

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



ΤΜΗΜΑ: ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ

## ΘΕΜΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ



### **ΕΦΑΡΜΟΓΗ ΕΥΡΕΣΗΣ WiFi ΔΙΚΤΥΩΝ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ANDROID**

ΦΟΙΤΗΤΡΙΑ:

ΠΟΛΥΜΕΡΗ ΑΙΚΑΤΕΡΙΝΗ Α.Μ: ΜΕ11089

ΕΠΙΒΛΕΠΩΝ:

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ ΕΥΘΥΜΟΓΛΟΥ ΓΕΩΡΓΙΟΣ

Αθήνα 2014

Πανεπιστήμιο Πειραιώς

## Περίληψη

Το αντικείμενο της παρούσας διπλωματικής είναι η μελέτη της διαδικασίας ανάπτυξης εφαρμογών για το λειτουργικό σύστημα Android. Το λειτουργικό σύστημα Android αποτελεί μια τεχνολογία η οποία μέρα με τη μέρα κερδίζει μεγαλύτερο μερίδιο στην αγορά των έξυπνων τηλεφώνων. Τον Νοέμβριο του 2013 το μερίδιο του Android άγγιξε το 81% με πωλήσεις 204,4 εκατομμυρίων συσκευών. Η κυριαρχία της Google είναι αδιαμφισβήτητη αφού τέσσερις στις πέντε συσκευές που πωλούνται «τρέχουν» Android. Το γεγονός αυτό καθιστά την αγορά των εφαρμογών Android μια πολλά υποσχόμενη περιοχή για ανάπτυξη επιχειρηματικής δραστηριότητας.

Η μελέτη ξεκινά με μια γενική επισκόπηση του κόσμου του Android καθώς και των εργαλείων που είναι απαραίτητα για την ανάπτυξη εφαρμογών. Στην συνέχεια γίνεται μια σύντομη ανάλυση της δομής του λειτουργικού συστήματος και των τμημάτων που το αποτελούν. Χρησιμοποιώντας τα παραπάνω εργαλεία, αναπτύχθηκε μια διαδικτυακή εφαρμογή σε περιβάλλον Android, η οποία επιτρέπει στον κάτοχο κινητού τηλεφώνου να εντοπίσει όλα τα Wi-Fi δίκτυα και να παίρνει πληροφορίες σχετικά με:

1. Λίστα με τα διαθέσιμα Wi-Fi δίκτυα όπου για το καθένα θα αναφέρεται το όνομα του δικτύου (SSID), η συχνότητα λειτουργίας και η ισχύς του σήματος (RSSI)
2. Γραφική απεικόνιση των διαθέσιμων δικτύων, όπου θα εμφανίζεται το όνομα κάθε δικτύου και το κανάλι που καταλαμβάνει (άξονας x: ισχύς και άξονας y: συχνότητα μεταφρασμένη σε ένα από τα κανάλια του Wi-Fi)
3. Να προτείνει στον χρήστη το καλύτερο Wi-Fi δίκτυο.

Τέλος, γίνεται περιγραφή της εφαρμογής, αναλύοντας τα δομικά συστατικά της καθώς και τον κώδικα βάσει του οποίου εξάγονται οι παραπάνω πληροφορίες. Συνοδευτικά δίνεται φωτογραφικό υλικό από όλα τα βήματα της εφαρμογής με τα αντίστοιχα αποτελέσματα.

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΚΕΦΑΛΑΙΟ 1: Εισαγωγή στο λειτουργικό σύστημα Android

1.1 Τι είναι το Android .....	6
1.2 Εφαρμογές Android .....	7
1.3 Εκδόσεις Android .....	7
1.4 Αρχιτεκτονική του Android .....	30
1.4.1 Πυρήνας Linux .....	31
1.4.2 Βιβλιοθήκες .....	32
1.4.3 Η εικονική μηχανή Dalvik .....	33
1.4.4 Χρόνος εκτέλεσης εφαρμογής (Android RunTime) .....	34
1.4.5 Πλαίσιο εφαρμογής (Application Framework) .....	34
1.5 Στο εσωτερικό μιας εφαρμογής Android .....	36
1.5.1 Το αρχείο AndroidManifest.xml .....	37
1.5.2 Οι φάκελοι src & res .....	37
1.5.3 Οι υπόλοιποι φάκελοι του project .....	38
1.5.4 Δομικά μέρη μιας εφαρμογής .....	39
1.6 Εργαλεία Ανάπτυξης .....	40
1.6.1 Εγκατάσταση Eclipse 3.5 .....	41
1.6.2 Εγκατάσταση Java SDK .....	41
1.6.3 Εγκατάσταση Android SDK .....	46
1.6.4 Εγκατάσταση AVD .....	48
1.6.5 Εγκατάσταση Android Development Tools (ADT) .....	51
1.6.6 Select Android SDK location .....	53

## ΚΕΦΑΛΑΙΟ 2: Ανάπτυξη εφαρμογών στο Android

2.1 Κύκλος ανάπτυξης της εφαρμογής .....	55
2.1.1 Εγκατάσταση λογισμικού .....	55
2.1.2 Ανάπτυξη πηγαίου κώδικα της εφαρμογής .....	56
2.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση της Εφαρμογής .....	56

2.1.3.1	Android Debug Bridge .....	58
2.1.3.2	Εικονικές συσκευές Android (AVD) .....	60
2.1.3.3	Εργαλεία καταγραφής συμβάντων – LogCat .....	60
2.1.3.4	Dalvik Debug Monitor Server (DDMS) .....	64
2.1.3.5	Application Crash Reporter for Android (ACRA) ....	65
2.1.4	Τελική έκδοση της εφαρμογής .....	66
<b>ΚΕΦΑΛΑΙΟ 3: Αναζήτηση Wi-Fi δικτύων σε πλατφόρμα Android</b>		
3.1	To Project .....	68
3.2	Περιγραφή του project .....	81
3.2.1	Φάκελος src .....	72
3.2.2	WiFiAnthi .....	83
3.2.2.1	Scan Networks .....	84
3.2.2.2	Report Networks .....	85
3.2.2.3	Choose Networks .....	86
3.2.2.4	Plot Networks .....	89
3.2.3	WifiPlotter .....	89
3.2.3.1	Σχεδίαση γραφικής παράστασης .....	90
3.2.4	Φάκελος res (resource) .....	92
3.2.5	Αρχείο AndroidManifest.xml .....	96
Manual	.....	99
Συμπεράσματα	.....	105
Βιβλιογραφία	.....	106

## 1.1 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο Linux, για φορητές συσκευές όπως smartphones και tablets. Αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις φορητές συσκευές. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache Licence, μιας ελεύθερης άδειας λογισμικού.

Τον Ιούλιο το 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες του Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Στην Google, η ομάδα με επικεφαλής τον Andy Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google είχε ήδη συγκεντρώσει μια σειρά από εταίρους hardware και software και επεσήμανε στους παρόχους ότι ήταν ανοιχτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυξε μια Google-branded συσκευή. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.

## 1.2 Εφαρμογές Android

Το Android έχει μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορεί κάποιος να κατεβάσει από το online κατάστημα της Google Play (πρώην Android market) της Google όπως και από άλλα sites. Μέχρι τον Φεβρουάριο του 2012 περισσότερες από 450000 εφαρμογές ήταν διαθέσιμες για Android ενώ εκτιμάται ότι ο αριθμός των downloads από τον Android Market μέχρι τον Δεκέμβριο του 2011 είχε υπερβεί μέχρι και τα 10 δισεκατομμύρια. Το Android είναι η πρώτη σε πωλήσεις παγκοσμίως πλατφόρμα για smartphones καθώς μέχρι το Φεβρουάριο του 2012 μετρούσε περισσότερες από 300 εκατομμύρια συσκευές σε χρήση.

## 1.3 Εκδόσεις Android

Η πρώτη έκδοση του λειτουργικού αυτού συστήματος ήταν η Android 1.0. Έκτοτε δημιουργήθηκαν αρκετές εκδόσεις με σκοπό την εξέλιξή του. Κάθε σημαντική έκδοση του Android έχει ένα χαρακτηριστικό όνομα επιδορπίου. Παρακάτω θα αναφερθούμε επιγραμματικά στις διάφορες εκδόσεις του Android<sup>1</sup>.

### Android 1.5 - Cupcake



Η έκδοση Android Cupcake ήταν η πρώτη σημαντική αναθεώρηση της πρώτης έκδοσης του Android. Το Android 1.5 SDK κυκλοφόρησε τον Απρίλιο του 2009 και έφερε αλλαγές που αφορούσαν τις διεπιφάνειες χρήστη. Υπήρξαν όμως και άλλες αλλαγές. Το cupcake έφερε

---

<sup>1</sup> Paul Deiter, Harvey Deitel, Abbey Deitel, Michael Morgano, 2011, Android for Programmers, An App-Driven Approach, 1st Edition , Prentice Hall, USA

χαρακτηριστικά όπως η βελτιωμένη υποστήριξη της τεχνολογίας Bluetooth, των λειτουργιών της βιντεοκάμερας και δημοφιλών υπηρεσιών βίντεο και εικόνας όπως το YouTube και το Picasa. Το Android 1.5 συνοδεύτηκε από την έκρηξη της σύγχρονης κινητής τηλεφωνίας και τη δημιουργία κινητών συσκευών όπως το HTC Hero/Eris, το Samsung Moment και το Motorola Cliq. Τα χαρακτηριστικά του λειτουργικού συστήματος φαίνονται παρακάτω.

## ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ

## ΠΕΡΙΓΡΑΦΗ

Πληκτρολόγιο	Υποστήριξη πληκτρολογίων από άλλους κατασκευαστές με δυνατότητα πρόβλεψης λέξεων κατά την πληκτρολόγηση και δημιουργία λεξικού με λέξεις που ορίζει ο χρήστης
Widgets	Εισαγωγή των Widgets, μικρογραφιών μιας εφαρμογής που μπορούν να ενσωματωθούν στην επιφάνεια εργασίας και να ανανεώνονται σε τακτά χρονικά διαστήματα
Video Codecs	Δυνατότητα εγγραφής και αναπαραγωγής βίντεο στα φορμά MPEG4 και 3GP
Auto-pairing & Stereo over Bluetooth	Αυτόματη ανίχνευση και σύνδεση με Bluetooth ακουστικά ή ηχεία και δυνατότητα αναπαραγωγής στέρεο ήχου
Copy & Paste in Browser	Εισαγωγή δυνατότητας copy & paste στον browser του Android
Εικόνες στις αγαπημένες επαφές	Δυνατότητα επιλογής εικόνων του χρήστη για εμφάνιση στις επαφές που βρίσκονται στα αγαπημένα



Λίστα κλήσεων

Αναγραφή ημερομηνίας και ώρας στις καταχωρήσεις της λίστας και δυνατότητα επίσκεψης της επαφής στην οποία αναφέρεται η συγκεκριμένη καταχώρηση

Animated Transitions

Δυνατότητα μετάβασης από τη μία οθόνη της εφαρμογής στην άλλη με χρήση εφέ (fade in, fade out κλπ)

Auto –Rotation

Αυτόματη προσαρμογή της οθόνης ανάλογα με την κλίση της συσκευής

Animation Εκκίνησης

Εισαγωγή του animation κατά την εκκίνηση του λειτουργικού

Μεταφόρτωση στο YouTube

Δυνατότητα μεταφόρτωσης βίντεο κατευθείαν στο YouTube

Μεταφόρτωση στο Picasa

Δυνατότητα μεταφόρτωσης εικόνων κατευθείαν στην υπηρεσία Picasa

### Android 1.6 – Donut



Η έκδοση Donut κυκλοφόρησε το Σεπτέμβριο του 2009 στηρίχθηκε πάνω στα χαρακτηριστικά του Android 1.5 και επέκτεινε κάποια από αυτά. Παρόλο που δεν έγιναν σημαντικές αλλαγές οι οποίες ήταν ορατές στο χρήστη, οι

αλλαγές στη βάση του λειτουργικού συστήματος προετοίμασαν το έδαφος για μελλοντικές εντυπωσιακές αλλαγές. Για τον τελικό χρήστη, οι δύο μεγαλύτερες αλλαγές αφορούσαν τη δυνατότητα αναζήτησης καθώς και την υπηρεσία εύρεσης και εγκατάστασης εφαρμογών Android (Android Market).

Επίσης το Android 1.6 έφερε υποστήριξη για οθόνες αφής ψηλότερης ανάλυσης από αυτές που χρησιμοποιούνταν μέχρι τότε, βελτίωνε την κάμερα και παρείχε την δυνατότητα σύνδεσης των κινητών τηλεφώνων μέσω δορυφόρου. Χωρίς αυτήν την έκδοση δεν θα είχαν δημιουργηθεί συσκευές όπως το Motorola Droid ή HTC Eno 4G. Τα χαρακτηριστικά αυτού του λειτουργικού συστήματος φαίνονται παρακάτω.

<b>ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ</b>	<b>ΠΕΡΙΓΡΑΦΗ</b>
Δυνατότητα Αναζήτησης	Βελτίωση της φωνητικής και μη αναζήτησης ώστε να περιλαμβάνει το ιστορικό των επισκέψεων στο διαδίκτυο, τις επαφές και το διαδίκτυο (με χρήση του Google.com)
Περιεχόμενα Αναζήτησης	Παρέχεται πλέον η δυνατότητα στους προγραμματιστές να περιλαμβάνουν το περιεχόμενο των εφαρμογών τους στην δυνατότητα αναζήτησης του Android
Πολυγλωσσική Σύνθεση Ομιλίας	Δυνατότητα σύνθεσης ομιλίας σε διαφορετικές γλώσσες ώστε κάθε συσκευή Android να μπορεί να διαβάσει μια πρόταση ή ένα κείμενο
Android Market	Ευκολότερη αναζήτηση εφαρμογών στην αγορά της Google και προβολή στιγμιότυπων από την εφαρμογή για καλύτερη επιλογή από τους χρήστες
Gallery, Camera, Camcorder	Βελτίωση των επιδόσεων των εφαρμογών πολυμέσων που είχαν πρόσβαση σε περιφερειακά όπως η κάμερα και οι κάρτες μνήμης της συσκευής
Gallery	Δυνατότητα επιλογής περισσότερων

Τεχνολογίες συνδεσιμότητας	της μίας φωτογραφιών για διαγραφή Αναβάθμιση των τεχνολογιών που χρησιμοποιούνταν για CDMA, EVDO, 802.1x, και VPN
Αναλύσεις οθόνης	Υποστήριξη για οθόνες WVGA

### Android 2.0 / 2.0.1 / 2.1 – Éclair



Η έκδοση Éclair ήταν ένα αρκετά σημαντικό βήμα για την εξέλιξη του λειτουργικού συστήματος σε σχέση με τις προηγούμενες εκδόσεις. Δημιουργήθηκε στα τέλη του 2009 και η πρώτη συσκευή που το ενσωμάτωνε

ήταν η Motorola Droid βελτιώνοντας χαρακτηριστικά όπως η πλοήγηση, την υπηρεσία Google Maps και τις διεπιφάνειες χρήστη. Επίσης δημιουργήθηκε η υπηρεσία Google Maps Navigation η οποία έδινε την δυνατότητα στις κινητές συσκευές να συγκριθούν με συσκευές πλοήγησης GPS.

Η έκδοση 2.0 γρήγορα αντικαταστάθηκε από την 2.0.1 η οποία βγήκε λίγους μήνες αργότερα και σκοπός της ήταν να διορθώσει προβλήματα της έκδοσης 2.0. Στη συνέχεια δημοσιοποιήθηκε η έκδοση 2.1 και πιο συγκεκριμένα τον Ιούνιο του 2010 και προσέφερε καλύτερες διεπιφάνειες χρήστη και καλύτερα γραφικά. Η έκδοση αυτή άργησε να εγκατασταθεί στις κινητές συσκευές λόγω των προσαρμογών που έπρεπε να γίνουν σ' αυτές. Η έκδοση αυτή έχει τα παρακάτω χαρακτηριστικά.

## **ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**

## **ΠΕΡΙΓΡΑΦΗ**

Γενικευμένος Μηχανισμός  
Συγχρονισμού

Πλέον ο χρήστης μπορεί να εισάγει  
πολλαπλούς λογαριασμούς από  
διάφορες υπηρεσίες και οι αντίστοιχες

	εφαρμογές μπορούν να χρησιμοποιούν τα στοιχεία αυτά για να συγχρονίζουν το περιεχόμενό τους
Exchange Mail Server	Υποστήριξη του Microsoft Exchange Mail Server για συγχρονισμό email
Αναβάθμιση Bluetooth	Υποστήριξη του πρωτοκόλλου Bluetooth 2.0
People App	Εισαγωγή της δυνατότητας να επιλέξει ο χρήστης την κλήση, την αποστολή email ή SMS σε μια επαφή απλά πατώντας την εικόνα της
Αναζήτηση στα SMS & MMS	Δυνατότητα αναζήτησης στα μηνύματα του χρήστη και αυτόματη διαγραφή μηνυμάτων που χρονολογικά περνούν κάποιο καθορισμένο όριο
Camera	Υποστήριξη flash, ψηφιακό ζουμ, scene mode λειτουργίας, ρύθμιση ισορροπίας λευκού, εισαγωγή εφέ χρώματος και δυνατότητα macro focus
Εικονικό πληκτρολόγιο	Βελτίωση της ταχύτητας δακτυλογράφησης στο πληκτρολόγιο με χρήση έξυπνου υλικού που μαθαίνει από τις πληκτρολογήσεις του χρήστη
Android Browser	Βελτίωση της διεπαφής του περιηγητή και εισαγωγή υποστήριξης HTML5 και double-tap ζουμ
Calendar	Βελτίωση της εμφάνισης της ατζέντας και δυνατότητα πρόσκλησης επιπλέον ατόμων σε κάποιο γεγονός
Βελτιώσεις Βασιζόμενες στο	Βελτιώσεις του λειτουργικού για

Υλικό	επίτευξη καλύτερων επιδόσεων και ανανέωση της διεπαφής χρήστη
Αναλύσεις οθόνης	Υποστήριξη περισσότερων μεγεθών και αναλύσεων οθονών με καλύτερο συντελεστή αντίθεσης
Google Maps	Αναβάθμιση του Google Maps στην έκδοση 3.1.2
MotionEvent class (SDK)	Η κλάση MotionEvent αναβαθμίστηκε ώστε να αναγνωρίζει και πολλαπλά αγγίγματα (multitouch)
Live Wallpapers	Κίνηση του φόντου της επιφάνειας εργασίας καθώς ο χρήστης αλλάζει οθόνες

### Android 2.2 – Froyo



Το Android 2.2 ανακοινώθηκε το Μάιο του 2010 από την Google στο Σαν Φρανσίσκο. Η μεγαλύτερη αλλαγή που εισήγαγε το Froyo ήταν η σημαντική αύξηση της επεξεργαστικής ισχύς της κινητής συσκευής που το είχε εγκατεστημένο. Επίσης, παρείχε υποστήριξη για το Adobe Flash και έδινε τη δυνατότητα στις κινητές συσκευές να παρέχουν σύνδεση στο Διαδίκτυο σε όποια συσκευή το επιθυμούσαν (Wi-Fi spot). Παρακάτω δίνονται τα χαρακτηριστικά του.

### **ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**

### **ΠΕΡΙΓΡΑΦΗ**

Βελτιωμένη διαχείριση μνήμης  
Και καλύτερες επιδόσεις

Ο διερμηνευτής Dalvik έγινε πέντε φορές πιο γρήγορος συγκρινόμενος με τον διερμηνευτή της έκδοσης 2.1

	<p>του Android, η μηχανή V8 του Chrome βοηθάει τον browser να φορτώνει γρηγορότερα σελίδες με JavaScript περιεχόμενο και αλλαγές στην διαχείριση μνήμης στο επίπεδο του πυρήνα αύξησαν ακόμη περισσότερο τις επιδόσεις</p>
<p>Προστασία Συσκευής με Κωδικό</p>	<p>Οι χρήστες μπορούν να κλειδώνουν τη συσκευή με κωδικό για να αποτρέψουν την πρόσβαση τρίτων σε προσωπικά δεδομένα</p>
<p>Remote Wipe</p>	<p>Δυνατότητα επαναφοράς της συσκευής στην εργοστασιακή της κατάσταση και διαγραφή όλων των προσωπικών δεδομένων σε περίπτωση απώλειας ή κλοπής της</p>
<p>Framework πολυμέσων</p>	<p>Το Stagefright Media Framework παρέχει την δυνατότητα αναπαραγωγής βίντεο καθώς και streaming βίντεο μέσω του HTTP. Στην δεύτερη περίπτωση η αναπαραγωγή μπορεί να αρχίσει και πριν την ολοκλήρωση της λήψης του αρχείου με το πολυμεσικό περιεχόμενο</p>
<p>Tethering &amp; Wi-Fi Hotspot</p>	<p>Με τις δυνατότητες αυτές ο χρήστης μπορεί να χρησιμοποιήσει την κινητή συσκευή του σαν ένα σημείο σύνδεσης στο διαδίκτυο συνδεδεμένος σε αυτήν με ένα απλό καλώδιο USB</p>
<p>Εφαρμογές σε εξωτερικό χώρο Αποθήκευσης</p>	<p>Πλέον οι εφαρμογές μπορούν να εγκατασταθούν και σε άλλες μνήμες της συσκευής πέραν της εσωτερικής</p>

	της μνήμης
Camera & Camcorder	Εισαγωγή του Camera Preview API που παρέχει 20fps ταχύτητα ανανέωσης, αλλαγή προσανατολισμού ανάλογα με τον προσανατολισμό της συσκευής, χειριστήρια για ζουμ, πληροφορίες έκθεσης καθώς και ένα βοηθητικό εργαλείο για δημιουργία μικρογραφιών των ληφθέντων φωτογραφιών. Επιπλέον υπάρχουν κλάσεις που μπορούν να χρησιμοποιηθούν για τον έλεγχο του υπάρχοντος βιντεοληπτικού υλικού προς χρήση
Backup δεδομένων	Δυνατότητα αποθήκευσης δεδομένων στο cloud και ανάκτησή τους από τους χρήστες ανεξάρτητα από την συσκευή στην οποία βρίσκονται
Γραφικά Framework πολυμέσων	Πρόσβαση στο Open GL ES 2.0 API API για την αναζήτηση αρχείων πολυμέσων, την αναπαραγωγή τους καθώς και την αυτόματη διακοπή και συνέχιση της αναπαραγωγής (σε περίπτωση διακοπής της εφαρμογής από, για παράδειγμα, μια εισερχόμενη κλήση
UI Framework	Ύπαρξη διαφόρων τρόπων λειτουργίας, όπως χρήση στο αυτοκίνητο, στο γραφείο κλπ. Πιο αξιόπιστη λειτουργία στα αγγίγματα και συγκεκριμένα στα πολλαπλά αγγίγματα

## Android 2.3 – Gingerbread



Το Android 2.3 με την κωδική ονομασία Gingerbread δημοσιεύθηκε τον Δεκέμβριο του 2010. Έφερε βελτιώσεις στις διεπιφάνειες χρήστη δίνοντας καλύτερη αίσθηση στο χειρισμό τους. Παρόλο αυτά το Android δεν άλλαξε ουσιαστικά από την προηγούμενη έκδοσή του.

Το Gingerbread παρέχει υποστήριξη για NFC (Near Field Communication) τεχνολογία. Πρόκειται για μια τεχνολογία ανταλλαγής πληροφοριών ανάμεσα σε κινητές συσκευές φέρνοντας τη μια δίπλα στην άλλη. Επίσης παρέχει και τεχνολογία SIP δηλαδή τηλεφωνία μέσω του διαδικτύου. Παρακάτω δίνονται τα χαρακτηριστικά.

