud και φυσικά είναι διαθέσιμα κάθε στιγμή και από οπουδήποτε.

- **Δομή μιας εφαρμογής**

Μια εφαρμογή στο Android έχει διαφορετική φιλοσοφία στη δομή της από την κλασική δομή. Συνήθως υπάρχει ένα εκτελέσιμο αρχείο που περιέχει την λογική του προγράμματος χρησιμοποιώντας εξωτερικούς πόρους, δηλαδή είτε τοπικά αρχεία εικόνων, είτε κείμενο που θα ανακτηθεί από το δίκτυο κλπ. Στο Android δεν υπάρχει ένα εκτελέσιμο αρχείο, αλλά ως εφαρμογή εννοούμε μια ομάδα αρχείων τα οποία αντιπροσωπεύουν τα Activities της εφαρμογής. Αυτά μπορεί να είναι προσπελάσιμα και από άλλες εφαρμογές.

Για να ξεκινήσει μία Activity θα πρέπει να έχει δηλωθεί ως startup αλλιώς θα πρέπει να περιμένει κάποιο Intent. Τα Intents είναι αντικείμενα που περιέχουν είτε κείμενο είτε άλλου τύπου δεδομένα όπως εικόνα, ήχος κλπ. Με αυτό τον τρόπο μπορεί να έχουμε φτιάξει μια εφαρμογή που να χρησιμοποιεί σε κάποιο σημείο τον PhoneDialer και περνώντας του το κατάλληλο Intent, το Android ξεκινάει τον PhoneDialer. Η λογική αυτή μας βοηθάει καθώς τα έτοιμα κομμάτια εφαρμογών δεν χρειάζεται να τα κατασκευάσουμε από την αρχή, αλλά μπορούμε να τα χρησιμοποιήσουμε χωρίς ο χρήστης του κινητού τηλεφώνου να αντιλαμβάνεται την εναλλαγή μεταξύ των εφαρμογών.



Εικόνα 2.14: Ο κύκλος ζωής μιας δραστηριότητας Android

Το σύστημα διαχειρίζεται τις Activities σαν μια στοίβα από Activities. Όταν μια νέα Activity ξεκινάει, τοποθετείται στην κορυφή της στοίβας και περνάει στην εκτελέσιμη κατάσταση. Η προηγούμενη Activity παραμένει πάντα πιο κάτω στην στοίβα και δεν έρχεται στο προσκήνιο αν δεν τελειώσει η καινούρια. Αξίζει να αναφερθεί ότι κάθε Activity έχει τον δικό της κύκλο ζωής. Οι καταστάσεις στις οποίες μπορεί να βρίσκεται μια Activity είναι:

- Αν μια Activity βρίσκεται στο προσκήνιο, δηλαδή στην κορυφή της στοίβας, τότε είναι ενεργή (active) ή αλλιώς εκτελείται (running).
- Αν μια Activity έχει χάσει την εστίασή της, δηλαδή κάποια άλλη είναι ενεργή πάνω από αυτή, τότε είναι σε αναστολή (paused). Μια Activity σε αναστολή είναι απολύτως «ζωντανή», διατηρεί όλες της τις πληροφορίες, αλλά το σύστημα μπορεί οποιαδήποτε στιγμή να τη «σκοτώσει» λόγω χαμηλών πόρων.
- Αν μια Activity επισκιαστεί από κάποια άλλη, τότε είναι σε διακοπή (stopped). Επίσης διατηρεί όλες της τις πληροφορίες, αλλά δεν είναι στο προσκήνιο. Το σύστημα μπορεί να τη «σκοτώσει» λόγω χαμηλών πόρων.
- Μια Activity έχει ανασταλεί ή διακοπεί, το σύστημα μπορεί να τη διαγράψει από τη μνήμη είτε καλώντας τη να τελειώσει (finish), ή απλά με τη «διακοπή» της διαδικασίας. Όταν εμφανίζεται και πάλι στο χρήστη, θα πρέπει να επανέλθει στην προηγούμενη κατάσταση.

Οι μέθοδοι που μπορούν να χρησιμοποιηθούν σε μία Activity είναι:

- onCreate (Bundle): καλείται όταν ξεκινάει η Activity για πρώτη φορά και μπορούμε να τη χρησιμοποιήσουμε για να προβάλλουμε το User Interface. Παίρνει μια παράμετρο, είτε είναι κενό (null) είτε την προηγούμενη κατάσταση στην οποία βρισκόταν, αφού την είχαμε «σώσει» με την μέθοδο onSaveInstanceState().

- onStart(): υποδεικνύει ότι η Activity πρόκειται να εμφανιστεί στον χρήστη.

- onResume(): καλείται όταν η Activity είναι έτοιμη για αλληλεπίδραση ,με τον χρήστη. Εδώ μπορούμε να ξεκινήσουμε τα animation και τη μουσική.

- onPause(): καλείται όταν η Activity περάσει σε αναστολή. Όταν χρησιμοποιούμε αυτή

την μέθοδο καλό είναι να αποθηκεύουμε τις πληροφορίες που θέλουμε, αφού μπορεί να είναι η τελευταία κατάσταση στην οποία θα βρεθεί η Activity.

- onStop(): καλείται όταν η Activity περάσει σε διακοπή (stopped). Αν οι πόροι του συστήματος είναι χαμηλοί, το σύστημα μπορεί να καταστρέψει την Activity χωρίς να την καλέσει.

- onStart(): καλείται όταν η Activity πρόκειται να ξαναεμφανιστεί μετά από μια κατάσταση διακοπής.

- onDestroy(): καλείται ακριβώς πριν η Activity καταστραφεί. Είναι πιθανό όμως να μην κληθεί ποτέ λόγω χαμηλών πόρων.

• Ασφάλεια στο Android

Γιατί μας ενδιαφέρει η ασφάλεια του Android;

Ως developers:

- Ανάπτυξη εφαρμογών που εξασφαλίζουν τα προσωπικά δεδομένα των χρηστών και δεν δημιουργούν κενά ασφάλειας στο λειτουργικό σύστημα.

Ως χρήστες:

- Τα smartphones πλέον φιλοξενούν μεγάλο πλήθος προσωπικών δεδομένων (Mail, Contacts, Facebook, Twitter κ.α.)
- Προστασία των δεδομένων μας και της ιδιωτικότητας μας
- Προστασία από κακόβουλες χρεώσεις
- Σταθερή λειτουργία της συσκευής μας

Θέλουμε να ελέγχουμε εμείς το τηλέφωνο μας και όχι αυτό εμάς!!

Η ραγδαία αύξηση της χρήσης των smartphones είναι άμεσα συνδεδεμένη με την αύξηση των αναδυόμενων απειλών. Καθώς τα smartphones χρησιμοποιούνται για εργασία και για διασκέδαση, οι συσκευές αυτές γίνονται ολοένα και πιο ελκυστικές για τους κυβερνοεγκληματίες. Μέχρι στιγμής, η λογική που ακολουθούν είναι αντίστοιχη με αυτή των απειλών εναντίον παλαιότερων εφαρμογών Windows, οι οποίες προσέφεραν στους

hackers πολλές δυνατότητες, ενώ παράλληλα επέτρεπαν τη δημιουργία πολλών διαφορετικών τύπων malware.

Το πρώτο κακόβουλο πρόγραμμα που εμφανίστηκε για το Android ήταν ένα Trojan-SMS το οποίο διείσδυε στα smartphones με λειτουργικό Android υπό τη μορφή media player και έστελνε SMS σε αριθμούς υψηλής χρέωσης, χωρίς να το καταλαβαίνει ο χρήστης. Σήμερα, οι αναλυτές εντοπίζουν πιο εξελιγμένες απειλές, οι οποίες έχουν ακόμη και δυνατότητες δημιουργίας κακόβουλων δικτύων botnet. Παράλληλα έχει τονιστεί ότι οι χρήστες θα πρέπει να δείχνουν ιδιαίτερη προσοχή κατά την εγκατάσταση εφαρμογών, ειδικά όσον αφορά στις υπηρεσίες στις οποίες ζητά πρόσβαση μια εφαρμογή. Σε αυτές περιλαμβάνονται υπηρεσίες υψηλής χρέωσης, οι οποίες αφορούν SMS και τηλεφωνικές κλήσεις.

Όταν κατά την εγκατάσταση μιας εφαρμογής, ο χρήστης συμφωνεί να επιτρέψει την πρόσβαση σε αυτές τις υπηρεσίες, το smartphone μπορεί να κάνει κλήσεις ή να στέλνει SMS χωρίς να απαιτείται έγκριση από το χρήστη. Ότι αφορά σε εταιρικό επίπεδο, η διαφορά του Android με άλλες ανταγωνιστικές πλατφόρμες, όπως το iOS της Apple ή η πλατφόρμα Blackberry, βρίσκεται στην υποδομή. Για τις προαναφερθείσες πλατφόρμες υπάρχουν policy servers στις επιχειρήσεις, οι οποίοι επιτρέπουν τον περιορισμό και τον έλεγχο των χρηστών. Αντίθετα ο ανοιχτός χαρακτήρας του Android, ο οποίος αποτελεί ένα από τα μεγαλύτερα πλεονεκτήματά του, μπορεί να αποτελέσει σοβαρό μειονέκτημα όσον αφορά στην εταιρική χρήση. Η ανοιχτή φιλοσοφία της πλατφόρμας Android δεν επιτρέπει τον έλεγχο των χρηστών, γεγονός που μπορεί να ευνοήσει τις ηλεκτρονικές απειλές.

Συμβουλές για τους χρήστες της πλατφόρμας Android

- Να μην κάνουν rooting ή jailbreaking στη φορητή συσκευή τους
- Να είναι πολύ προσεκτικοί όσον αφορά στην πρόσβαση που απαιτούν οι εφαρμογές σε διάφορες υπηρεσίες
- Να έχουν υπόψη ότι οι εφαρμογές του Android Market είναι άμεσα συνδεδεμένες με τους λογαριασμούς Gmail. Αυτό σημαίνει ότι σε περίπτωση που παραβιαστεί ένας λογαριασμός Gmail, οι κυβερνοεγκληματίες μπορούν να αποκτήσουν εύκολα πρόσβαση και σε εφαρμογές Android
- Να είναι προσεκτικοί στις διαδικασίες εγκατάστασης εφαρμογών. Καθώς η έγκριση εγκατάστασης γίνεται online, οι εφαρμογές μπορούν να εγκατασταθούν άμεσα και αυτόματα στο smartphone, χωρίς την έγκριση του χρήστη. Η μόνη ένδειξη είναι το εικονίδιο της εφαρμογής που δείχνει ότι γίνεται download. Δυστυχώς, δεν υπάρχει τρόπος να διακοπεί η αυτόματη εγκατάσταση εφαρμογών.

• Android & Antivirus

Στο Android δεν υπάρχουν ιοί (virus) ακόμα, όπως και σε όλα τα Linux-based συστήματα. Αυτό που κάνουν λοιπόν τα antivirus για να μας προστατέψουν είναι μεταξύ άλλων να διαβάζουν τη λίστα των αδειών και να μας λένε αν βρίσκουν κάτι ύποπτο. Αυτό όμως είναι τελείως υποκειμενικό καθώς ύποπτη μπορεί να θεωρείται και η σύνδεση με το Google

λογαριασμό μας, κατ' επέκταση και το Gmail βλαβερό. Όταν λοιπόν ένα antivirus μας προειδοποιεί για μία κακόβουλη εφαρμογή, το καλύτερο που έχουμε να κάνουμε είναι να διαβάσουμε το λόγο και να ενημερωθούμε από τις άδειές της αν είναι λογικό ή όχι.

Από τα παραπάνω λοιπόν καταλαβαίνουμε ότι οι άδειες είναι ένα τεράστιο αλλά ανεκμετάλλετο εργαλείο στην καταπολέμηση των βλαβερών εφαρμογών. Είναι στο χέρι μας να τις διαβάζουμε πριν την εγκατάσταση αλλά και μετά, αν θεωρούμε ότι κάτι δεν πάει καλά. Το Android λοιπόν μας δίνει τη δύναμη, αρκεί να την εκμεταλλευτούμε και να θυμάστε πάντα ότι συνήθως η απροσεξία μας είναι αυτή που προκαλεί τη μόλυνση του συστήματος.

• Πλεονεκτήματα και μειονεκτήματα του Android

Πλεονεκτήματα

1. **Multitasking** – Οι συσκευές με Android λογισμικό μπορούν να τρέξουν πολλές εφαρμογές την ίδια στιγμή. Ο χρήστης δηλαδή μπορεί να περιηγηθεί στο διαδίκτυο ενώ παράλληλα ακούει την αγαπημένη του μουσική.
2. **Ease of Notification** – Για κάθε sms, email, ακόμα και τα τελευταία άρθρα από τον RSS Reader, θα υπάρχει πάντα μια ειδοποίηση στην αρχική οθόνη του τηλεφώνου, εκτός φυσικά από την ενδεικτική λυχνία LED που αναβοσβήνει, με αποτέλεσμα να μην χάνει ο χρήστης καμία κλήση, sms, email κτλ.
3. **Easy access to thousands of applications via the Android Market** – Υπάρχουν χιλιάδες εφαρμογές και παιχνίδια που ο χρήστης μπορεί να κατεβάσει οποιαδήποτε στιγμή θελήσει, δωρεάν ή και επι πληρωμή.
4. **Phone options are diverse** – Το Android είναι διαθέσιμο σε κινητά από διάφορους κατασκευαστές όπως Sony Ericsson, Motorola, Samsung, HTC κ.α. και έτσι εξυπηρετείται το στυλ και οι διαφορετικές ανάγκες του κάθε χρήστη.
5. **Google Maniac** – Στο Android έχουν ενσωματωθεί αρκετές από τις υπηρεσίες της Google, όπως το Gmail, ώστε και οι πιο πιστοί χρήστες να μπορούν να λαμβάνουν το email τους ακόμα και από το κινητό τους τηλέφωνο.
6. **Price** – Σαν πλατφόρμα είναι η πιο γρήγορα αναπτυσσόμενη ενώ στην αγορά υπάρχουν πολλές διαθέσιμες συσκευές σε όλες τις κατηγορίες τιμών.
7. **Open source** – Λογισμικό ανοιχτού κώδικα στο οποίο μπορεί κάποιος απλός χρήστης να επέμβει στον κώδικα για να επιτελέσει διάφορες αλλαγές σε λειτουργίες ώστε να ανταποκρίνεται πλήρως στις ανάγκες του.

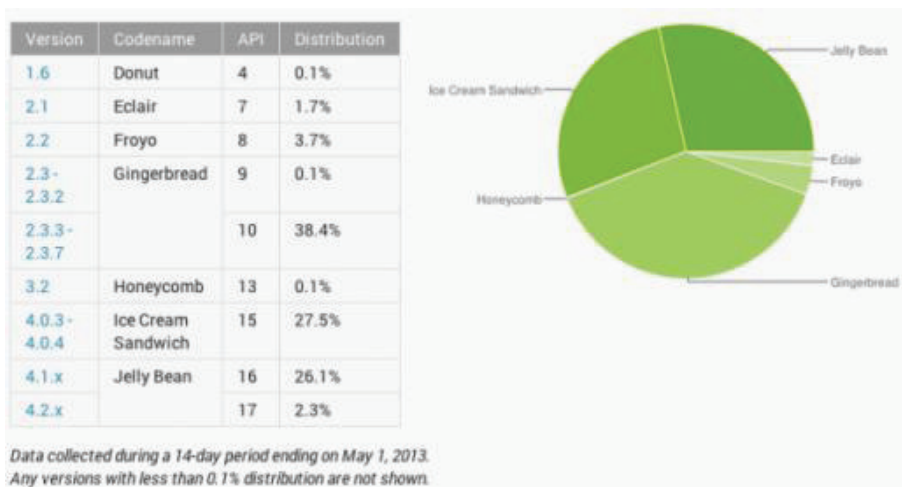
Μειονεκτήματα

1. **Advertising** – Η εγκατάσταση εφαρμογών στο λειτουργικό Android μπορεί να επιτευχθεί πολύ εύκολα και γρήγορα, όμως θα υπάρχουν πάντα διαφημίσεις ή στο επάνω ή στο κάτω μέρος της οθόνης.
2. **Battery** – Τα smartphones με λειτουργικό Android παρουσιάζουν μια ακόμα αδυναμία στην διάρκεια της μπαταρίας τους.

• Στατιστικά χρήσης του Android

Τα στατιστικά κατανομής εκδόσεων Android του μήνα Μαρτίου έγιναν διαθέσιμα από την Google και δείχνουν κάτι πολύ θετικό: Σημαντική **μείωση του ποσοστού του Android 2.3 Gingerbead** κατά 4,4%, με παράλληλη **αύξηση του Jelly Bean κατά 8,8%**. Το Gingerbead καταλαμβάνει πλέον το 39,8% των συσκευών με πρόσβαση στο Google Play, ενώ το Jelly Bean (4.1/4.2) είναι πλέον στο 25%. Η πτωτική τάση του Android 2.3 είχε αρχίσει από τους προηγούμενους μήνες, αλλά για πρώτη φορά είναι τόσο μεγάλη, κάτι το οποίο είναι θετικό. Επίσης το Android 1,6 "χάνεται" καθότι το ποσοστό του από 0,2% έπεσε στο 0,1%, όπως και το 2.1 Eclair έπεσε από 1,9% σε 1,7%. Ναι λοιπόν **οι "παλιές" εκδόσεις του Android σιγά σιγά εξαφανίζονται**. Συνεπώς **το 94,3% των συσκευών τρέχει Android 2.3** και άνω, και στην ουσία ένας developer θα πρέπει να ασχοληθεί με αυτό το μερίδιο των συσκευών.

Αξίζει επίσης να αναφερθεί ότι σύμφωνα με τις δηλώσεις του επικεφαλής της Google, Eric Schmidt τον περασμένο Σεπτέμβριο περίπου 1.3 εκατομμύρια Android συσκευές ενεργοποιούνται κάθε μέρα ενώ δεν έχασε την ευκαιρία να επανέλθει στο θέμα στο συνέδριο All Things Dive Into Mobile κάνοντας λόγο ότι ο αριθμός αυτός έχει αγγίξει το **1.5 εκατομμύριο την ημέρα**. Προσέθεσε μάλιστα ότι βρισκόμαστε πολύ κοντά στη στιγμή που **θα δούμε, περίπου 1 δισεκατομμύριο ενεργοποιηθείς συσκευών Android σε παγκόσμιο επίπεδο** σημαντικά υψηλότερο από τα 750 εκατομμύρια συσκευές που έχουν ενεργοποιηθεί μέχρι σήμερα όπως έχει δηλώσει ο πρόεδρος της εταιρείας Larry Page.

