



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**" ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΟΥ
ΤΗΛΕΦΩΝΟΥ ΜΕ ΧΡΗΣΗ ΤΗΣ
ΠΛΑΤΦΟΡΜΑΣ GOOGLE ANDROID "**



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΚΩΝΣΤΑΝΤΙΝΟΣ ΓΙΑΚΟΥΣΤΙΔΗΣ
ΚΑΘΗΓΗΤΗΣ ΕΦΑΡΜΟΓΩΝ ΑΤΕΙΘ

ΔΙΔΑΣΚΑΛΟΥ ΠΑΝΑΓΙΩΤΑ (ΑΜ 05/2842)

ΘΕΣΣΑΛΟΝΙΚΗ, ΑΠΡΙΛΙΟΣ 2013

Περίληψη

Στην παρούσα πτυχιακή αναλύεται και περιγράφεται η ανάπτυξη εφαρμογής για κινητά τηλέφωνα χρησιμοποιώντας την πλατφόρμα ανάπτυξης Android SDK για Android συσκευές.

Στην εισαγωγή περιγράφεται η ραγδαία ανάπτυξη της αγοράς των εφαρμογών για mobile πλατφόρμες και τις προοπτικές που ανοίγονται για τους νέους προγραμματιστές την σύγχρονη εποχή.

Γίνεται αναφορά πάνω στις πλέον αναπτυσσόμενες πλατφόρμες και σύγκριση αυτών καθώς και των πλεονεκτημάτων και μειονεκτημάτων που παρουσιάζουν, δίνοντας έμφαση στην πλατφόρμα ανοιχτού κώδικα Android SDK.

Στην συνέχεια περιγράφεται η χρήση και ενσωμάτωση των εργαλείων στο σύστημα ανάπτυξης εφαρμογών του Android, και εμβαθύνουμε πάνω στις πιο χρήσιμες παραμέτρους τους χρησιμοποιώντας την αντικειμενοστρεφής γλώσσα προγραμματισμού JAVA.

Χρησιμοποιώντας τα παραπάνω εργαλεία και τις εμπειρίες που αποκτήθηκαν από την μελέτη αυτών, αναπτύχθηκε εφαρμογή η οποία λειτουργεί ως σύστημα ειδοποίησης (notifier) βασιζόμενο στην γεωγραφική τοποθεσία του χρήστη (geolocation). Πλησιάζοντας ο χρήστης σε ένα συγκεκριμένο γεωγραφικό σημείο το σύστημα τον ενημερώνει-υπενθυμίζει με την εμφάνιση μηνύματος κα δόνηση για το συγκεκριμένο γεωγραφικό σημείο, το οποίο ο ίδιος θα έχει ορίσει ως σημείο υπενθύμισης. Το σύστημα που έχει αναπτυχθεί εκμεταλλεύεται τις υποσυσκευές A-GPS και την διασύνδεση στο Διαδίκτυο (GPRS, WiFi) του smartphone.

Επιπλέον, μελλοντικά και με την εξέλιξη αυτής της εφαρμογής θα υπάρχει πρόσβαση σε περιβάλλον web ώστε να ενημερώνεται ο χρήστης real-time για την κατάσταση των ειδοποιήσεων. Δηλαδή ποιές έχουνε εκπληρωθεί και ποιές εκκρεμούν.