### **ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**

### **ΠΕΡΙΓΡΑΦΗ**

Διαχείριση ενέργειας

Το Android έχει το δικαίωμα να τερματίσει οποιαδήποτε εφαρμογή τρέχει στο παρασκήνιο και καταναλώνει πολύ ενέργεια ή τρέχει στο προσκήνιο για περισσότερο χρόνο του κανονικού (συνήθως πέντε δευτερόλεπτα) προκειμένου να διασφαλίσει την μέγιστη διάρκεια λειτουργίας

Διαχείριση εφαρμογών

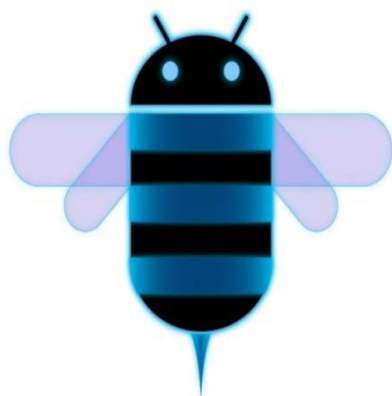
Η συντόμευση διαχείρισης εφαρμογών στο μενού επιλογών της επιφάνειας εργασίας δίνει τη δυνατότητα στο χρήστη να δει ποιες εφαρμογές εκτελούνται. Για κάθε εφαρμογή υπάρχουν διαθέσιμες πληροφορίες που αφορούν το χώρο που χρησιμοποιεί στη μνήμη ή σε



	<p>κάποιο αποθηκευτικό μέσο, τους περιορισμούς στους οποίους υπόκειται η εφαρμογή κα. Ο χρήστης έχει την δυνατότητα να τερματίσει την εφαρμογή</p>
<p>Near Field Communication (NFC)</p>	<p>Το πρωτόκολλο NFC είναι ένα πρότυπο ασύρματης επικοινωνίας μεταξύ δύο συσκευών ή μιας συσκευής και μιας καρτέλας. Το Android δίνει την δυνατότητα στη συσκευή να λειτουργήσει ως αναγνώστης πληροφορίας ή ως εγγραφέας πληροφορίας</p>
<p>Βελτιωμένες λειτουργίες για Αντιγραφή και επικόλληση</p>	<p>Με ένα απλό άγγιγμα μιας λέξης αυτή επιλέγεται. Η επιλογή μπορεί να τροποποιηθεί, ώστε να περιλαμβάνει και γειτονικές λέξεις, σέρνοντας τους δείκτες που θα εμφανιστούν, στα επιθυμητά σημεία. Αγγίζοντας την επιλεγμένη περιοχή αυτή αντιγράφεται και είναι έτοιμοι για να επικολληθεί σε κάποιο άλλο σημείο πιέζοντας παρατεταμένα και επιλέγοντας επικόλληση από το εμφανιζόμενο μενού</p>
<p>Κλήσεις με Camera</p>	<p>Οι εφαρμογές έχουν πρόσβαση σε όλες τις κάμερες της συσκευής και έτσι μπορούν να τις χρησιμοποιούν για κλήσεις μέσω του διαδικτύου. Το Android υποστηρίζει το πρωτόκολλο SIP και έτσι αν οι χρήστες διαθέτουν λογαριασμούς σε κάποιον πάροχο αντίστοιχης υπηρεσίας μπορούν να</p>

	χρησιμοποιήσουν τις κινητές συσκευές τους σαν SIP τηλέφωνα
Εφαρμογή Downloads	Με τη εφαρμογή downloads οι χρήστες μπορούν να δουν όλα τα αρχεία που έχουν λάβει στη συσκευή τους από email, από τον περιηγητή κλπ
Τηλεφωνία μέσω Διαδικτύου	Η υποστήριξη μέσω πρωτοκόλλου SIP δίνει την δυνατότητα ενσωμάτωσης στις εφαρμογές δυνατοτήτων τηλεφωνίας
Near Field Communication (NFC) API	Δίνει τη δυνατότητα ανάγνωσης και εγγραφής δεδομένων με χρήση του πρωτοκόλλου αυτού που επιτρέπει την ανταλλαγή πληροφοριών μεταξύ συσκευών. Η χρήση του χαρακτηριστικού αυτού εξαρτάται από την συσκευή
API για εφέ ήχου	Δυνατότητα ισοστάθμισης του αναπαραγόμενου ήχου, ενίσχυσης των μπάσων, δημιουργία ηχώ κλπ
Υποστήριξη νέων φορμά ήχου	Υποστήριξη για αρχεία τύπου AAC και AMR-WB για ηχογράφηση ήχου υψηλής πιστότητας
Camera API	Δυνατότητα επιλογής κάμερας (σε περίπτωση που η συσκευή διαθέτει περισσότερες από μια) και επιλογή της επιθυμητής κάμερας για χρήση

## Android 3.X – Honeycomb



Το Android 3.0 κυκλοφόρησε το Φεβρουάριο του 2011. Ήταν η πρώτη έκδοση του λειτουργικού συστήματος που δημιουργήθηκε αποκλειστικά για κινητές συσκευές τύπου tablet και επέφερε σημαντικές αλλαγές στις διεπιφάνειες του χρήστη. Επίσης αναβαθμίστηκαν εφαρμογές που παρέχει η Google όπως το Gmail και το Google Talk. Σημαντική ήταν και η βελτίωση του υλικού των συσκευών μέσω των χαρακτηριστικών του Honeycomb.

Αξίζει να αναφερθούμε στο γεγονός ότι η Google εφάρμοσε έναν διαφορετικό τρόπο στην διανομή του κώδικα του λειτουργικού της στους κατασκευαστές κινητών συσκευών. Προκειμένου οι κατασκευαστές να πάρουν και να χρησιμοποιήσουν το λειτουργικό έπρεπε η Google να εγκρίνει τις επιλογές του υλικού που θα έκαναν για τις κινητές συσκευές τους. Αυτό είχε ως αποτέλεσμα να περιοριστεί η ελευθερία του λογισμικού το οποίο θεωρείται ανοιχτό και χρησιμοποιούμενο από όλους.

Βελτιώσεις στο Android 3.0, Android 3.1 και Android 3.2 ανακοινώθηκαν το Μάιο του 2011.

### **ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**

### **ΠΕΡΙΓΡΑΦΗ**

Διεπαφή χρήστη

Ελκυστική 3D-looking διεπαφή χρήστη πλήρως ρυθμιζόμενη ως προς το περιεχόμενό της

Επανασχεδιασμένο πληκτρολόγιο

Σχεδιασμός εκ νέο του πληκτρολογίου λόγω της μεγαλύτερης διαθέσιμης επιφάνειας στα tablets

Μπάρα συστήματος

Γρήγορη επιλογή συχνά χρησιμοποιούμενων λειτουργιών διαθέσιμες πάντα στο κάτω μέρος της

	οθόνης
Μπάρα εφαρμογής (Action bar)	Μπάρα με επιλογές ειδικά για τρέχουσα εφαρμογή, που βρίσκεται στο πάνω μέρος της οθόνης
Multitasking	Δυνατότητα επισκόπησης των ταυτόχρονα εκτελούμενων εφαρμογών και μετάβαση σε οποιαδήποτε από αυτές ο χρήστης επιθυμεί
Επιλογές Συνδεσιμότητας	Δυνατότητα σύνδεσης του tablet με εξωτερικές συσκευές όπως πληκτρολόγιο με χρήση είτε USB, είτε Bluetooth
Photo Transfer Protocol (PTP) & Media Transfer Protocol (MTP)	Με χρήση των πρωτοκόλλων αυτών γίνεται δυνατή η μεταφορά πολυμέσων από συσκευή σε συσκευή με μεγαλύτερη ευκολία χρησιμοποιώντας εφαρμογές που τα υποστηρίζουν
Bluetooth Tethering	Δυνατότητα χρήσης της σύνδεσης της κινητής συσκευής από κάποιον προσωπικό υπολογιστή που συνδέεται σε αυτή με Bluetooth
Περιηγητής	Εισαγωγή tabs, όπως και στους προσωπικούς υπολογιστές αφού πλέον υπάρχει ο διαθέσιμος χώρος. Παρέχει υποστήριξη JavaScript και δυνατότητες συγχρονισμού με τον λογαριασμό Google του χρήστη
Camera	Επανασχεδιασμένη διεπαφή χρήστη για τον έλεγχο της κάμερας με περισσότερες επιλογές άμεσα διαθέσιμες και την δυνατότητα λήψης

	time-lapse βίντεο
Επαφές	Λόγω και πάλι του μεγαλύτερου διαθέσιμου μεγέθους η επισκόπηση των επαφών και η αναζήτησή τους είναι ευκολότερη από ποτέ χάρη στη διεπαφή δύο στηλών (αριστερά οι επαφές και δεξιά οι τρέχουσες πληροφορίες της επιλεγμένης)
Συμβατότητα με παλαιότερες Εκδόσεις	Το Android 3.x είναι πλήρως συμβατό με εφαρμογές που έχουν γραφτεί για παλαιότερες εκδόσεις του Android
Holographic UI	Με μια απλή καταχώρηση στο manifest αρχείο της εφαρμογής αυτή χρησιμοποιεί το theme του Android 3.0 και απολαμβάνει του νέου βελτιωμένου look & feel
Πρόσθετες περιγραφές Διεπαφών	Για την υποστήριξη οθονών μεγαλύτερου μεγέθους δίνεται η δυνατότητα προσθήκης εναλλακτικών περιγραφών διεπαφών. Με τον τρόπο αυτό μπορούν κάθε φορά να χρησιμοποιούνται αυτές που ταιριάζουν με το μέγεθος της οθόνης της συσκευής στην οποία εκτελούνται οι εφαρμογές
Activity Fragments	Δυνατότητα χωρισμού μιας οθόνης της εφαρμογής σε μικρότερα κομμάτια και χρήση τους σε διάφορους συνδυασμούς. Η Google διαμορφώνει το API αυτό προκειμένου να είναι συμβατό από το Android 1.6 και μετά

Βελτιωμένα Widgets	Δυνατότητα αναγνώρισης κινήσεων ακόμη και στα widgets ώστε ο χρήστης να μπορεί να πλοηγείται στο περιεχόμενό τους όπως θα έκανε αν αλληλεπιδρούσε μια κανονική εφαρμογή
Action bar	Κάθε εφαρμογή έχει τη δική της μπάρα εφαρμογών από όπου μπορεί να παρέχει στον χρήστη διάφορες επιλογές κατά την εκτέλεσή της
Βελτιώσεις για ανάπτυξη παιχνιδιών	Εισαγωγή παράλληλου συλλέκτη απορριμμάτων και αναβάθμιση των οδηγών γραφικών. Χρήση περισσότερων διαθέσιμων αισθητήρων στις συσκευές. Υποστήριξη του προτύπου OpenGL ES για την υποστήριξη ήχου υποβοηθούμενου από το υλικό. Υποστήριξη της βιβλιοθήκης Khronos EGL για την διαχείριση των γραφικών που υποβοηθούνται από το υλικό
Clipboard	Δυνατότητα αντιγραφής και επικόλλησης δεδομένων μεταξύ διαφορετικών εφαρμογών
Drag & Drop	Εισαγωγή του DragEvent framework που δίνει την δυνατότητα στις εφαρμογές να παρέχουν drag & drop χαρακτηριστικά
Επεξεργαστές πολλαπλών πυρήνων	Υποστήριξη επεξεργαστών με περισσότερους του ενός επεξεργαστικούς πυρήνες
RenderScript 3D	Δημιουργία τρισδιάστατων γραφικών για εφαρμογές, widgets με

Δισδιάστατα γραφικά	υποβοήθηση της GPU Χρήση του προτύπου OpenGL για αύξηση των επιδόσεων σε συχνές δισδιάστατες διαδικασίες απεικόνισης που υποβοηθούνται από το υλικό
Digital Rights Management API	Δυνατότητα προστασίας του περιεχομένου των εφαρμογών με χρήση DRM API
Πολιτικές προστασίας της συσκευής	Πλέον οι κωδικοί προστασίας μπορεί να έχουν ημερομηνία λήξεως και να απαιτούν την αλλαγή τους μετά την παρέλευσή τους. Η συνεχής αλλαγή του κωδικού κλειδώματος μιας συσκευής συμβάλει στην αύξηση της ασφάλειας έναντι πρόσβασης τρίτων σε προσωπικά δεδομένα

### Android 4.0 – Ice Cream Sandwich



Η συνέχεια στις εκδόσεις του λειτουργικού συστήματος Android ανακοινώθηκε από την Google τον Μάιο του 2011 και κυκλοφόρησε το Δεκέμβριο του ίδιου χρόνου. Το Ice Cream Sandwich ή αλλιώς Android 4.0 έφερε τα χαρακτηριστικά του Android 3.0 που όπως προαναφέραμε ήταν αποκλειστικά για tablets, στα smartphones.

Η πρώτη συσκευή που χρησιμοποίησε την έκδοση 4.0 ήταν το Samsung Galaxy Nexus. Παρόλο που κυκλοφόρησε σχετικά πρόσφατα ήδη υπάρχουν νέες βελτιωμένες εκδόσεις.

### Android 4.1/4.2/4.3 – Jelly Bean



Το Jelly Bean είναι η κυρίαρχη έκδοση του Android αυτή τη στιγμή αφήνοντας πίσω του το Ice Cream Sandwich και το Gingerbread και φτάνοντας κοντά στο να βρίσκεται εγκατεστημένο σχεδόν στις μισές συσκευές με λειτουργικό σύστημα της Google.

Η έκδοση του Android 4.1 ανακοινώθηκε από την Google στις 27 Ιουνίου του 2012. Η έκδοση αυτή είναι βασισμένη στον πυρήνα Linux 3.1.10 και κύριος σκοπός ήταν η βελτίωση της διεπαφής του χρήστη. Το νέο λειτουργικό, εκτός από πιο ομαλό στη λειτουργία του, είναι ταχύτερο με αμεσότερη απόκριση στις εντολές του χρήστη. Ένα άλλο στοιχείο είναι η ενισχυμένη προσιτότητα, η οποία εστιάζει που τοποθετεί ο χρήστης το χέρι του καθώς εκτελεί εντολές στην οθόνη αφής. Αυξημένη είναι η υποστήριξη πολλών γλωσσών, ενώ οι developers μπορούν να χρησιμοποιούν δύο κατευθύνσεις γραφής, έτσι ώστε να μην υπάρχουν περιορισμοί στις γλώσσες που μπορούν να υποστηρίξουν οι εφαρμογές. Οι χρήστες μπορούν να χρησιμοποιούν εναλλακτικά πληκτρολόγια ή ακόμη να δημιουργούν δικές τους παραλλαγές.

## **ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**

Νέο user interface

## **ΠΕΡΙΓΡΑΦΗ**

Υπάρχουν νέοι τρόποι να εμφανίζεται το User Interface, με μεγαλύτερες ειδοποιήσεις, υποστήριξη περισσότερων μορφών περιεχομένου στο Notifications bar. Μερικά από τα βασικά στοιχεία είναι:



- App widgets με ικανότητα αλλαγής μεγέθους
- Easy animations for Activity launch
- Transitions to Lights Out and Full Screen Modes
- Νέες μορφές απεικόνισης
- Προεπισκόπηση Live wallpaper
- Φωτογραφίες επαφών μεγαλύτερης ανάλυσης

Οι νέες μορφές εισαγωγής κειμένου και οι δυνατότητες που προσφέρουν

- Γρήγορη εύρεση συσκευών που προστίθενται ή αφαιρούνται
- Εντοπισμός των δυνατοτήτων της κάθε συσκευής
- Χρήση δόνησης για τις συσκευές εισόδου

Animation και γραφικά

- Vsync for apps
- Νέα κινούμενα εικονίδια και νέοι τρόποι μετάβασης

Νέες μορφές συνδεσιμότητας

- Android Beam, βασισμένο στο NFC
- Εύρεση δικτύου Wi-fi, με υποστήριξη multicast DNS-based υπηρεσιών
- Υπηρεσία εύρεσης υπηρεσιών Wifi-Direct
- Διαχείριση εύρους δικτύου

Νέες δυνατότητες πολυμέσων

- Πρόσβαση σε Media codecs
- Έξοδος USB Audio
- Έναρξη ηχογράφησης Audio
- Πολυκάναλος ήχος και υποστήριξη κωδικοποίησης /αποκωδικοποίησης ήχου ACC 5.1

	<ul style="list-style-type: none"> <li>• Προεπεξεργασία ήχου</li> <li>• Συνεχής αναπαραγωγή χωρίς παύσεις</li> <li>• Media Router</li> </ul>
Renderscript Computation	Με τη δυνατότητα αυτή οι developers έχουν μεγαλύτερη ευελιξία στα scripts, ενώ μπορούν να εφαρμόσουν ταχύτερη vector επεξεργασία και άλλα στοιχεία που δεν ήταν εφικτά με το IEEE754-2008 standard.
Android Browser και WebView	<ul style="list-style-type: none"> <li>• Καλύτερη εμπειρία HTML5 video, με touch-to-play/pause και ομαλή μετάβαση σε full screen mode.</li> <li>• Μεγαλύτερες ταχύτητες rendering και μειωμένη χρήση μνήμης.</li> <li>• Καλύτερη απόδοση στο HTML5/CSS3/Canvas animation</li> <li>• Βελτιωμένη εισαγωγή κειμένου</li> <li>• Ανανεωμένη JavaScript Engine</li> <li>• Υποστήριξη HTML5 Media Capture</li> </ul>
Google APIs και υπηρεσίες	<ul style="list-style-type: none"> <li>• Google Cloud Messaging for Android</li> <li>• Κρυπτογράφηση εφαρμογών</li> <li>• Έξυπνα App Updates</li> <li>• Google Play services, όπως η πιστοποίηση και η ενσωμάτωση του Google+ στις εφαρμογές</li> </ul>

Εκτός από την έκδοση Android 4.1 Jelly Bean, η Google ανακοίνωσε τη νέα έκδοση Android 4.2 Jelly Bean στις 29 Οκτωβρίου 2012, βασισμένη στο Linux Kernel 3.4.0 μαζί με τα νέα Nexus 4 και Nexus 10, τα οποία κυκλοφόρησαν

στις 13 Νοεμβρίου 2012. Όλα τα νέα του χαρακτηριστικά συγκεντρώνονται παρακάτω:

Εκτός από το Photo Spheres που είναι μια νέα λειτουργία για τη λήψη φωτογραφιών σε 360 μοίρες έτσι ώστε να δημιουργείται μια ενιαία φωτογραφία τύπου Google Street View, στο αναβαθμισμένο λειτουργικό η πληκτρολόγηση είναι σαφώς πιο εύκολη με το Swype, ενώ υπάρχει η υποστήριξη για πολλούς χρήστες σε κάθε συσκευή, ο καθένας από τους οποίους μπορεί να έχει το δικό του προφίλ.

Το νέο λειτουργικό υποστηρίζει επίσης το wireless display, το οποίο θα επιτρέπει το διαμοιρασμό της εικόνας της συσκευής με μια HDTV, η οποία θα μπορεί να συνδεθεί με έναν ειδικό αντάπτορα και να μετατραπεί σε δεύτερη ασύρματα συνδεδεμένη οθόνη.

Το Daydream από τη μεριά του είναι ένα χαρακτηριστικό που κάνει το Android να προβάλλει όμορφες εικόνες ή άλλες χρήσιμες πληροφορίες όταν είναι σε κατάσταση αναμονής. Από την άλλη μεριά οι notifications έχουν ανανεωθεί και είναι όλες σε ένα σημείο συγκεντρωμένες, επιτρέποντας την άμεση εκτέλεση εργασιών από εκεί.

Αναβάθμιση υπάρχει τόσο και για το Google Now, όσο και για το Google Search, με το πρώτο να έχει γίνει πιο έξυπνο να ανταποκρίνεται αυτόματα στις συνήθειες και ανάγκες των χρηστών, ενώ το δεύτερο έχει ενσωματώσει το Google Knowledge Graph ώστε ο χρήστης να μπορεί να παίρνει όλο τον πλούτο πληροφοριών που ζητάει ανά πάσα στιγμή.

Στα ενδιαφέροντα χαρακτηριστικά πρέπει να προστεθεί ο εύκολος διαμοιρασμός φωτογραφιών και video με το Android Beam, η εξελιγμένη χρήση των widgets και τα εργαλεία μεγέθυνσης των όσων βλέπει ο χρήστης στην οθόνη.

Η Google ανακοίνωσε και την Τρίτη έκδοση Android 4.3 Jelly Bean στις 24 Ιουλίου του 2013 κατά τη διάρκεια μιας εκδήλωσης στο Σαν Φρανσίσκο. Από τα κύρια χαρακτηριστικά του Android 4.3 Jelly Bean ξεχωρίζουν:

- **Πολλαπλά προφίλ χρηστών:** Από τα χαρακτηριστικά που ζητούσαν οι χρήστες tablet, το Android πλέον δίνει τη δυνατότητα δημιουργίας πολλαπλών προφίλ, με αντίστοιχους περιορισμούς που θα μπορούν να ρυθμιστούν ανάλογα με το προφίλ. Για παράδειγμα θα μπορούν οι χρήστες να δίνουν πρόσβαση σε συγκεκριμένες εφαρμογές και υπηρεσίες για κάποιο άλλο άτομο που χρησιμοποιεί τη συσκευή τους καθώς και να αποκλείουν την αγορά εφαρμογών ή παιχνιδιών.
- **Open GLes 3.0:** Πολύ σημαντική προσθήκη για τους κατασκευαστές παιχνιδιών, υπόσχεται πολύ καλύτερα παιχνίδια με rendering σε πραγματικό χρόνο και ανάλυση 1080p.
- **Υποστήριξη Bluetooth Smart:** η οποία προορίζεται για σημαντικά μειωμένη ανάγκη μπαταρίας με μια νέα σειρά Bluetooth αξεσουάρ, χαρίζοντας σημαντικά μεγαλύτερη αυτονομία στη συσκευή.
- **Dial Pad Autocomplete:** Χρήσιμο χαρακτηριστικό που έλειπε, η αυτόματη εμφάνιση επαφών που συμβαδίζουν με τον αριθμό που έχετε ξεκινήσει να πληκτρολογείται στην εφαρμογή τηλεφωνικών κλήσεων.
- **Υποστήριξη Bluetooth AVRCP 1.3:** για εμφάνιση των τίτλων των τραγουδιών σε ένα στερεοφωνικό αυτοκινήτου, ανανεωμένο περιβάλλον στην εφαρμογή λήψης φωτογραφιών και βίντεο καθώς και βελτιωμένος αλγόριθμος για καλύτερη αναγνώριση του κειμένου που εισάγουμε.

Το Android 4.3 Jelly Bean είναι διαθέσιμο για τα smartphones Nexus 4 Galaxy Nexus καθώς και για τα tablets Nexus 7 και Nexus 10.

#### Android 4.4 – KitKat



Η νέα έκδοση του λειτουργικού για smartphones της Google, το Android KitKat ή αλλιώς Android KitKat 4.4 είναι εδώ και παρουσιάστηκε επίσημα από την Google στις 3 Σεπτεμβρίου 2013 μαζί με το νέο smartphone της, το Google Nexus 5 στις

30 Οκτωβρίου 2013. Η νέα έκδοση είναι ελαφριά και φέρνει ένα νέο “ανάλαφρο” στυλ στο design του interface του αλλά και κάποια νέα χαρακτηριστικά που υπόσχονται να κάνουν την ζωή ενός χρήστη με Android κινητό ακόμα πιο όμορφη.

Η έκδοση του Android KitKat “σπάει” το φράγμα των περιορισμών μνήμης που υπήρχε έως τώρα και εμποδίζει low-end μοντέλα να επωφεληθούν από αναβαθμίσεις. Πλέον, μπορεί να “τρέξει” σε συσκευές με RAM από 512MB και πάνω, φέρνοντας τα τελευταία “καλούδια” του Android” ακόμη και σε οικονομικά προσιτά μοντέλα. Όπως αναφέρει εξάλλου η Google, χρησιμοποιεί 16 τοις εκατό λιγότερη μνήμη από το Jelly Bean, με αποτέλεσμα ένα συνολικά πιο ελαφρύ λειτουργικό σε ότι μοντέλο και αν ενσωματώνεται. Οι αλλαγές που φέρνει η νέα έκδοση του KitKat στο Android είναι οι παρακάτω:

Καταρχάς , όλος ο σχεδιασμός του OS έχει υποστεί ένα μίνι “λίφτινγκ”, ώστε το αποτέλεσμα να είναι πιο καθαρό και πιο απλό από ποτέ. Το status και το notification bar της οθόνης έναρξης είναι πλέον διάφανα, ενώ εξαφανίζονται εντελώς όταν βρίσκεστε μέσα σε μια εφαρμογή.

Το “παρών” δίνει ένας νέος launcher, καθώς και μια νέα version του γνώριμου Roboto font της Google που προσφέρει πιο “καθαρό” look, χωρίς ωστόσο να αλλάζει εντελώς το χαρακτήρα του λειτουργικού.

Το Android KitKat διαθέτει μια εντελώς ανανεωμένη Phone app, που κάνει τις κλήσεις πιο εύκολες από ποτέ, βοηθώντας να κάνετε εύκολα και γρήγορα αναζητήσεις στις επαφές σας, σε κοντινά μέρη, ακόμη και σε Google apps accounts μέσα από την ίδια την εφαρμογή.

Ενημέρωση είναι, ωστόσο, και η εφαρμογή των Hangouts, που πλέον διαχειρίζεται όλα τα SMS και MMS στο ίδιο σημείο με τις “internet” επικοινωνίες σας, ώστε να μη χάνεται κανένα μήνυμα από τους φίλους σας σε όποια μορφή και αν σας το στείλουν.