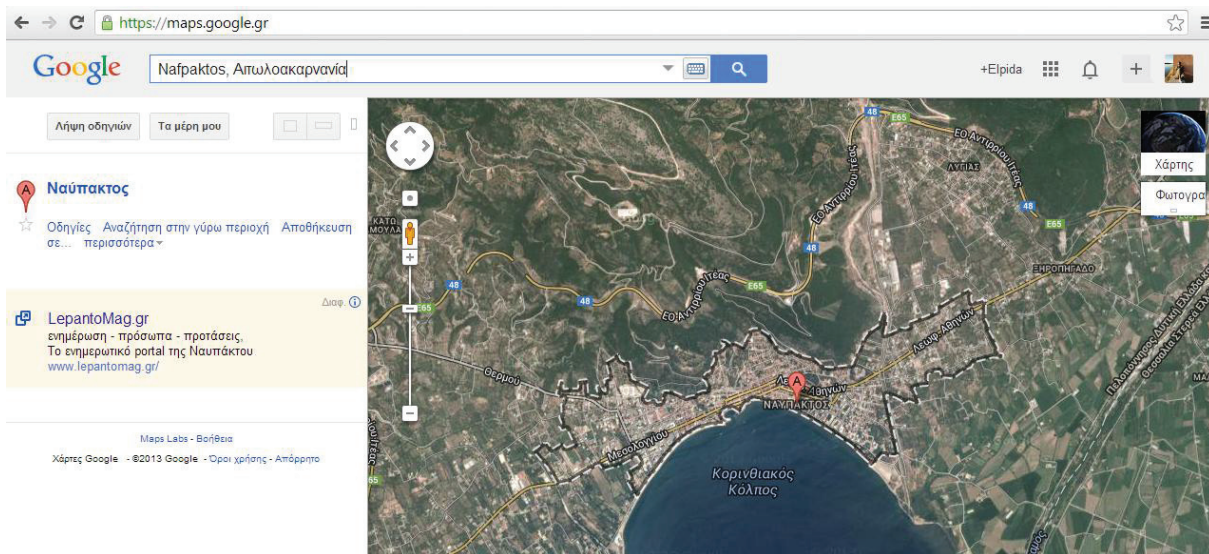


Εικόνα 2.15: Στατιστικά χρήσης συσκευών Android

• Χάρτες Google – Google Maps

Οι χάρτες Google, είναι μια διαδικτυακή εφαρμογή υπηρεσιών χαρτογράφησης. Η τεχνολογία αυτή παρέχεται δωρεάν για προσωπική χρήση από την Google. Ανάμεσα στις υπηρεσίες που προσφέρει, είναι η ιστοσελίδα των χαρτών Google, η Google Transit και χάρτες ενσωματωμένους σε ιστοσελίδες τρίτων μέσω των Google Maps API. Παρέχει οδικούς χάρτες, εφαρμογή δρομολόγησης για ταξίδι με τα πόδια, το αυτοκίνητο ή μέσα μαζικής μεταφοράς και εντοπισμό αστικών επιχειρήσεων για πολλές χώρες σε όλο τον κόσμο. Η τεχνολογία των χαρτών Google χρησιμοποιεί μια στενή παραλλαγή της προβολής

Mercator (ένας κυλινδρικής προβολής χάρτης του κόσμου), με αποτέλεσμα να μην μπορεί να δείξει περιοχές γύρω από τους πόλους. Ένα σχετικό προϊόν είναι το Google Earth, ένα αυτόνομο πρόγραμμα το οποίο προσφέρει περισσότερα χαρακτηριστικά ως προς την παγκόσμια προβολή, συμπεριλαμβανομένων των πολικών περιοχών.



Εικόνα 2.16: Χάρτες Google

Η τεχνολογία αυτή χρησιμοποιεί εκτενώς την γλώσσα Javascript. Όταν ο χρήστης σύρει τον χάρτη τότε φορτώνονται από τον εξυπηρετητή τα τετράγωνα του πλέγματος, τα οποία ακολούθως εισάγονται στην σελίδα. Όταν ο χρήστης ψάχνει για μια επιχείρηση τα αποτελέσματα φορτώνονται στο παρασκήνιο για εισαγωγή στο πλαίσιο και τον χάρτη, με αποτέλεσμα να αποφεύγεται η επαναφόρτωση της σελίδας. Οι τοποθεσίες καθορίζονται δυναμικά, με την τοποθέτηση μιας κουκκίδας (pin) πάνω από τις εικόνες του χάρτη. Επιπρόσθετα, η σελίδα χρησιμοποιεί JSON για μεταφορά δεδομένων, αντί XML, για σκοπούς απόδοσης. Το γεγονός ότι οι χάρτες Google είναι τεχνολογία γραμμένη σχεδόν εξολοκλήρου σε Javascript και XML, επιτρέπει σε τελικούς χρήστες να παράγουν client-side scripts και server-side hooks, τα οποία επιτρέπουν με την σειρά τους σε ένα χρήστη ή μια ιστοσελίδα να εισάγουν επεκταμένες ή τροποποιημένες λειτουργίες στην διεπαφή των χαρτών Google.

Τον Ιούνιο του 2005, η Google διέθεσε το Google Maps API, με σκοπό να επιτρέψει στους προγραμματιστές να ενσωματώσουν τους Χάρτες Google στις ιστοσελίδες τους. Το API αυτό είναι δωρεάν για εμπορική χρήση, με την προϋπόθεση ότι η ιστοσελίδα για την οποία θα χρησιμοποιηθεί, δεν χρεώνει τους χρήστες της για πρόσβαση σε αυτήν. Ιστοσελίδες που χρεώνουν και θέλουν να χρησιμοποιήσουν το Google Maps API, μπορούν να αγοράσουν το Google Maps Premier. Με την χρήση της διεπαφής, μπορεί κάποιος να ενσωματώσει την ιστοσελίδα Χαρτών Google σε εξωτερική ιστοσελίδα με την δυνατότητα τοποθέτησης δικών του δεδομένων πάνω από τους χάρτες. Αν και αρχικά το Google Maps API ήταν ένα API για Javascript, μετέπειτα επεκτάθηκε ώστε να συμπεριλάβει ένα API για Adobe Flash και μια υπηρεσία για ανάκτηση στατικών εικόνων χαρτών. Επίσης συμπεριέλαβε δικτυακές υπηρεσίες για την εκτέλεση γεωκωδικοποιήσεων και δημιουργία κατευθύνσεων οδήγησης.

- Χάρτες Google για κινητά

Οι Χάρτες Google για κινητά είναι μια εφαρμογή σε Java, η οποία εκδόθηκε από την Google το 2006, με σκοπό να τρέχει σε οποιοδήποτε βασισμένο σε Java κινητό ή φορητή συσκευή. Πολλά από τα χαρακτηριστικά των Χαρτών Google υπάρχουν και σε αυτή την εφαρμογή. Στις 28 Νοεμβρίου του 2008 κυκλοφόρησε και η δεύτερη έκδοση της εφαρμογής αυτής, οι Χάρτες Google για Κινητά 2.0. Σε αυτήν την καινούργια έκδοση, η Google υλοποίησε μια υπηρεσία παρόμοια με το παγκόσμιο σύστημα εντοπισμού θέσης (GPS-like), η οποία όμως δεν χρειάζοταν δέκτη GPS. Η λειτουργία 'η τοποθεσία μου' (my location), λειτουργεί με χρήση της τοποθεσίας GPS του κινητού, αν αυτή είναι διαθέσιμη. Αυτό γίνεται με τον εξής τρόπο:

- Το λογισμικό ψάχνει για το πιο κοντινό ασύρματο δίκτυο ή κύτταρο/κυψέλη. Ακολούθως ψάχνει για την τοποθεσία του κυττάρου, χρησιμοποιώντας μια βάση δεδομένων με γνωστά ασύρματα δίκτυα και κύτταρα. Ο προσδιορισμός θέσης του κυττάρου γίνεται με μια τριγωνική μέθοδο όπου μελετούνται διαφορετικές εντάσεις του σήματος από διαφορετικούς πομπούς που βρίσκονται στους Σταθμούς Κυττάρων. Ακολούθως χρησιμοποιώντας την τοποθεσία η οποία ανακτάται από μια διαδικτυακή βάση δεδομένων, διορθώνεται 'η τοποθεσία μου', καθορίζοντας την τρέχουσα θέση του χρήστη. Η μέθοδος εντοπισμού ασύρματου δικτύου, πραγματοποιείται βρίσκοντας τις πιο κοντινές δυναμικές ζώνες WiFi (WiFi hotspots). Ακολούθως η τοποθεσία αυτών των ζωνών που ανακτάται από αντίστοιχη βάση δεδομένων χρησιμοποιείται για την εξακρίβωση της θέσης του χρήστη. Η εφαρμογή κάνει χρήση υπηρεσιών βασισμένες στο GPS, υπηρεσιών βασισμένες στο WiFi και στο WLAN, και υπηρεσιών βασισμένων στους πομπούς των κυττάρων. Στον χάρτη φαίνονται οι δρόμοι και η εκτιμώμενη θέση του κινητού τηλεφώνου, η οποία δεν είναι απόλυτη καθώς περιβάλλεται από ένα κύκλο που απεικονίζει το εύρος χώρου στον οποίο μπορεί να βρίσκεται το κινητό. Το εκτιμώμενο εύρος χώρου, υπολογίζεται με βάση την ισχύ του σήματος του τηλεφώνου που υποδεικνύει πόσα κοντά αυτό είναι στον Σταθμό του κυττάρου, άρα και στον πομπό. Η υπηρεσία αυτή είναι διαθέσιμη σε πολλές πλατφόρμες όπως για παράδειγμα τις ακόλουθες:

1. Android (η σχέση του Android με την υπηρεσία αυτή μελετάται στην συνέχεια αναλυτικότερα).
2. iOS (iPhone/iPod Touch/iPad)
3. Windows Mobile
4. Nokia/Symbian (σε συγκεκριμένα μόνο μοντέλα)
5. Symbian OS (UIQ v3)
6. BlackBerry

- 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 υπάρχων 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>). 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.

ΚΕΦΑΛΑΙΟ 3

Βασικά εργαλεία ανάπτυξης εφαρμογών Android

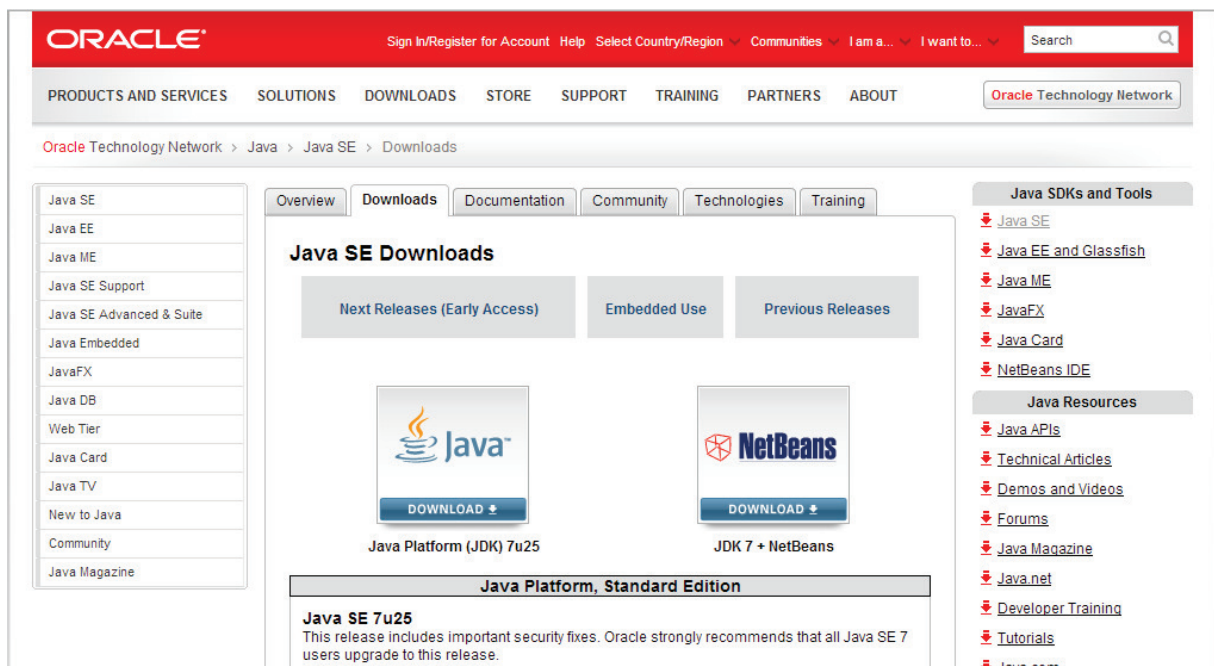
- Ανάπτυξη εφαρμογής Android

Σε αυτό το σημείο, θα παρουσιάσουμε πως αναπτύσσετε μια εφαρμογή Android. Σε πρώτη φάση θα πρέπει να δημιουργήσουμε το κατάλληλο περιβάλλον για ανάπτυξη εφαρμογών Android. Τα τρία βασικά εργαλεία αυτού του περιβάλλοντος είναι η Java, το Eclipse και το Android. Τα παραπάνω εργαλεία πρέπει να εγκατασταθούν την σειρά που προαναφέρθηκαν γιατί το κάθε ένα από αυτά έχει σαν προϋπόθεση την ύπαρξη του προηγούμενου στο σύστημα. Παρακάτω θα δούμε αναλυτικά κάθε ένα από αυτά τα εργαλεία.

- Εγκατάσταση Java

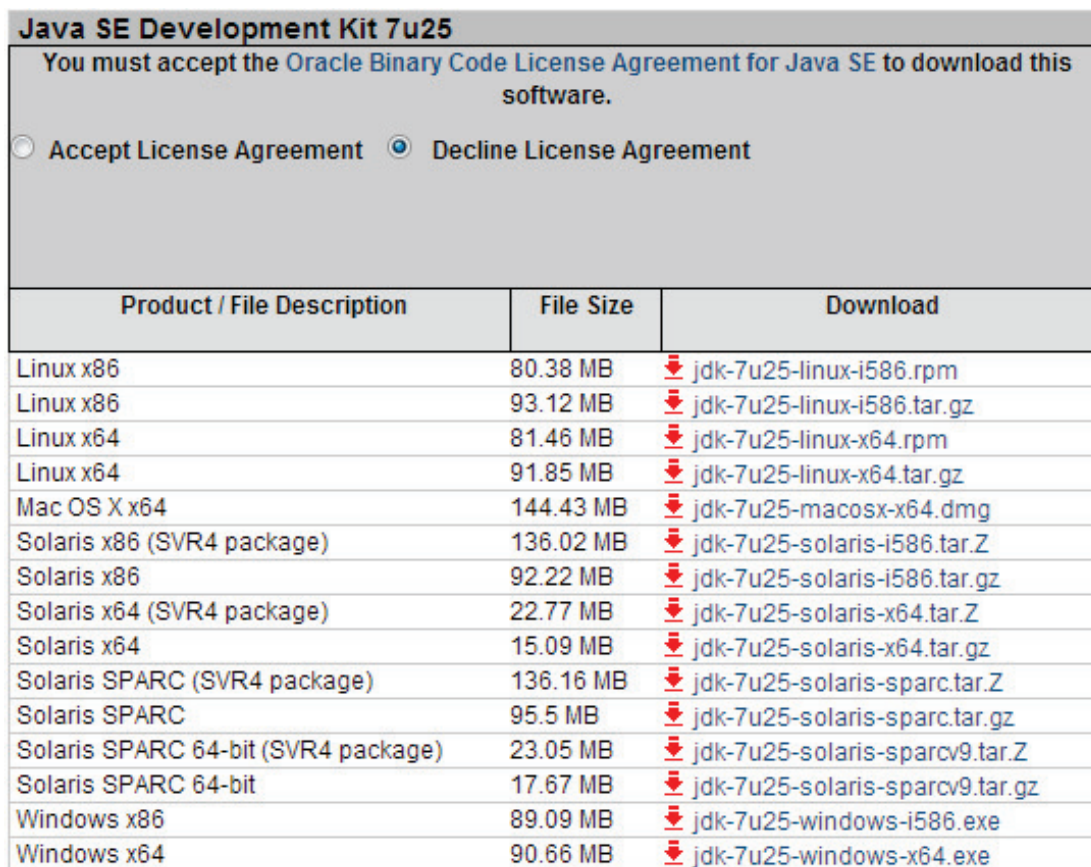
Όλη η ανάπτυξη εφαρμογών Android στηρίζεται στην στάνταρ έκδοση γλώσσας προγραμματισμού Java (Java Platform Standard Edition – Java SE) και για αυτό θα πρέπει να κατεβάσουμε την τελευταία έκδοση του JDK (Java Development Kit) από την σελίδα της Oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.



Εικόνα 3.1: Κατέβασμα τελευταίας έκδοσης JDK

Στην συνέχεια επιλέγουμε την κατάλληλη πλατφόρμα που αντιστοιχεί στο λειτουργικό μας σύστημα και κατεβάζουμε το αρχείο. Αφού αποθηκεύσουμε το αρχείο, το εκτελούμε και το εγκαθιστούμε δεχόμενοι τις προκαθορισμένες ρυθμίσεις σε κάθε παράθυρο.