Περιεχόμενα

Κεφάλαιο 1 : Εισαγωγή	5
1.1 Εισαγωγή.....	5
1.2 Android.....	5
1.2.1 Τι είναι το Android;	7
1.2.2 Ιστορικά-Εκδόσεις και χαρακτηριστικά	10
1.2.3 Αρχιτεκτονική του Android	19
1.2.3.1 Πυρήνας Linux (Linux kernel).....	20
1.2.3.2 Εγγενής Βιβλιοθήκες – Native Libraries και εικονική μηχανή Dalvik.....	21
1.2.3.3 Χρόνος Εκτέλεσης – Android Runtime	22
1.2.4 Πλαίσιο Εφαρμογής – Application Framework.....	23
1.2.5 Εφαρμογές και Widgets.....	26
1.2.6 Στο εσωτερικό μιας εφαρμογής του Android	26
1.2.7 Το αρχείο AndroidManifest.xml	26
1.2.8 Οι φάκελοι src & res	27
1.2.9 Οι υπόλοιποι φάκελοι του project	28
1.2.10 Δομικά Μέρη μιας Εφαρμογής	28
1.2.11 Ασφάλεια στο Android.....	30
1.3 Κινητά τηλέφωνα Android	31
1.3.1. Η Πρώτη Συσκευή Android.....	31
1.3.2 Εναλλακτικά Πληκτρολόγια	32
1.3.3 Αυτοματοποίηση	32
1.3.4 Προσαρμοσμένο Γραφικό Περιβάλλον	33
1.3.5 Αναιρούμενα Μέσα Αποθήκευσης και Μπαταρία.....	33
1.3.6 Ασύρματη Εγκατάσταση Εφαρμογών	34
1.3.7 Προσαρμοσμένες ROM	35
1.3.8 Έλεγχος του Τηλεφώνου από τον Υπολογιστή σου	35
1.3.9 Flash Player	35
1.3.10 Πραγματική Ενσωμάτωση Εφαρμογών	36
1.4 Ταμπλέτες Android	37
1.4.1 Tablet PC: Με δυο Λόγια.....	37
1.4.2 Ιστορική αναδρομή.....	38
1.4.2.1 Πώς εξελίσσεται έως σήμερα.....	39
1.4.3 Η Adobe φέρνει τις Touch Apps στις ταμπλέτες Android.....	40
1.4.4 Η εξέλιξη των Tablets	41
1.4.4.1 Χρήση 4-πύρηνων επεξεργαστών	42
1.4.4.2 Υψηλότερη ανάλυση.....	43
1.5 Χάρτες Google - Google Maps.....	43
1.5.1 Υλοποίηση	44
1.5.2 Επεκτασιμότητα	45
1.5.3 Προγραμματιζόμενη διεπαφή εφαρμογής χαρτών Google (Google Maps API)	45
1.5.4 Χάρτες Google για κινητά	46
1.5.5 Λοιπές χρήσεις των χαρτών Google από την Google	47
1.5.6 Οι χάρτες Google και το Android.....	48
Κεφάλαιο 2 : Ανάπτυξη Εφαρμογών στο Android	50
2.1 Κύκλος Ανάπτυξης Εφαρμογής.....	50

2.1.1 Εγκατάσταση Λογισμικού	50
2.1.2 Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής	51
2.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής.....	52
2.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό	53
2.2 Android SDK.....	54
2.3 Χρήση του Eclipse IDE μαζί με το ADT (Android Development Tools).....	55
2.4 Προκλήσεις ανάπτυξης εφαρμογών στο Android.....	57
2.4.1 Android Design Guidelines	57
2.4.2 Υποστήριξη πολλαπλών συσκευών	59
2.4.2.1 Υποστήριξη παλαιότερων εκδόσεων του Android	60
2.4.2.2 Υποστήριξη πολλαπλών διαστάσεων οθόνης και πυκνότητας pixel .	62
2.5 Δοκιμή και Αποσφαλμάτωση (Debugging) της Εφαρμογής.....	66
2.5.1 Android Debug Bridge (ADB).....	67
2.5.2 Εικονικές Συσκευές Android (Android Virtual Devices – AVD)	68
2.5.2.1 Δημιουργία διαφορετικών εικονικών συσκευών	69
2.5.3 Εργαλείο καταγραφής συμβάντων – LogCat	71
2.5.4 Dalvik Debug Monitor Server (DDMS)	75
2.5.5 Application Crash Reporter for Android (ACRA).....	77
2.6 Κατακερματισμός του Android.....	80
Κεφάλαιο 3: Ανάλυση απαιτήσεων εφαρμογής	83
3.1 Ανάπτυξη UI (User Interface) σε smartphones συσκευές.....	83
3.2 Δημιουργία φόρμας.....	83
3.3 Χρησιμοποίηση GPS δεδομένων	83
3.4 Σύγκριση γεωγραφικών σημείων.....	83
3.5 Χρησιμοποίηση του emulator.....	84
Κεφάλαιο 4: Σχεδιασμός Εφαρμογής	85
Κεφάλαιο 5: Υλοποίηση Εφαρμογής	86
5.1 Ιδιομορφία hardware κινητών συσκευών	86
5.2 Σχεδιασμός UI.....	86
5.2.1 Activity_main.xml	87
5.2.2 Set_layout.xml	87
5.3 Activities.....	88
5.4 Navigation	89
5.5 Λειτουργία GPS	90
5.6 Googlemaps	94
Κεφάλαιο 6: Δοκιμές Λειτουργίας.....	95
Κεφάλαιο 7 : Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης.....	99
Συντομογραφίες.....	101
Βιβλιογραφία	103
Ιστότοποι.....	103