Βελτιώσεις υπάρχουν και στην υπηρεσία φωνητικών αναζητήσεων της Google. Το Voice Search είναι τώρα 25% πιο ακριβές, ενώ μπορεί να ενεργοποιείται και με keyword χωρίς καν να ακουμπήσετε τη συσκευή. Και

συνολικά βέβαια, οι αναζητήσεις σας στο νέο λειτουργικό είναι καλύτερες από ποτέ και πλέον μπορούν να σας οδηγούν ακόμη και σε εφαρμογές και όχι μόνο σε web pages. Το Google Now πλέον ενεργοποιείται με ένα swipe από τα αριστερά στα δεξιά της homescreen σας, έχοντας και αυτό σημαντικές βελτιώσεις και στα χαρακτηριστικά των καρτελών του.

Πλέον η κάμερα λειτουργεί σαν HDR+. Εσείς τραβάτε μια φωτογραφία και η κάμερα επιλέγει τον καλύτερο συνδυασμό χρωμάτων και φτιάχνει το καλύτερο δυνατό αποτέλεσμα. Η δυνατότητα αυτή υπάρχει για την ώρα μόνο στο Google Nexus 5.

Το Android KitKat διαθέτει πολλά ακόμη νέα χαρακτηριστικά. Ανάμεσά τους ξεχωρίζουν:

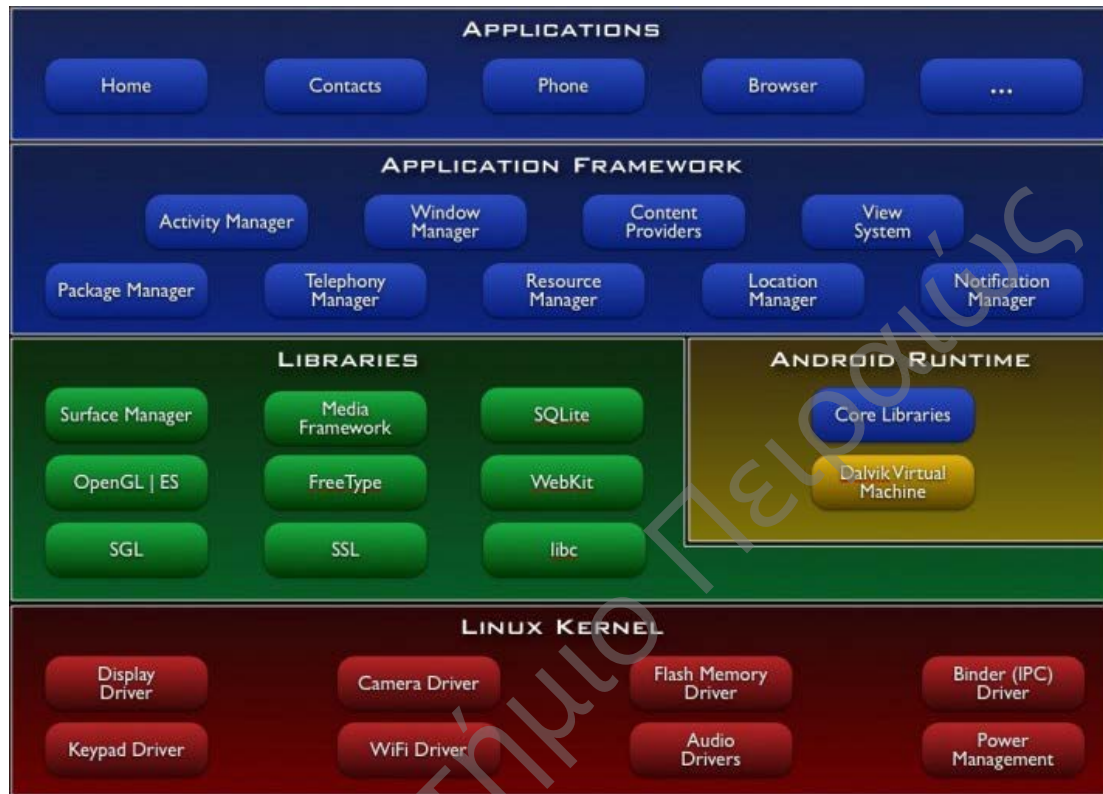
- Καλύτερη ενσωμάτωση των υπηρεσιών cloud μέσα στις εφαρμογές (όπως π.χ στη gallery της συσκευής)
- Εμπλουτισμένη υποστήριξη printing και υποστήριξη νέων εκτυπωτών της HP.
- Δυνατότητα να κάνει κάθε εφαρμογή να ενεργεί σαν μια NFC smart card
- Ενσωμάτωση υποστήριξης για αισθητήρες μέτρησης βημάτων
- Ένα νέο API για υποστήριξη IR καθώς και ένα
- Νέο utility screen recording.

Το Android KitKat, εκτός από το νέο Nexus 5, θα είναι διαθέσιμο για τα Nexus 4, 7, 10 αλλά και για την Developer έκδοση του Galaxy S4 και HTC One.

## 1.4 Αρχιτεκτονική του Android

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από το λειτουργικό σύστημα, τις υπηρεσίες διασύνδεσης με τις εφαρμογές (middleware) και τέλος από τις κύριες εφαρμογές, μεταξύ αυτών, ενός email client, μιας εφαρμογής διαχείρισης SMS,

ενός browser, εφαρμογές διαχείρισης επαφών, και άλλες οι οποίες έρχονται δεμένες με την υπόλοιπη στοιβάδα λογισμικού του Android. Στην παρακάτω εικόνα φαίνεται η αρχιτεκτονική αυτή.



Η αρχιτεκτονική, όπως βλέπουμε αποτελείται από 5 βασικά επίπεδα τα οποία είναι<sup>2</sup>:

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

<sup>2</sup> Wei – Meng Lee, 2012, Beginning Android 4 Application Development, 1st Edition, Wrox

### 1.4.1 Πυρήνας Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux Kernel και υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μια πλατφόρμα στην άλλη.

### 1.4.2 Βιβλιοθήκες

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η κάθε μια από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. Μερικές από τις βιβλιοθήκες του Android είναι:

- **System C Library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.



- **Βιβλιοθήκες πολυμέσων** – υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG και PNG
- **Surface Manager** – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλές εφαρμογές.
- **LibWebCore** – μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο η οποία χρησιμοποιείται και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- **SGL** – η γνωστή μηχανή δισδιάστατων γραφικών
- **Βιβλιοθήκες 3D** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 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.

### 1.4.3 Η εικονική μηχανή Dalvik

Σχεδόν το σύνολο των APIs του Android βασίζονται στην γλώσσα προγραμματισμού Java. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείται ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει κάτι παρόμοιο και δεν είναι άλλο από την εικονική μηχανή Dalvik.

Η Dalvik λοιπόν είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android. Η κάθε εφαρμογή τρέχει μέσω της δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik δεν υποστηρίζει

τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία .dex ώστε να τρέξουν στην VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου στην εκτέλεση.

Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο. Χάρη στο λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

#### **1.4.4 Χρόνος εκτέλεσης εφαρμογής (Android RunTime)**

Ο χρόνος εκτέλεσης των εφαρμογών Android, βρίσκεται στο ίδιο επίπεδο με τις βιβλιοθήκες και τη μηχανή Dalvik. Εδώ βρίσκουμε το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της εικονικής μηχανής Dalvik.

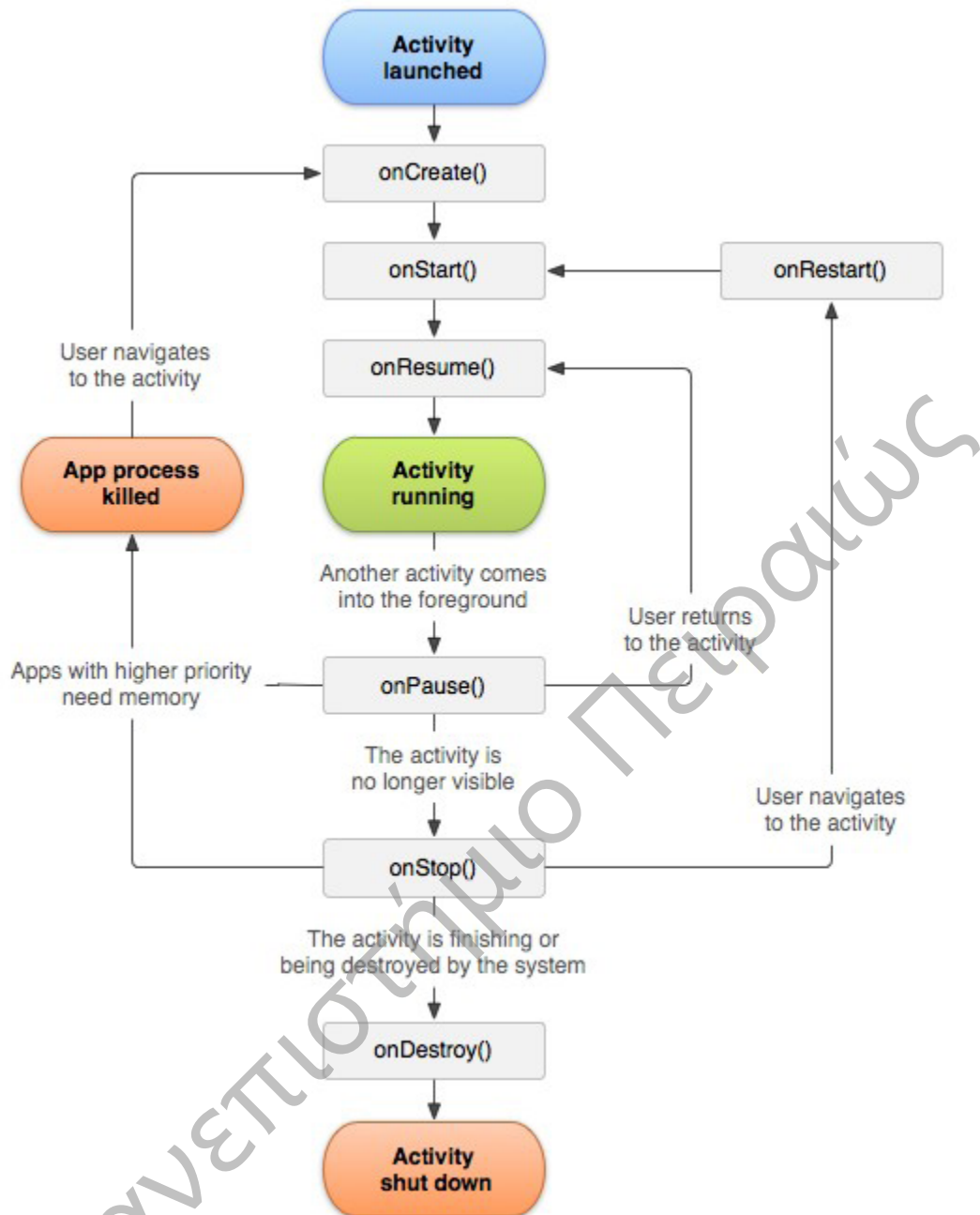
#### **1.4.5 Πλαίσιο εφαρμογής (Application Framework)**

Το Android παρέχει στους developers μια ανοικτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό εφαρμογές. Οι developers έχουν στη διάθεσή τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα.

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση των δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων

της μιας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές της ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

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



## 1.5 Στο εσωτερικό μιας εφαρμογής Android

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να εγκαταστήσουμε στις συσκευές μας.

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

### 1.5.1 Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομά του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από τις πληροφορίες αυτές είναι:

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

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

### 1.5.2 Οι φάκελοι src & res

Στο φάκελο src ( εκ του source) περιέχονται τα αρχεία κλάσης της Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία κλπ. Ο

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

Ο φάκελος `res` (εκ του `resources`) περιέχει όλα τα αρχεία εικόνας, κειμένου, `xml` layout κλπ τα οποία χρησιμοποιούνται από τα `Activities` που βρίσκονται σε φάκελο `src`. Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων σε ένα φάκελο αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κυρίου φακέλου `res` είναι ο φάκελος `drawable` ο οποίος περιέχει τα αρχεία εικόνας (`.png`, `.jpg`, `.gif`) τα οποία χρησιμοποιεί η εφαρμογή μας, ο φάκελος `layout` ο οποίος περιέχει όλα τα αρχεία `xml` τα οποία ορίζουν τα διάφορα `layouts` που υπάρχουν στην εφαρμογή και τέλος ο φάκελος `values` στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται στην εφαρμογή.

### 1.5.3 Οι υπόλοιποι φάκελοι του project

Ένα `project` αποτελείται από τους τρεις βασικούς φακέλους, κάποιιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο `project` λοιπόν περιλαμβάνεται και ο φάκελος με τα διαθέσιμα `APIs` αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο `built path` του `project` μας και τέλος τις διαβαθμίσεις του φακέλου `res` όπως είναι οι φάκελοι `drawable-hdpi`, `drawable-mdpi`, `layout-port`, `menu` κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίπτωση.

## 1.5.4 Δομικά μέρη μιας εφαρμογής

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

- **Δραστηριότητες (Activities)** – πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες `Activities` όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στο χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στο χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.
- **Προθέσεις (Intents)** - οι δραστηριότητες επικοινωνούν και εναλλάσσουν τη λειτουργία τους μέσω των `Intents`. Ουσιαστικά τα `Intents` εξασφαλίζουν την μετάβαση από μια δραστηριότητα σε άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων μπορεί να γίνει είτε μεταξύ των `Activities` μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη.
- **Υπηρεσίες (Services)** – πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματα ακόμη κι όταν η εφαρμογή δεν είναι στο προσκήνιο, πχ μια εφαρμογή `media player` μπορεί μέσω μιας υπηρεσίας να συνεχίζει να παίζει μουσική ακόμη κι όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.
- **Πάροχος περιεχομένου (Content Providers)** – η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός `Intent`, ένας πάροχος περιεχομένου έχει όμως πιο σύνθετη λειτουργία. Οι `content providers` μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω `content providers` είναι οι βάσεις δεδομένων `SQLite` μιας εφαρμογής και οι επαφές του χρήστη.
- **Δέκτες μετάδοσης (Broadcast Receivers)** – πρόκειται για μια υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και

αναλαμβάνει να ενημερώσει το σύστημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθώς μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στο χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

## 1.6 Εργαλεία Ανάπτυξης

Όπως έχουμε δει μέχρι τώρα οι εφαρμογές για το λειτουργικό σύστημα Android γράφονται στη γλώσσα προγραμματισμού Java. Απαραίτητο είναι, για όποιον επιθυμεί να αναπτύξει εφαρμογές για Android, να διαθέτει λειτουργικό σύστημα Windows, Linux ή Mac OS. Επιπρόσθετα, χρειάζονται τα παρακάτω εργαλεία, στις αντίστοιχες εκδόσεις του λειτουργικού στο οποίο θα πραγματοποιηθεί η διαδικασία ανάπτυξης:

1. **Java Development Kit**
2. **Eclipse IDE για Java**
3. **Android SDK**
4. **ADT (Android Development Tools) Plug-in για το Eclipse**

Όλα τα απαραίτητα εργαλεία είναι διαθέσιμα δωρεάν και η εγκατάστασή τους είναι αρκετά εύκολη και σύντομη διαδικασία ( η χρονική διάρκεια της εγκατάστασης εξαρτάται σημαντικά από την ταχύτητα της σύνδεσης στο διαδίκτυο).



### 1.6.1 Εγκατάσταση Eclipse 3.5

Το Eclipse IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών. Δεν είναι απαραίτητο να έχει κάποιος ένα περιβάλλον ανάπτυξης προκειμένου να αναπτύξει εφαρμογές, ένα απλό σημειωματάριο αρκεί. Οι τεράστιες όμως διευκολύνσεις που παρέχει ένα τέτοιο περιβάλλον το καθιστούν απαραίτητο για μεσαίου και μεγάλου μεγέθους εφαρμογές.

Το Eclipse παρέχει ένα περιβάλλον για την οργάνωση του κώδικα ενός ολόκληρου project. Λειτουργίες όπως, διατήρηση εκδόσεων, διαμοιρασμός αλλαγών στον κώδικα μεταξύ των διαφόρων μελών της ομάδας ανάπτυξης, εικονικοί εξυπηρετητές για έλεγχο ορθότητας web εφαρμογών και άλλες παρέχονται εντός του ίδιου του περιβάλλοντος. Επιπλέον, με την υποστήριξη των plug-in το Eclipse μπορεί να επεκτείνει τις λειτουργίες που προσφέρει ακόμα παραπέρα.

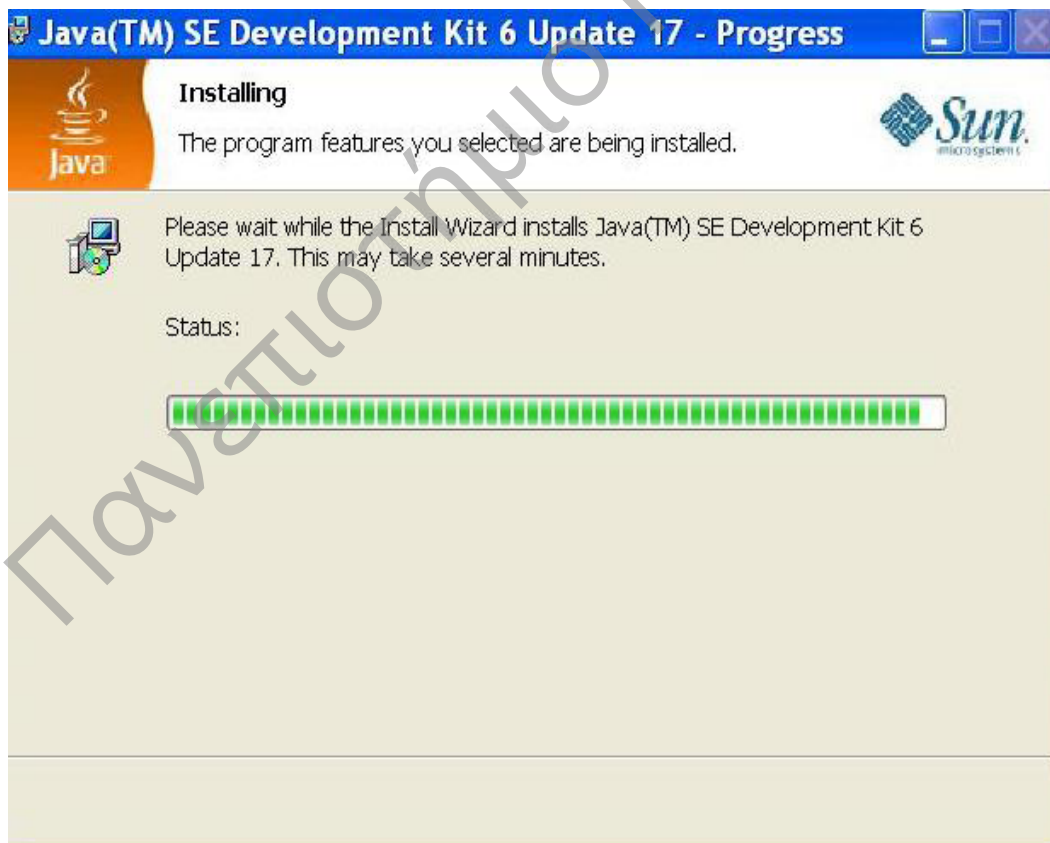
Η εγκατάσταση του Eclipse είναι πολύ εύκολη υπόθεση. Το μόνο που χρειάζεται να γίνει είναι να αποθηκευτεί ο φάκελος του Eclipse που λαμβάνουμε από την ιστοσελίδα <http://www.eclipse.org/downloads/> σε μια επιθυμητή διαδρομή στο δίσκο μας. Από κει και πέρα το πρόγραμμά μας είναι πλήρως λειτουργικό.

### 1.6.2 Εγκατάσταση Java SDK

Αρχικά θα γίνει κατέβασμα του προγράμματος από την ιστοσελίδα <http://java.sun.com/javase/downloads/> και εγκατάσταση στο program files/java



*Άδεια χρήσης του JDK*



*To InstallShield Wizard για το JDK*

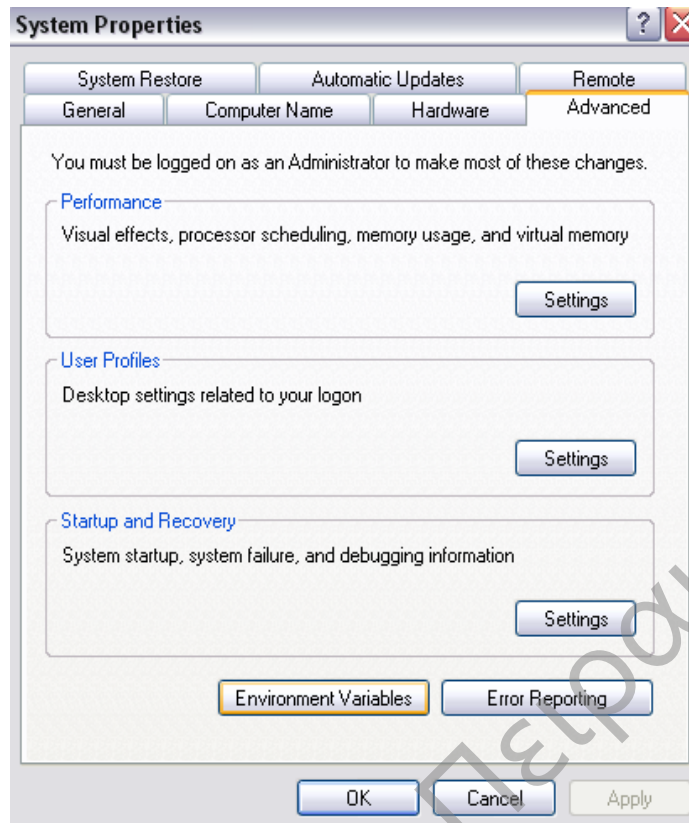
Το τελευταίο βήμα πριν να είναι έτοιμο προς λειτουργία το JDK είναι ο ορισμός της μεταβλητής path. Το Path δείχνει που είναι εγκατεστημένα τα εργαλεία του JDK.

**Άνοιγμα του παραθύρου System Properties:** δεξιό κλικ στο εικονίδιο My Computer στην επιφάνεια εργασίας και επιλογή Properties από το μενού.



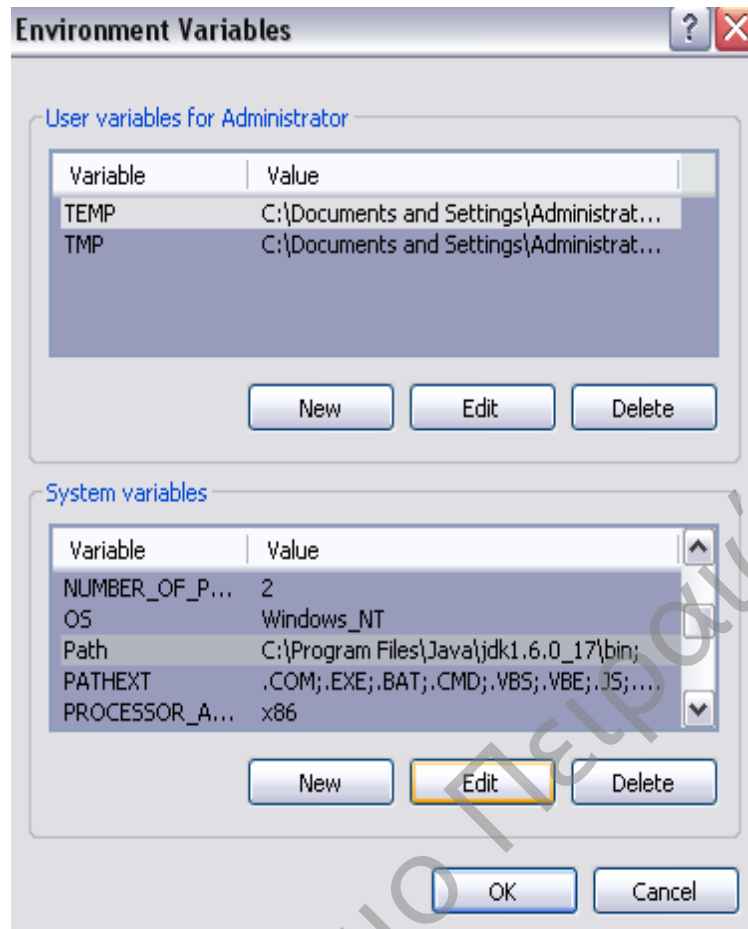
*System Properties*

Εμφανίζεται το παράθυρο System Properties όπως φαίνεται παρακάτω.