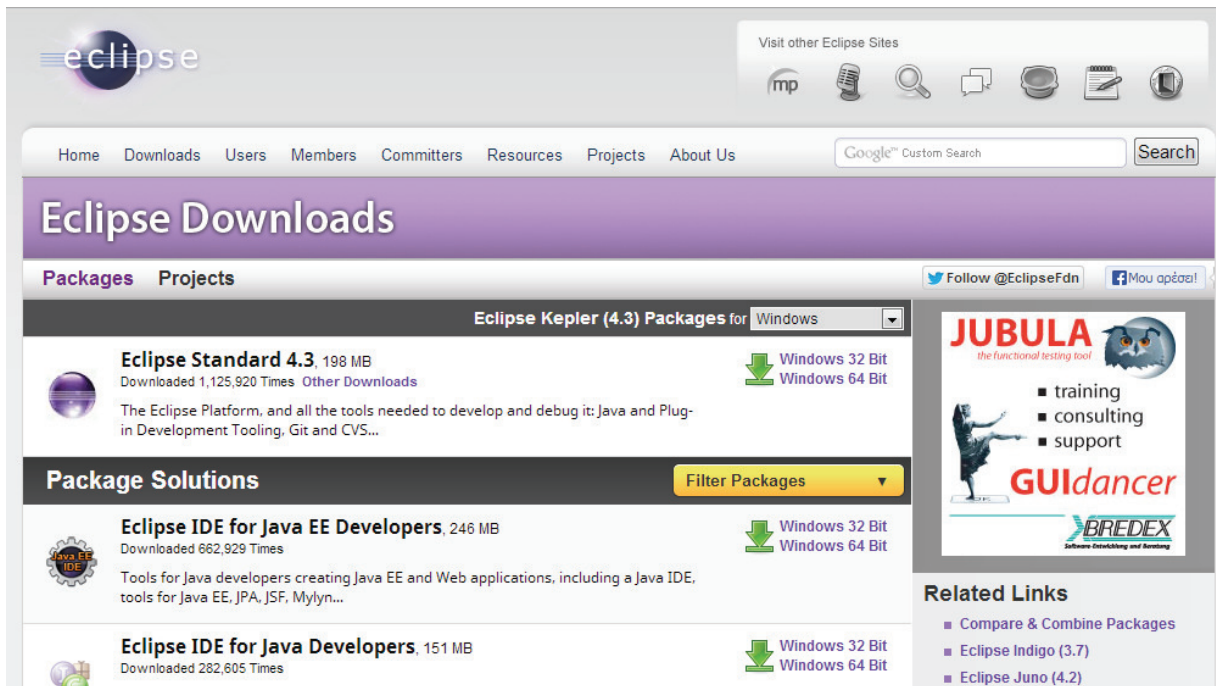


Product / File Description	File Size	Download
Linux x86	80.38 MB	jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	jdk-7u25-linux-x64.rpm
Linux x64	91.85 MB	jdk-7u25-linux-x64.tar.gz
Mac OS X x64	144.43 MB	jdk-7u25-macosx-x64.dmg
Solaris x86 (SVR4 package)	136.02 MB	jdk-7u25-solaris-i586.tar.Z
Solaris x86	92.22 MB	jdk-7u25-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.77 MB	jdk-7u25-solaris-x64.tar.Z
Solaris x64	15.09 MB	jdk-7u25-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.16 MB	jdk-7u25-solaris-sparc.tar.Z
Solaris SPARC	95.5 MB	jdk-7u25-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	jdk-7u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.67 MB	jdk-7u25-solaris-sparcv9.tar.gz
Windows x86	89.09 MB	jdk-7u25-windows-i586.exe
Windows x64	90.66 MB	jdk-7u25-windows-x64.exe

Εικόνα 3.2: Επιλογή κατάλληλης πλατφόρμας

- Εγκατάσταση Eclipse IDE

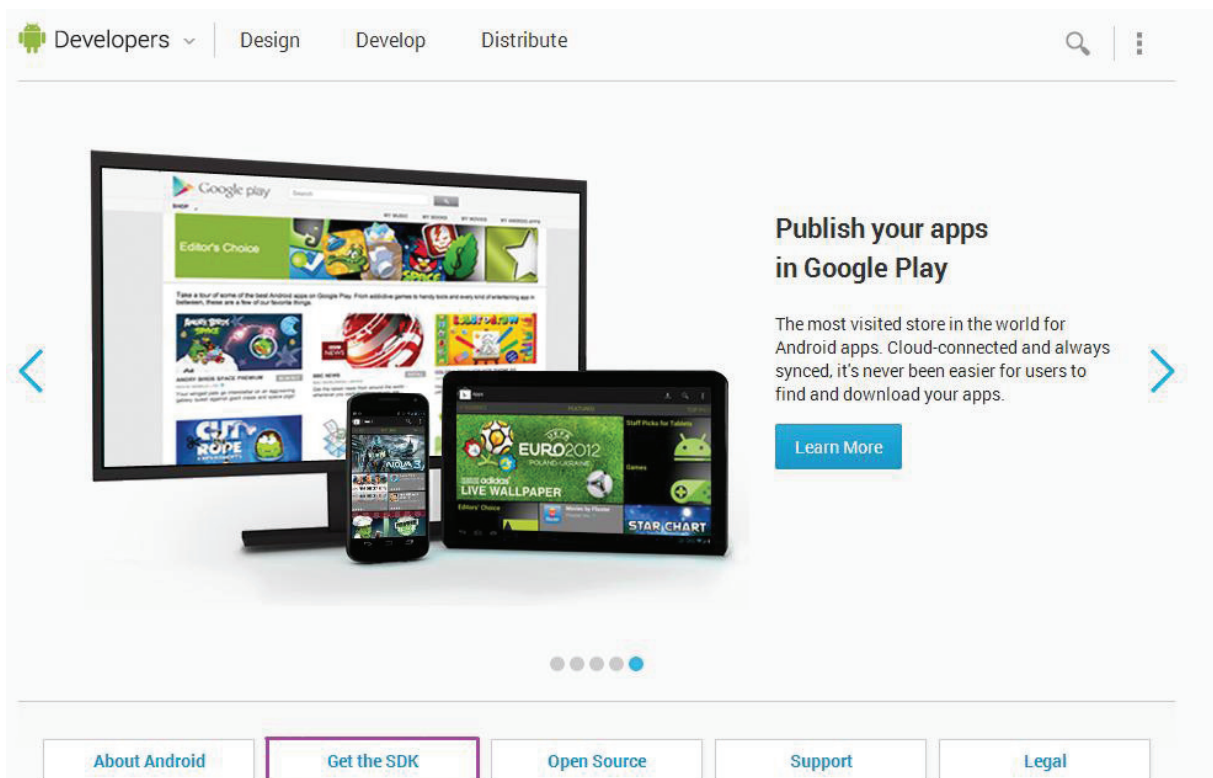
Το Eclipse αποτελεί το περιβάλλον ανάπτυξης εφαρμογών Java και Android. Από την επίσημη σελίδα: <http://www.eclipse.org/downloads/> κατεβάζουμε την έκδοση Eclipse IDE for Java Developers για το λειτουργικό σύστημα το οποίο χρησιμοποιούμε. Το πρόγραμμα Eclipse δεν χρειάζεται εγκατάσταση απλά μόλις το κατέβασμα τελειώσει, αποσυμπιέζουμε το αρχείο και μεταφέρουμε τον φάκελο eclipse μέσα στο Program Files. Δημιουργούμε μια συντόμευση στην επιφάνεια εργασίας μας και τρέχουμε το αρχείο εφαρμογής.



Εικόνα 3.3: Κατέβαμα Eclipse IDE for Java EE Developers

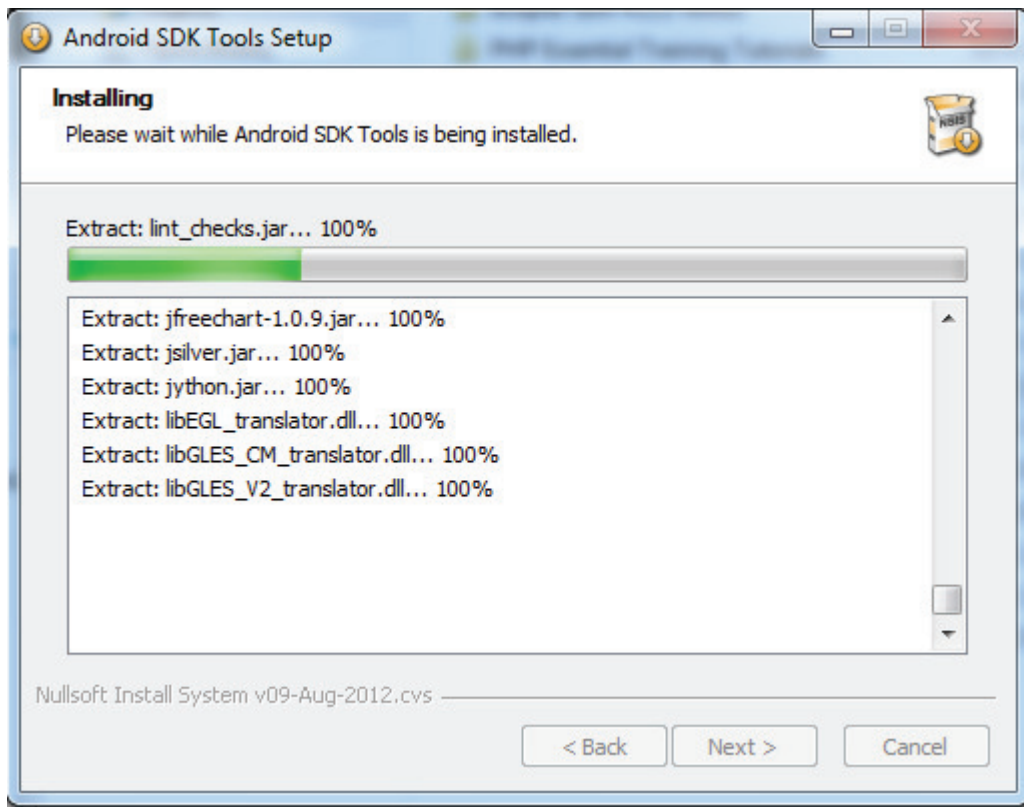
- Εγκατάσταση Android SDK

Το τελευταίο εργαλείο που χρειαζόμαστε είναι το Android SDK, το οποίο μπορούμε να το κατεβάσουμε από την επίσημη σελίδα Android: <http://developer.android.com/index.html>. επιλέγουμε από την αρχική σελίδα το <Get the SDK> και στην συνέχεια επιλέγουμε το αντίστοιχο SDK για το λειτουργικό μας σύστημα.



Εικόνα 3.4: Κατέβαμα Android SDK

Αφού κατέβει το αρχείο το εκτελούμε και κάνουμε εγκατάσταση.

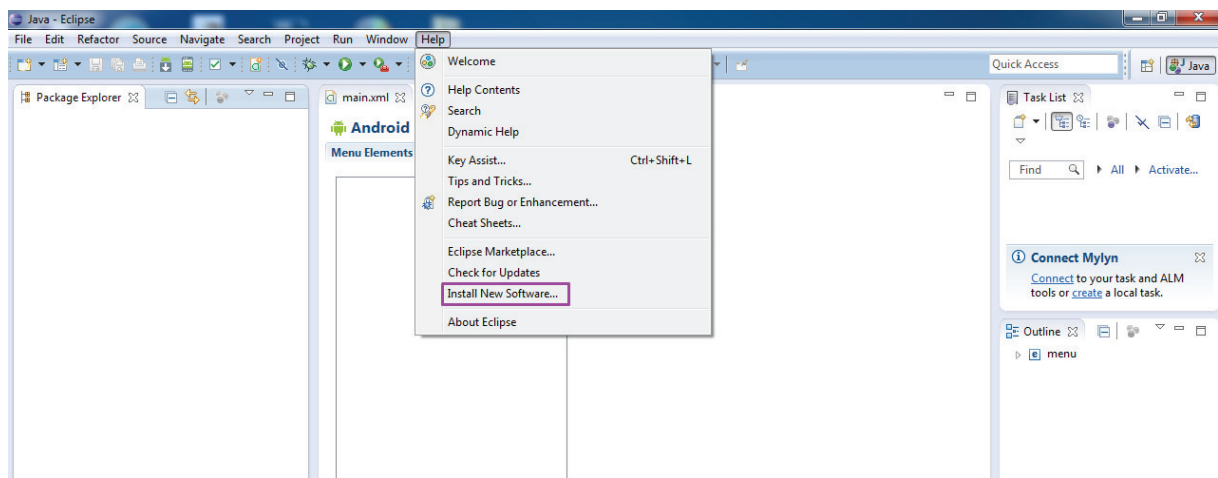


Εικόνα 3.5: Εγκατάσταση Android SDK

Για να ενσωματωθεί η λειτουργία του SDK μέσα στο Eclipse θα πρέπει να δηλώσουμε σε ποια τοποθεσία βρίσκεται και αυτό θα το πετύχουμε με την εγκατάσταση του Android Development

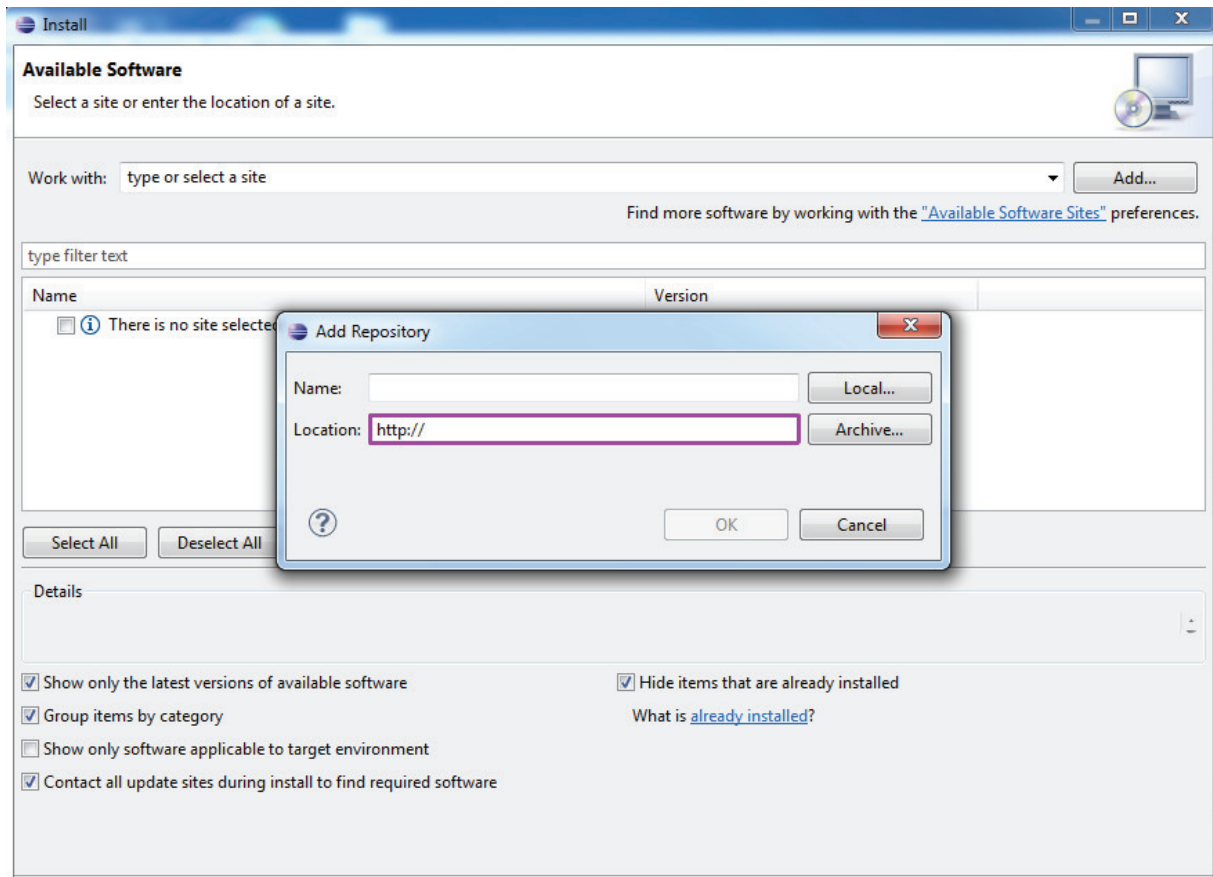
Tool plug-in που υπάρχει για το Eclipse. Αυτό θα το πετύχουμε με τα παρακάτω απαραίτητα βήματα:

- Πρώτο βήμα: Ανοίγουμε το Eclipse και από το κύριο μενού επιλέγουμε Help και μετά Install New Software.



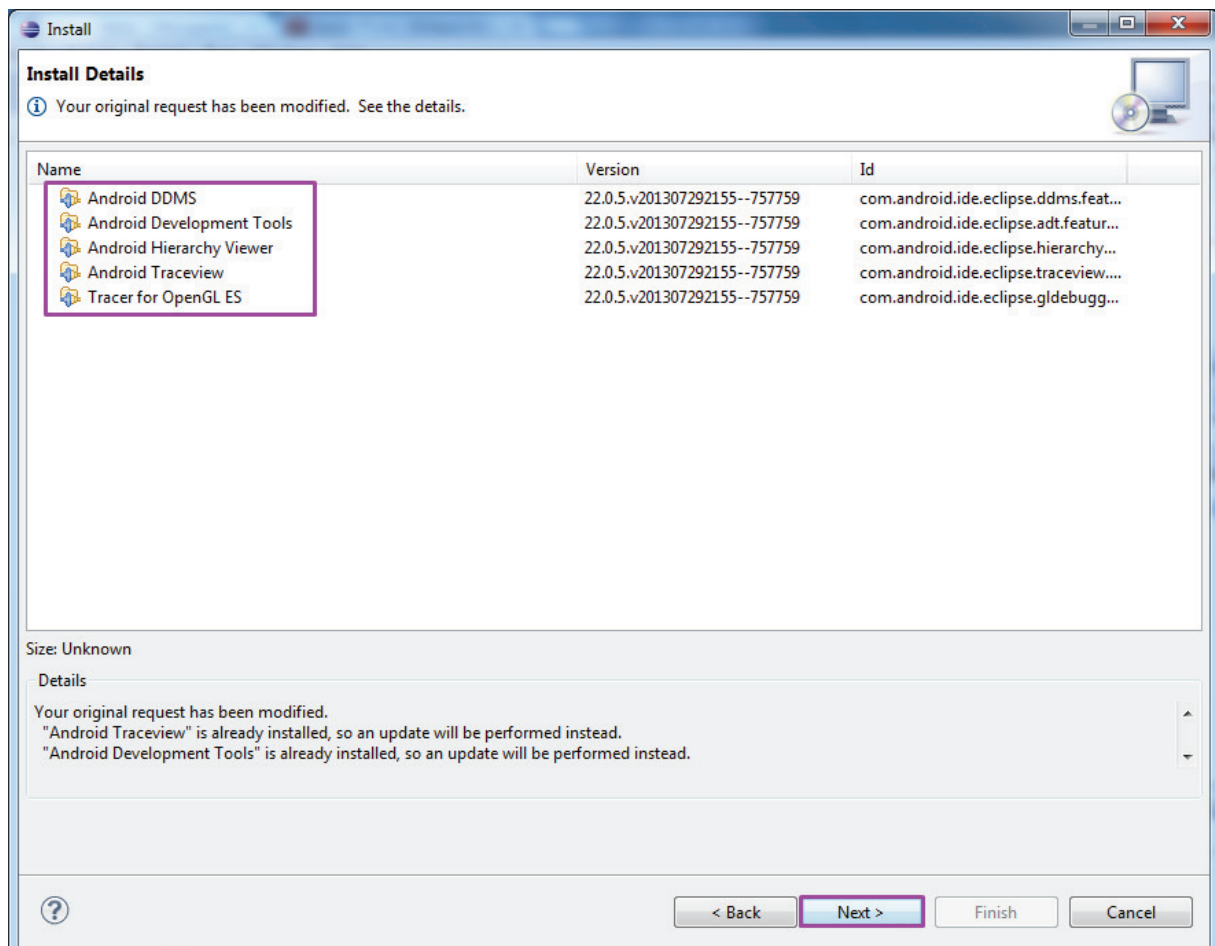
Εικόνα 3.6: Πρώτο βήμα ενσωμάτωσης εργαλείων Android στο Eclipse

- Δεύτερο βήμα: Στο παράθυρο που εμφανίζεται πατάμε το πλήκτρο Add. Στο νέο παράθυρο Add Repository στην περιοχή Location εισάγουμε τη διεύθυνση <http://dl-ssl.google.com/android/eclipse/>.