Κεφάλαιο 1 : Εισαγωγή

1.1 Εισαγωγή

Στα πλαίσια αυτής της πτυχιακής εργασίας, για την υλοποίηση της εφαρμογής για τις κινητές συσκευές, μελετήθηκε και χρησιμοποιήθηκε μια στοίβα λογισμικού, το Android. Παρακάτω αναλύονται τα βασικά χαρακτηριστικά του.

1.2 Android

Η ανάπτυξη αυτής της εργασίας οδήγησε στην αναζήτηση της καλύτερης πλατφόρμας για υλοποίηση και σχεδιασμό αντίστοιχων εφαρμογών για κινητά τηλέφωνα.

Ταυτόχρονα αναζητήθηκε κάτι το καινούργιο στον χώρο που θα άξιζε την μελέτη του και παράλληλα θα ήταν πολλά υποσχόμενο. Η έρευνα οδήγησε σε δύο υποψήφιες πλατφόρμες, το iPhone και το Android. Οι δύο αυτές πλατφόρμες φαίνεται ότι θα πρωταγωνιστήσουν στον χώρο των έξυπνων τηλεφώνων (smart phones), αφού διαθέτουν εξαιρετικά χαρακτηριστικά και απίστευτες δυνατότητες.

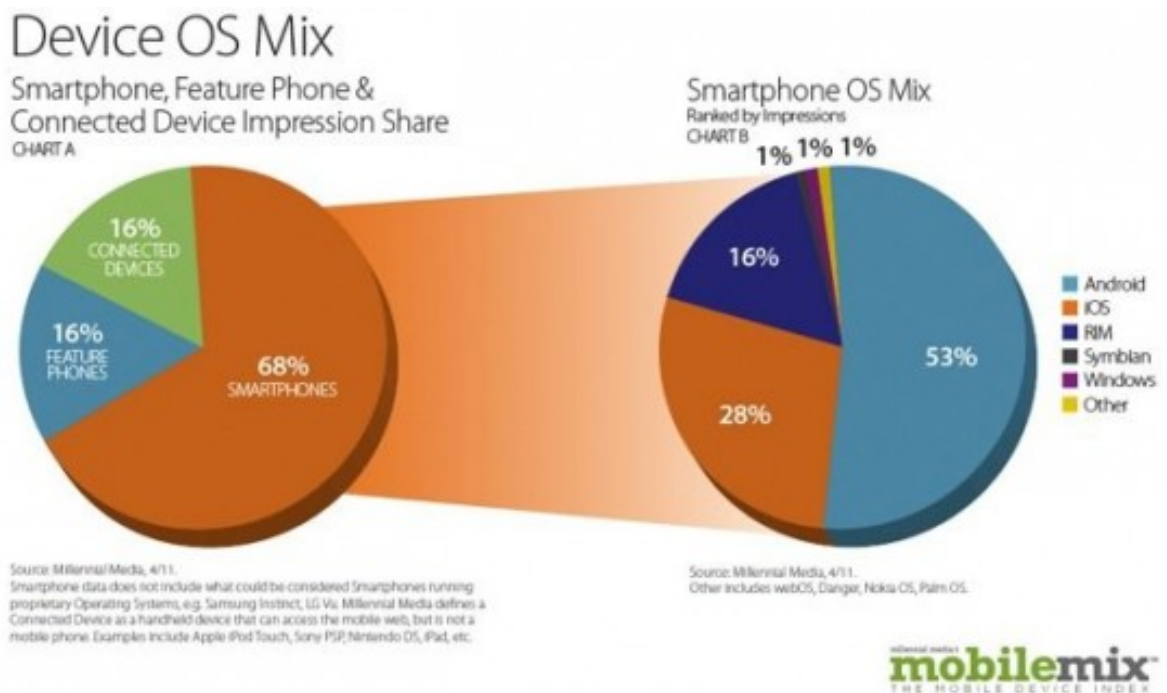


Εικόνα 1.1: Η μάχη μεταξύ Android και Iphone βρίσκεται σε εξέλιξη

Στις αρχές Οκτωβρίου 2009, το iPhone κατείχε τα σκήπτρα στις πωλήσεις. Παρόλα αυτά με την ραγδαία ανάπτυξη που είχε πάρει το Android, οι ερευνητές

μιλούσαν ότι τα δεδομένα θα έχουν ανατραπεί μέχρι το 2012 το γρηγορότερο. Προς έκπληξη αρκετών, την ώρα που γράφεται αυτή η εργασία (Μάιος 2011), το Android έχει ήδη ξεπεράσει προπολλού σε πωλήσεις το iPhone και συνεχίζει να επεκτείνεται. Ωστόσο είναι μια μάχη η οποία αναμένετε να συνεχιστεί, με τις δύο πλατφόρμες να εκσυγχρονίζουν ανά τακτά χρονικά διαστήματα τα χαρακτηριστικά τους. Χαρακτηριστική η έρευνα που ανακοίνωσε η Millennial Media.