*Η καρτέλα Advanced του παραθύρου διαλόγου System Properties*

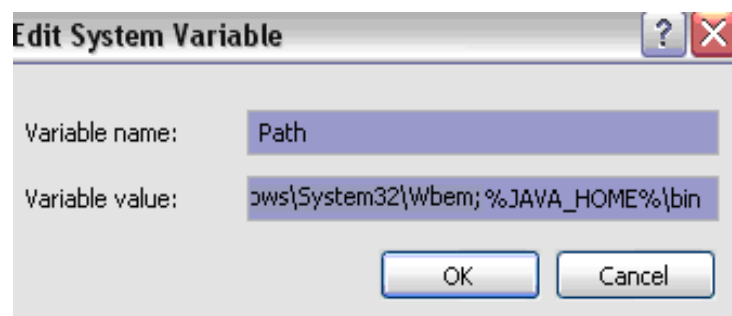
**Άνοιγμα του παραθύρου διαλόγου Environment Variables:** Επιλογή της καρτέλας Advanced στο πάνω μέρος του παραθύρου διαλόγου. Πατώντας το κουμπί Environment Variables, εμφανίζεται το παράθυρο διαλόγου Environment Variables, όπως φαίνεται παρακάτω:



Παράθυρο διαλόγου *Environment Variables*

**Επεξεργασία της μεταβλητής Path:** Χρησιμοποιώντας τη γραμμή κύλισης στο πλαίσιο System Variables γίνεται η επιλογή της μεταβλητής path. Πατώντας το κουμπί Edit. Αυτό θα προκαλέσει την εμφάνιση του παραθύρου Edit System Variable.

**Αλλαγή των περιεχομένων της μεταβλητής Path:** Χρειάζεται η προσθήκη του καταλόγου C:/Program Files\Java\jdk1.6.0\_17\bin; στη μεταβλητή path του συστήματος



Path πριν:

`C:\Windows\System32;C:\Windows;C:\Windows\System32\Wbem`

Path μετά:

`C:\Windows\System32;C:\Windows;C:\Windows\System32\Wbem;C:/ProgramFiles\Java\jdk1.6.0_17\bin;`

ή

`C:\Windows\System32;C:\Windows;C:\Windows\System32\Wbem;%JAVA_HOME%\bin`

### 1.6.3 Εγκατάσταση Android SDK

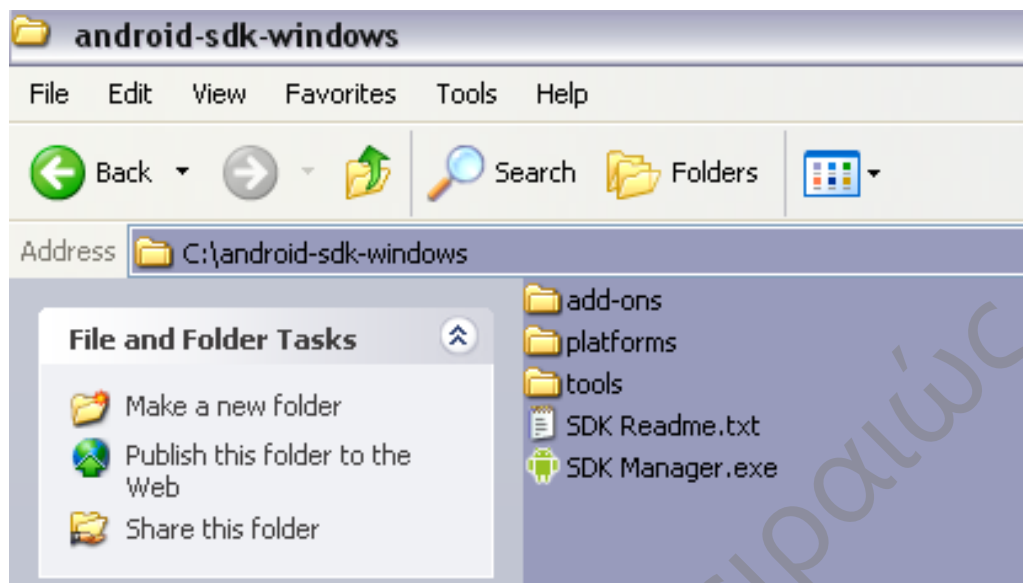
Το Android SDK αποτελεί απαραίτητο εργαλείο για την ανάπτυξη εφαρμογών Android. Το Android SDK περιλαμβάνει τις απαραίτητες βιβλιοθήκες για την ανάπτυξη εφαρμογών Android, παραδείγματα εφαρμογών με τον πηγαίο κώδικά τους, τον εξομοιωτή συσκευών καθώς και άλλες βιβλιοθήκες που έχουν να κάνουν με την τιμολόγηση των εφαρμογών και την πώλησή τους στο Google play.

Για την εγκατάστασή του ακολουθούμε τα παρακάτω βήματα:

Εύρεση του Android SDK από την ιστοσελίδα:  
<http://developer.android.com/sdk>.

Προτείνεται η αποθήκευση του Android SDK σε έναν εύκολα προσβάσιμο φάκελο όπως για παράδειγμα τον δίσκο C:\ ..

Μετά το unzip θα πρέπει να εμφανιστούν τα αρχεία όπως φαίνονται παρακάτω:



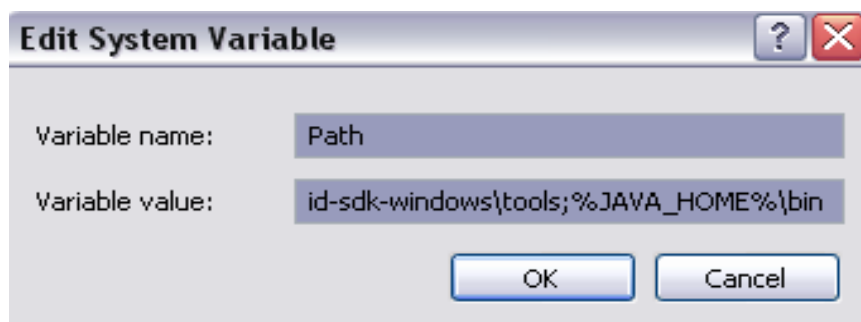
*Περιεχόμενα φακέλου android-sdk-windows*

Το Android SDK περιέχει ένα φάκελο με το όνομα tools ο οποίος θα πρέπει να συμπεριληφθεί στο Environment Variables με τον τρόπο που δείχθηκε προηγουμένως.

Χρειάζεται η προσθήκη του καταλόγου όπου είναι αποθηκευμένος ο φάκελος tools στην μεταβλητή path του συστήματος.

Στην παρούσα κατάσταση είναι C:\android-sdk-windows\tools;

Οπότε:



Το path πριν:

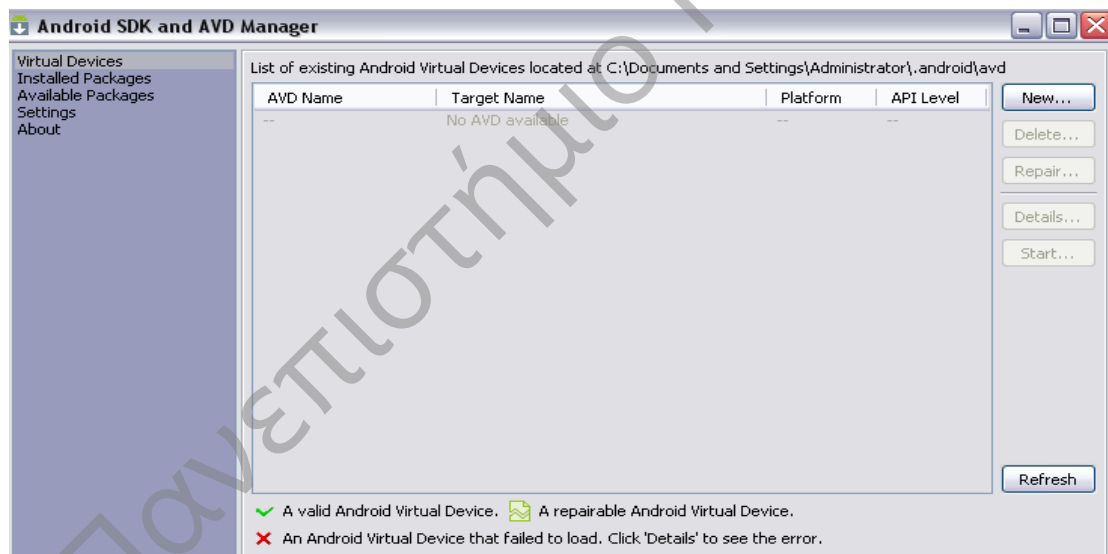
```
C:\Windows\System32;C:\Windows\;C:\Windows\System32\Wbem;%JAVA_HOME%\bin
```

Το path μετά:

```
C:\Windows\System32;C:\Windows\;C:\Windows\System32\Wbem;%JAVA_HOME%\bin
```

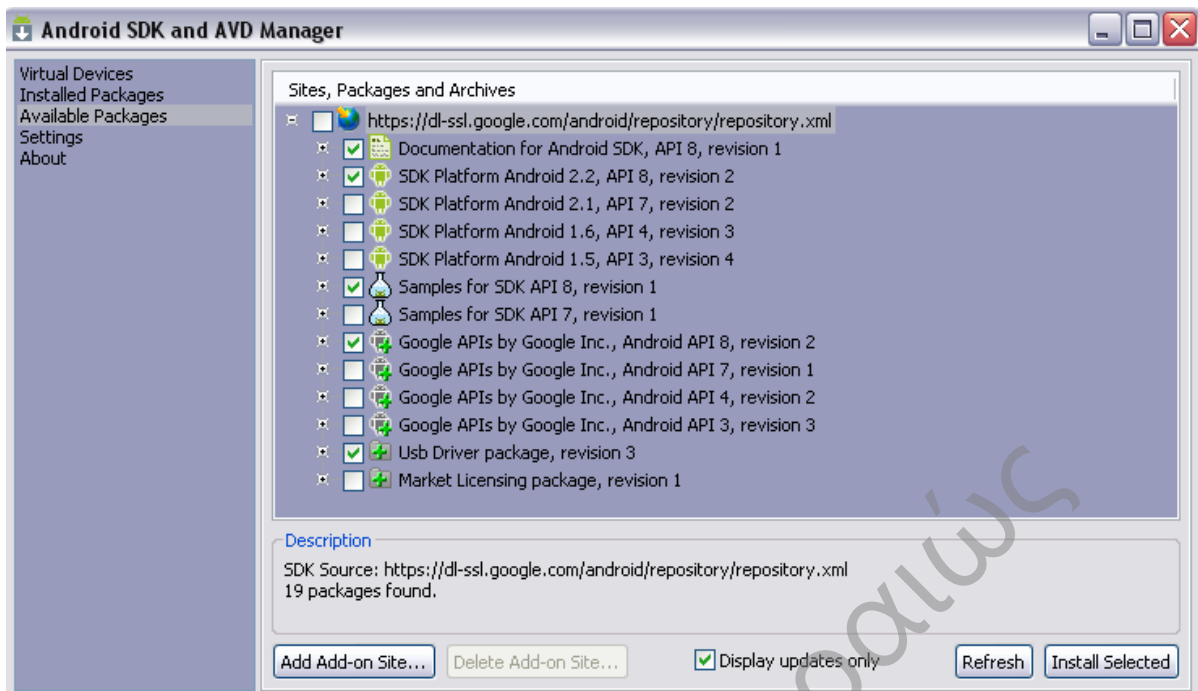
## 1.6.4 Εγκατάσταση AVD

Μέσα στο φάκελο tools υπάρχει ένα bat αρχείο με όνομα android. Με διπλό κλικ θα εμφανιστεί η αρχική οθόνη της εφαρμογής.

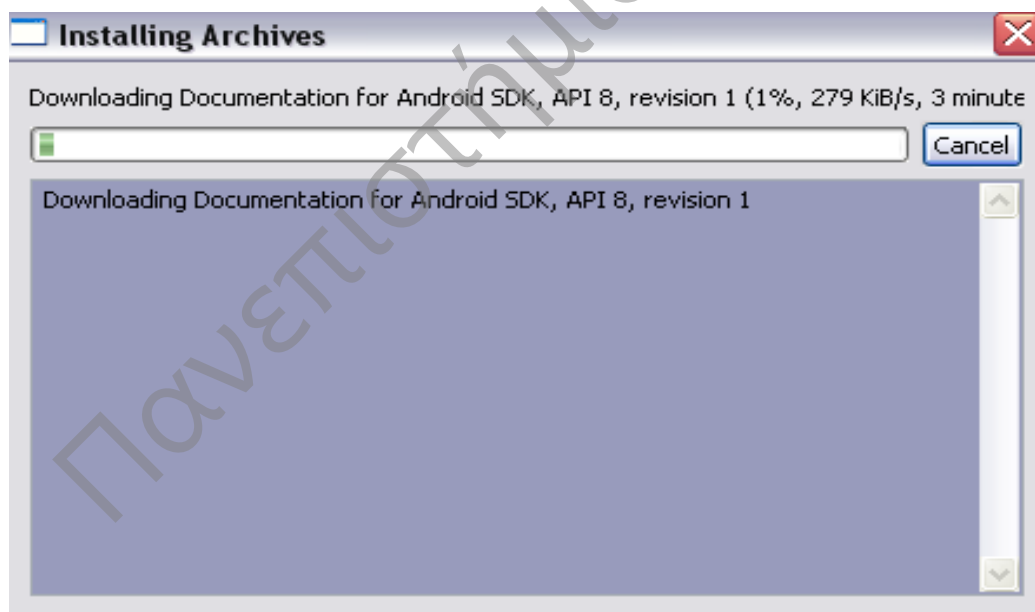


Επιλέγοντας αριστερά **Available packages** και έπειτα **install selected** θα εγκατασταθούν τα προγράμματα που είναι διαθέσιμα.



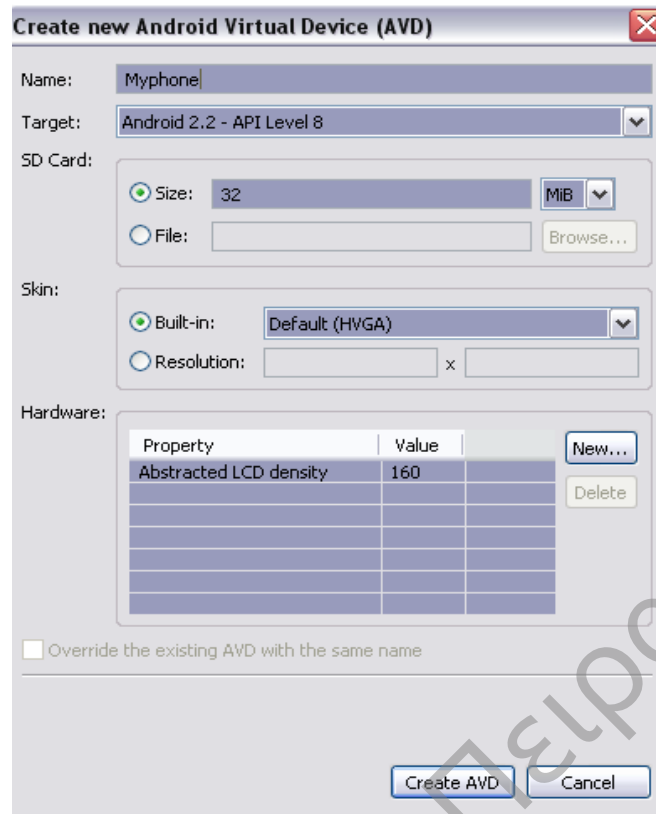


*Επιλογή Available Packages*



*Εγκατάσταση Available Packages*

Στην αρχική οθόνη και πάλι επιλέγοντας New δημιουργείται ένα emulator με ειδικά χαρακτηριστικά που επιλέγονται από το χρήστη.



Επιλογή χαρακτηριστικών Emulator



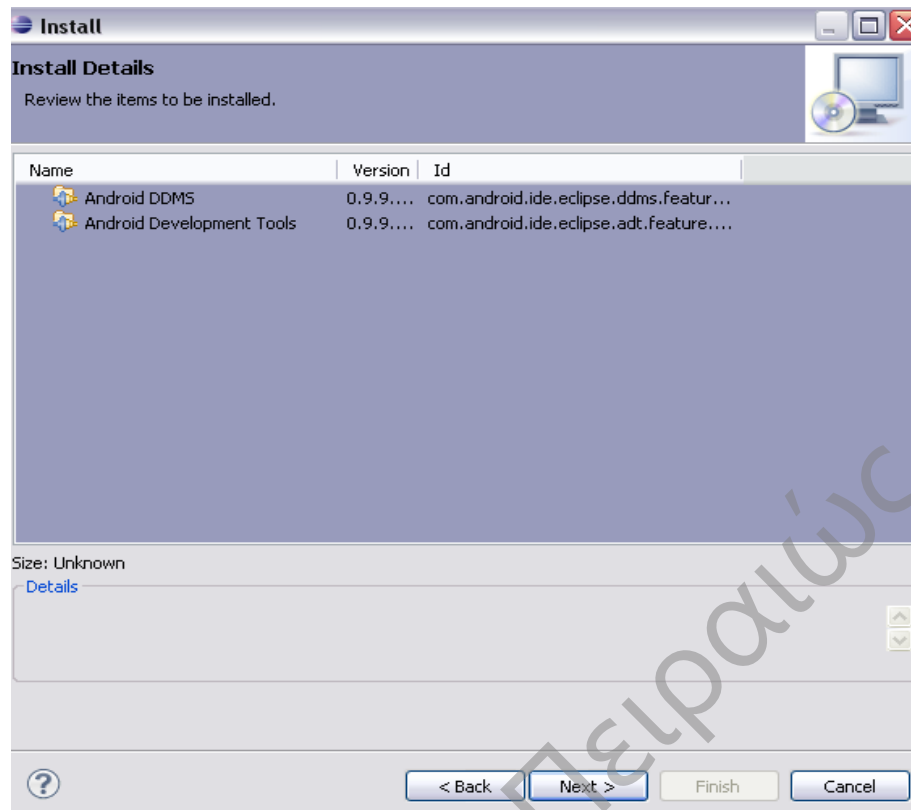
## 1.6.5 Εγκατάσταση Android Development Tools (ADT)

Χάρη στα plug-in το Eclipse μπορεί να επεκτείνει τις λειτουργικότητες που προσφέρει. Εκμεταλλευόμενη το γεγονός αυτό η Google ανέπτυξε το ADT (Android Development Tools) Plugin για το Eclipse. Με το plug-in αυτό το Eclipse προσφέρει τη δυνατότητα σχεδίασης διεπαφών χρήστη για τις εφαρμογές Android με γραφικό τρόπο. Επεκτείνει τις επιλογές των μενού εισαγωγής του Eclipse έτσι ώστε, για παράδειγμα, να μπορεί κανείς να δημιουργήσει ένα καινούριο project Android με τη σωστή δομή και αρχεία αυτοματοποιημένα εντός του περιβάλλοντος.

Η εγκατάσταση του plug-in μπορεί να γίνει εντός του ίδιου του Eclipse ακολουθώντας την παρακάτω διαδικασία:

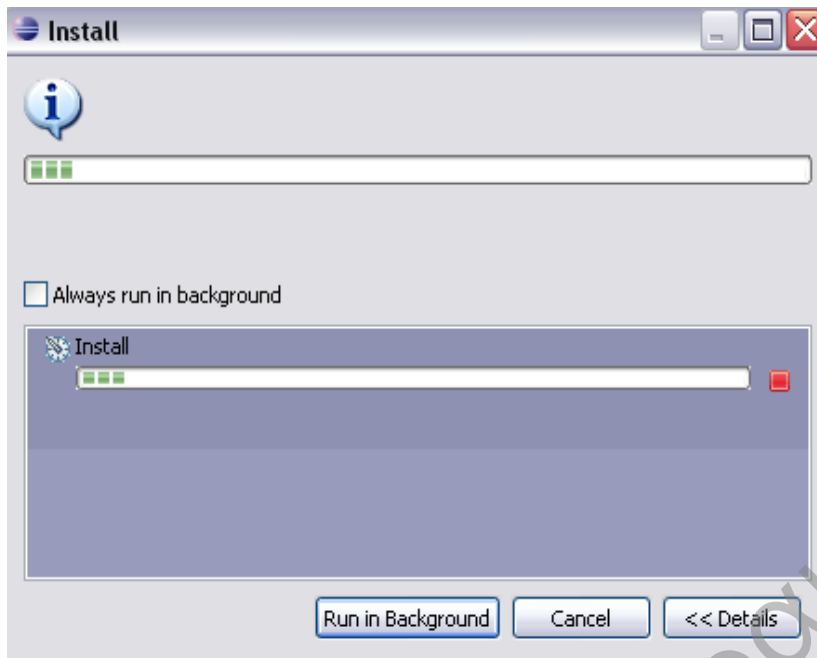
Άνοιγμα του Eclipse και επιλογή **Help** → **Install New Software**.

1. Επιλογή add στο παράθυρο Available Software
2. Στο **Add Site** παράθυρο που εμφανίζεται πρέπει να οριστεί ένα όνομα στο Name Field για παράδειγμα, "Android Plug-in" και στο Location Field πρέπει να εισαχθεί το παρακάτω URL: <https://dl-ssl.google.com/android/eclipse/>
3. Πίσω στο **Available Software** παράθυρο, πρέπει να έχουν προστεθεί 2 "Developer Tools" στη λίστα το 'Android DDMS' και το 'Android Development Tools'. Επιλέγοντάς τα με το checkbox πρόκειται να προστεθούν για εγκατάσταση.



*Install details*

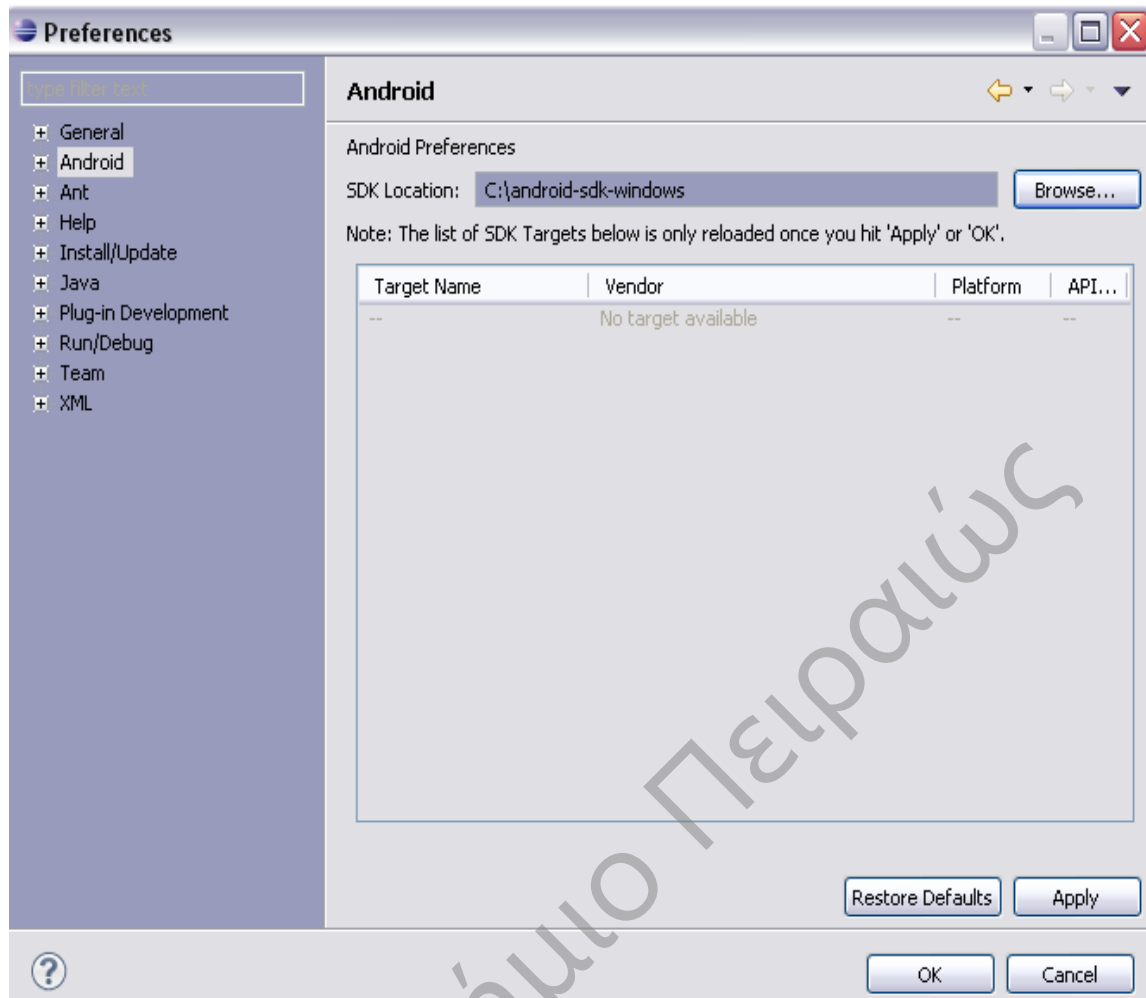
4. Στο παράθυρο Install Details, το Android DDMS και το Android Development Tools έχουν προστεθεί. Στη φάση αυτή είναι αναγκαίο να αποδεχτούν οι όροι χρησιμοποίησης των προγραμμάτων. Πατώντας Next θα αρχίσει η εγκατάσταση.