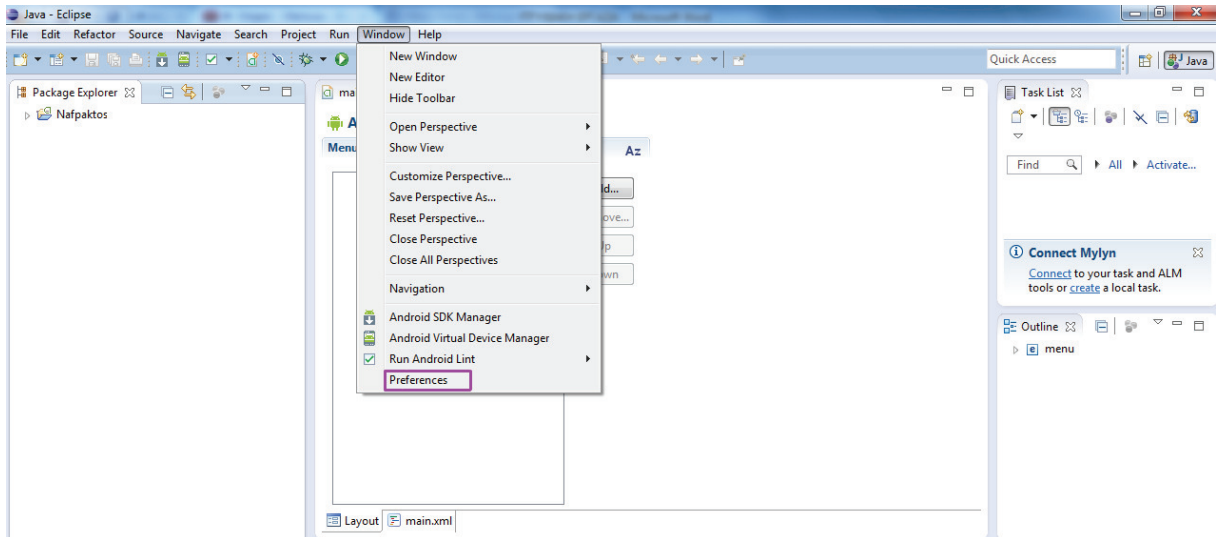
Εικόνα 3.7: Δεύτερο βήμα ενσωμάτωσης εργαλείων Android στο Eclipse

- Τρίτο βήμα: Από το παράθυρο Available Software που εμφανίζεται επιλέγουμε το Developer Tools το οποίο περιλαμβάνει τα βασικά εργαλεία ανάπτυξης για Android Εφαρμογές και πατάμε το κουμπί Next. Αν η εγκατάσταση αποτύχει, δοκιμάζουμε να εγκαταστήσουμε ένα ένα τα εργαλεία.



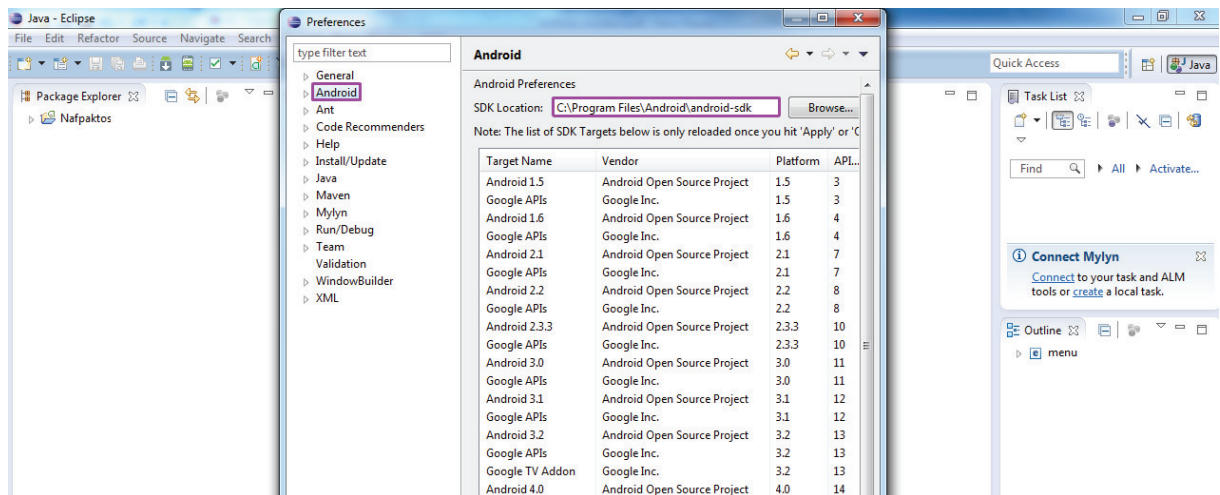
Εικόνα 3.8: Τρίτο βήμα ενσωμάτωσης εργαλείων Android στο Eclipse

- Τέταρτο βήμα: Κάνουμε επανεκκίνηση του Eclipse και αφού ξανανοίξει θα εμφανιστεί το παράθυρο Welcome to Android Development μέσα από το οποίο μπορούμε να κατεβάσουμε όλες τις εκδόσεις για τα Android APIs. Εμείς όμως θα πατήσουμε Cancel και θα ενημερώσουμε τα παρακάτω APIs μόνοι μας μέσω του Android SDK Manager.
- Πέμπτο βήμα: Παρατηρούμε ότι η ενσωμάτωση του ADT plug-in στο περιβάλλον του Eclipse έχει ήδη πραγματοποιηθεί. Αυτό που μένει να κάνουμε εμείς είναι να δηλώσουμε το ADT plug-in να δείχνει στην Android SDK εγκατάσταση. Για να το κάνουμε αυτό θα πρέπει από το κύριο μενού του Eclipse να επιλέξουμε Window και μετά Preferences.



Εικόνα 3.9: Πέμπτο βήμα ενσωμάτωσης εργαλείων Android στο Eclipse

- Έκτο βήμα: Από το παράθυρο Preferences που θα εμφανιστεί επιλέγουμε την κατηγορία Android από την αριστερή λίστα και στην περιοχή του SDK Location αναζητάμε την τοποθεσία όπου κάναμε εγκατάσταση το Android SDK.



Εικόνα 3.10: Έκτο βήμα ενσωμάτωσης εργαλείων Android στο Eclipse

- Έβδομο βήμα: Και τελευταίο είναι η εγκατάσταση βιβλιοθηκών Android μέσω του Android SDK Manager, για το οποίο θα μιλήσουμε αναλυτικότερα παρακάτω.

• Android SDK Manager και AVD Manager

Έπειτα από την ενσωμάτωση και του ADT plug-in στο περιβάλλον του Eclipse, παρατηρούμε ότι εμφανίστηκαν 2 εικονίδια που παραπέμπουν στο Android SDK Manager και στο AVD Manager.



Εικόνα 3.11: Android Development Tools – ADT plug-in

• Android SDK Manager

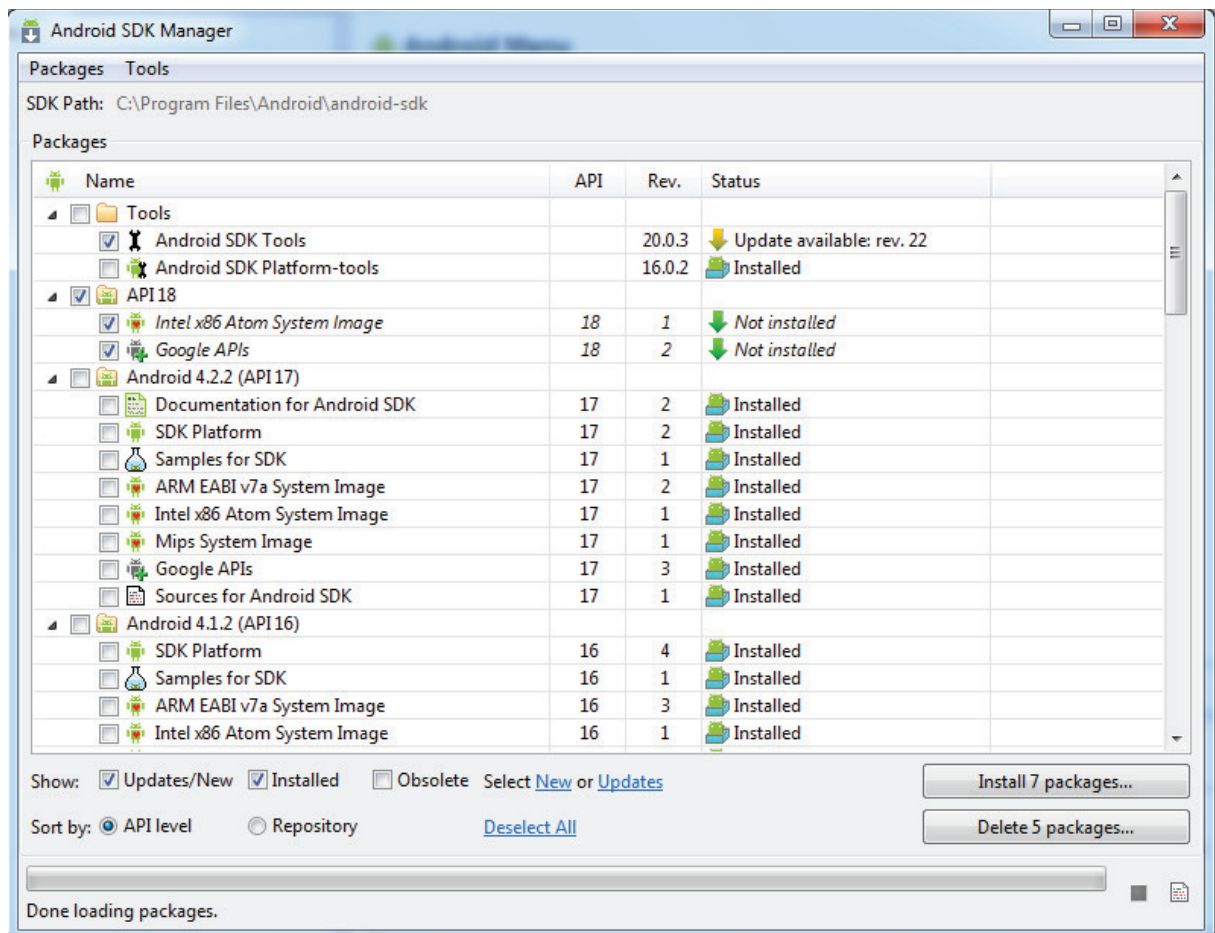
Πατώντας το αριστερό εικονίδιο από το ADT plug-in θα εμφανιστεί το παράθυρο του Android SDK Manager το οποίο περιλαμβάνει μια λίστα η οποία αποτελείται από διαθέσιμες προγραμματιστικές βιβλιοθήκες. Πιο συγκεκριμένα αποτελείται από εργαλεία, εκδόσεις του Android και κάποια επιπλέον στοιχεία. Στα εργαλεία (Tools) περιέχονται δύο πακέτα, το πρώτο είναι το Android SDK Tools και το δεύτερο το Android SDK Platform tools. Το Android SDK Tools έχει να κάνει με εργαλεία εντοπισμού και ελέγχου, καθώς και άλλα βοηθητικά προγράμματα που έχουν να κάνουν με την ανάπτυξη μιας εφαρμογής, ενώ το Android SDK Platform tools περιέχει μια πλατφόρμα που εξαρτιέται από εργαλεία που έχουν να κάνουν με την ανάπτυξη και τον εντοπισμό της εφαρμογής μας. Τα εργαλεία αυτά υποστηρίζουν τις πιο πρόσφατες δυνατότητες της πλατφόρμας Android και συνήθως ενημερώνεται μόνο όταν μια νέα πλατφόρμα είναι διαθέσιμη.

Στην λίστα ακολουθούν οι εκδόσεις Android όπου δίπλα στο καθένα βρίσκεται και το αντίστοιχο API (Application Programming Interface). Τα APIs περιέχουν δομές δεδομένων, μεταβλητές και κλάσεις αντικειμένων που μπορούν να χρησιμοποιηθούν. Μια εφαρμογή που φτιάχτηκε για να τρέξει πάνω σε έκδοση Android με μικρότερο API (π.χ. Gingerbread με API 10), αυτή θα μπορεί να τρέξει σε μια πιο πρόσφατη έκδοση με μεγαλύτερο API (π.χ. Jelly Bean με API 16) το αντίθετο όμως δεν μπορεί να συμβεί. Τα σημαντικότερα πακέτα που περιέχει η κάθε έκδοση και πρέπει να εγκαταστήσουμε είναι:

- Το SDK Platform το οποίο πρέπει οπωσδήποτε να εγκατασταθεί για να μπορούμε να χρησιμοποιήσουμε το συγκεκριμένο API.
- Το Samples for SDK που περιέχει έτοιμες εφαρμογές για να δούμε τον κώδικα τους, πώς λειτουργούν κ.τ.λ.
- Το Google APIs που περιέχει εφαρμογές, βιβλιοθήκες και υπηρεσίες της Google που μπορούμε να χρησιμοποιήσουμε στην εφαρμογή μας.

Τελευταίο στην λίστα είναι τα επιπλέον στοιχεία (Extras), τα οποία είναι επιπλέον πακέτα που υλοποιούν διάφορες υπηρεσίες και είναι προαιρετικά. Για παράδειγμα: το Google USB Driver που μας επιτρέπει να μεταφέρουμε τις εφαρμογές μας στο κινητό μας τηλέφωνο με την χρήση της USB θύρας του υπολογιστή, το Android Support Library που περιέχει στατικές βιβλιοθήκες υποστήριξης που μπορούμε να προσθέσουμε στην εφαρμογή μας για να μπορέσουμε να χρησιμοποιήσουμε APIs που δεν είναι διαθέσιμα για παλαιότερες εκδόσεις πλατφόρμας και πολλά άλλα.

Για την σωστή ανάπτυξη εφαρμογών καλό θα ήταν να επιλέξουμε την έκδοση Android 2.3.3 ή αλλιώς Gingerbread με API 10 για να είμαστε σίγουρη πως θα είναι συμβατή με τις περισσότερες συσκευές. Σε περίπτωση που η μαζική εγκατάσταση των παραπάνω πακέτων είναι ανεπιτυχής προσπαθούμε να τα εγκαταστήσουμε ένα-ένα. Τέλος συνίσταται η τακτική ενημέρωση και λήψη των πρόσφατων ενημερώσεων από το Android SDK Manager.