Τα αποτελέσματα είναι λίγο πολύ αναμενόμενα, δηλαδή το Android OS κυριαρχεί με παρουσία στο 53% των smartphones παγκοσμίως, με το iOS να ακολουθεί με 28% και το BlackBerry OS με 16%. (Σχήμα 1.1).



Σχήμα 1.1: Πωλήσεις Λειτουργικών Συστημάτων για τα Smartphone

Επίσης ο Hugo Barra, Product Management Director του Android ανακοίνωσε στο πρόσφατο Google I/O που έγινε, 10 Μαΐου 2011, στο San Francisco ορισμένα εντυπωσιακά στατιστικά για την πλατφόρμα. Η Google δεν έχασε την

ευκαιρία να ρίξει τα “καρφιά” της σε οποιοδήποτε αμφισβητεί την κυριαρχία του Android OS και ανακοίνωσε τα ιλιγγιώδη στατιστικά στοιχεία που συνοδεύουν το υπερ-επιτυχημένο λειτουργικό σύστημα:

- 100 εκατ. ενεργοποιήσεις από την πρώτη ημέρα κυκλοφορίας του Android OS
- 400.000 νέες ενεργοποιήσεις κάθε ημέρα
- 4.5 δισεκατομμύρια downloads εφαρμογών
- 200.000 εφαρμογές στο Android Market
- 36 κατασκευαστές συσκευών
- 215 πάροχοι σε 112 χώρες

Οι δυνατότητες και οι προοπτικές του Android οδήγησαν στην επιλογή του και την προτίμηση του έναντι του iOS, ως πλατφόρμα ανάπτυξης αυτής εδώ της εφαρμογής. Μεγάλο ρόλο στην απόφαση αυτή, έπαιξε το γεγονός ότι το Android είναι ένα πρόγραμμα ανοικτού κώδικα, κάτι το οποίο θα προσέφερε την δυνατότητα εύκολης πρόσβασης σε πηγαίο κώδικα και τις εφαρμογές αυτής της πλατφόρμας. Αυτή η εργασία θα αρκестεί στο να παρουσιάσει τα βασικά χαρακτηριστικά του Android, ως την επιλεγθείσα πλατφόρμα, αποκλείοντας αυτή του iOS.