5. Το Eclipse χρειάζεται επανεκκίνηση προτού τεθεί σε λειτουργία.

### 1.6.6 Select Android SDK location

Με το Eclipse σε λειτουργία ακολουθούμε τα εξής βήματα: **eclipse** → **preferences** → **Android** → **SDK Location**, εκεί πρέπει να προστεθεί το directory του android-sdk-windows (στην περίπτωση του οδηγού αυτού είναι C://android-sdk-windows).



Επιλογή Android SDK Location

# Ανάπτυξη εφαρμογών στο Android

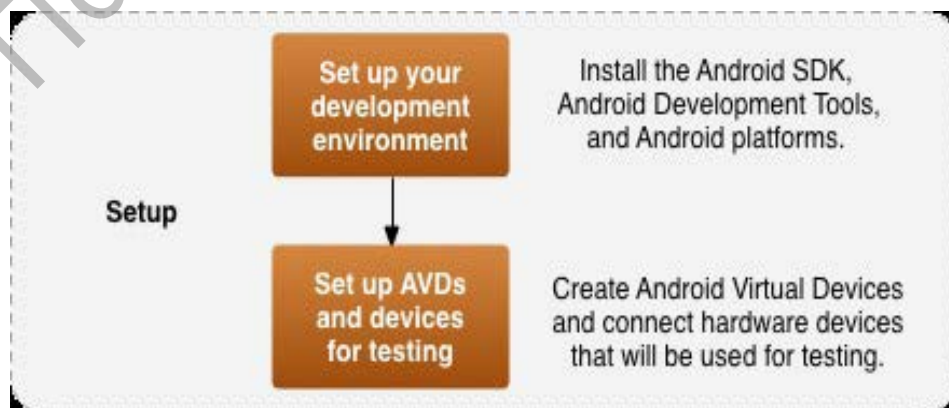
## 2.1 Κύκλος ανάπτυξης της εφαρμογής

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

### 2.1.1 Εγκατάσταση λογισμικού

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείται να στήσει το περιβάλλον εργασίας στο οποίο θα γίνει ο σχεδιασμός, η ανάπτυξη, ο έλεγχος και η λειτουργία των εφαρμογών. Μπορεί να επιλέξει όποιο περιβάλλον ανάπτυξης (IDE) τον εξυπηρετεί καλύτερα και να χρησιμοποιήσει όλα τα εργαλεία του Android SDK μηδενός εξαιρουμένου.

Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στη διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει τη λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας. Ιδανικά ο developer θα διαθέτει έναν αριθμό διαφορετικών φυσικών συσκευών ώστε να δοκιμάσει ο ίδιος πως συμπεριφέρεται η εφαρμογή του σε κάθε περίπτωση, όμως αυτή η πρακτική μπορεί να αποδειχθεί πολυδάπανη και χρονοβόρα.

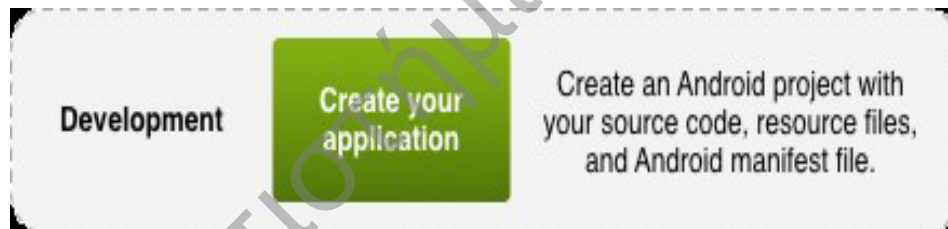


Πρώτο βήμα – Εγκατάσταση λογισμικού

## 2.1.2 Ανάπτυξη πηγαίου κώδικα της εφαρμογής

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

Η διαδικασία ξεκινάει με ένα νέο Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά ότι χρειάζεται η εφαρμογή για να τρέξει ως οφείλει. Στο project του ο developer θα πρέπει να φροντίσει ώστε το υλικό του να είναι τακτοποιημένο και ο κώδικάς του ευανάγνωστος ώστε να ακολουθήσει η διαδικασία του Debugging.



Δεύτερο βήμα - Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής

## 2.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής

Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείται από αρκετά επιμέρους στάδια.

Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode. Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης προϋποθέτουν ότι ο κώδικας δεν έχει

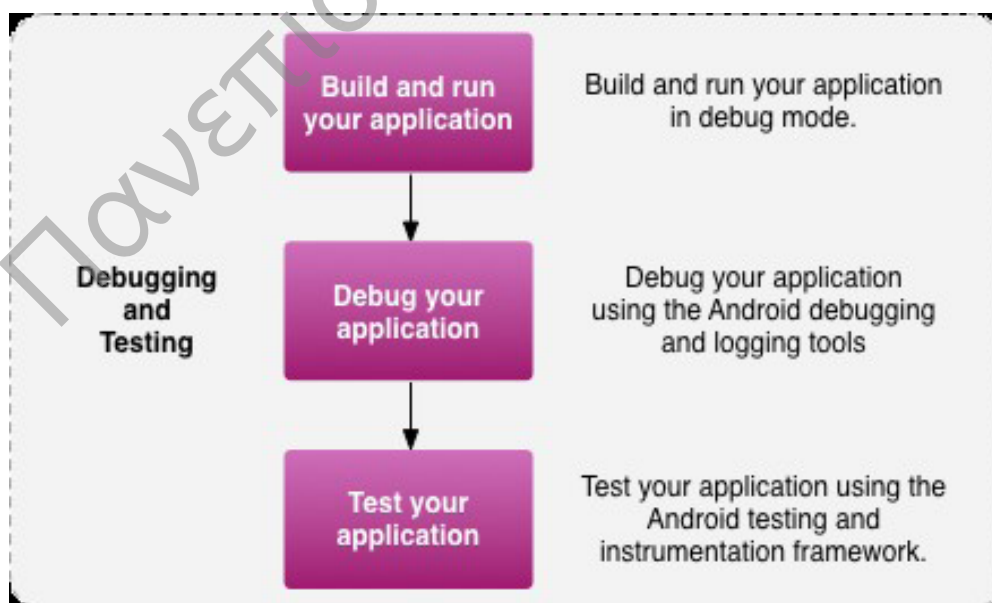


κανένα συντακτικό λάθος, αλλιώς ειδοποιούν το χρήστη να τα διορθώσει. Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε συσκευή μέσω ADB push εντολής.

Στο δεύτερο στάδιο ο προγραμματιστής καλείται να αντιμετωπίσει τα λειτουργικά και αισθητικά προβλήματα της εφαρμογής του, πρώτα εντοπίζοντάς τα στη λειτουργία της συσκευής και μετά διορθώνοντας τα κομμάτια του κώδικα που δημιουργούν τα σφάλματα. Το κύριο εργαλείο που κάνει αυτή τη διαδικασία εφικτή είναι το “LogCat” το οποίο μας επιστρέφει το stack trace του κώδικα στο σημείο εκείνο που συνέβη το σφάλμα.

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

Φυσικά η διαδικασία του debugging είναι ένας βρόχος (loop) που επαναλαμβάνεται συνέχεια μέχρι να εντοπιστούν και να διορθωθούν όλα τα σφάλματα της εφαρμογής.



Τρίτο βήμα – Δοκιμή και Debugging της εφαρμογής

Πριν να εκδώσει την εφαρμογή του στο κοινό, ο προγραμματιστής του Android, αλλά και οποιασδήποτε άλλης πλατφόρμας, οφείλει να δοκιμάσει ενδελεχώς κάθε λειτουργική και αισθητική παράμετρο της εφαρμογής ώστε να εξασφαλίσει ότι ο χρήστης θα έχει όσο το δυνατόν καλύτερη εμπειρία χρήσης.

Ο αυτοσκοπός της ανάπτυξης εφαρμογής είναι η ικανοποίηση των αναγκών όσο το δυνατόν μεγαλύτερου αριθμού χρηστών. Αυτό φυσικά δεν επιτυγχάνεται αμέσως. Πρέπει να ακολουθήσει μια εκτενής περίοδος δοκιμών για να διαπιστωθεί ότι η εφαρμογή θα αποδίδει όπως οφείλει σε όλες τις καθημερινές της χρήσεις, και αφού περάσει επιτυχώς το στάδιο των δοκιμών, ακολουθεί η ώρα της δημοσίευσης.

Πως γίνεται λοιπόν η διαδικασία της αποσφαλμάτωσης<sup>3 4</sup>; Μια εφαρμογή μπορεί να αποτελείται από μερικές απλές κλάσεις ή από δεκάδες κλάσεις χωρισμένες σε πακέτα, αποτελούμενες από δεκάδες γραμμές κώδικα η καθεμία. Η έρευνα για σφάλματα στη δεύτερη περίπτωση έχει πολύ χειρότερη αναλογία πιθανοτήτων.

Η ανάγκη για εργαλεία ελέγχου, καλύπτεται στο έπακρο με μια σειρά εφαρμογών τα οποία δένοντας αρμονικά με το περιβάλλον ανάπτυξης μας (IDE) μας λύνουν τα χέρια, στο δύσκολο έργο του εντοπισμού σφαλμάτων. Τα εργαλεία αυτά αφορούν τον έλεγχο της εφαρμογής σε εικονικές μηχανές (Android Virtual Devices), καταγραφή σφαλμάτων (logcat), εργαλεία ελέγχου μνήμης και άλλων λειτουργιών της συσκευής (DDMS) κλπ.

### 2.1.3.1 Android Debug Bridge

Για να λειτουργήσουν σωστά τα εργαλεία που αναφέραμε παραπάνω, χρειάζεται κάποιο είδος προγράμματος client-server που να συνδέει τον υπολογιστή με τις συσκευές μας, εικονικές και μη. Το ρόλο αυτό αναλαμβάνει το Android Debug Bridge (ADB). Πρόκειται για ένα εργαλείο γραμμής εντολών και το οποίο αποτελείται από τρία μέρη:

---

<sup>3</sup> Reto Meier, 2012, Professional Android 4 Application Development, Updated Edition, John Wiley & Sons

<sup>4</sup> J.Steele,2010,The Android Developer's Cookbook, Addison & Wesley

- Έναν client ο οποίος τρέχει στον υπολογιστή που έχουμε στήσει το SDK. Μπορούμε είτε να τον εκκινήσουμε χειροκίνητα είτε να χρησιμοποιήσουμε κάποιο εργαλείο το οποίο ξεκινάει αυτόματα το δικό του client, όπως το DDMS ή το ADT Plugin.
- Έναν server ο οποίος τρέχει σαν υπηρεσία παρασκηνίου στον υπολογιστή που βρίσκεται το SDK, όπως και ο client. Ο server εξασφαλίζει την επικοινωνία μεταξύ του client και του “δαίμονα” (daemon) που τρέχει στη συσκευή.
- Ο “δαίμονας” που τρέχει σαν διεργασία παρασκηνίου στην εικονική ή πραγματική συσκευή που χρησιμοποιείται για εξομοίωση.

Όταν ξεκινάει το ADB, ο client ελέγχει αν υπάρχει κάποια διεργασία του server που να εκτελείται ήδη, αλλιώς δημιουργεί μια νέα. Μετά δημιουργεί μια τοπική TCP σύνδεση στην θύρα 5037 και είναι έτοιμος να δεχτεί εντολές. Μετά ελέγχει το εύρος θυρών TCP από 5555 μέχρι 5585, στο οποίο επικοινωνούν οι συσκευές εξομοίωσης, και ελέγχει αν υπάρχουν διαθέσιμες και πόσες είναι αυτές. Αφού εντοπίσει κάποια συσκευή ελέγχει αν σε αυτή τη συσκευή τρέχει ο “δαίμονας”, και αν ναι δημιουργείται σύνδεση adb με την συσκευή.

Αφού πραγματοποιηθεί σύνδεση μεταξύ συσκευής και client μπορούμε να χρησιμοποιήσουμε όλες τις δυνατότητες που μας παρέχει ο ADB για να ασκήσουμε πλήρη έλεγχο της συσκευής. Οι εντολές μέσω γραμμής εντολών δίνονται ως εξής:

**adb [-d|-e|-s <serialNumber>] <command>**

Όπου -d είναι η απευθείας εντολή εάν υπάρχει μόνο μια συνδεδεμένη συσκευή, -e η απευθείας εντολή αν υπάρχει μια συνδεδεμένη εικονική συσκευή, -s <serialNumber> η απευθείας εντολή στη συσκευή που έχει το <serialNumber>, και <command> είναι η εντολή που θέλουμε να εκτελεστεί.

Παράδειγμα χρήσης:

**adb -s emulator-5556 install helloWorld.apk**

Στο παραπάνω παράδειγμα γίνεται εγκατάσταση της εφαρμογής με όνομα “helloWorld.apk” στη συσκευή εξομοίωσης που ακούει στη θύρα 5556.

### 2.1.3.2 Εικονικές συσκευές Android (Android Virtual Devices – AVD)

Όπως γράψαμε και παραπάνω, ο developer πρέπει πριν εκδώσει την εφαρμογή του να την δοκιμάσει σε έναν αριθμό συσκευών για να εξασφαλίσει την ομαλή λειτουργία της σε όλες τις συνθήκες. Φυσικά το κόστος είναι αρκετά μεγάλο για να αποθαρρύνει τον προγραμματιστή να έχει στην κατοχή του 10 με 20 συσκευές για να ελέγξει σε όλες τις λειτουργίες και την εμφάνιση της εφαρμογής του. Το πρόβλημα αυτό έρχεται να λύσει η ύπαρξη των εικονικών συσκευών Android.

Πρόκειται για μια συσκευή εξομοίωσης η οποία μας επιτρέπει να εξομοιώσουμε την λειτουργία και συμπεριφορά μιας εικονικής συσκευής, ορίζοντας τις επιλογές υλικού και λογισμικού που θέλουμε τον εξομοιωτή του Android. Με τον τρόπο αυτό ο developer μπορεί να ελέγξει την εφαρμογή του σε μια σειρά από πραγματικά σενάρια λειτουργίας και να πάρει γρήγορα και άμεσα feedback για τη λειτουργία της εφαρμογής του. Μια εικονική συσκευή αποτελείται από:

- **Το προφίλ του υλικού:** Σε αυτό προσδιορίζονται οι ιδιότητες και τα χαρακτηριστικά της εικονικής συσκευής. Μπορούμε παραδείγματος χάρη να ορίσουμε την ανάλυση της οθόνης και την πυκνότητα σε pixels, το μέγεθος της μνήμης RAM, αν η συσκευή θα έχει κάμερα, υποστήριξη GPS κλπ.
- **Την έκδοση του Android:** Επιλογή της έκδοσης της πλατφόρμας του Android που θέλουμε να εξομοιώσει η εικονική συσκευή.
- **Έξτρα χώρος αποθήκευσης:** Εδώ αποθηκεύονται όλα τα δεδομένα της εφαρμογής και επίσης μπορούμε να ορίσουμε μια εικονική κάρτα μνήμης ώστε να επεκτείνουμε τον αποθηκευτικό χώρο, όπως θα κάναμε σε μια πραγματική συσκευή.

### 2.1.3.3 Εργαλεία καταγραφής συμβάντων – LogCat

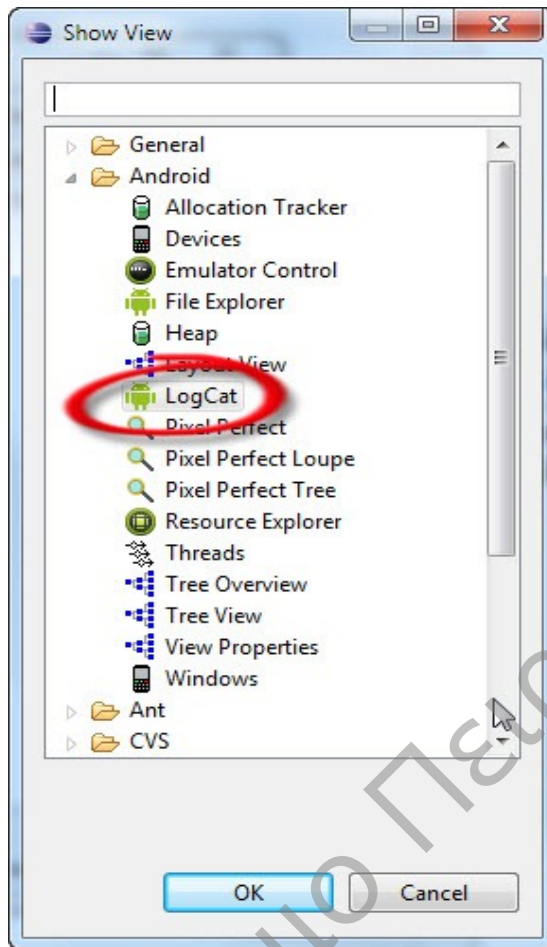
Το Android διαθέτει έναν μηχανισμό καταγραφής συμβάντων, σκοπός του οποίου είναι η συλλογή και προβολή των αρχείων αποσφαλμάτωσης του

συστήματος. Τα δεδομένα των διαφόρων εφαρμογών συγκεντρώνονται σε μια σειρά από buffers, τους οποίους μετά μπορούμε να προβάσουμε και να φιλτράρουμε με την εντολή 'logcat'.

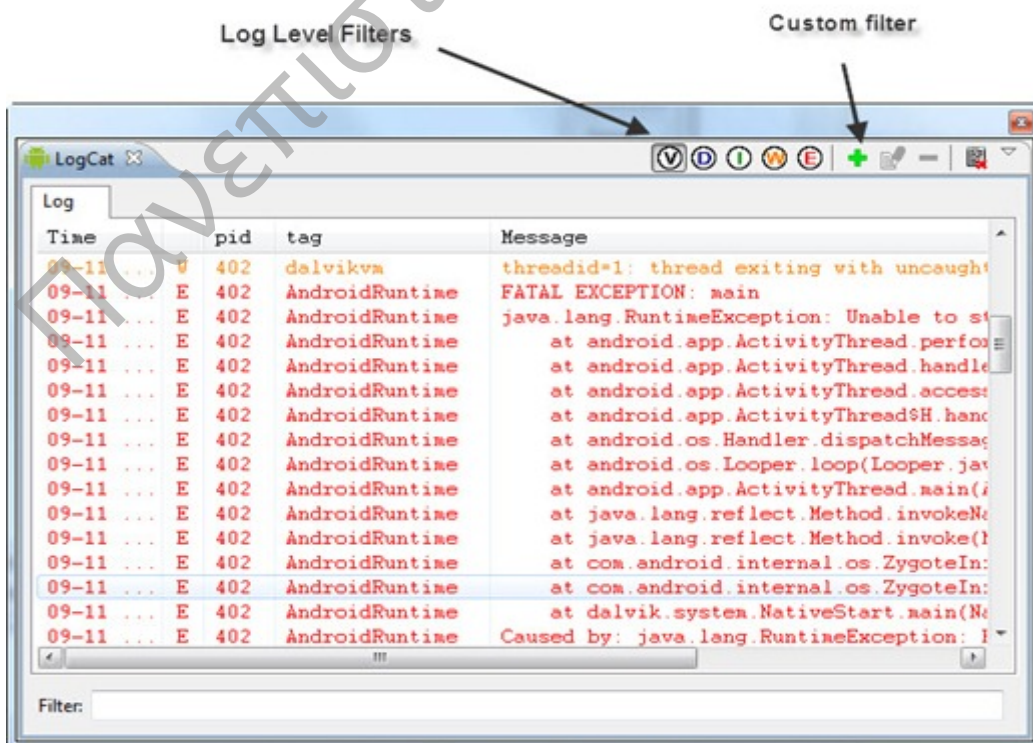
Στον προγραμματισμό υπάρχουν οι λεγόμενες εξαιρέσεις (exceptions), καταστάσεις δηλαδή που προκύπτουν όταν κάτι δεν πάει καλά, και αυτό έχει σαν αποτέλεσμα την διακοπή λειτουργίας του προγράμματος σε περίπτωση που δεν έχουμε φροντίσει να 'χειριστούμε' την εξαίρεση. Συνηθισμένο παράδειγμα εξαίρεσης λειτουργίας του Android είναι η "NullPointerException" η οποία μας εμφανίζεται όταν προσπαθούμε να προσπελάσουμε κάποια μεταβλητή ή αντικείμενο που έχει μηδενική (Null) τιμή.

Τα σφάλματα λειτουργίας μιας εφαρμογής στο Android, που προκύπτουν από εξαιρέσεις λειτουργίας, συνήθως προκαλούν τον άμεσο τερματισμό εφαρμογής προβάλλοντας ένα παράθυρο με το όνομα της εφαρμογής που τερματίστηκε, και ένα απλό μήνυμα σφάλματος.

Το LogCat λοιπόν είναι μια εντολή γραμμής εντολών η οποία μπορεί να χρησιμοποιηθεί μέσω του ADB για να δούμε τα debug logs της συσκευής που δουλεύουμε και άρα της εφαρμογής που αναπτύσσουμε ώστε να εντοπίσουμε τις πηγές των σφαλμάτων, οι οποίες συνήθως αν όχι πάντα, είναι exception στον κώδικά μας. Το Eclipse ενσωματώνει μια GUI έκδοση του LogCat για αποτελεσματικότερη αποσφαλμάτωση του κώδικά μας. Στο Eclipse για να μεταφερθούμε στο LogCat παράθυρο ακολουθούμε την παρακάτω διαδρομή: επιλέγουμε *Window* → *Show View* → *Other* → *Android* → *LogCat*.



Open the Android LogCat View in Eclipse



Στην παραπάνω εικόνα βλέπουμε την τυπική διάταξη του logcat τη στιγμή μάλιστα που έχει συμβεί μια εξαίρεση λειτουργίας στην εφαρμογή μας και η οποία εμφανίζεται με κόκκινα χαρακτηριστικά γράμματα ξεκινώντας με τη φράση “Fatal Excerption”. Οι κόκκινες γραμμές που προβάλλονται είναι όλες εξαιρετικά σημαντικές, κυρίως όμως μας ενδιαφέρουν αυτές που κάνουν λόγο για τη φύση της εξαίρεσης, για τη γραμμή του κώδικα και την κλάση στην οποία συνέβη το σφάλμα.

Βέβαια υπάρχουν πολλές άλλες εξαιρέσεις οι οποίες προκύπτουν σε διάφορες ανύποπτες στιγμές και χάρη στο logcat μπορούμε αρχικά να τις εντοπίσουμε και μετά να τις αντιμετωπίσουμε κάνοντας τις απαραίτητες διορθώσεις/αλλαγές στον κώδικά μας.

Φυσικά υπάρχουν και τα σφάλματα στον κώδικα τα οποία δεν προκαλούν αναγκαστικό κλείσιμο της εφαρμογής αλλά παρόλα αυτά συμβάλουν στη μη σωστή λειτουργία της. Αυτού του είδους τα σφάλματα φυσικά δεν τα πιάνει ο compiler, ούτε εμφανίζονται με τη μορφή που εμφανίζονται τα σφάλματα που προκύπτουν από εξαιρέσεις. Το αποτέλεσμα συνήθως αυτών των σφαλμάτων είναι μια κενή λίστα, μια λάθος τοποθετημένη εικόνα, κάποιο λάθος κείμενο κλπ.