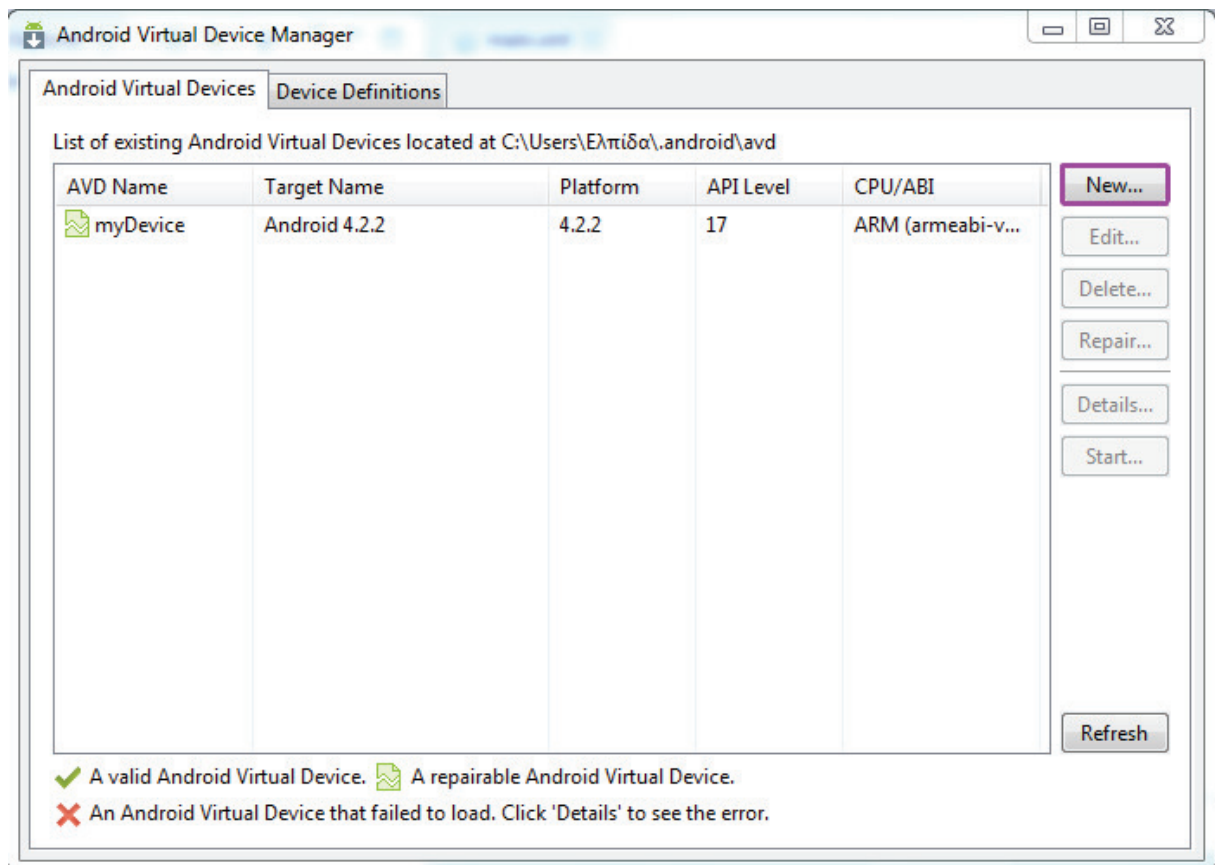


Εικόνα 3.12: Android SDK Manager

• Android AVD Manager

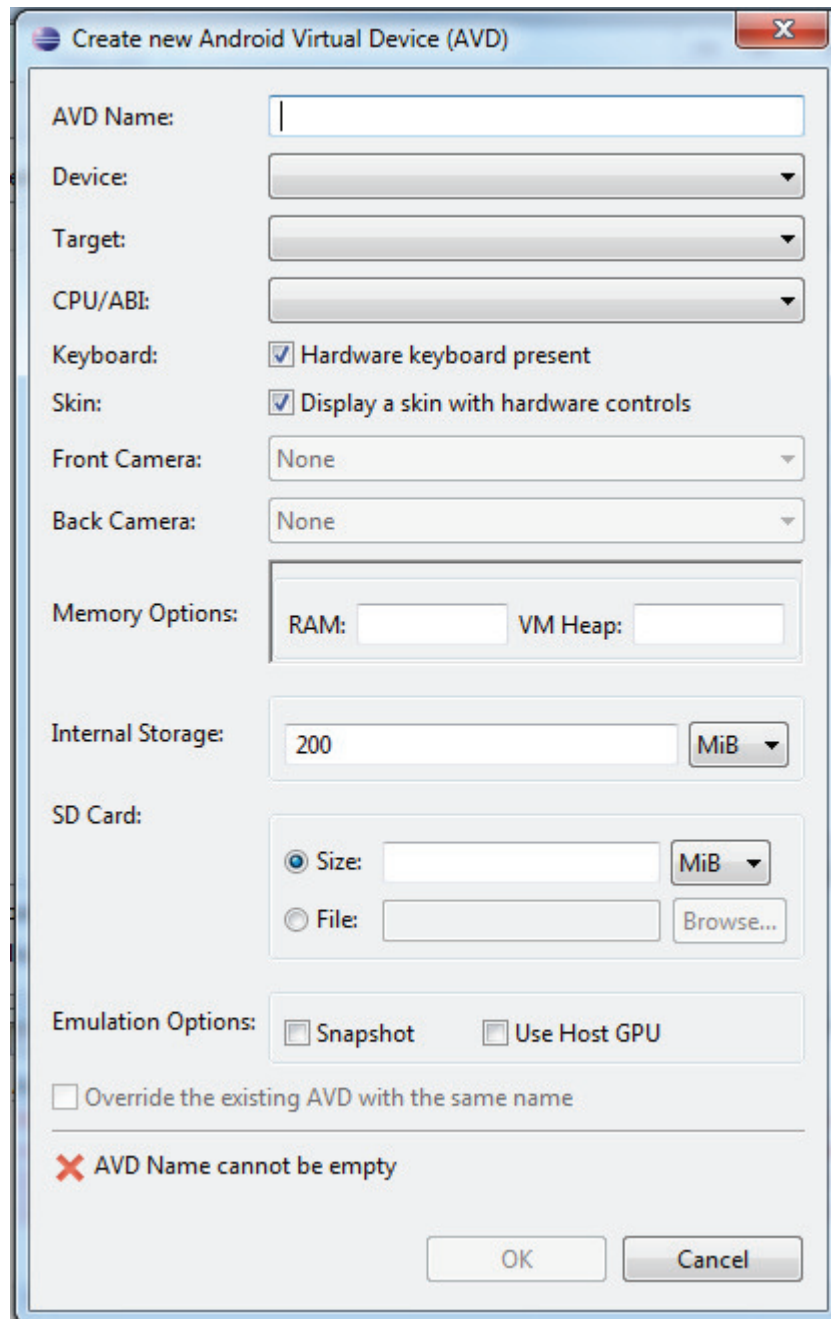
Πατώντας το δεξί εικονίδιο από το ADT plug-in θα εμφανιστεί το παράθυρο του Android Virtual Device Manager μέσα στο οποίο μπορούμε να φτιάξουμε έναν εξομοιωτή του Android και να δοκιμάζουμε τις εφαρμογές μας στον υπολογιστή. Είναι στην ουσία μια εικονική συσκευή που ανταποκρίνεται σε πραγματικά δεδομένα. Μπορούμε να φτιάξουμε περισσότερες από μια εικονικές συσκευές με τα χαρακτηριστικά που θα τους ορίσουμε εμείς. Παρακάτω θα δούμε πως μπορούμε να δημιουργήσουμε έναν τέτοιο εξομοιωτή:

- Πρώτο βήμα: Αρχικά από το παράθυρο Android Virtual Device Manager κάνουμε κλικ στο κουμπί New.



Εικόνα 3.13: Δημιουργία AVD

- Δεύτερο βήμα: Από το παράθυρο Create New Android Virtual Device, στην περιοχή Name γράφουμε το όνομα που θέλουμε να δώσουμε.



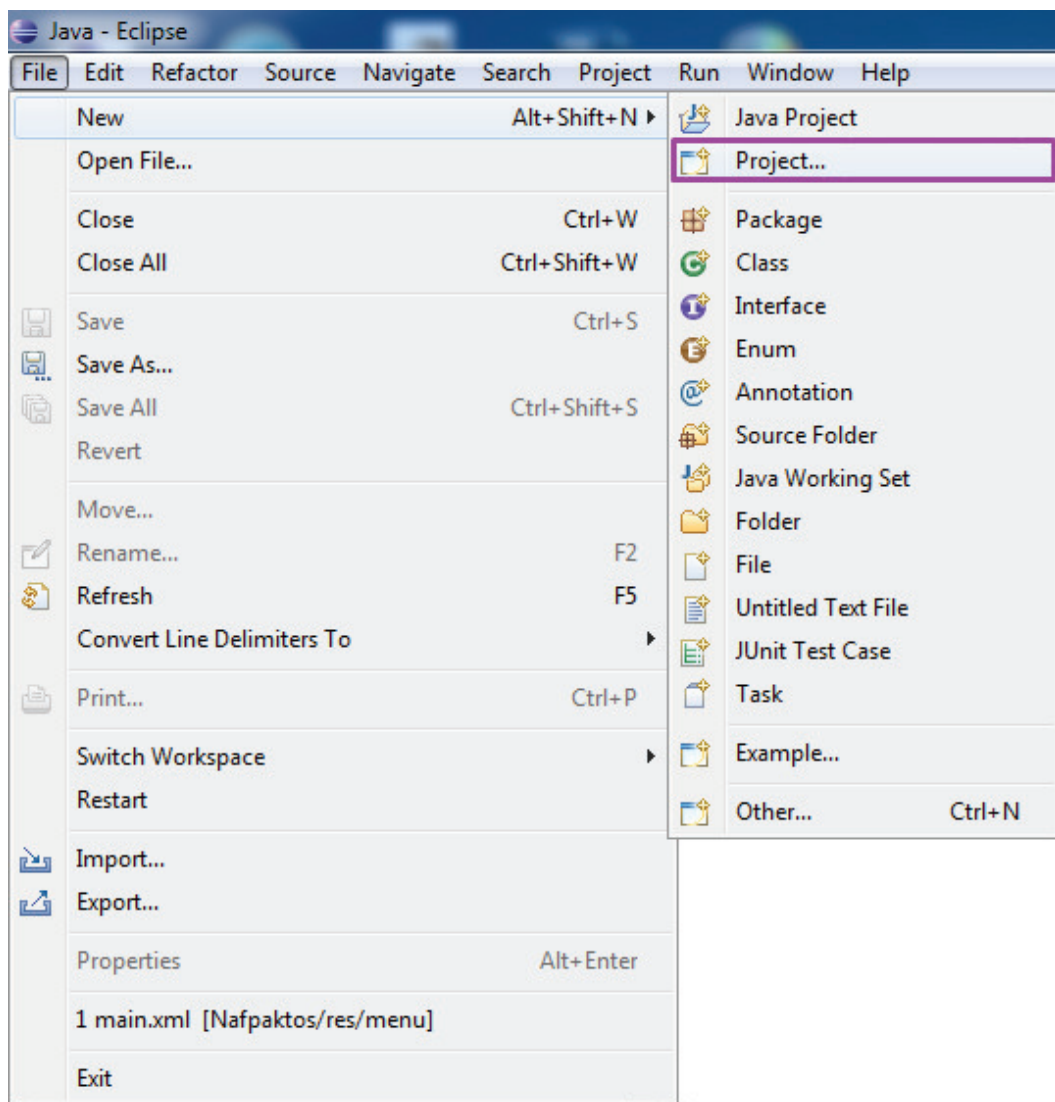
Εικόνα 3.14: Δημιουργία AVD

- Τρίτο βήμα: Στην περιοχή Target ορίζουμε την πλατφόρμα στην οποία θέλουμε να δουλέψουμε.
- Τέταρτο βήμα: Η περιοχή CPU/ABI ρυθμίζεται αυτόματα με την επιλογή Target και καθορίζει πως αναμένεται να αλληλεπιδράσει η εφαρμογή με το σύστημα χρόνου εκτέλεσης.
- Πέμπτο βήμα: Στην περιοχή SD Card ορίζουμε αν ο εξομοιωτής μας θα έχει κάρτα μνήμης και τη χωρητικότητά της.
- Έκτο βήμα: Με την επιλογή ενεργοποίησης στο Snapshot έχουμε τη δυνατότητα να αποθηκεύσουμε και να επαναφέρουμε ένα στιγμιότυπο μιας κατάστασης για κάθε ένα AVD, με σκοπό να αποφύγουμε την εκκίνηση όταν ξεκινάμε τον προσομοιωτή.

- Έβδομο βήμα: Στην περιοχή Skin μπορούμε να επιλέξουμε προκαθορισμένες διαστάσεις οθόνης ή να ορίσουμε τις δικές μας και τέλος.
- Όγδοο βήμα: Στο Hardware μπορούμε να ορίσουμε τα χαρακτηριστικά που θέλουμε να έχει η συσκευή μας, όπως π.χ. η μνήμη RAM, αν υποστηρίζει κάρτα μνήμης, αν διαθέτει κάμερα, αν μπορεί να αναπαράγει ήχους και να ηχογραφεί ήχους.

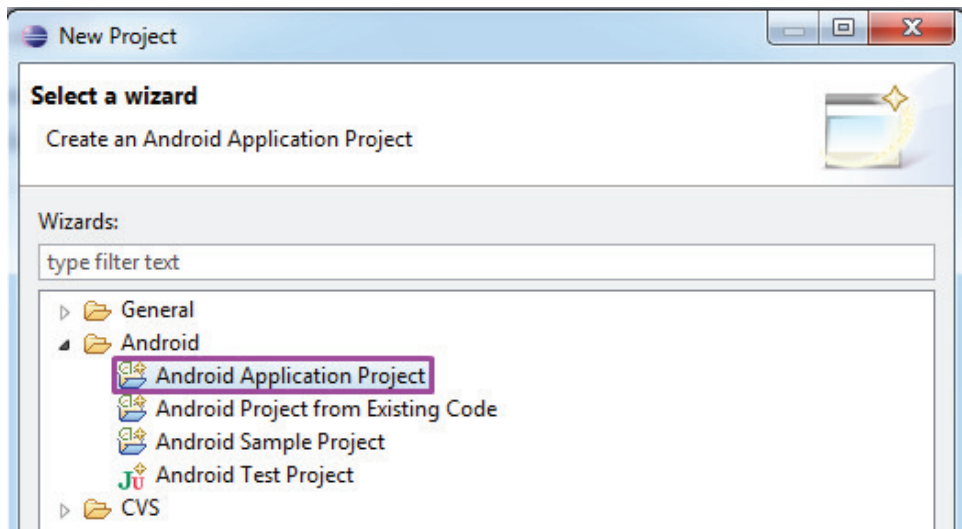
• Δημιουργία Project στο Eclipse

Αφού έχουμε ορίσει τις απαιτούμενες ανάγκες της εφαρμογής και έχουμε στο μυαλό μας τον βασικό κορμό, θα προχωρήσουμε στην υλοποίηση της. Όλα ξεκινάνε από τη δημιουργία ενός νέου Android project στο Eclipse. Το όνομα που δηλώνουμε κατά τη δημιουργία του project δεν είναι το όνομα πακέτου της εφαρμογής, αλλά είναι το όνομα του φακέλου στον οποίο θα αποθηκευτεί το project, δηλαδή το όνομα του project. Στα επόμενα βήματα επιλέγουμε την έκδοση και το είδος των APIs που θα χρησιμοποιήσουμε για να χτίσουμε την εφαρμογή μας, για καλύτερη συμβατότητα επιλέγουμε την έκδοση 4.0 των Google APIs για να έχουμε πρόσβαση και στο API των Google maps και γενικά συμπληρώνουμε κάποιες σημαντικές ρυθμίσεις. Παρακάτω βλέπουμε αναλυτικά τα βήματα δημιουργίας project. Από το κύριο μενού του Eclipse επιλέγουμε File-New-Project.



Εικόνα 3.15: Δημιουργία project στο Eclipse

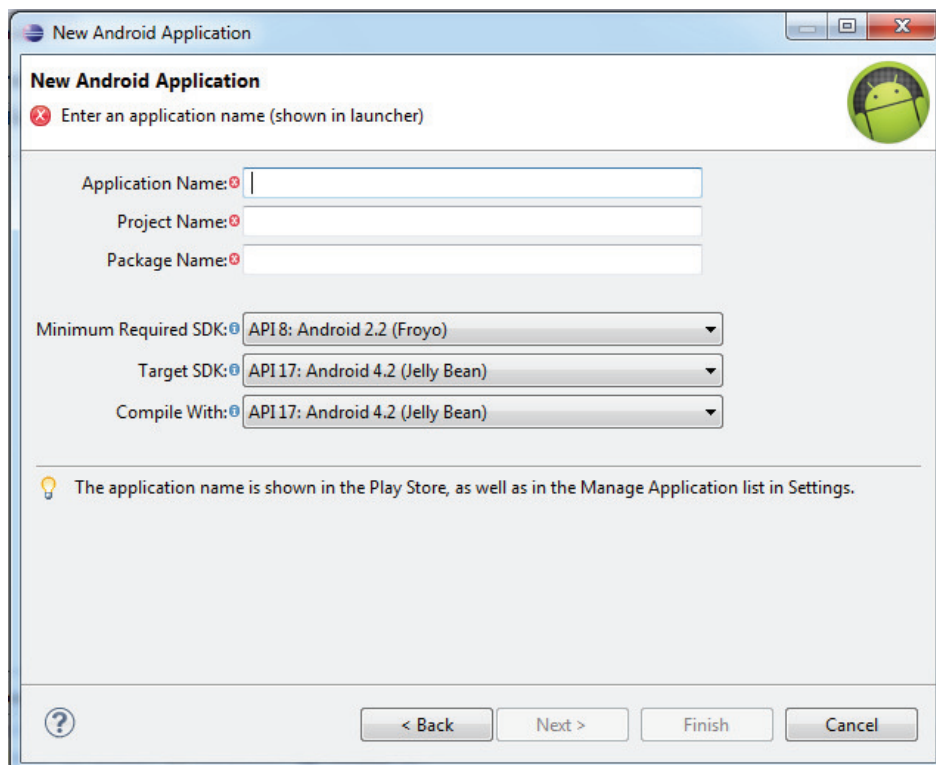
Από το παράθυρο New Project που θα ανοίξει επιλέγουμε Android-Android Application Project όπως φαίνεται στην εικόνα παρακάτω



Εικόνα 3.16: Δημιουργία project στο Eclipse

Στο παράθυρο New Android App που ανοίγει:

- Συμπληρώνουμε το όνομα που θέλουμε να έχει η εφαρμογή μας, το project και το πακέτο.
- Συμπληρώνουμε την παλαιότερη και τη νεότερη έκδοση που θέλουμε να τρέχει η εφαρμογή μας.
- Επιλέγουμε αν θέλουμε να δημιουργηθεί ένα προσαρμοσμένο εικονίδιο εκκίνησης άμα το project αυτό θα χρησιμοποιηθεί σαν βιβλιοθήκη και σε ποιο περιβάλλον εργασίας θέλουμε να το δημιουργήσουμε.



Εικόνα 3.17: Δημιουργία project στο Eclipse

Στο επόμενο μας βήμα διαμορφώνουμε όπως εμείς θέλουμε το εικονίδιο εκκίνησης μας (Launcher icon) και ορίζουμε τη δραστηριότητα (Activity) που θέλουμε. Από τις διαθέσιμες επιλογές, επιλέγουμε το Blank Activity που δημιουργεί μια κενή δραστηριότητα χωρίς δυνατότητα πλοήγησης και στη συνέχεια συμπληρώνουμε το όνομα της κύριας δραστηριότητας (Activity Name), του layout και του τίτλου εφαρμογής μας και πατάμε Finish. Τα περισσότερα χαρακτηριστικά της εφαρμογής που δηλώσαμε παραπάνω, θα χρησιμοποιηθούν για να δημιουργηθεί αυτόματα το αρχείο AndroidManifest.xml. Αυτό σημαίνει ότι αργότερα μπορούμε να τροποποιήσουμε κάποιο από τα παραπάνω πεδία που δηλώσαμε, όπως πχ το όνομα της εφαρμογής, εφόσον αυτό κριθεί απαραίτητο.