1.2.1 Τι είναι το Android;

Το Android είναι μια στοίβα λογισμικού για κινητές συσκευές η οποία περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό (middleware) και βασικές εφαρμογές. Το Android τρέχει τον πυρήνα του λειτουργικού Linux και μέσω της δικιά του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), επιτρέπει στους κατασκευαστές να δημιουργούν πρωτοποριακές εφαρμογές. Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance (OHA). Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού,

οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας.

Ενδεικτικά αναφέρουμε μερικά μέλη του οργανισμού αυτού (Εικόνα 1.2) για να δείξουμε την τεράστια προοπτική που δημιουργείται:

- Sprint Nextel
- T-Mobile
- Motorola
- Samsung
- Sony Ericsson
- Vodafone
- Google
- Verizon
- Texas Instruments
- Htc



Εικόνα 1.2: Εταιρίες λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας

Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android, υπό τους

όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.



Εικόνα 1.3: Λογότυπο πλατφόρμας Android

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορείς να κατεβάσει από το online κατάστημα 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 εκτιμάτε ότι ο αριθμός των downloads από το Android Market μέχρι το Δεκέμβριο του 2011 είχε υπερβεί 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.

1.2.2 Ιστορικά-Εκδόσεις και χαρακτηριστικά

Στην πληροφορική συνηθίζεται τα προϊόντα hardware και software να κυκλοφορούν εκτός από τον αριθμό έκδοσης τους, και με μία κωδική ονομασία. Η κωδική ονομασία μπορεί να είναι πχ ονόματα πόλεων (Windows Viena, Chicago), ονόματα ζώων (OSX Leopard, Lion), στην περίπτωση όμως του Android τα κώδικα ονόματα έρχονται στη μορφή επιδόρπιου!

Η πρώτη έκδοση του Android SDK τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής, παρά μόνο την χρήση των κλασσικών 'σκληρών' κουμπιών της συσκευής.

Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5, εν ονόματι 'Cupcake'.



Εικόνα 1.4: Λογότυπο Android 1.5 CUPCAKE

Το 'Cupcake' εισάγει κάποια καινούργια χαρακτηριστικά και ανανεώσεις στην διεπιφάνεια χρήστη (User Interface):

- Ικανότητα για καταγραφή και παρακολούθηση βίντεο μέσα από την λειτουργία της βιντεοκάμερας, μεταφόρτωση βίντεο στο YouTube και φωτογραφιών στο Picasa απευθείας από το τηλέφωνο, καινούργιο μαλακό πληκτρολόγιο (αφής) με πρόβλεψη κειμένου
- Υποστήριξη προτύπου Bluetooth A2DP και AVRCP
- Ικανότητα αυτόματης σύνδεσης σε μικροσυσκευή Bluetooth από μια συγκεκριμένη απόσταση
- Καινούργια widgets και φάκελοι που μπορούν να δημοσιευτούν στην αρχική οθόνη Κινούμενες μεταβάσεις οθόνης

Το 'Donut', Android 1.6, ήρθε τον Σεπτέμβριο του 2009.



Εικόνα 1.5: Λογότυπο Android 1.6 DONUT

]

Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως:

- Βελτιωμένο Android Market
- Ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα και διεπαφή (interface) γκαλερί
- Η γκαλερί επιτρέπει πλέον στους χρήστες την επιλογή πολλαπλών φωτογραφιών για διαγραφή
- Ανανεωμένη αναζήτηση με φωνή, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενής (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας να καλούμε επαφές
- Ανανεωμένη αναζήτηση με την δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών και στο διαδίκτυο από την αρχική οθόνη
- Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech)
- Υποστήριξη για ανάλυση οθονών WVGA
- Βελτιώσεις στην ταχύτητα για αναζήτηση και για εφαρμογές φωτογραφικής μηχανής

Ακολουθεί το 'Eclair', Android 2.0 τον Νοέμβριο 2009, με τις επανεκδόσεις του σε Android 2.0.1 τον Δεκέμβριο 2009 (Eclair 0.1) και τον Ιανουάριο 2010 με το Android 2.1 (Eclair MR1).