Πως αντιμετωπίζουμε λοιπόν ένα σφάλμα το οποίο δεν εμφανίζει κάποιο stack trace με ακριβές σημείο κώδικα προς διόρθωση, όπως συμβαίνει στην περίπτωση των εξαιρέσεων; Η κλάση log υπάρχει γι αυτό ακριβώς το σκοπό, περιλαμβάνει έναν αριθμό από διαφορετικές μεθόδους αναλόγως με τον τύπο του σφάλματος που ψάχνουμε. Μερικές από τις μεθόδους της log είναι οι παρακάτω<sup>5</sup>:

Log.e( ): error

Log.w( ): warn

Log.d( ): debug

Log.i( ): info

---

<sup>5</sup> Mike Wolfson, 2013, Android Developer Tools Essentials, 1st Edition, O'Reilly Media

Log.v( ): verbose

Log.wtf( ): failure

#### 2.1.3.4 Dalvik Debug Monitor Server ( DDMS)

Το Android συνοδεύεται μεταξύ των άλλων, από ένα εργαλείο αποσφαλμάτωσης το οποίο ονομάζεται Dalvik Debug Monitor Server (DDMS), και το οποίο περιέχει:

- Υπηρεσίες προώθησης θυρών (port forwarding)
- Λήψη εικόνας της επιφάνειας εργασίας της συσκευής
- Πληροφορίες για τις διεργασίες και τα νήματα (threads) της συσκευής
- Το εργαλείο logcat
- Πληροφορίες δικτύων και εισερχόμενων κλήσεων
- Δημιουργία ψευδών SMS και πληροφοριών τοποθεσίας και άλλα πολλά

Το DDMS είναι ενσωματωμένο στο Eclipse και επίσης συμπεριλαμβάνεται στο Android SDK. Δουλεύει κανονικά είτε είναι συνδεδεμένο σε εικονική συσκευή μέσω του emulator είτε είναι συνδεδεμένο σε πραγματική συσκευή. Από προεπιλογή αν είναι συνδεδεμένες δύο συσκευές ταυτόχρονα και η μία είναι εικονική, τότε αυτό θα επιλέξει σαν προεπιλεγμένη την εικονική.

Κατά την εκκίνησή του το DDMS συνδέεται στο ADB. Αφού συνδεθεί κάποια συσκευή στον υπολογιστή μέσω ADB, αυτόματα δημιουργείται μια υπηρεσία παρακολούθησης μεταξύ του DDMS και του ADB η οποία ειδοποιεί το DDMS τότε ξεκινάει και τότε σταματάει η λειτουργία μιας εικονικής μηχανής. Όταν η εικονική μηχανή (VM) βρίσκεται σε λειτουργία το DDMS παίρνει το process ID της και μέσω του ADB δημιουργεί μια σύνδεση με τον debugger της εικονικής μηχανής, μέσω του adb daemon.

Το DDMS αναθέτει μια μοναδική θύρα επικοινωνίας σε κάθε εικονική μηχανή στην συσκευή ή στις συσκευές τις οποίες είναι συνδεδεμένο. Η ανάθεση ξεκινά από τη θύρα 8600 για την πρώτη εικονική μηχανή και



συνεχίζει για όσες εικονικές μηχανές τρέχουν ταυτόχρονα στη συσκευή μας. Ισχύει ότι κάθε εικονική μηχανή έχει μια θύρα επικοινωνίας για debugging αλλά το DDMS μπορεί να ακούσει σε πολλές θύρες ταυτόχρονα ώστε να λάβει δεδομένα από παντού.

Το DDMS είναι ίσως το χρησιμότερο από όλα τα debugging tools μιας και ενσωματώνει από τα πιο απλά (πχ logcat) μέχρι εξειδικευμένα εργαλεία debugging του Android το network statistics και απευθύνεται κυρίως σε πιο έμπειρους developers οι οποίοι μπορούν να αξιολογήσουν αποτελεσματικότερα τις ενδείξεις που επιστρέφει το εργαλείο.

#### **2.1.3.5 Application Crash Reporter for Android (ACRA)**

Η συλλογή των debugging logs, είναι μια πολύ απλή διαδικασία η οποία πραγματοποιείται πολύ γρήγορα εφόσον συνδέσουμε την συσκευή μας μέσω του Android Debug Bridge. Τι γίνεται όμως όταν θέλουμε να συλλέξουμε τα logs μιας συσκευής χρήστη η οποία εμφάνισε κάποιο σφάλμα;

Οι μηχανικοί του Android έχουν μεριμνήσει γι αυτό και γι αυτό έχουν ενσωματώσει στις εφαρμογές που είναι συνδεδεμένες με το marketplace (έκδοση Android 2.2 και πάνω) έναν μηχανισμό απομακρυσμένης συλλογής logs. Τα stack traces των logs γίνονται προσβάσιμα μέσω της developer console του developer στο Google Play, και μέσω αυτών ο προγραμματιστής μπορεί να εντοπίσει το σφάλμα στο κώδικα του και να φροντίσει να το διορθώσει το συντομότερο δυνατό.

Τι γίνεται όμως στην περίπτωση που κάποιος developer δεν έχει κάνει ακόμη διαθέσιμη την εφαρμογή του στο κοινό, μέσω του Google Play; Το κενό λοιπόν αυτό έρχονται να καλύψουν κάποιες βιβλιοθήκες οι οποίες έχουν αναπτυχθεί με τα εργαλεία του Android SDK και σκοπός τους είναι ακριβώς αυτός: η συλλογή και αποστολή των stack traces από το κινητό του χρήστη σε εμάς για περαιτέρω ανάλυση. Ένα παράδειγμα τέτοιας βιβλιοθήκης η οποία χρησιμοποιείται από πολλές δημοφιλείς εφαρμογές είναι η βιβλιοθήκη ACRA.

Η ACRA είναι μια βιβλιοθήκη του Android η οποία αυτόματα συλλέγει τα δεδομένα των αναφορών σφάλματος και μας τα αποστέλλει σε μορφή εγγράφου GoogleDoc. Η βιβλιοθήκη είναι open source και φιλοξενείται στη σελίδα [code.google.com/p/acra/](http://code.google.com/p/acra/).

Τα χαρακτηριστικά της βιβλιοθήκης είναι τα εξής:

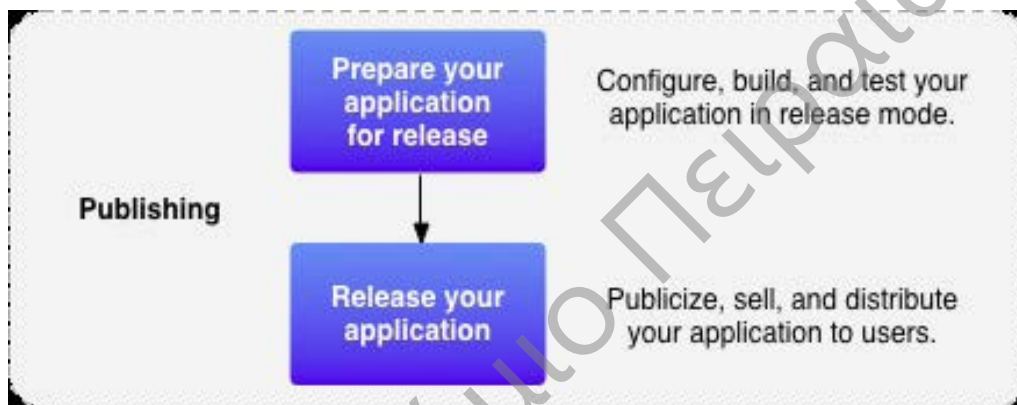
- Δυνατότητα παραμετροποίησης ειδοποιήσεων στη συσκευή του χρήστη (silent report, ειδοποίηση toast, ειδοποίηση στο Notification Bar και παράθυρο διαλόγου)
- Είναι δυνατόν να χρησιμοποιηθεί σε όλες τις εκδόσεις του Android
- Λεπτομερείς πληροφορίες για τη συσκευή στην οποί συνέβη το σφάλμα λειτουργίας, πολύ περισσότερες από αυτές που δίνει η επίσημη βιβλιοθήκη.
- Μπορούμε να προσθέσουμε και τις δικές μας μεταβλητές περιεχομένου στις αναφορές που στέλνει η βιβλιοθήκη.
- Είναι εφικτή η αποστολή αναφορών χωρίς να έχει συμβεί κάποιο σφάλμα λειτουργίας της εφαρμογής.
- Δουλεύει σε όλες τις εφαρμογές είτε αυτές έχουν κυκλοφορήσει μέσω του Google Play, είτε βρίσκονται σε beta testing οπότε η βιβλιοθήκη της Google δεν είναι διαθέσιμη.
- Μπορούμε να επιλέξουμε εκτός από κάποιο έγγραφο GoogleDoc η ACRA να μας στέλνει απευθείας τις αναφορές σε κάποιον http server, είτε να μας τις στέλνει απευθείας μέσω email.
- Τα reports που έχουμε ρυθμίσει να αποθηκεύονται σε κάποιο έγγραφο GoogleDoc μπορεί να γίνει προσβάσιμο άμεσα από μια ομάδα developers ώστε να αντιμετωπισθεί γρηγορότερα το σφάλμα.

#### **2.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό**

Στο τέταρτο και τελευταίο στάδιο της ανάπτυξης ο προγραμματιστής έχει να κάνει μερικές τελευταίες κινήσεις. Πρώτον πρέπει να έχει διορθώσει όλα τα

σφάλματα της εφαρμογής που προέκυψαν από τη διαδικασία της αποσφαλμάτωσης, να κάνει τις τελευταίες ρυθμίσεις και να κάνει το τελικό compile της εφαρμογής σε κανονική λειτουργία αυτή τη φορά και όχι debug.

Στη συνέχεια ακολουθεί η διάθεση της εφαρμογής με το μέσο της επιλογής του developer. Μπορεί να την διαθέσει στο Google Play, αφού πρώτα κάνει λογαριασμό ή να την διαθέσει σε κάποιο εναλλακτικό market όπως το marketplace της Amazon. Μπορεί κατά τη δημοσίευση σε οποιοδήποτε να ορίσει τιμή πώλησης ή να την διαθέσει δωρεάν.



Τέταρτο βήμα – Δημοσίευση της εφαρμογής

### 3.1 Project

Εφαρμογή εύρεσης WiFi δικτύων στην πλατφόρμα Android, η οποία θα είναι σε θέση να παρέχει τις παρακάτω πληροφορίες για το δίκτυο Wi-Fi:

1. Λίστα με τα διαθέσιμα Wi-Fi δίκτυα όπου για το καθένα θα αναφέρεται το όνομα του δικτύου (SSID), η συχνότητα λειτουργίας και η ισχύς του σήματος (RSSI)
2. Γραφική απεικόνιση των διαθέσιμων δικτύων, όπου θα εμφανίζεται το όνομα κάθε δικτύου και το κανάλι που καταλαμβάνει (άξονας x: ισχύς και άξονας y: συχνότητα μεταφρασμένη σε ένα από τα κανάλια του Wi-Fi)
3. Να προτείνει στον χρήστη το καλύτερο Wi-Fi δίκτυο (tradeoff μεταξύ ισχύς και παρεμβολής – λιγότερα δίκτυα σε μια συχνότητα –).

Οι παραπάνω επιλογές θα παρέχονται στο χρήστη μέσα από κατάλληλο γραφικό περιβάλλον.

Ο κώδικας με τα αντίστοιχα σχόλια φαίνεται παρακάτω:

```
1 package anthi.java.utilities;
2
3
4 /***** import std java libs *****/
5 import java.util.EmptyStackException;
6 import java.util.List;
7 import java.lang.String;
8 import java.util.ArrayList;
9
10 /***** import android java libs *****/
11 import android.app.Activity;
12 import android.os.Bundle;
13 import android.widget.TextView;
14 import android.widget.Toast;
15 import android.net.wifi.WifiManager;
16 import android.content.Context;
```

```
16 import android.content.Intent;
17 import android.graphics.Color;
18 import android.graphics.Typeface;
19 import android.net.wifi.ScanResult;
20 import android.view.View;
21 import android.view.View.OnClickListener;
22 import android.widget.Button;
23
24 /*
25 * WARNING !!!
26 * Running on an emulator may cause an uncaught NullPointerException
27 * List<ScanResults> = NULL
28 */
29 public class WiFiAnthi extends Activity implements OnClickListener {
30 /*
31 * WiFiAnthi inherits Activity
32 * WiFiAnthi inherits OnClickListener (interface class)
33 */
34 /*
35 * member declaration
36 */
37 WifiManager wifimanager;
38 TextView textStatus;
39 Button buttonScan;
40 Button reportNetwks;
41 Button plotNetwks;
42 Button chooseNetwks;
43 List<ScanResult> configs;
44
45 /** Called when the activity is first created. */
46 @Override /*
47 * override base onCreate() function.
```

```

48 * The onCreate() method is called by the Android system when WiFiAnthi
starts
49 */
50 public void onCreate(Bundle savedInstanceState) {
51     super.onCreate(savedInstanceState);
52     /*
53     * Instead of passing setContentView() a View object, we give it a reference
to the layout resource.
54     * The resource is identified as R.layout.main, which is actually a compiled
object representation
55     * of the layout defined in /res/layout/main.xml. The Eclipse plugin
automatically creates this reference inside the project's R.java class.
56     */
57     /*
58     setContentView(R.layout.main);
59     */
60     * text to appear on welcome; before any interaction is ignited
61     */
62     String text = "\nWelcome to myWifi Application!\n";
63     /*
64     * Setup all declared member variables.
65     * All UI connected members (buttons, textview) are already
66     * placed in the main.xml file
67     */
68     wifimanager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
69     textStatus = (TextView) findViewById(R.id.textStatus);
70     buttonScan = (Button) findViewById(R.id.buttonScan);
71     buttonScan.setOnClickListener(this);
72     reportNetwks = (Button) findViewById(R.id.reportNetwks);
73     reportNetwks.setOnClickListener(this);
74     plotNetwks = (Button) findViewById(R.id.plotNetwks);
75     chooseNetwks = (Button) findViewById(R.id.chooseNetwks);
76     chooseNetwks.setOnClickListener(this);
77     */

```

```

78 * this button will cause another View to appear on
79 * main screen; namely a WifiPlotter object
80 */
81 plotNetwks.setOnClickListener( new View.OnClickListener()
82 { public void onClick(View arg0)
83 { Intent i = new Intent(WiFiAnthi.this, WifiPlotter.class);
84 startActivity(i); } });
85 /*
86 * get current state and basic information;
87 * scan for wifi network points
88 */
89 boolean scanstart = wifimanager.startScan();
90 text = text + "\nWiFi : starting network scan ...";
91 if (scanstart)
92 text+="ok";
93 else
94 text+="oops! scan not started";
95 try {
96 /*
97 * getScanResults() may cause an exception to be thrown when run on an
emulator!
98 * fill list(configs) with network connections found via the scan (getScanRe
sults).
99 * wifimanager calls the scan function
100 */
101 int numofnetworks = 0;
102 configs = wifimanager.getScanResults();
103 for (int i=0; i<configs.size(); i++) {
104 //text = text + "\n+1 network!";
105 ++numofnetworks;
106 }
107 text+="\nNumber of wifi networks found: "+numofnetworks;
108 } catch (NullPointerException e) {

```

```

109 text = text + "\nexception caught! No scan results available\n";
110 }
111 /*
112 * report information collected (and saved in text variable)
113 * on screen; format text.
114 */
115 textStatus.setText(text);
116 textStatus.setTextColor(Color.LTGRAY);
117 textStatus.setTypeface(null, Typeface.BOLD);
118 }// -- end of onCreate() --
119
120 public void clearText()
121 {
122 textStatus.setText("");
123 }
124
125 public String reportWifiNetworks()
126 {
127 /*
128 * Function to Scan for Wifi network points and report them;
129 * reports: network SSID, frequency and level in dB.
130 * returns a String, which is in turn shown on screen.
131 */
132 String information="\n";
133 if (configs.size()==0)/*this line will throw a NullPointerException when
running on emulator
*/
134 throw new EmptyStackException();
135 int counter = 1;
136 for (ScanResult config : configs) {
137 information = information + "Net. "+(counter++) + " SSID: "+config.SSID+"
FREQ: "+config.frequency+ " LEVEL: "+config.level + "\n";
138 }

```



```

139 return information;
140 }
141
142 public List<ScanResult> chooseOptimal()
143 {
144 /*
145 * Function to choose optimal network list from the set of networks saved
in List<ScanResult> configs.
146 * Two new lists are created; one holding (seperate) frequencies from all
ScanResults, and a second
147 * holding the number of times they occur. Depending on how many
networks operate on the same
148 * frequency, the ScanResults are ordered in a new list (orderedList),
which is returned to calling function.
149 */
150 if (configs.size()==0)/*this line will throw a NullPointerException when
running on emulator
*/
151 throw new EmptyStackException();
152
153 List<Integer> freqHolder = new ArrayList<Integer>();
154 List<Integer> countHolder = new ArrayList<Integer>();
155 List<ScanResult> orderedList = new ArrayList<ScanResult>();
156 for (int i=0; i<configs.size(); i++) {
157 int currentFrequency = configs.get(i).frequency;
158 if (freqHolder.contains(currentFrequency)==false) {
159 freqHolder.add(currentFrequency);
160 countHolder.add(1);
161 } else {
162 int index = freqHolder.indexOf(currentFrequency);
163 countHolder.set(index, countHolder.get(index)+1);
164 }
165 }

```

```

166 while (countHolder.size() != 0) {
167     int minOccurrences = countHolder.size()+1000;
168     int minIndex = 100;
169     for (int i=0; i<countHolder.size(); i++) {
170         if (countHolder.get(i)<minOccurrences) {
171             minOccurrences = freqHolder.get(i);
172             minIndex = i;
173         }
174     }
175     for (int i=0; i<configs.size(); i++)
176         if (configs.get(i).frequency == minOccurrences){
177             orderedList.add(configs.get(i));
178         }
179     if (countHolder.size()==0)
180         break;
181     else {
182         countHolder.remove(minIndex);
183         freqHolder.remove(minIndex);
184     }
185     if (countHolder.size()==0)
186         break;
187 }
188 return orderedList;
189 }
190 public void throwToast()
191 {
192     /*
193     * throws a toast message on screen, informing
194     * the user that no valid connections are found
195     */
196     Context context = getApplicationContext();
197     CharSequence text = "\nNo WiFi Connections found!\n";
198     int duration = Toast.LENGTH_SHORT;
199     Toast toast = Toast.makeText(context, text, duration);

```

```

200 toast.show();
201 }
202 /*
203 * override base onClick method
204 * a View object is passed as input
205 * i.e. one of the classe's Buttons
206 */
207 @Override
208 public void onClick(View v) {
209     switch (v.getId()) {
210         case R.id.reportNetwks:
211             /*
212              * print WiFi network list, and respective info.
213              */
214             configs.clear();
215             clearText();
216             configs = wifimanager.getScanResults();
217             textStatus.setText("\nAvailable WiFi Networks: " + configs.size());
218             try {
219                 textStatus.append(reportWifiNetworks());
220             } catch (EmptyStackException exc) {
221                 throwToast();
222             }
223             break;
224         case R.id.chooseNetwks:
225             /*
226              * print WiFi network list, in ascending availability order.
227              */
228             clearText();
229             configs.clear();
230             configs = wifimanager.getScanResults();
231             textStatus.append("\nNetwork Availability (in ascending order):\n");
232             try {

```

```

233 List<ScanResult> ordered = chooseOptimal();
234 for (int i=0; i<ordered.size(); i++)
235 textStatus.append("\nNet. id: "+ordered.get(i).SSID);
236 } catch (EmptyStackException exc) {
237 throwToast();
238 }
239 break;
240 case R.id.buttonScan:
241 /*
242 * scan WiFi network points.
243 */
244 clearText();
245 textStatus.setText("\nRefreshing WiFi Networks...");
246 configs = wifimanager.getScanResults();
247 try {
248 if (configs.size()== 0)
249 textStatus.append("\nNO WiFi NETWORKS FOUND (empty list)!");
250 } catch (NullPointerException e) {//see comment (WARNING) on line 28
251 throwToast();
252 }
253 textStatus.append("done!");
254 break;
255 }
256 }
257 }

```

Για την γραφική απεικόνιση των διαθέσιμων δικτύων χρησιμοποιείται μια δεύτερη class, η wifiplotter. Η οποία χρησιμοποιεί δικό της layout, το οποίο ορίζεται στο plot.xml.

Για την επικοινωνία των δύο διαφορετικών layout έχει προστεθεί στο AndroidManifest.xml η παρακάτω γραμμή κωδικά:

```
<activity android:name="WifiPlotter" android:label="Network Plot">
```

Το layout του wifiplotter εμφανίζεται στην κυρία οθόνη του project με την επιλογή του Plot.

Ο κώδικας του wifiplotter με τα αντίστοιχα σχόλια φαίνεται παρακάτω:

```
1 package anthi.java.utilities;
2
3 /***** import std java libs *****/
4 import java.util.EmptyStackException;
5 import java.util.List;
6 import java.lang.String;
7 import java.util.ArrayList;
8 import java.util.Arrays;
9 import java.lang.Number;
10 import android.graphics.Color;
11
12 /***** import android java libs *****/
13 import android.app.Activity;
14 import android.content.Context;
15 import android.os.Bundle;
16 import android.view.View;
17 import android.widget.Button;
18 import android.net.wifi.ScanResult;
19 import android.net.wifi.WifiManager;
20
21 /***** import androidplot java libs *****/
22 import com.androidplot.xy.BoundaryMode;
23 import com.androidplot.xy.LineAndPointFormatter;
24 import com.androidplot.xy.SimpleXYSeries;
25 import com.androidplot.xy.XYPlot;
26 import com.androidplot.xy.XYStepMode;
27 import com.androidplot.series.XYSeries;
28
29 public class WifiPlotter extends Activity {
```

```

30 XYPlot networkPlot;
31 WifiManager wifimanager;
32 List<ScanResult> configs;
33
34 /** Called when the activity is first created. */
35 @Override
36 public void onCreate(Bundle savedInstanceState) {
37     super.onCreate(savedInstanceState);
38     setContentView(R.layout.plot);
39 /*
40 * this button will restore the main layout of WiFiAnthi
41 */
42 Button b = (Button) findViewById(R.id.close);
43 b.setOnClickListener(new View.OnClickListener() {
44     public void onClick(View arg0) {
45         setResult(RESULT_OK);
46         finish();
47     }
48 });
49
50 wifimanager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
51 configs = wifimanager.getScanResults();
52
53 if (configs.size()==0)/*this line will throw a NullPointerException when
running on emulator
*/
54 throw new EmptyStackException();
55 Number[] freqs = new Number[configs.size()];
56 Number[] levels = new Number[configs.size()];
57 List<String> ssidHolder = new ArrayList<String>();
58 Number maxchanel, minchanel;
59 Number maxlevel, minlevel;
60 for (int i=0; i<configs.size(); i++) {

```

```

61 freqs[i] = configs.get(i).frequency;
62 freqs[i] = frequency2Channel(freqs[i].intValue());
63 levels[i] = configs.get(i).level;
64 ssidHolder.add(configs.get(i).SSID);
65 }
66 maxchannel = minchannel = freqs[0];
67 maxlevel = minlevel = levels[0];
68 for (int i=1; i<configs.size(); i++) {
69 if (freqs[i].intValue()>maxchannel.intValue())
70 maxchannel=freqs[i];
71 if (freqs[i].intValue()<minchannel.intValue())
72 minchannel=freqs[i];
73 if (levels[i].intValue()>maxlevel.intValue())
74 maxlevel=levels[i];
75 if (levels[i].intValue()<minlevel.intValue())
76 minlevel=levels[i];
77 }
78 minchannel= minchannel.intValue()-1;
79 maxchannel= maxchannel.intValue()+1;
80 minlevel= minlevel.intValue()-1;
81 maxlevel= maxlevel.intValue()+1;
82 networkPlot = (XYPlot) findViewById(R.id.networkPlot);
83 XYSeries series = new SimpleXYSeries(Arrays.asList(levels),
Arrays.asList(freqs), "Network Plot");
84 LineAndPointFormatter seriesFormat = new
LineAndPointFormatter(Color.rgb(0, 200, 0), Color.rgb(0, 100, 0),
Color.rgb(150, 190, 150));
85 networkPlot.addSeries(series, seriesFormat);
86 networkPlot.disableAllMarkup();
87 networkPlot.setDomainStep(XYStepMode.SUBDIVIDE, 5);
88 networkPlot.setRangeStep(XYStepMode.SUBDIVIDE, 5);
89 networkPlot.setDomainLabel("level in dB");
90 networkPlot.setRangeLabel("channel");

```

```
91 networkPlot.getDomainLabelWidget().pack();
92 networkPlot.getRangeLabelWidget().pack();
93 networkPlot.setRangeBoundaries(minchannel, maxchannel,
BoundaryMode.FIXED);
94 networkPlot.setDomainBoundaries(minlevel, maxlevel,
BoundaryMode.FIXED);
95 }
96 Number frequency2Channel(int freq) {
97 /*
98 * reference http://en.wikipedia.org/wiki/List\_of\_WLAN\_channels
99 */
100 switch (freq) {
101 case 2412:
102 return 1;
103 case 2417:
104 return 2;
105 case 2422:
106 return 3;
107 case 2427:
108 return 4;
109 case 2432:
110 return 5;
111 case 2437:
112 return 6;
113 case 2442:
114 return 7;
115 case 2447:
116 return 8;
117 case 2452:
118 return 9;
119 case 2457:
120 return 10;
121 case 2462:
```



```
122 return 11;
123 case 2467:
124 return 12;
125 case 2472:
126 return 13;
127 case 2484:
128 return 14;
129 default:
130 return 0;
131 }
132 }
133 }
```

### 3.2 Περιγραφή του project

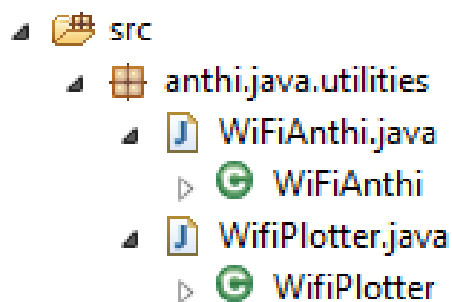
Το project περιλαμβάνει τους ακόλουθους φακέλους

- src
- gen
- assets
- bin
- lib
- res
- Τις απαραίτητες android βιβλιοθήκες
- Τα αρχεία
  - AndroidManifest.xml
  - proguard.cfg
  - project.properties

Στις επόμενες παραγράφους περιγράφονται τα σημαντικότερα από αυτά.