• Δημιουργία των Activities της εφαρμογής

Αφού έχουμε ξεκαθαρίσει τις ανάγκες της εφαρμογής μας ήρθε η ώρα να δηλώσουμε τις απαραίτητες οθόνες διεπαφής χρήστη, δηλαδή τις Activities. Έχουμε ήδη δημιουργήσει αυτόματα την Activity Main.java η οποία θα εμφανίζεται στον χρήστη αμέσως μόλις ανοίξει την εφαρμογή, και θα τη χρησιμοποιήσουμε ως κεντρικό σημείο επαφής με τις υπόλοιπες λειτουργίες της εφαρμογής. Για να δημιουργήσουμε μια νέα Activity στο project μας, πηγαίνουμε στη διαδρομή File/New/Class στο Eclipse, και βλέπουμε το παράθυρο δημιουργίας νέας κλάσης. Στη συνέχεια δηλώνουμε το όνομα της εφαρμογής και στο πεδίο "Superclass" εισάγουμε την κλάση android.app.Activity έτσι ώστε η δική μας κλάση να ενσωματώσει όλα τα χαρακτηριστικά της superclass Activity. Τελειώνοντας με τις υπόλοιπες ρυθμίσεις της νέα μας Activity πατάμε το κουμπί finish για να προστεθεί η κλάση Guide στον φάκελο src του project. Με τον ίδιο τρόπο θα δημιουργήσουμε και τις υπόλοιπες Activities της εφαρμογής μας.

Η activity Guide θα χρησιμοποιηθεί όταν ο χρήστης θα κάνει κλικ στην επιλογή Οδηγός από το βασικού μενού και θα εμφανίσει μια λίστα, έναν πίνακα ουσιαστικά με τις επόμενες επιλογές που είναι διαθέσιμες στον χρήστη. Συνολικά η εφαρμογή διαθέτει 6 κλάσεις που ανήκουν στην κατηγορία των Activities για την διεπαφή με τον χρήστη:

- **Guide:** Η οθόνη αντιπροσωπεύει την κύρια οθόνη του οδηγού
- **SubGuide:** Η οθόνη αντιπροσωπεύει την κύρια οθόνη του υπο-οδηγού
- **Map:** Η οθόνη αντιπροσωπεύει την κύρια οθόνη του χάρτη
- **Nearby:** Η οθόνη αντιπροσωπεύει την κύρια οθόνη της τρέχουσας τοποθεσίας
- **Show:** Η οθόνη αντιπροσωπεύει την κύρια οθόνη της εμφάνισης των πληροφοριών
- **Main:** Η κεντρική οθόνη που τρέχει κατά την εκκίνηση της εφαρμογής.

• Δήλωση των Activities στο αρχείο AndroidManifest.xml

Όλες οι Activities της εφαρμογής πρέπει να είναι δηλωμένες στο Αρχείο AndroidManifest.xml, στο οποίο δηλώνονται και οι υπόλοιπες ιδιότητες τους όπως το Intent-filter. Με το Intent-filter ξεκαθαρίζεται ποια θα είναι η πρώτη activity που θα τρέξει όταν θα παίζει η εφαρμογή.

```
<application  
    android:allowBackup="true"
```



```
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="Library.example.nafpaktos.Main"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity android:name="Library.example.nafpaktos.Guide"></activity>
    <activity android:name="Library.example.nafpaktos.SubGuide"></activity>
    <activity android:name="Library.example.nafpaktos.Show"></activity>
    <activity android:name="Library.example.nafpaktos.Map"></activity>
    <activity android:name="Library.example.nafpaktos.NearBy"></activity>

    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyCUoJlvkbbkvnqngwetSN3ZJtZIDeIB3t4jY"/>
</application>
```

Όπως βλέπουμε μεταξύ των tags <application> υπάρχουν όλες οι activities, δηλωμένες με τα αντίστοιχα tags.

```
<activity
    android:name="Library.example.nafpaktos.Main"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Στο πρώτο πεδίο name δηλώνουμε το όνομα της Activity ακριβώς όπως και στο αρχείο κλάσης που δημιουργήσαμε παραπάνω. Στο δεύτερο πεδίο name δηλώνουμε το όνομα του intent-filter το οποίο αποτελείται από το όνομα πακέτου και το όνομα της κλάσης. Στο τρίτο πεδίο name του tag category αλλάζουμε το τελευταίο κομμάτι και γράφουμε LAUNCHER που δηλώνει πως η κλάση που ορίσαμε θα εμφανίζεται πρώτη στον χρήστη κατά την εκκίνηση της εφαρμογής. Παρομοίως δηλώσαμε και τις υπόλοιπες 5 δραστηριότητες όπως δείξαμε παραπάνω και τις εισάγαμε ανάμεσα στα application tags του AndroidManifest.

• Ανάλυση στοιχείων Android

Ξεκινώντας, η κάθε εφαρμογή αποτελείται από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν μοναδικό όνομα πακέτου (package name), το οποίο χρησιμοποιείται από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής. Μια εφαρμογή μπορεί να αποτελείται από πολλά υποπακέτα, λόγω της πολυπλοκότητας της εφαρμογής, αλλά από μόνο ένα κύριο.

○ Το αρχείο AndroidManifest.xml

Είναι το κύριο συστατικό κάθε εφαρμογής Android εφαρμογή διότι περιέχει τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

1. Ονομάζει και αναγνωρίζει τα java πακέτα της εφαρμογής μέσα στα οποία βρίσκονται οι java κλάσεις.
2. Περιγράφει όλα τα στοιχεία από τα οποία αποτελείται μια εφαρμογή (activities, services...).
3. Προσδιορίζει τις διαδικασίες που θα χρειαστούν να υποστηρίξουν την εφαρμογή.
4. Ορίζει τα δικαιώματα τα οποία πρέπει να έχει η εφαρμογή για να μπορεί να αλληλεπιδρά με άλλες.
5. Ορίζει τα δικαιώματα που πρέπει να έχουν οι άλλες εφαρμογές για να χρησιμοποιήσουν την δική μας.
6. Δηλώνει το ελάχιστο επίπεδο Android API που απαιτεί η εφαρμογή.
7. Παραθέτει τις βιβλιοθήκες που χρειάζεται η εφαρμογή.

○ Φάκελος src

Ο φάκελος src (από την λέξη source) περιέχει τον πηγαίο κώδικα της εφαρμογής μας. Όλος ο κώδικας ενσωματώνεται αυτόματα σε ένα Library του οποίου το όνομα πρέπει να είναι μοναδικό για τον υπολογιστή και το οποίο το έχουμε δηλώσει εμείς κατά την δημιουργία της εφαρμογής.

Μέσα σε αυτό το Library περιέχονται όλα τα αρχεία JAVA της εφαρμογής μας.

○ Φάκελος gen

Στον φάκελο gen περιέχεται το αρχείο R.* το οποίο καταγράφεται αυτοματοποιημένα και περιέχει τα αντικείμενα XML που χρησιμοποιούνται και τα activities. Τα αρχεία αυτά παράγονται από το ADT και καλό είναι να μην τα επεξεργαζόμαστε.

- Βιβλιοθήκες

Στον φάκελο αυτό παρατηρούμε δύο βιβλιοθήκες, Android και Android Dependencies, οι οποίες περιέχουν αρχεία java τα οποία ορίζουν ένα μορφότυπο συνάθροισης πολλών αρχείων σε ένα και χρησιμοποιούνται για την διανομή κλάσεων java.

- Φάκελος assets

Στον φάκελο αυτό, ο οποίος είναι αρχικά άδειος, μπορούμε να αποθηκεύσουμε αρχεία και καταλόγους της εφαρμογής τους οποίους όμως δεν θέλουμε να διαχειριστούμε σαν πόρους της εφαρμογής.

- Φάκελος bin

Ο φάκελος bin παράγεται από το compiler που περιέχει όλα τα αρχεία με κατάληξη .class από τα οποία παράγεται το αρχείο .apk. μέσα σε αυτόν τον φάκελο βρίσκουμε το τελικό εκτελέσιμο αρχείο .apk με το οποίο μπορούμε να εκτελέσουμε την εφαρμογή μας σε κάποια συσκευή Android.

- Φάκελος res

Στον φάκελο res περιέχονται όλα τα αρχεία που αφορούν εικόνα και ήχο και γενικά ότι έχει να κάνει με τη γραφική διεπαφή. Σε αυτό τον φάκελο βρίσκονται και οι παρακάτω σημαντικοί υποφάκελοι:

1. Anim που περιέχει αρχεία xml που αναφέρονται σε αντικείμενα κινούμενων σχεδίων.
2. Color που περιέχει αρχεία xml που αναφέρονται σε λίστες περιγραφής χρωμάτων.
3. Drawable που περιέχει αρχεία bitmap και αρχεία xml που περιγράφουν σχεδιάσιμα σχήματα ή αντικείμενα.
4. Layout που περιέχει xml αρχεία που αναφέρονται στον τρόπο διάταξης της οθόνης.
5. Menu που περιέχει xml αρχεία που έχουν να κάνουν με το μενού της εφαρμογής μας.
6. Raw που περιέχει ακατέργαστα αρχεία όπως και στο φάκελο assets. Η διαφορά εδώ είναι ο τρόπος με τον οποίο έχουμε πρόσβαση σε αυτά.
7. Values που περιέχει xml αρχεία που έχουν να κάνουν με: τιμές Boolean, χρώματα, διαστάσεις, ακεραίους, συμβολοσειρές, συστοιχεία συμβολοσειρών κτλ.
8. Xml που περιέχει xml αρχεία που διαμορφώνουν τα στοιχεία της εφαρμογής.

- Φάκελος layout

Ο φάκελος Layout αφορά τις διεπαφές-φόρμες της εφαρμογής. Κάθε αρχείο μέσα στον φάκελο Layout είναι ένα αρχείο XML που περιγράφει την φόρμα που θα εμφανιστεί. Πάντα μια φόρμα ή αλλιώς Activity πρέπει να συνδιάζεται με ένα αρχείο java.

- Φάκελος menu

Σε αυτόν τον φάκελο περιέχεται ένα αρχείο main.xml που περιέχει πληροφορίες σχετικά με το κεντρικό μενού της εφαρμογής. Όταν ο χρήστης πατήσει το κουμπί menu του κινητού του και η εφαρμογή τρέχει εκείνη την στιγμή θα εμφανιστούν οι 3 διαθέσιμες επιλογές (GUIDE,MAP,NEARBY).

- Φάκελος values

Ο φάκελος values περιέχει xml αρχεία που περιγράφουν τα μενού του Οδηγού (GUIDE) και των υπομενού του οδηγού (SUBGUIDE). Κάθε xml αρχείο ουσιαστικά περιέχει πληροφορίες που αφορούν τον τίτλο, το όνομα της εικόνας, μια σύντομη περιγραφή και τέλος το όνομα του αρχείου xml που περιέχει τις περισσότερες πληροφορίες για κάθε αντικείμενο.

- Φάκελος menu

Σε αυτόν τον φάκελο βρίσκονται οι εξωτερικές βιβλιοθήκες που μπορεί να χρησιμοποιήσει η εφαρμογή. Δίνει την δυνατότητα να ενσωματωθούν κλάσεις και επιπλέον χαρακτηριστικά στην εφαρμογή.

- Αρχείο projects.properties

Αυτό το αρχείο περιέχει τις ρυθμίσεις του project και είναι αναπόσπαστο κομμάτι του.

ΚΕΦΑΛΑΙΟ 4

Σχεδιασμός εφαρμογής και στόχοι υλοποίησης

- Σχεδιασμός εφαρμογής περιήγησης στην πόλη της Ναυπάκτου

Έχοντας ζήσει στην πόλη της Ναυπάκτου κατά τη διάρκεια των φοιτητικών μας χρόνων, μας ήρθε η ιδέα να φτιάξουμε μια εφαρμογή για τις ανάγκες της πτυχιακής εργασίας. Η συγκεκριμένη εφαρμογή θεωρούμε πως είναι πολύ εύχρηστη για όλους τους τουρίστες και μη που επισκέπτονται την Ναύπακτο, οι οποίοι θα έχουν την δυνατότητα να περιηγούνται και να συλλέγουν όσες απαραίτητες πληροφορίες που χρειάζονται για τα σημεία ενδιαφέροντος της πόλης όπως για παράδειγμα που βρίσκεται το Λιμάνι της Ναυπάκτου, το Κάστρο, το Ρολόι κτλ, για διάφορα εστιατόρια και καφέ, ξενοδοχεία, καθώς και μικρές περιγραφές για το κάθε ένα με σκοπό την ευκολία της μετακίνησης τους και την κατάλληλη επιλογή τους ανάλογα πάντα με τις ανάγκες τους.

- Καθορισμός χαρακτηριστικών γνωρισμάτων της εφαρμογής

Μια καλή εφαρμογή περιήγησης θα πρέπει να περιέχει εκτός από τον βασικό χάρτη, και άλλα χαρακτηριστικά γνωρίσματα όπως για παράδειγμα:

- Μια οθόνη έναρξης
- Μια οθόνη που θα εξυπηρετεί σαν τουριστικός οδηγός που θα παρέχει πληροφορίες για τα σημεία ενδιαφέροντος
- Έναν τρόπο να προβάλλονται ποια σημεία ενδιαφέροντος βρίσκονται κοντά στον χρήστη

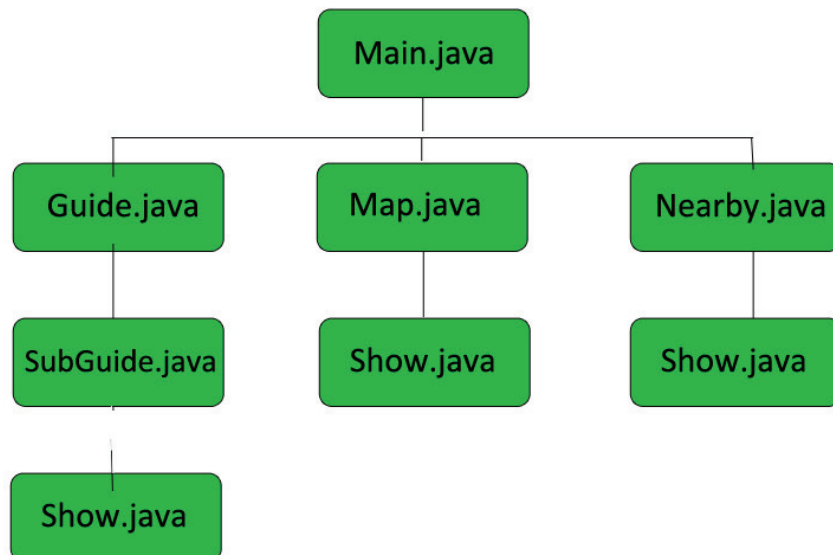
Χρειάζεται επίσης ένας τρόπος για να γίνεται η μετάβαση μεταξύ αυτών των διαφορετικών χαρακτηριστικών γνωρισμάτων. Ένας τρόπος να γίνει αυτό είναι να δημιουργηθεί μια κύρια οθόνη του βασικού μενού όπου ο χρήστης μπορεί να χρησιμοποιεί για να έχει πρόσβαση στα χαρακτηριστικά γνωρίσματα της εφαρμογής. Αναθεωρώντας λοιπόν τις απαιτήσεις της εφαρμογής μας, διαπιστώνουμε ότι χρειαζόμαστε έξι αρχικές οθόνες της εφαρμογής περιήγησης στην Ναύπακτο:

- Μια οθόνη έναρξης
- Μια οθόνη του οδηγού
- Μια οθόνη του υπο-οδηγού
- Μια οθόνη του χάρτη
- Μια οθόνη εμφάνισης τρέχουσας τοποθεσίας
- Μια οθόνη εμφάνισης πληροφοριών

- Καθορισμός απαιτήσεων των δραστηριοτήτων (Activities)

Κάθε οθόνη της εφαρμογής περιήγησης έχει την δική της κλάση δραστηριότητας (Activity). Η τακτική που ακολουθήσαμε, θεωρώντας την φυσικά και πιο ορθή, είναι να εφαρμοστεί αρχικά μια κύρια κλάση δραστηριότητας την Main.java η οποία καλείται από το αρχείο των ρυθμίσεων AndroidManifest.xml κατά την εκκίνηση της εφαρμογής. Παρακάτω ακολουθούν οι υπόλοιπες κλάσεις δραστηριότητας (Activities) οι οποίες είναι:

- Guide.java - η κλάση αντιπροσωπεύει την κύρια οθόνη του οδηγού
- SubGuide.java – αυτήν η κλάση αντιπροσωπεύει την κύρια οθόνη του υπο-οδηγού
- Map.java - η κλάση αντιπροσωπεύει την κύρια οθόνη του χάρτη
- Nearby.java - η κλάση αντιπροσωπεύει την κύρια οθόνη της τρέχουσας τοποθεσίας
- Show.java – η κλάση αυτή εμφανίζει με μια συγκεκριμένη σειρά της πληροφορίες



Εικόνα 4.1: Πρόχειρος σχεδιασμός των ροών δεδομένων των δραστηριοτήτων

- Καθορισμός χαρακτηριστικών γνωρισμάτων οθονών της εφαρμογής

- Καθορισμός χαρακτηριστικών γνωρισμάτων οθόνης έναρξης

Η Οθόνη Έναρξης χρησιμεύει ως το αρχικό σημείο εισόδου για την εφαρμογή. Η λειτουργία της πρέπει να ενσωματωθεί στον κώδικα του αρχείου Main.java. Αυτή η οθόνη πρέπει να κάνει τα εξής:

- ❖ Να εμφανίζει το όνομα της εφαρμογής
- ❖ Να εμφανίζει ένα ενδιαφέρον γραφικό ή λογότυπο για την εφαρμογή
- ❖ Να μεταβαίνει αυτόματα στο μενού επιλογών μετά από ορισμένη χρονική περίοδο

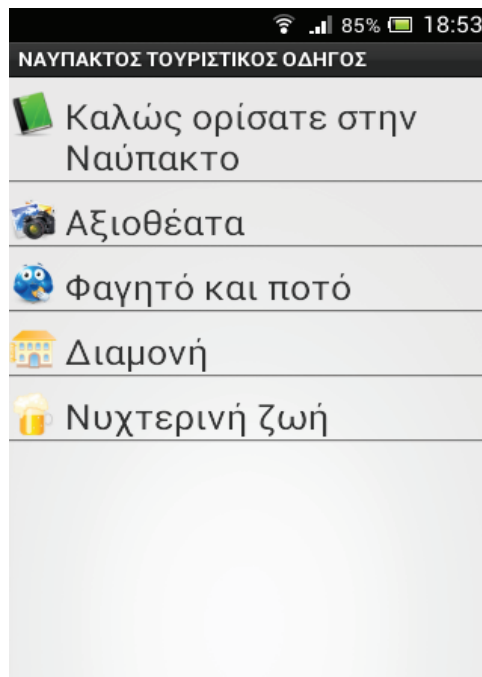


Εικόνα 4.2: Πρότυπο οθόνης έναρξης

- Καθορισμός χαρακτηριστικών γνωρισμάτων κύριας οθόνης οδηγού

Η Κύρια Οθόνη Οδηγού εμφανίζει μια λίστα με τις κύριες επιλογές της εφαρμογής, όπου ο χρήστης μπορεί να λάβει τις πληροφορίες που χρειάζεται. Η λειτουργία της πρέπει να ενσωματωθεί μέσα στον κώδικα του αρχείου Guide.java. Αυτή η οθόνη πρέπει να κάνει τα εξής:

- ❖ Να εμφανίζει τη λίστα με τις επιλογές στο χρήστη
- ❖ Να επιτρέπει την μετάβαση σε κάποια άλλη οθόνη στο χρήστη

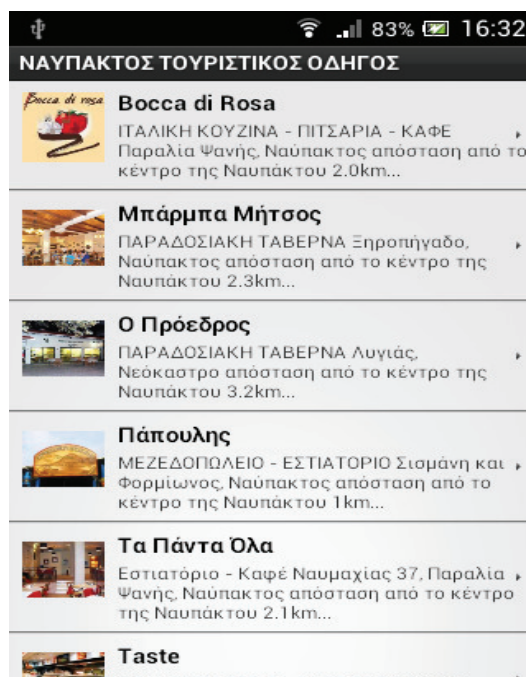


Εικόνα 4.3: Πρότυπο κύριας οθόνης επιλογών

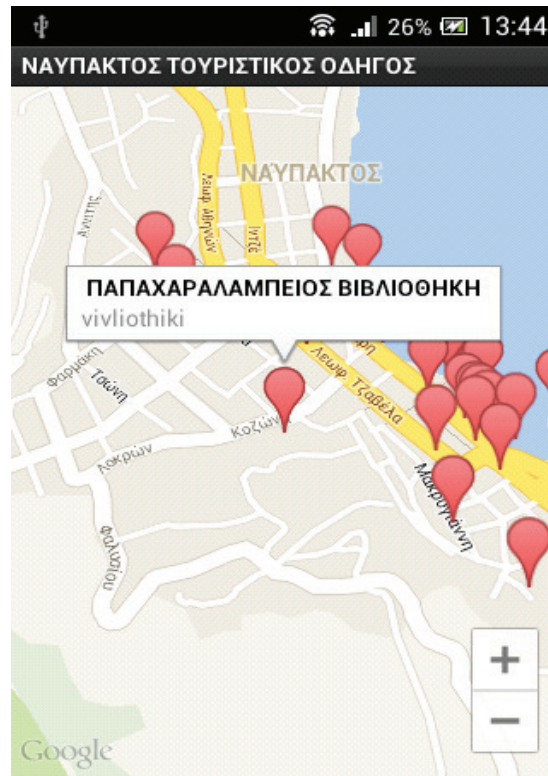
- Καθορισμός χαρακτηριστικών γνωρισμάτων οθόνης Υπο-οδηγού

Η Οθόνη Υπο-οδηγού εμφανίζει στο χρήστη μια λίστα με επιλογές που ανάλογα με τις ανάγκες του χρήστη οδηγείτε σε κάποια άλλη οθόνη. Η λειτουργία της πρέπει να ενσωματωθεί μέσα στον κώδικα του αρχείου SubGuide.java. Αυτή η οθόνη πρέπει να κάνει τα εξής:

- ❖ Να εμφανίζει τη λίστα με τις επιλογές στο χρήστη
- ❖ Να επιτρέπει την μετάβαση σε κάποια άλλη οθόνη στο χρήστη



Εικόνα 4.4: Πρότυπο οθόνης υπο-οδηγού



Εικόνα 4.5: Πρότυπο οθόνης τρέχουσας τοποθεσίας

- Καθορισμός χαρακτηριστικών γνωρισμάτων οθόνης εμφάνισης πληροφοριών

Η Οθόνη Εμφάνισης Πληροφοριών εμφανίζει στον χρήστη όλες τις απαραίτητες πληροφορίες που χρειάζονται για τα σημεία ενδιαφέροντος της πόλης. Η λειτουργία της πρέπει να ενσωματωθεί στον κώδικα του αρχείου Show.java. Αυτή η οθόνη πρέπει να κάνει τα εξής:

- ❖ Να εμφανίζει την εικόνα και ένα κείμενο πληροφοριών
- ❖ Να επιτρέπει την κύλιση του κειμένου στον χρήστη



Εικόνα 4.6: Πρότυπο οθόνης εμφάνισης πληροφοριών

- Εφαρμογή ενός πρωτοτύπου εφαρμογής

Τώρα που έχουμε μια ιδέα τι θα κάνει και πως θα φαίνεται η εφαρμογή μας, είναι η ώρα να αρχίσει ο προγραμματισμός της. Αυτό περιλαμβάνει τα ακόλουθα βήματα:

- Δημιουργία ενός νέου Android προγράμματος στο προγραμματιστικό περιβάλλον Eclipse
- Προσθήκη πόρων στο πρόγραμμά μας, συμπεριλαμβανομένων strings και γραφικών (drawable)
- Δημιουργία ενός σχεδιαγράμματος (layout) για κάθε οθόνη
- Εφαρμογή μιας Δραστηριότητας για κάθε οθόνη

- Δημιουργία Android προγράμματος στο Eclipse

Το πρόγραμμά μας έχει τις ακόλουθες ρυθμίσεις:

- Project name: Nafpaktos
- Build target: Android 2.2 + Google APIs
- Application name: Ναύπακτος Τουριστικός Οδηγός
- Package name: library.example.nafpaktos
- Create activity: Main.java

- **Στόχοι υλοποίησης εφαρμογής**

Ο στόχος της πτυχιακής εργασίας είναι η ανάπτυξη και ο σχεδιασμός ενός λογισμικού κινητού τηλεφώνου για το λειτουργικό σύστημα Android της Google με σκοπό την ανάδειξη των δυνατοτήτων αυτής της ραγδαία αναπτυσσόμενης πλατφόρμας. Για την επίτευξη του στόχου χρησιμοποιήθηκε το λογισμικό Eclipse το οποίο είναι ένα πρόγραμμα ανοιχτού κώδικα που σε συνεργασία με τα δωρεάν εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google (Android Development Tools) αποτελεί ένα πολύ ισχυρό εργαλείο ανάπτυξης εφαρμογών Android.

Ο κύριος στόχος υλοποίησης της εφαρμογής περιήγησης είναι η ανάπτυξη μιας εφαρμογής η οποία θα δίνει πληροφορίες για την πόλη της Ναυπάκτου σε κατοίκους, φοιτητές και ακόμα περισσότερο σε τουρίστες να γνωρίσουν την πόλη και να περιηγηθούν σ'αυτήν. Ο σχεδιασμός της εφαρμογής λοιπόν θα πρέπει να είναι τέτοιος ώστε ο χρήστης αμέσως να εντοπίζει που βρίσκεται η πληροφορία που τον ενδιαφέρει, να χρησιμοποιήσει τα εργαλεία της εφαρμογής και μέσω αυτών να έχει πρόσβαση στα υπόλοιπα παρεμφερή αντικείμενα πληροφοριών της πόλης.

Στόχος μας είναι να διευκόλυνουμε τον χρήστη δίνοντάς του πρόσβαση στο μενού της εφαρμογής μέσω ενός κουμπιού και στη συνέχεια μέσω ενός παράθυρου με μία λίστα από πληροφορίες ο χρήστης θα επιλέγει σε ποιες επιθυμεί να έχει πρόσβαση. Το μενού περιέχει τα 3 βασικά πεδία της εφαρμογής, τον Οδηγό, τον Χάρτη και την Τρέχουσα Τοποθεσία όπου πατώντας το κουμπί του μενού μπορεί ανα πάσα στιγμή να έχει πρόσβαση σ'αυτό.

Η εφαρμογή θα παρέχει πρόσβαση σε σημεία ενδιαφέροντος της πόλης, όπως τα αξιοθέατα, το φαγητό και ποτό, τη διαμονή και τη νυχτερινή ζωή. Επίσης την προβολή του χάρτη της πόλης όπου θα μπορεί να έχει μια ολοκληρωμένη εικόνα για το που βρίσκεται, ώστε να έχει ένα γρήγορο σημείο αναφοράς αφού υπάρχει η δυνατότητα εντοπισμού της τρέχουσας τοποθεσίας και τι υπάρχει γύρω του, κάνοντας έτσι πιο εύκολη την περιήγηση του στη πόλη.

ΚΕΦΑΛΑΙΟ 5

Ανάλυση προβλήματος εφαρμογής

- Δομή της εφαρμογής

Η δομή της εφαρμογής περιήγησης στην πόλη της Ναυπάκτου αποτελείται από τα αρχεία με κατάληξη .xml και .java. Συγκεκριμένα η εφαρμογή περιλαμβάνει:

- Έξι αρχεία διάταξης (layout), τα οποία είναι το `guide.xml`, το `main.xml`, το `map.xml`, το `nearby.xml`, το `show.xml` και το `subguide.xml`
- Το αρχείο `main.xml` το οποίο βρίσκεται στο φάκελο `menu` και περιέχει την διάταξη του μενού επιλογής της εφαρμογής
- Τον φάκελο `values` που περιέχει όλους τους πίνακες (arrays) τους οποίους με την χρήση του κώδικα τους διαβάζουμε και φτιάχνουμε τις λίστες για το μενού. Μέσα στο φάκελο αυτό περιέχονται 8 αρχεία το `accommodation.xml`, το `dimens.xml`, που δημιουργείται αυτόματα από το Android, το `eatanddrink.xml`, το `guide_menu.xml`, το `nightlife.xml`, το `seeanddo.xml`, το `strings.xml` με τα ονόματα των μεταβλητών και το `styles.xml` που δημιουργείται επίσης αυτόματα
- Ο φάκελος `xml` περιέχει όλα τα xml αρχεία μέσα στα οποία βρίσκονται οι περιγραφές του κάθε σημείου ενδιαφέροντος

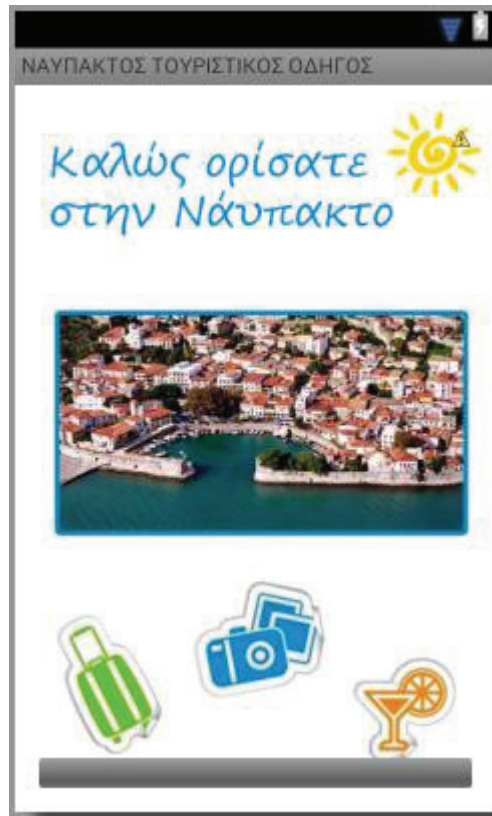
Επιλέξαμε xml αρχεία διότι λόγω της δομής τους μπορούμε πολύ εύκολα να αναζητήσουμε τα δεδομένα που θέλουμε με βάση κάποια κριτήρια, είναι πολύ εύκολο να τα δημιουργήσουμε και ο χώρος που πιάνουν είναι μικρός. Πιο αναλυτικά την δομή του κάθε αρχείου θα την δούμε παρακάτω.

- Υλοποίηση της κεντρικής οθόνης της εφαρμογής

Η κεντρική οθόνη εμφανίζεται στον χρήστη μόλις εκείνος πατήσει το κουμπί της εφαρμογής. Η οθόνη που εμφανίζεται αρχικά περιλαμβάνει το αρχικό εικονίδιο της εφαρμογής καθώς και μια μπάρα εξέλιξης προόδου τα οποία εμφανίζονται μόλις 3 δευτερόλεπτα. Αμέσως μετά εμφανίζεται η οθόνη που περιλαμβάνει τα 3 βασικά μενού πλοήγησης τα οποία είναι:

- Το κουμπί `GUIDE`
- Το κουμπί `MAP`
- Και το κουμπί `NEARBY`

Η κεντρική οθόνη με το αρχικό εικονίδιο και την μπάρα προόδου δημιουργείται στο αρχείο `main.xml` στον φάκελο `layout` και σε αυτό ορίζεται η διάταξη που θα έχουν τα αντικείμενα στην οθόνη, τον προσανατολισμό και το φόντο που θα έχει η κεντρική οθόνη.



Εικόνα 4.1: Κεντρική οθόνη εφαρμογής περιήγησης

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".Main" >
```