### 3.2.1 Φάκελος src

Στο φάκελο αυτό περιέχονται δύο java αρχεία που υλοποιούν την λειτουργικότητα της εφαρμογής.



Εικόνα 1: Φάκελος src

Κάθε αρχείο αντιστοιχεί σε ένα activity. Κάθε εφαρμογή android αποτελείται από ένα τουλάχιστον activity. Activity είναι εκείνο το στοιχείο της εφαρμογής που υποστηρίζει την διεπαφή του χρήστη. Πάντα ένα από αυτά είναι το κύριο activity. Όταν ένα Activity εκκινεί, προκαλεί τον τερματισμό εκτέλεσης του προηγούμενου. Υλοποιούνται με java κλάσεις που επεκτείνουν την κλάση Activity. Κάθε activity διαμορφώνει την παρουσίαση του βασισμένο σε ένα xml αρχείο που είναι αποθηκευμένο στον φάκελο res/layout. Επίσης η ύπαρξη του στην εφαρμογή δηλώνεται στο αρχείο AndroidManifest.xml. Τα events που μπορούν να προγραμματιστούν σε ένα activity είναι:

- onCreate: Εκτελεί τις καθορισμένες εντολές όταν το Activity ξεκινάει.
- onStart: Εκτελεί τις καθορισμένες εντολές όταν το Activity γίνεται ορατό στον χρήστη.
- onResume: Εκτελεί τις καθορισμένες εντολές όταν το Activity χρησιμοποιείται από τον χρήστη.
- onPause: Εκτελεί τις καθορισμένες εντολές όταν το μέχρι εκείνη την στιγμή εκτελούμενο έχει σταματήσει προσωρινά και ένα activity που εκτελούνταν σε προηγούμενο χρόνο ξεκινάει πάλι.

- `onStop`: Εκτελεί τις καθορισμένες εντολές όταν παύει να είναι ορατό από τον τελικό χρήστη.
- `onDestroy`: Εκτελεί τις καθορισμένες εντολές όταν το `activity` έχει διαγραφεί από την μνήμη του συστήματος.
- `onRestart`: Εκτελεί τις καθορισμένες εντολές όταν ένα `activity` που έχει σταματήσει, εκκινεί πάλι.

### 3.2.2 WifiAnthi

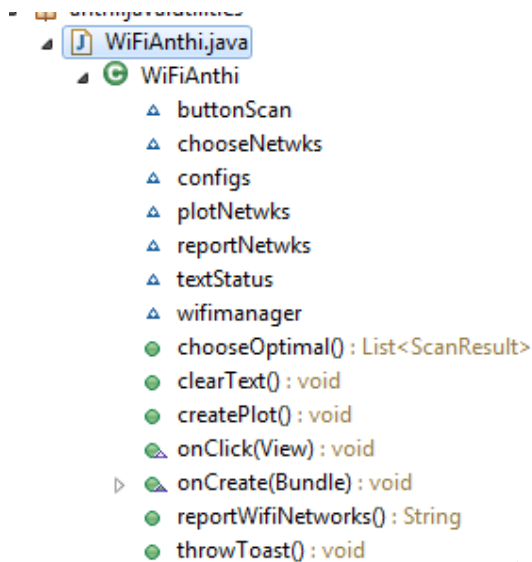
Η διεπαφή που αντιστοιχεί στην κλάση `WifiAnthi` ακολουθεί το `layout` που περιγράφεται από το αρχείο `res/layout/main.xml`. Αρχικοποιούνται τα στοιχεία της διεπαφής (`textboxes`, `buttons`). Το `button plotNetwks` προγραμματίζεται έτσι ώστε με κλικ να ανοίγει την οθόνη `WifiPlotter`.

Εκκινεί διαδικασία αναζήτησης ασυρμάτων δικτύων με την μέθοδο `startScan` του αντικειμένου `wifimanager`. Δημιουργείται μήνυμα επιτυχίας ή αποτυχίας, ανάλογα με το αποτέλεσμα που επιστρέφει η μέθοδος. Αν βρεθούν ασύρματα δίκτυα τότε στο μήνυμα προστίθεται και το πλήθος τους.

Στην συνέχεια ορίζονται οι ακόλουθες μέθοδοι:

- `clearText()`: Αρχικοποιεί το `textbox` με όνομα `textStatus`.
- `reportWifiNetworks()`: Συγκεντρώνει σε ένα `string` τα στοιχεία των `wifi` δικτύων που βρέθηκαν με την αναζήτηση και το επιστρέφει.
- `chooseOptimal()`: Ταξινομεί τα δίκτυα σε λίστα ανάλογα με το πόσα εμφανίζεται να λειτουργούν στην ίδια συχνότητα.
- `createPlot()`: Παρουσιάζει σχηματικά τα εντοπισμένα δίκτυα.
- `throwToast()`: Εμφανίζει προκαθορισμένο συγκεκριμένο μήνυμα.
- `onClick(View v)`: Εντοπίζει σε ποιο πλήκτρο γίνεται κλικ κάθε φορά και ανάλογα εκτελεί ενέργειες. Έτσι:
  - Αν πιεστεί το `button` με `id reportNetwks` τότε εμφανίζει πληροφορίες για τα εντοπισμένα δίκτυα ενώ τρέχει και την μέθοδο `reportWifiNetworks()`

- Αν πιεστεί το button με id chooseNetwks τότε εμφανίζονται τα εντοπισμένα δίκτυα μετά την κλήση της μεθόδου chooseOptimal().
- Αν πιεστεί το button με id buttonScan τότε γίνεται μία νέα αναζήτηση δικτύων και εμφανίζεται αντίστοιχο με το αποτέλεσμα μήνυμα.



Εικόνα 2: WifiAnthi.java tree

### 3.2.2.1 Scan networks

Στην παράγραφο αυτή θα περιγράψουμε τις δράσεις που λαμβάνουν χώρα όταν πατηθεί το κουμπί scan. Ο σχετικός κώδικας είναι

```

case R.id.buttonScan:
    /*
     * scan WiFi network points.
     */
    clearText();
    textStatus.setText("\nRefreshing WiFi Networks...");
    configs = wifimanager.getScanResults();
    try {
        if (configs.size()== 0)
            textStatus.append("\nNO WiFi NETWORKS FOUND (empty list)!");
    } catch (NullPointerException e) { //see comment (WARNING) on line 28
        throwToast();
    }
    textStatus.append("done!");
    break;

```

Αρχικά λοιπόν καθαρίζεται η text area από τυχόν προηγούμενες εγγραφές. Στη συνέχεια καλείται από τις βιβλιοθήκες του android η μέθοδος `getScanResults` (κλάση `WifiManager`), η οποία και επιστρέφει μια λίστα από αντικείμενα `ScanResult`, δηλαδή πρακτικά πληροφορίες για όλα τα διαθέσιμα ασύρματα δίκτυα. Εάν δε βρεθεί κανένα διαθέσιμο δίκτυο, τότε εμφανίζεται κατάλληλο μήνυμα στο χρήστη, καλώντας τη συνάρτηση `throwToast()`.

```
public void throwToast()
{
    /*
     * throws a toast message on screen, informing
     * the user that no valid connections are found
     */
    Context context = getApplicationContext();
    CharSequence text = "\nNo WiFi Connections found!\n";
    int duration = Toast.LENGTH_SHORT;
    Toast toast = Toast.makeText(context, text, duration);
    toast.show();
}
```

### 3.2.2.2 Report networks

Στην παράγραφο αυτή θα περιγράψουμε τις δράσεις που λαμβάνουν χώρα όταν πατηθεί το κουμπί report. Ο σχετικός κώδικας είναι

```
case R.id.reportNetworks:
/*
 * print WiFi network list, and respective info.
 */
    configs.clear();
    clearText();
    configs = wifimanager.getScanResults();
    textStatus.setText("\nAvailable WiFi Networks: " + configs.size());
    try {
        textStatus.append(reportWifiNetworks());
    } catch (EmptyStackException exc) {
        throwToast();
    }
    break;
```

```
public String reportWifiNetworks()
{
    /*
     * Function to Scan for Wifi network points and report them;
     * reports: network SSID, frequency and level in dB.
     * returns a String, which is in turn shown on screen.
     */
}
```

```

    */
    String information="\n";
    if (configs.size()==0)/*this line will throw a NullPointerException when running
on emulator*/
        throw new EmptyStackException();
    int counter = 1;
    for (ScanResult config : configs) {
        information = information + "Net. "+(counter++) + " SSID:
"+config.SSID+ " FREQ: "+config.frequency+ " LEVEL: "+config.level + "\n";
    }
    return information;
}

```

Όπως και στην περίπτωση του κουμπιού scan, αρχικά καθαρίζεται η text area και καλείται η μέθοδος getScanResults (κλάση WifiManager), η οποία και επιστρέφει μια λίστα από αντικείμενα ScanResult. Στην περίπτωση αυτή όμως επιπλέον τυπώνεται ο αριθμός των δικτύων που εντοπίστηκαν, καθώς και συγκεκριμένες πληροφορίες για κάθε ένα από αυτά, δηλαδή όπως αυτό καθορίζεται από τη μέθοδο reportWifiNetworks: το μοναδικό αναγνωριστικό του δικτύου (SSID), η συχνότητα και το επίπεδο. Οι πληροφορίες αυτές προστίθενται στην text area που αναφέραμε προηγουμένως.

### 3.2.2.3 Choose networks

Στην παράγραφο αυτή θα περιγράψουμε τις δράσεις που λαμβάνουν χώρα όταν πατηθεί το κουμπί choose. Ο σχετικός κώδικας είναι

```

case R.id.chooseNetwks:
    /*
    * print WiFi network list, in ascending availability order.
    */
    clearText();
    configs.clear();
    configs = wifimanager.getScanResults();
    textStatus.append("\nNetwork Availability (in ascending order):\n");
    try {
        List<ScanResult> ordered = chooseOptimal();
        for (int i=0; i<ordered.size(); i++)
            textStatus.append("\nNet. id: "+ordered.get(i).SSID);
    } catch (EmptyStackException exc) {
        throwToast();
    }
    break;

```

```

    public List<ScanResult> chooseOptimal()
    {
        /*
         * Function to choose optimal network list from the set of networks saved in
         List<ScanResult> configs.
         * Two new lists are created; one holding (seperate) frequencies from all
         ScanResults, and a second
         * holding the number of times they occur. Depending on how many networks
         operate on the same
         * frequency, the ScanResults are ordered in a new list (orderedList), which is
         returned to calling function.
         */
        if (configs.size()==0)/*this line will throw a NullPointerException when running
        on emulator*/
            throw new EmptyStackException();

        List<Integer> freqHolder    = new ArrayList<Integer>();
        List<Integer> countHolder  = new ArrayList<Integer>();
        List<ScanResult> orderedList = new ArrayList<ScanResult>();
        for (int i=0; i<configs.size(); i++) {
            int currentFrequency = configs.get(i).frequency;
            if (freqHolder.contains(currentFrequency)==false) {
                freqHolder.add(currentFrequency);
                countHolder.add(1);
            } else {
                int index = freqHolder.indexOf(currentFrequency);
                countHolder.set(index, countHolder.get(index)+1);
            }
        }
        while (countHolder.size() != 0) {
            int minOccurances = countHolder.size()+1000;
            int minIndex     = 100;
            for (int i=0; i<countHolder.size(); i++) {
                if (countHolder.get(i)<minOccurances) {
                    minOccurances = freqHolder.get(i);
                    minIndex     = i;
                }
            }
            for (int i=0; i<configs.size(); i++)
                if (configs.get(i).frequency == minOccurances){
                    orderedList.add(configs.get(i));
                }
            if (countHolder.size()==0)
                break;
            else {
                countHolder.remove(minIndex);
                freqHolder.remove(minIndex);
            }
            if (countHolder.size()==0)
                break;
        }
        return orderedList;
    }
}

```

Όπως και προηγουμένως, αρχικά καθαρίζεται η text area και καλείται η μέθοδος getScanResults (κλάση WifiManager), η οποία και επιστρέφει μια λίστα από αντικείμενα ScanResult. Στην περίπτωση αυτή όμως επιπλέον καλείται η μέθοδος chooseOptimal, σκοπός της οποίας είναι να κάνει sort τα δίκτυα ανάλογα με τα ποιοτικά χαρακτηριστικά τους. Πιο συγκεκριμένα, αρχικά γεμίζουν οι λίστες freqHolder και countHolder με την τιμή της κάθε συχνότητας και τον αριθμό εμφάνισής της αντίστοιχα.

```
if (freqHolder.contains(currentFrequency)==false) {
    freqHolder.add(currentFrequency);
    countHolder.add(1);
} else {
    int index = freqHolder.indexOf(currentFrequency);
    countHolder.set(index, countHolder.get(index)+1);
}
```

Στη συνέχεια διατρέχεται η λίστα countHolder και εντοπίζονται σειριακά οι συχνότητες εκείνες με το μικρότερο αριθμό εμφανίσεων, οι οποίες και τοποθετούνται τελικά στη λίστα orderedList.

```
for (int i=0; i<countHolder.size(); i++) {
    if (countHolder.get(i)<minOccurances) {
        minOccurances = freqHolder.get(i);
        minIndex = i;
    }
}
for (int i=0; i<configs.size(); i++)
    if (configs.get(i).frequency == minOccurances){
        orderedList.add(configs.get(i));
    }
}
```

Τέλος προστίθενται στο κείμενο του text area τα μοναδικά αναγνωριστικά των δικτύων (SSID) διατρέχοντας σειριακά τη λίστα, κάτι που σημαίνει ότι εμφανίζονται πρώτα τα στοιχεία με τη μικρότερη συχνότητα εμφάνισης συχνότητας (αύξουσα σειρά).

```
for (int i=0; i<ordered.size(); i++)
    textStatus.append("\nNet. id: "+ordered.get(i).SSID);
```



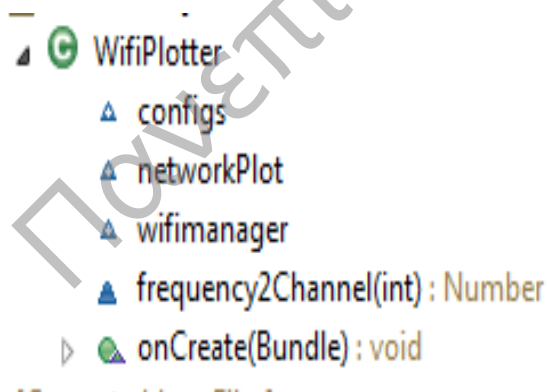
### 3.2.2.4 Plot networks

Το κουμπί plot δεν αποκτά handler όπως τα προηγούμενα κουμπιά, αλλά όταν πατηθεί πρακτικά ξεκινάει το δεύτερο activity, δηλαδή το WifiPlotter, και άρα φορτώνεται νέα οθόνη.

```
plotNetwks.setOnClickListener( new View.OnClickListener() {  
    public void onClick(View arg0) {  
        Intent i = new Intent(WiFiAnthi.this, WifiPlotter.class);  
        startActivity(i);  
    }  
});
```

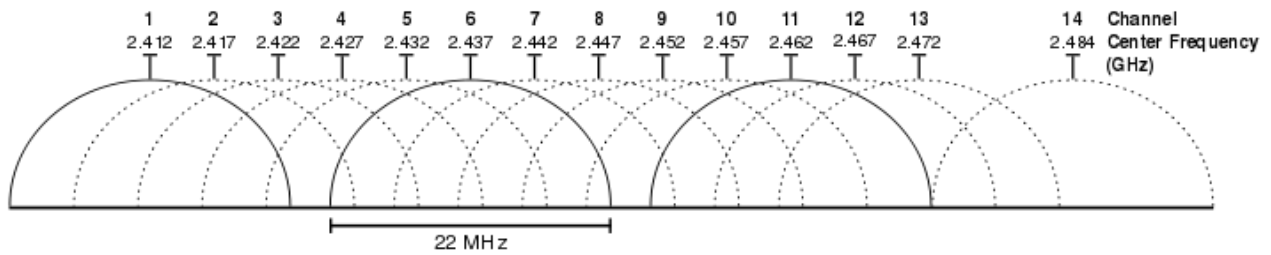
### 3.2.3 WifiPlotter

Η διεπαφή που αντιστοιχεί στην κλάση WifiPlotter ακολουθεί το layout που περιγράφεται από το αρχείο res/layout/plot.xml. Αρχικοποιούνται οι μεταβλητές που χρησιμοποιούνται στο activity, προγραμματίζεται το button με id close έτσι ώστε να κλείνει το activity με το κλικ. Στην συνέχεια ανιχνεύονται τα υπάρχοντα ασύρματα δίκτυα γίνεται επεξεργασία των ανακτηθέντων στοιχείων και διαμορφώνεται το στοιχείο του layout με id networkPlot το οποίο χρησιμοποιείται για να παραστήσει γραφικά τα στοιχεία των δικτύων που βρέθηκαν.



Εικόνα 3: WifiPlot.java tree

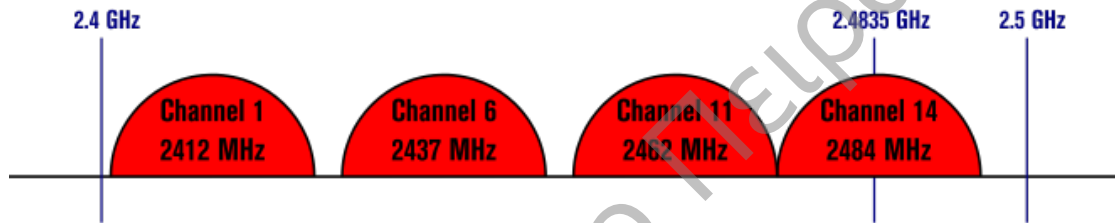
Η μέθοδος frequency2Channel μετατρέπει τις συχνότητες της μπάντας των 2,4 GHz (IEEE 802.11b/g/n) σε κανάλια.



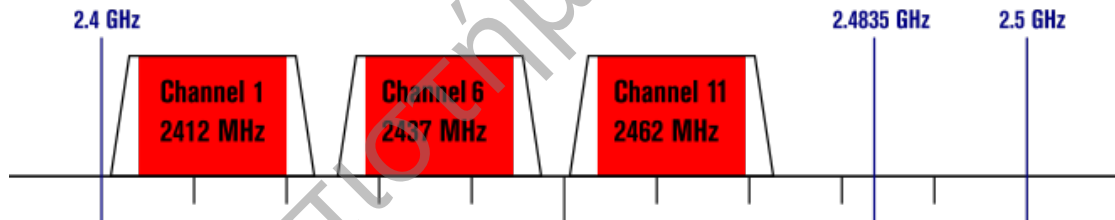
Εικόνα 6: Γραφική αναπαράσταση των 2,4 GHz band

## Non-Overlapping Channels for 2.4 GHz WLAN

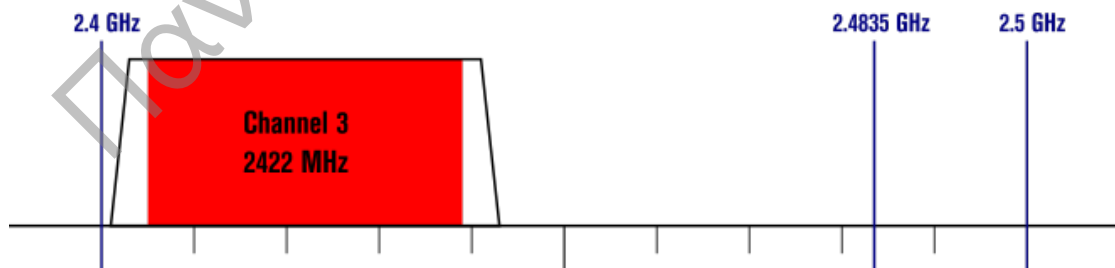
802.11b (DSSS) channel width 22 MHz



802.11g/n (OFDM) 20 MHz ch. width - 16.25 MHz used by sub-carriers



802.11n (OFDM) 40 MHz ch. width - 33.75 MHz used by sub-carriers



Εικόνα 7: Γραφική αναπαράσταση των μη επικαλυπτόμενων καναλιών για τα ασύρματα τοπικά δίκτυα των 2,4 GHz

### 3.2.3.1 Σχεδίαση γραφικής παράστασης

Αρχικά εντοπίζονται τα διαθέσιμα ασύρματα δίκτυα και για κάθε ένα από αυτά (αντικείμενα `ScanResult`) καταγράφονται πληροφορίες σε σχετικές λίστες. Πιο συγκεκριμένα, καταγράφονται η τιμή `level`, το μοναδικό αναγνωριστικό του δικτύου `ssid`, καθώς και το κανάλι. Για το τελευταίο χρησιμοποιείται η συνάρτηση `mapping frequency2Channel`, η οποία επιστρέφει τον αριθμό του καναλιού για την τιμή της συχνότητας που φέρει το δίκτυο.

```
for (int i=0; i<configs.size(); i++) {  
    freqs[i] = configs.get(i).frequency;  
    freqs[i] = frequency2Channel(freqs[i].intValue());  
    levels[i] = configs.get(i).level;  
    ssidHolder.add(configs.get(i).SSID);  
}
```

Το επόμενο βήμα είναι να βρεθούν οι μέγιστες και ελάχιστες τιμές του καναλιού (`channel`) και του επιπέδου (`level`) για όλα τα διαθέσιμα δίκτυα, ώστε να οριοθετηθούν κατάλληλα οι άξονες του γραφήματος. Οι επόμενες γραμμές δημιουργούν το γράφημα, όπου παρατηρούμε ότι στον άξονα X τοποθετείται η λίστα `levels`, ενώ στον άξονα Y η λίστα `freqs`, συνεπώς απεικονίζουμε τη συσχέτιση καναλιών με επίπεδο. Τέλος, ορίζεται ένας `LineAndPointFormatter` για την προσθήκη χρωμάτων στο γράφημα, ο οποίος δέχεται ως όρισμα τρεις ακεραίους αριθμούς που αντιστοιχούν σε χρώματα RGB. Οι τελευταίοι καθορίζουν το χρώμα της γραμμής, το χρώμα των εφέ και το χρώμα του γεμίσματος αντίστοιχα.

```
networkPlot = (XYPlot) findViewById(R.id.networkPlot);  
XYSeries series = new SimpleXYSeries(Arrays.asList(levels), Arrays.asList(freqs),  
"Network Plot");  
LineAndPointFormatter seriesFormat = new LineAndPointFormatter(Color.rgb(0,  
200, 0), Color.rgb(0, 100, 0), Color.rgb(150, 190, 150));  
networkPlot.addSeries(series, seriesFormat);
```

Αφού δημιουργηθεί το γράφημα, καθορίζονται επιπλέον λεπτομέρειες για τους δύο άξονες (που αναφέρονται ως `domain` και `range`), όπως:

- Το βήμα ανάμεσα στις απεικονιζόμενες τιμές
- Οι ετικέτες των αξόνων
- Τα όρια τιμών

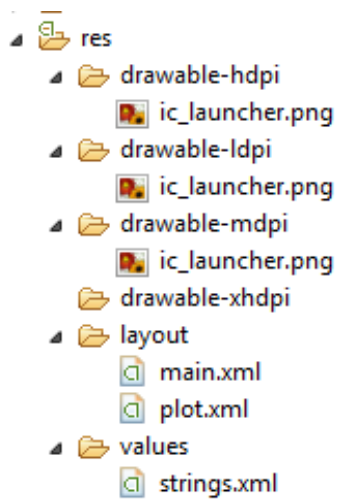
```

networkPlot.disableAllMarkup();
networkPlot.setDomainStep(XYStepMode.SUBDIVIDE, 5);
networkPlot.setRangeStep(XYStepMode.SUBDIVIDE, 5);
networkPlot.setDomainLabel("level in dB");
networkPlot.setRangeLabel("channel");
networkPlot.getDomainLabelWidget().pack();
networkPlot.getRangeLabelWidget().pack();
networkPlot.setRangeBoundaries(minchannel, maxchannel, BoundaryMode.FIXED);
networkPlot.setDomainBoundaries(minlevel, maxlevel, BoundaryMode.FIXED);

```

### 3.2.4 Φάκελος res(resource)

Περιλαμβάνει όλους τους πόρους που χρησιμοποιεί η εφαρμογή.



Εικόνα 8: Φάκελος res

Στους φακέλους drawable-xxxx περιλαμβάνεται εικόνα σε διάφορες εκδοχές που αντιστοιχούν στις δυνατότητες απεικόνισης της συσκευής – κυρίως σε μέγεθος εικόνας. Η συγκεκριμένη εικόνα είναι το εικονίδιο της εφαρμογής.

- drawable-hdpi: Υψηλές δυνατότητες απεικόνισης
- drawable-xhdpi: Πολύ υψηλές δυνατότητες απεικόνισης
- drawable-mdpi: Μέσες δυνατότητες απεικόνισης
- drawable-ldpi: Χαμηλές δυνατότητες απεικόνισης

Στον φάκελο layout περιλαμβάνονται οι περιγραφές των διεπαφών σε μορφή xml. Έτσι:

Στο αρχείο main.xml περιγράφεται η αρχική οθόνη της εφαρμογής. Τα στοιχεία ακολουθούν γραμμική διάταξη. Περιλαμβάνονται ένα textbox με id textStatus και περιεχόμενο το string με id Scan, ένα button με id buttonScan στο οποίο αναγράφεται το ίδιο string καθώς και ένα group στοιχείων τοποθετημένα σε διάταξη σχετικών θέσεων μεταξύ τους. Το group αυτό περιλαμβάνει τρία buttons τα οποία είναι τοποθετημένα το ένα δίπλα στο άλλο με id reportNetwks, plotNetwks, chooseNetwks και έχουν ετικέτες strings με id report, plot, choose αντίστοιχα. Είναι στοιχημμένα μεταξύ τους στο κάτω δεξιά μέρος της οθόνης.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<TextView
    android:id="@+id/textStatus"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/Scan">
</TextView>
<Button
    android:id="@+id/buttonScan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/Scan" >
</Button>
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:text="@string/report"
        android:id="@+id/reportNetwks"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true">
    </Button>
```

```

<Button
    android:text="@string/plot"
    android:id="@+id/plotNetwks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@id/reportNetwks">
</Button>
<Button
    android:text="@string/choose"
    android:id="@+id/chooseNetwks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@id/plotNetwks">
</Button>
</RelativeLayout>
</LinearLayout>

```

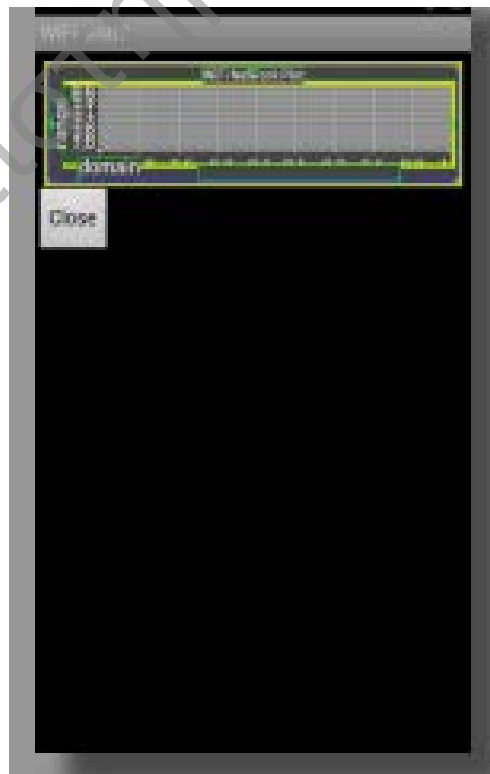


Εικόνα 9: main.xml

Στο αρχείο plot.xml περιγράφεται η οθόνη στην οποία αποδίδονται τα χαρακτηριστικά των δικτύων γραφικά. Περιέχονται δύο στοιχεία σε γραμμική διάταξη. Ένα στοιχείο απεικόνισης διαγραμμάτων και ένα button. Για να είναι δυνατή η χρησιμοποίηση του πρώτου χρειάστηκε να προστεθεί η βιβλιοθήκη

AndroidPlot. Το id του είναι networkPlot ενώ ρυθμίζονται οι διαστάσεις του και τα περιθώρια στην θέση που τοποθετείται. Το button έχει id close ενώ η ετικέτα του αποδίδεται απ' ευθείας ως «close».

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <com.androidplot.xy.XYPlot
        android:id="@+id/networkPlot"
        android:layout_width="fill_parent"
        android:layout_height="130px"
        android:layout_marginTop="10px"
        android:layout_marginLeft="10px"
        android:layout_marginRight="10px"
        title="WiFi Network Plot"/>
    <Button
        android:id="@+id/close"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Close"
    />
</LinearLayout>
```



Εικόνα 10: plot.xml

Στο φάκελο strings.xml περιέχονται όλα τα string που χρησιμοποιούνται στις διεπαφές της εφαρμογής. Πρόκειται για xml αρχείο το οποίο περιλαμβάνει tags <string> με attribute name στο οποίο αποδίδεται το id του ενώ σαν περιεχόμενο του αποδίδεται η συμβολοσειρά που βλέπει ο τελικός χρήστης.

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

    <string name="hello">Hello World, WiFiAnthi!</string>
    <string name="Scan">Start Scan!</string>
    <string name="app_name">WiFi-Anthi</string>
    <string name="report">Report</string>
    <string name="plot">Plot</string>
    <string name="choose">Choose</string>
    <string name="close">Close</string>

</resources>
```

Εικόνα 11: string.xml

### 3.2.5 Αρχείο AndroidManifest.xml

Το αρχείο αυτό είναι ζωτικής σημασίας για κάθε εφαρμογή Android. Είναι το πρώτο αρχείο το οποίο αναζητά το λειτουργικό σύστημα προκειμένου να το προσπελάσει και να λάβει τις απαραίτητες πληροφορίες για την εκτέλεση του. Οι πληροφορίες που παρέχει σχετίζονται

- Με τα πακέτα java στα οποία περιλαμβάνονται οι προς εκτέλεση κλάσεις java.
- Με τα στοιχεία από τα οποία αποτελείται η εφαρμογή όπως
  - Activities
  - Services
  - Broadcast Receivers
  - Content Provides
- Με τις διαδικασίες που υποστηρίζουν την εκτέλεση της
- Με τα απαραίτητα δικαιώματα με τα οποία πρέπει να είναι εφοδιασμένη η εφαρμογή ώστε να μπορεί να αλληλεπιδρά με άλλες εφαρμογές και διαδικασίες.



- Με δικαιώματα που χρειάζεται να έχουν άλλες εφαρμογές και διαδικασίες ώστε να επικοινωνούν με την υπό εξέταση εφαρμογή.
- Με τις ελάχιστες απαιτήσεις του android API ώστε να είναι εφικτή η εκτέλεση της.
- Με τις βιβλιοθήκες που χρειάζεται η εφαρμογή για να εκτελεστεί.
- Στο AndroidManifest.xml της συγκεκριμένης εφαρμογής: Ορίζεται το Android namespace (xmlns:android="http://schemas.android.com/apk/res/android")
- Το package στο οποίο βρίσκονται οι java κλάσεις που προσδιορίζουν την λειτουργικότητα της (package="anthi.java.utilities").
- Ο κωδικός της έκδοσης της (android:versionCode="1") και το όνομα της (android:versionName="1.0")
- Το ελάχιστο απαιτητό Android API (android:minSdkVersion="7")
- Τα δικαιώματα που χρειάζεται ώστε να εκτελεστεί (<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" /><uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE" />)
- Το εικονίδιο της εφαρμογής (android:icon="@drawable/ic\_launcher")
- Το όνομα της εφαρμογής όπως θα εμφανίζεται στον τελικό χρήστη σαν string id (android:label="@string/app\_name")
- Προσδιορίζεται αν στην εφαρμογή μπορεί να εφαρμοστεί debug (android:debuggable="true")
- Προσδιορίζονται επίσης τα δύο activities που περιλαμβάνει με δύο αντίστοιχα tags. Για το Activity main καθορίζεται ότι είναι το κύριο activity (<category android:name="android.intent.category.LAUNCHER" />)

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="anthi.java.utilities"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <application
        android:debuggable="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".WiFiAnthi"
            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="WifiPlotter" android:label="Network Plot">
        </activity>
    </application>
</manifest>

```

Εικόνα 12: AndroidManifest.xml

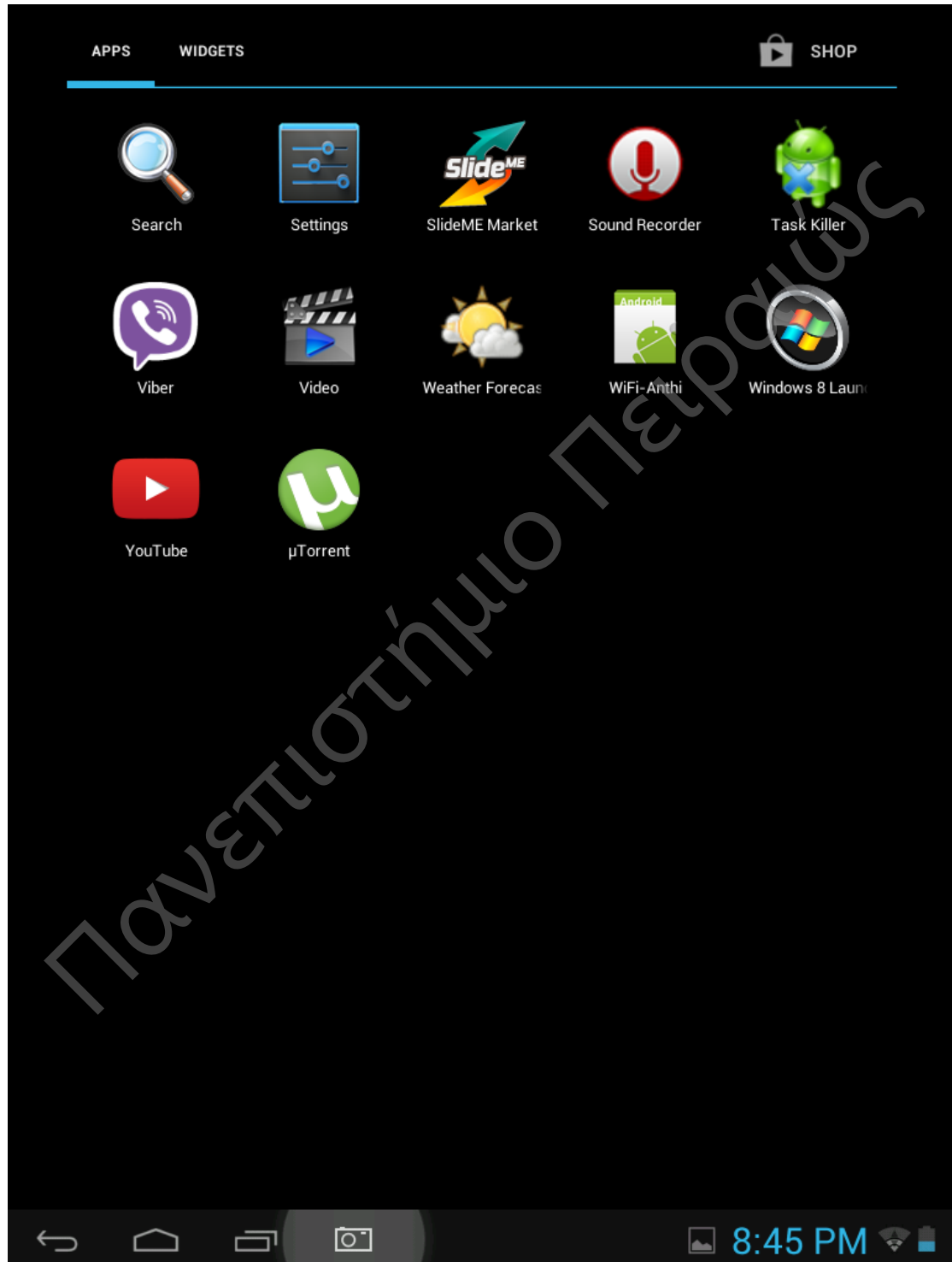
Η ανάπτυξη εφαρμογών που χρησιμοποιούν Wi-Fi είναι υλοποιήσιμες, αλλά είναι δύσκολο να γίνουν δοκιμές καθώς ο emulator δεν υποστηρίζει τη χρήση του Wi-Fi.

Για την δοκιμή των εφαρμογών απαιτείται η χρήση πραγματικής συσκευής με λειτουργικό σύστημα Android

Στον φάκελο bin/res αποθηκεύεται το αρχείο WifiAnthi.apk που είναι αυτό που πρέπει να εγκατασταθεί στο smartphone/tablet. Η εγκατάσταση υλοποιήθηκε σε συσκευή tablet.

## Manual

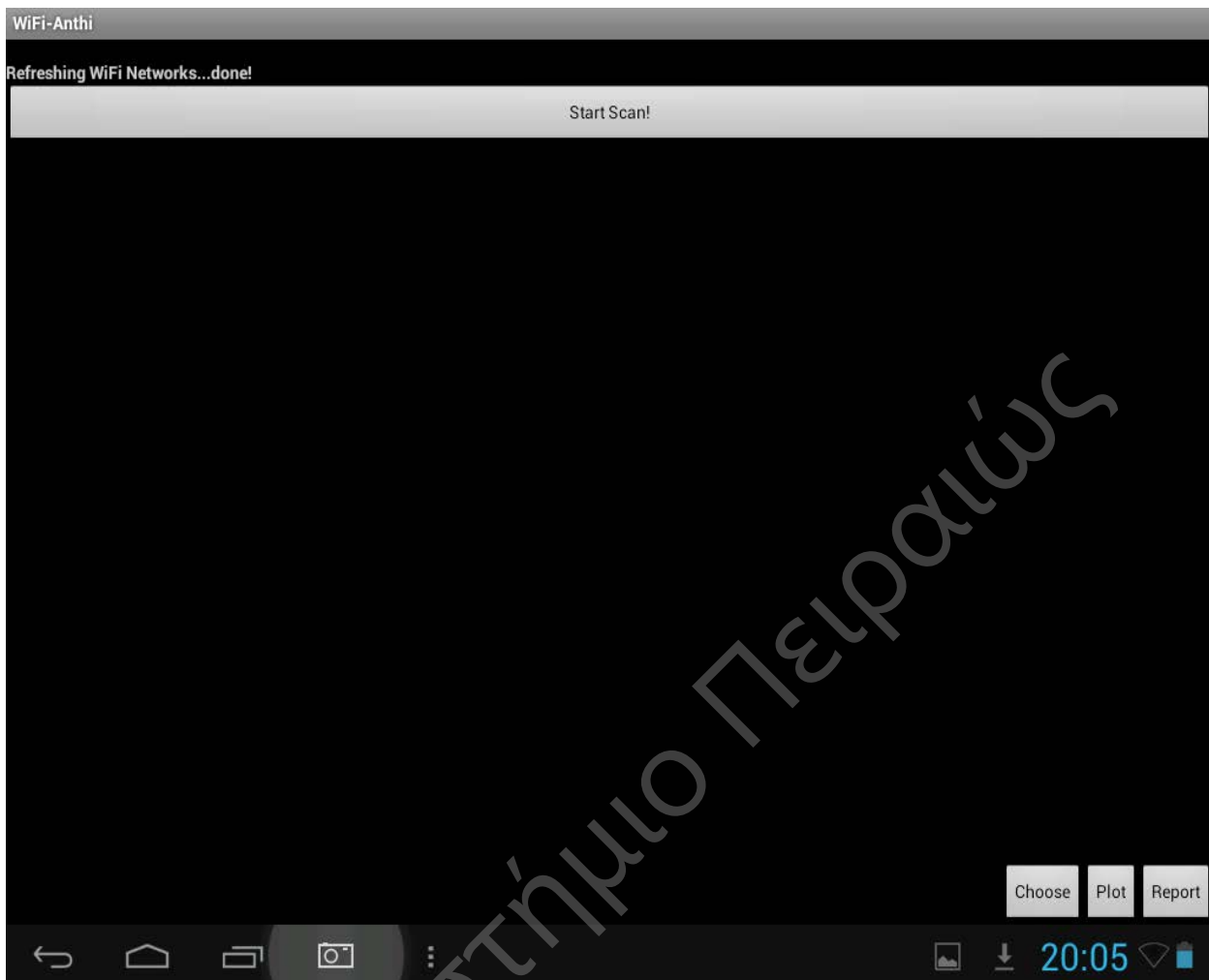
**Βήμα 1:** Επιλέγουμε την εφαρμογή WifiAnthi



**Βήμα 2:** Στην αρχική οθόνη μας ενημερώνει ότι έχει ολοκληρωθεί το scan για την ανίχνευση διαθέσιμων δικτύων και το πλήθος τους.



**Βήμα 3:** Εάν επιθυμούμε μπορούμε να επαναλάβουμε την διαδικασία ανίχνευσης διαθέσιμων ασύρματων δικτύων επιλέγοντας **Start Scan!** Μόλις ολοκληρωθεί η διαδικασία θα μας ενημερώσει εμφανίζοντας το παρακάτω μήνυμα *'Refreshing WiFi Networks...done!'*



**Βήμα 4:** Επιλέγοντας **Report** μας εμφανίζει επιπρόσθετες πληροφορίες σχετικά με τα ανιχνεύσιμα δίκτυα.

Οι πληροφορίες που μας επιστρέφει όπως απεικονίζεται και στο ακόλουθο σχήμα, είναι:

- SSID: Όνομα δικτύου
- FREQ: Συχνότητα λειτουργίας
- LEVEL: Ισχύς του σήματος



**Βήμα 5:** Επιλέγοντας **Choose** μας εμφανίζει μια λίστα με όλα τα διαθέσιμα δίκτυα τα οποία ανιχνεύθηκαν ταξινομημένα.

Το κριτήριο βάση του οποίου γίνεται η ταξινόμηση και επιλέγεται το καλύτερο δίκτυο, το οποίο είναι και το πρώτο στη λίστα μας, είναι να μην χρησιμοποιείται η ίδια συχνότητα λειτουργίας

Σε περίπτωση όπου δεν υπάρχουν δίκτυα τα οποία χρησιμοποιούν την ίδια συχνότητα λειτουργίας, η εφαρμογή θεωρεί ότι το δίκτυο με την μεγαλύτερη συχνότητα λειτουργίας είναι και το βέλτιστο και το τοποθετεί πρώτο στη λίστα.



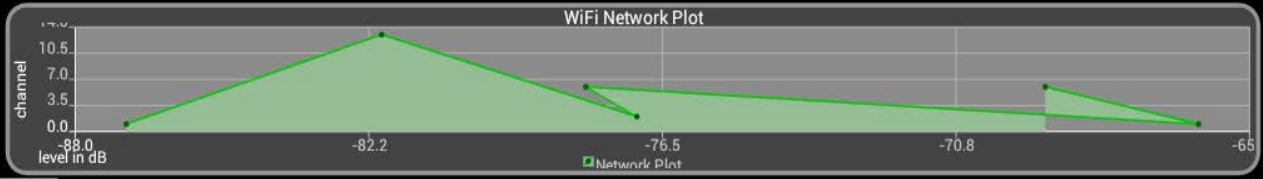
**Βήμα 6:** Τέλος επιλέγοντας **Plot** η εφαρμογή μας επιστρέφει την γραφική παράσταση των διαθέσιμων δικτύων που έχει ανιχνεύσει.

Στον άξονα xχ' απεικονίζεται η ισχύς του κάθε δικτύου σε dB ενώ στον άξονα yy' η συχνότητα λειτουργίας μεταφρασμένη στο αντίστοιχο κανάλι.

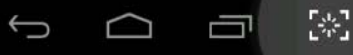
Κάθε δίκτυο τοποθετείται στο σύστημα συντεταγμένων με μια κουκίδα.

Επιλέγοντας **Close** η εφαρμογή μας επιστρέφει στην προηγούμενη οθόνη που βρισκόμασταν πριν επιλέξουμε **Plot**.

Network Plot



Close



Saving screenshot...  
Screenshot is being saved. 10:08

Πανεπιστήμιο Πειραιώς



## Συμπεράσματα

Οι δυνατότητες των κινητών συσκευών πλέον μπορούν όχι μόνο να καλύψουν τις ανάγκες του σύγχρονου ανθρώπου αλλά και να βελτιώσουν τη ζωή του επεμβαίνοντας σε τομείς τους οποίους κανείς δεν μπορούσε να φανταστεί. Οι κινητές συσκευές σε συνδυασμό με το διαδίκτυο άνοιξαν τον δρόμο σε ένα σύνολο εφαρμογών οι οποίες παρέχουν ευκολίες που οδηγούνται καθαρά από την καινοτομία των μηχανικών.

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

Η παρούσα εργασία σαν σκοπό είχε μια γρήγορη ξενάγηση και μια σύντομη επισκόπηση αυτού του κόσμου. Σε καμία περίπτωση δεν ήταν πλήρης και εξαντλητική αλλά αποσκοπούσε στο να βοηθήσει κάποιον που θέλει να αναπτύξει εφαρμογές για Android να ξεκινήσει.

Κάποια πράγματα που δεν πρέπει ποτέ να ξεχνάμε όταν αναπτύσσουμε εφαρμογή για το Android είναι να σχεδιάζουμε πάντα κάτι πριν το υλοποιήσουμε, να δίνουμε στην διεπαφή χρήστη την σημασία που της αξίζει και να ελέγχουμε οτιδήποτε αναπτύσσουμε εξονυχιστικά. Αν κάνουμε συνήθεια αυτές τις τρεις βασικές αρχές, είμαστε στο σωστό δρόμο όχι μόνο για την ανάπτυξη σωστών εφαρμογών Android αλλά και για να γίνουμε καλύτεροι προγραμματιστές.

## Βιβλιογραφία

### Βιβλιογραφικές πηγές

- Wei - Meng Lee, 2011, *Beginning Android Application Development*, 1<sup>st</sup> Edition, Wrox
- Wei – Meng Lee, 2012, *Beginning Android 4 Application Development*, 1<sup>st</sup> Edition, Wrox
- Jason Morris, 2011, *Android User Interface Development Beginner's Guide*, 1<sup>st</sup> Edition, Packt Publishing
- Reto Meier, 2012, *Professional Android 4 Application Development*, Updated Edition, John Wiley & Sons
- Mike Wolfson, 2013, *Android Developer Tools Essentials*, 1<sup>st</sup> Edition, O'Reilly Media
- Paul Deiter, Harvey Deitel, Abbey Deitel, Michael Morgano, 2011, *Android for Programmers, An App-Driven Approach*, 1st Edition , Prentice Hall, USA
- J.Steele, 2010, *The Android Developer's Cookbook*, Addison & Wesley

### Διαδικτυακές πηγές

- Ανάλυση της αρχιτεκτονικής του Android  
<http://developer.android.com/guide/basics/what-is-android.html>
- Ανάλυση των βασικών συστατικών μιας εφαρμογής του Android, και ο συσχετισμός τους με την αρχιτεκτονική του συστήματος  
<http://developer.android.com/guide/basics/what-is-android.html>
- Οδηγίες εγκατάστασης του Android SDK  
<http://developer.android.com/sdk/installing.html>
- Οδηγίες εγκατάστασης java SDK  
<http://java.sun.com/javase/downloads/>
- Η επίσημη ιστοσελίδα για την εγκατάσταση του Eclipse IDE,  
<http://www.eclipse.org/downloads>

- Οδηγίες εγκατάστασης του ADT plugin  
<http://developer.android.com/sdk/eclipse-adt.html#installing>
- Αντιστοίχιση συχνότητας σε κανάλι  
[http://en.wikipedia.org/wiki/List\\_of\\_WLAN\\_channels](http://en.wikipedia.org/wiki/List_of_WLAN_channels)

Πανεπιστήμιο Πειραιώς