Εικόνα 4.2: Ορισμός διάταξης αντικειμένων

Στον παραπάνω κώδικα βλέπουμε πως ορίζεται η διάταξη των αντικειμένων, το ύψος και το πλάτος περιέχουν την τιμή `wrap_content` και το περιεχόμενο του `RelativeLayout` θα εμφανίζεται στο κέντρο της οθόνης. Στο `RelativeLayout` περιέχονται επίσης η αρχική εικόνα της εφαρμογής καθώς και η μπάρα εξέλιξης προόδου όπως φαίνεται στην παρακάτω εικόνα.

```

<ImageView
    android:id="@+id/IntroImage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:contentDescription="@string/IntroImage"
    android:src="@drawable/intro" />

<ProgressBar
    android:id="@+id/progressBar1"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/IntroImage"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="false"
    android:layout_alignRight="@+id/IntroImage"
    android:layout_marginBottom="35dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp" />

```

Εικόνα 4.3: Κώδικας για την αρχική οθόνη και την μπάρα εξέλιξης προόδου

Το Android χρησιμοποιεί το όρισμα `id` που παρέχει ένα μοναδικό αναγνωριστικό για την προβολή και μπορεί να χρησιμοποιηθεί στον πηγαίο κώδικα της εφαρμογής για να συνδέσει αρχεία τύπου `java` με τα `xml` αρχεία. Έτσι όπως βλέπουμε παραπάνω στον κώδικα `android:id="@+id/IntroImage"` και `android:id="@+id/progressBar1"` ορίζεται ότι η αρχική οθόνη και η μπάρα εξέλιξης προόδου έχουν αναγνωριστικό τον πόρο `IntroImage` και `progressBar1` αντίστοιχα που είναι τύποι `id`. Επειδή ο τύπος του πόρου ορίζεται για πρώτη φορά χρησιμοποιείται το `+`. Από το `AndroidManifest.xml` αρχείο γνωστοποιείται πως το πρώτο `activity` που θα τρέξει είναι το `Main.java`. Αυτό το βλέπουμε στο σημείο του παρακάτω κώδικα όπου βλέπουμε πως η `Main` είναι η `Launcher activity` της εφαρμογής:

```

<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>

```

Εικόνα 4.4: Κώδικας για το αρχικό `activity`

- Το πρώτο αρχείο που θα τρέξει η εφαρμογή (`Main.java`)

Παρακάτω θα δούμε τι περιέχει το αρχείο `Main.java`. Όπως αναφέρθηκε και προηγουμένως το `Main.java` είναι το πρώτο αρχείο που θα τρέξει η εφαρμογή καθώς και ξεκινάει και έτσι ορίζεται και στο αρχείο ρυθμίσεων, `AndroidManifest.xml`. Μέσα στον `Constructor` της `Main` έχουμε ορίσει έναν `timer`, ο οποίος είναι στην ουσία ένα αντικείμενο και χρησιμεύει για το οπτικό εφέ μετακίνησης της μπάρας εξέλιξης προόδου. Ο `timer` έχει προγραμματιστεί έτσι ώστε μετά από 3 δευτερόλεπτα να εμφανίζει το κύριο μενού της εφαρμογής με την βοήθεια της συνάρτησης `openOptionsMenu`. Πιο συγκεκριμένα έχουμε ορίσει μια συνάρτηση `run` η οποία θα τρέχει ανά δευτερόλεπτο και μέσα σε αυτήν βρίσκεται ένα αντικείμενο το `ProgressBar` το οποίο βρίσκει το `id` του και το επιστρέφει.

Με αυτό το αντικείμενο δηλώνουμε μέχρι που θα φτάσει η μπάρα εξέλιξης προόδου. Έτσι κάθε φορά που θα τρέχει η συνάρτηση `run`, ο `timer` αυξάνεται κατά 1 και το `ProgressBar` πολλαπλασιάζεται επί 33. Για παράδειγμα στο πρώτο δευτερόλεπτο η μπάρα

εξέλιξης προόδου έχει φτάσει στο 33%, στο δεύτερο δευτερόλεπτο η μπάρα θα φτάσει στο 66% κ.ο.κ. Όταν ο *timer* θα είναι 3 με την εντολή *PB.setProgress(100)*; δηλώνουμε πως η μπάρα έχει φτάσει στο 100% ενώ με την εντολή *t.cancel()*; αναιρούμε τον *timer* για να μην καταναλώνει άσκοπα πόρους. Επίσης εφόσον ανοίξει το βασικό μενού, με την εντολή *PB.setVisibility(View.INVISIBLE)*;; μετατρέπουμε την μπάρα εξέλιξης προόδου σε αόρατη. Εφόσον εμφανιστεί το κύριο μενού στην οθόνη, ο χρήστης είναι έτοιμος να διαλέξει τι θέλει να κάνει: 1) Να εμφανίσει τον οδηγό, 2) Να εμφανίσει τον χάρτη, 3) Να εμφανίσει την τρέχουσα τοποθεσία στην οποία βρίσκεται.

```
final Timer t = new Timer();
t.scheduleAtFixedRate(new TimerTask() {

    public void run() {
        runOnUiThread(new Runnable() {
            public void run() {

                ProgressBar PB= (ProgressBar) findViewById(R.id.progressBar1);
                time += 1;

                PB.setProgress(time*33);
                if(time == 3)
                {
                    PB.setProgress(100);
                    t.cancel();

                    openOptionsMenu();

                    ImageView introImage= (ImageView) findViewById(R.id.IntroImage);
                    introImage.setVisibility(View.INVISIBLE);

                    PB.setVisibility(View.INVISIBLE);

                }

            }

        });
    }

}, 0, 1000);
}
```

Εικόνα 4.5: Κώδικας για τον timer της μπάρας εξέλιξης προόδου

- Εμφάνιση βασικού μενού (OptionsMenu.java και OptionMenu2.java)

Το βασικό μενού της εφαρμογής περιγράφεται στα δύο αρχεία *OptionsMenu.java* και *OptionMenu2.java* και τα δύο αυτά αρχεία αφορούν το μενού απλά το κάθε ένα με διαφορετική προσέγγιση. Τα δύο αυτά αρχεία αποτελούν τον βασικό κώδικα του βασικού μενού (Οδηγός, Χάρτης, Γύρω μου). Ο λόγος που χρησιμοποιούνται δύο αρχεία είναι ότι το ένα μενού κληρονομεί την βιβλιοθήκη *ListActivity* και το δεύτερο την *FragmentActivity*. Ας δούμε πρώτα τον κώδικα για το αρχείο *OptionsMenu.java*. Όπως αναφέραμε και προηγουμένως το αρχείο *OptionsMenu.java* κληρονομεί την *FragmentActivity*. Στον

constructor της υπάρχει η συνάρτηση `onCreateOptionsMenu` με την οποία αντλούμε στοιχεία από το αρχείο `main.xml`, που βρίσκεται στο φάκελο `menu`, και ουσιαστικά μέσα σε αυτό το αρχείο βρίσκονται τα 3 αντικείμενα που αφορούν το βασικό μενού της εφαρμογής μας. Αν η άντληση των στοιχείων γίνει επιτυχώς τότε επιστρέφεται η τιμή `true` όπως φαίνεται και στον κώδικα παρακάτω.

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}
```

Εικόνα 4.6: Κώδικας του `OptionsMenu`

Στην συνέχεια, με την βοήθεια της συνάρτησης `onOptionsItemSelected` ο χρήστης οδηγείται στο κατάλληλο γι' αυτόν Activity. Μέσα στην συνάρτηση αυτή καθορίζεται με την συνθήκη `if,else if` ποιο Activity θα ακολουθήσει. Αν το αντικείμενο που επιλέχθηκε, το οποίο το βρίσκουμε από το `id` του, είναι ίσο με το `guide` τότε δημιουργείται μια καινούργια Activity, με την εντολή `Intent`, η `Guide.java`. Αν το `id` είναι ίσο με το `map` τότε η καινούργια Activity που δημιουργείται είναι η `Map.java` και τέλος αν το `id` είναι ίσο με το `nearby` τότε η Activity που δημιουργείται είναι η `Nearby.java`. Περίπου η ίδια διαδικασία ακολουθείται και στο `OptionsMenu2.java`. Ο λόγος που κληρονομεί το κάθε μενού διαφορετική Activity (`ListActivity` και `FragmentActivity`), είναι γιατί όλες οι υπόλοιπες κλάσεις κληρονομούν τις `OptionsMenu` και `OptionsMenu2`. Αυτό γίνεται για να είναι το μενού ορατό καθ' όλη τη διάρκεια της εφαρμογής.

- Εμφάνιση οδηγού (`Guide.java`)

Όταν ο χρήστης επιλέξει την εμφάνιση του οδηγού από το βασικό μενού της εφαρμογής, τότε ο constructor της `guide.java` θα τρέξει δημιουργώντας μια λίστα `MyAdapter` με αντικείμενα η οποία αντλεί στοιχεία από το αρχείο `guide_menu.xml`. Στην συνέχεια βλέπουμε πως έχουμε ορίσει την συνάρτηση `getView` η οποία είναι υπεύθυνη για την εμφάνιση των αντικειμένων (Καλώς ορίσατε στην Ναύπακτο, Αξιοθέατα κτλ) καθώς και τις αντίστοιχες εικόνες τους. Αργότερα ορίζουμε την συνάρτηση `setOnClickListener` η οποία ενεργοποιείται όταν ο χρήστης κάνει κλικ σε μια επιλογή. Στην ουσία πρόκειται για έναν `Listener` ο οποίος περιμένει να «ακούσει» την κίνηση του χρήστη. Αν λοιπόν ο χρήστης επιλέξει την πρώτη επιλογή τότε δημιουργείται μια μεταβλητή η `fname` στην οποία δίνουμε την τιμή `welcome` και ταυτόχρονα δημιουργείται ένα `intent` (μια συνάρτηση) η οποία καλεί και εμφανίζει μια άλλη Activity, την `Show.java`. Σε όλες τις υπόλοιπες περιπτώσεις η διαδικασία είναι περίπου ίδια μόνο που η Activity που καλείται και εμφανίζεται είναι η `SubGuide.java`.

```

public void onClick(View v) {

    if(tv.getText().equals("Καλώς ορίσατε στην Ναύπακτο"))
    {
        Intent i = new Intent(Guide.this, Show.class);
        i.putExtra("fname","welcome");

        startActivity(i);
    }
    else if(tv.getText().equals("Αξιοθέατα"))
    {

        Intent i = new Intent(Guide.this, SubGuide.class);
        i.putExtra("fname","seeanddo");

        startActivity(i);

    }

    else if(tv.getText().equals("Φαί και ποτό"))
    {

        Intent i = new Intent(Guide.this, SubGuide.class);
        i.putExtra("fname","eatanddrink");

        startActivity(i);

    }
}

```

Εικόνα 5.7: Κώδικας για το Guide.java

- Εμφάνιση υπο-οδηγού (SubGuide.java)

Όταν ο χρήστης επιλέξει να δει κάποια από τις υποκατηγορίες, εκτός βέβαια της πρώτης, από τον τότε θα τρέξει ο κώδικας του αρχείου *SubGuide.java*. Σε αυτό το σημείο, ο κώδικας παίρνει την τιμή που στάλθηκε από την προηγούμενη Activity και την μετατρέπει κατάλληλα έτσι ώστε η μεταβλητή *name_of_submenu* να δημιουργηθεί το όνομα του υπομενού. Για παράδειγμα τον τίτλο Eat And Drink τον μετατρέπει σε *eatanddrink* ώστε να μπορεί να καλέσει το αρχείο *xml* με αυτό το όνομα.

```

String cat=getIntent().getExtras().getString("fname");
name_of_submenu=cat.replace(" ", "");
name_of_submenu=name_of_submenu.toLowerCase();

```

Εικόνα 5.8: Κώδικας για την μετατροπή του τίτλου

Στην συνέχεια βλέπουμε πως ο κώδικας περιέχει σχεδόν τα ίδια με το αρχείο *Guide.java*. Εμφανίζεται πάλι η συνάρτηση *getView* η οποία εμφανίζει τα αντικείμενα με τους αντίστοιχους τίτλους, τις αντίστοιχες εικόνες, ένα ακόμα αντικείμενο που προστίθεται τώρα, ένα εικονίδιο που χρησιμοποιείται απλά για οπτικό εφέ και ένα αντικείμενο που περιέχει κάποιες πληροφορίες. Στην συνέχεια έχουμε πάλι μια συνάρτηση *setOnClickListener* η οποία λειτουργεί με τον ίδιο τρόπο, με την μόνη αλλαγή πως τώρα η Activity που καλείται και εμφανίζεται είναι σε όλες τις περιπτώσεις, *Show.java*, η οποία παίρνει την μεταβλητή που της στάλθηκε, την τοποθετεί στην μεταβλητή *name* και ανοίγει και διαβάζει το αντίστοιχο *xml* αρχείο.

```

        public void onClick(View v) {

            Intent i = new Intent(SubGuide.this, Show.class);
            i.putExtra("fname",fileText.getText().toString());

            startActivity(i); //start new activity

        }
    });

    infoText.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {

            Intent i = new Intent(SubGuide.this, Show.class);
            i.putExtra("fname",fileText.getText().toString());

            startActivity(i); //start new activity
        }
    });

```

Εικόνα 5.9: Κώδικας από το SubGuide.java

- Εμφάνιση πληροφοριών (Show.java)

Το αρχείο αυτό περιλαμβάνει τον κώδικα που αντλεί δεδομένα από τα xml αρχεία και τον κώδικα που τα εμφανίζει στην οθόνη. Η Activity Show αντλεί δεδομένα από την προηγούμενη activity με την αυτήν την εντολή `name=getIntent().getExtras().getString("fname")`. Τα δεδομένα αυτά αφορούν το όνομα του xml αρχείου που είναι προς εμφάνιση. Με την εντολή `TextView myXmlContent = (TextView)findViewById(R.id.my_xml)` γίνεται η προετοιμασία και η δήλωση των οπτικών συστατικών της φόρμας, δηλαδή που θα εμφανιστούν τα τελικά δεδομένα. Στην συγκεκριμένη περίπτωση τα δεδομένα θα εμφανιστούν στο `my_xml` αντικείμενο που είναι ένα TextView του Layout Show. Αργότερα βλέπουμε πως γίνεται μια διαχώριση των δεδομένων ανάλογα με τον τύπο τους, εικόνα, τίτλος ή κείμενο. Ανάλογα με το τύπο του καθενός γίνεται και η τοποθέτηση του στην αντίστοιχη θέση του Activity. Ουσιαστικά σε αυτό το αρχείο παίρνουμε από τα xml αρχεία τα tags τους (τίτλος, κείμενο, εικόνα) και τα βάζουμε με την σειρά για να μας εμφανιστεί η Activity.

```

if(eventType == XmlPullParser.START_TAG) //in start of Tag
{
    //stringBuffer.append("\n"+xpp.getName());
    if(xpp.getName().equals("picture"))
    {
        isPicture=true;
    }
    else if(xpp.getName().equals("title"))
    {
        isTitle=true;
    }
    else if(xpp.getName().equals("text"))
    {
        isText=true;
    }
}

```

```

else if(eventType == XmlPullParser.TEXT) //in main text
{
    if(isTitle)
    {
        TextView tx=(TextView) findViewById(R.id.MyTitle);
        tx.setText(xpp.getText());
    }
    else if(isPicture)
    {
        ImageView img=(ImageView) findViewById(getResources().getIdentifier
            ("imageView1", "id", getPackageName()));

        img.setImageDrawable(getResources().getDrawable(getResources().getIdentifier
            (xpp.getText(), "drawable", getPackageName())));
    }
    else if(isText)
    {
        stringBuffer.append("\n"+xpp.getText());
    }
}

```

Εικόνα 5.10: Κώδικας για Show.java

- Εμφάνιση χάρτη (Map.java)

Για να προγραμματίσουμε με τους χάρτες της Google θα πρέπει να φορτωθούν οι βιβλιοθήκες `Google_play_services_lib.java` και `android_support_v4.jar`. Ένα επίσης πολύ σημαντικό στοιχείο που μας χρειάζεται είναι το API Key. Για την δημιουργία του API Key χρειαζόμαστε δύο κλειδιά. Το ένα είναι το όνομα του πακέτου μας και το άλλο βγαίνει από τον υπολογιστή μας όταν τρέχουμε την εφαρμογή μας. Όταν δώσουμε αυτά τα δύο κλειδιά στην Google μας παράγει ένα API Key το οποίο το δηλώνουμε στο αρχείο `AndroidManifest.xml`.

Στο αρχείο `Map.java` ορίζουμε έναν πίνακα με `Markers`, τις κόκκινες κουκίδες που εμφανίζονται επάνω στον χάρτη. Κάθε `Marker` αφορά μόνο ένα αντικείμενο (πχ ξενοδοχείο, εστιατόριο κτλ). Ορίζουμε και έναν ακόμη πίνακα στον οποίο περιέχονται όλα τα αντικείμενα που θα ορίζονται πάνω στον χάρτη, ουσιαστικά τα ονόματα των xml αρχείων μέσα στα οποία περιέχονται και οι γεωγραφικές πληροφορίες (`latitude`, `longitude`). Όταν ο χρήστης κάνει κλικ πάνω σε κάποιον `Marker`, τότε εκτελείται ο παρακάτω κώδικας.

```

myGoogle.setOnInfoWindowClickListener(new OnInfoWindowClickListener() {
    @Override
    public void onInfoWindowClick(Marker arg0) {
        // TODO Auto-generated method stub
        Intent i = new Intent(Map.this, Show.class);
        i.putExtra("fname", arg0.getSnippet().toString());

        startActivity(i); //start new activity
    }
});

```

Εικόνα 5.11: Κώδικας για τους Markers

- Εμφάνιση τρέχουσας τοποθεσίας (Nearby.java)

Όταν ο χρήστης επιλέγει από το βασικό μενού την εμφάνιση τρέχουσας τοποθεσίας, τότε εκτελείται ο κώδικας του αρχείου Nearby.java. Αυτός ο κώδικας είναι σχεδόν ίδιος με τον κώδικα του αρχείου Map.java συν το ότι περιέχει μια ακόμα σημαντική εντολή, `myGoogle.setMyLocationEnabled(true)` η οποία δίνει την δυνατότητα του εντοπισμού της τρέχουσας περιοχής του χρήστη μέσω της τεχνολογίας GPS.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.nearby);

    SupportMapFragment nf=(SupportMapFragment)getSupportFragmentManager().
        findFragmentById(R.id.map3);

    myGoogle=nf.getMap();

    myGoogle.setMyLocationEnabled(true);

    myGoogle.setOnInfoWindowClickListener(new OnInfoWindowClickListener() {
        @Override
        public void onInfoWindowClick(Marker arg0) {
            // TODO Auto-generated method stub
            Intent i = new Intent(NearBy.this, Show.class);
            i.putExtra("fname",arg0.getSnippet().toString());

            startActivity(i); //start new activity
        }
    });
}
```

Εικόνα 5.12: Κώδικας για το αρχείο Nearby.java

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΠΕΡΑΙΤΕΡΩ ΑΝΑΠΤΥΞΗΣ

Στην παρούσα πτυχιακή εργασία μελετήσαμε και παρουσιάσαμε εκτενώς το λειτουργικό σύστημα Android. Διαπιστώσαμε ότι 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 επί πληρωμή εφαρμογές διαθέσιμες για όλους τους χρήστες του Android. Στην συνέχεια παρουσιάσαμε τα απαραίτητα εργαλεία ανάπτυξης και τη δομή μιας εφαρμογής Android. Τέλος αναλύσαμε τον τρόπο ανάπτυξης και την λειτουργία της εφαρμογής μας.

Τώρα πια που έχει υλοποιηθεί πλήρως η εφαρμογή μπορούμε να βγάλουμε κάποια βασικά συμπεράσματα για όλη την διαδικασία υλοποίησης και να προτείνουμε κάποιες ιδέες για περαιτέρω ανάπτυξη της εφαρμογής.

Είδαμε ο σχεδιασμός μιας Android εφαρμογής 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 αντικειμενοστραφείς γλώσσα προγραμματισμού, που χρησιμοποιείται κυρίως στην ανάπτυξη εφαρμογών.

Η εφαρμογή μας παρόλο που είναι μια ολοκληρωμένη εφαρμογή περιήγησης, έχει ακόμα πολλά περιθώρια ανάπτυξης. Μερικές κατευθύνσεις για μελλοντική επέκτασή της είναι:

- Πρώτον να δημιουργηθεί η δυνατότητα οριζόντιου προσανατολισμού της οθόνης, να λειτουργεί δηλαδή η εφαρμογή μας και κατά οριζόντιο τρόπο.
- Δεύτερον υπάρχει ικανότητα να μετατραπεί σε πολυγλωσσική ώστε να εξυπηρετεί περισσότερους χρήστες
- Τρίτον να ενισχυθεί ίσως το μενού ώστε να περιέχει ίσως κάποιες ρυθμίσεις για να μπορεί ο χρήστης να συνδέεται με το προφίλ του σε διάφορα κοινωνικά δίκτυα κ.α, να προστεθεί μια δυνατότητα όπου ο χρήστης να μπορεί να αποθηκεύει κάποια σημεία ενδιαφέροντος που θα ήθελε να επισκεφτεί, να κρατάει δηλαδή κάτι σαν σημειωματάριο. Επίσης θα μπορούσαν ίσως να προστεθούν και κάποιες ακόμα πληροφορίες για την ευρύτερη περιοχή της Ναυπάκτου.
- Τέταρτον μια ακόμη βελτίωση θα ήταν στην εμφάνιση της τρέχουσας τοποθεσίας να υπολογίζεται παράλληλα και η απόσταση και οι διαδρομές από το σημείο που βρισκόμαστε μέχρι το σημείο ενδιαφέροντος που έχουμε επιλέξει να επισκεφτούμε.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Σύγγραμμα: «Η Βίβλος της Java 2» , Walsh, Couch & Steinberg, 2002
- Πληροφορίες για την πρώτη συσκευή σε περιβάλλον Android
<http://didyouknow.org/worlds-first-android-phone/>
- Τα ονόματα των εκδόσεων και η σημασία τους <http://osarena.net/news/key-lime-pie-t-onoma-tou-android-4-2-pou-tha-kikloforisi-to-2013.html>
- Πληροφορίες σχετικά με τις εκδόσεις του Android
<http://www.allaboutandroid.gr/?p=6362>
- Πληροφορίες για την έκδοση Jelly Bean
<http://tech.in.gr/news/article/?aid=1231219982>
- Πληροφορίες για την νέα έκδοση Key Lime Pie
<http://www.whos.gr/android/android-50-lime-pie-erhetai-ton-oktobrio>
- Πληροφορίες για την ασφάλεια στο Android
<http://www.slideshare.net/spyrosel/android-sec-itp12401>
- Πληροφορίες για την ασφάλεια στο Android
<http://www.allaboutandroid.gr/?p=2247>
- Πλεονεκτήματα και μειονεκτήματα στο Android <http://mobilecon.info/advantages-and-disadvantages-android-mobile-phone.html>
- Αρχιτεκτονική του Android <http://androidmag.gr/h-%CE%B1%CF%81%CF%87%CE%B9%CF%84%CE%B5%CE%BA%CF%84%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE-%CF%84%CE%BF%CF%85-android/>
- Βιβλιοθήκες Android <http://ventrix.nsd.c.gr/stuff/AndroidFosscommWorkshop.pdf>