Εικόνα 1.6: Λογότυπο Android 2.0 ÉCLAIR

Ανάμεσα στις άλλες αλλαγές είναι και:

- Βέλτιστη ταχύτητα υλικού
- Υποστήριξη για περισσότερες οθόνες και αναλύσεις Βελτιωμένη διεπιφάνεια χρήστη
- Καινούργια διεπιφάνεια χρήσης για την μηχανή αναζήτησης και υποστήριξη του προτύπου HTML5
- Καινούργιες λίστες επαφών
- Καλύτερος λόγος άσπρου – μαύρου για φόντα
- Βελτιωμένοι χάρτες Google (google maps) 3.1.2
- Υποστήριξη Microsoft Exchange
- Ενσωματωμένη υποστήριξη flash για την Camera
- Ψηφιακή μεγέθυνση (zoom)
- Κλάση MotionEvent βελτιωμένη ώστε οι κατασκευαστές να μπορούν να παρακολουθούν αποτελεσματικότερα τα γεγονότα πολλαπλής αφής
- Ανανεωμένο εικονικό πληκτρολόγιο Bluetooth 2.1

Ακολουθεί το Android 2.2 με το όνομα 'Froyo' τον Μάιο του 2010.



Εικόνα 1.7: Λογότυπο Android 2.2 FROYO

Η έκδοση FROYO ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση
- Ενσωμάτωση στην μηχανή αναζήτησης, της μηχανής Javascript του Chrome V8
- Αυξημένη υποστήριξη Microsoft Exchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GAL look-up, remote wipe)
- Βελτιωμένος προωθητής εφαρμογής (application launcher), με συντομεύσεις προς τις εφαρμογές τηλεφώνου και εφαρμογές της Μηχανής Αναζήτησης
- Πρόσδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi Ανανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση
- Επιλογή για απαγόρευση πρόσβασης δεδομένων πάνω από ένα δίκτυο κινητής τηλεφωνίας
- Γρήγορη εναλλαγή ανάμεσα σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους
- Φωνητική κλήση και διαμοιρασμός επαφών με Bluetooth
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς
- Η μηχανή αναζήτησης μπορεί να αποτυπώσει κινούμενα GIFs Υποστήριξη για πεδία μεταφόρτωσης αρχείων στην μηχανή αναζήτησης Υποστήριξη για εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη
- Υποστήριξη Adobe Flash 10.1

Η τρέχουσα έκδοση για κινητά smartphones, από τον Δεκέμβριο του 2010 μέχρι και την στιγμή που γράφεται αυτή η εργασία, είναι η έκδοση Android 2.3 με το όνομα “Gingerbread” με την επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.



Εικόνα 1.8: Λογότυπο Android 2.3 GINGERBREAD

Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:

- Βελτιωμένο UI για απλότητα και ταχύτητα
- Πιο γρήγορη, πιο διαισθητική εισαγωγή κειμένου
- Επιλογή λέξεων και αντιγραφή/επικόλληση με ένα άγγιγμα Βελτιωμένη ενεργειακή διαχείριση
- Υποστήριξη NFC (Near Field Communication)
- Υποστήριξη video κλήσης
- Υποστήριξη του πρωτόκολλου WebM για αναπαραγωγή video

Η πιο πρόσφατη έκδοση, είναι η έκδοση Android 3.0 με το όνομα “Honeycomb” όπου είναι στην διάθεση των χρηστών και προγραμματιστών από τον Φεβρουάριο του 2011, λίγες μέρες μετά την επανέκδοση του Android 2.3.3, και προορίζεται αποκλειστικά για ταμπλέτες, για τις οποίες να αναφερθούμε αναλυτικότερα στην συνέχεια.



Εικόνα 1.9: Λογότυπο Android 3.0 HONEYCOMB

Μερικά από τα χαρακτηριστικά του είναι:

- Υποστηρίζει διπύρηνους και τετραπύρηνους επεξεργαστές
- Βελτιωμένη υποστήριξη των ταμπλετών
- Ανάπτυξη λογισμικού (scripting) για 3D, σε γλώσσα η οποία καλείται "Renderscript" Video chat μέσω Google Talk
- Google eBooks
- "Ιδιωτική περιήγηση"

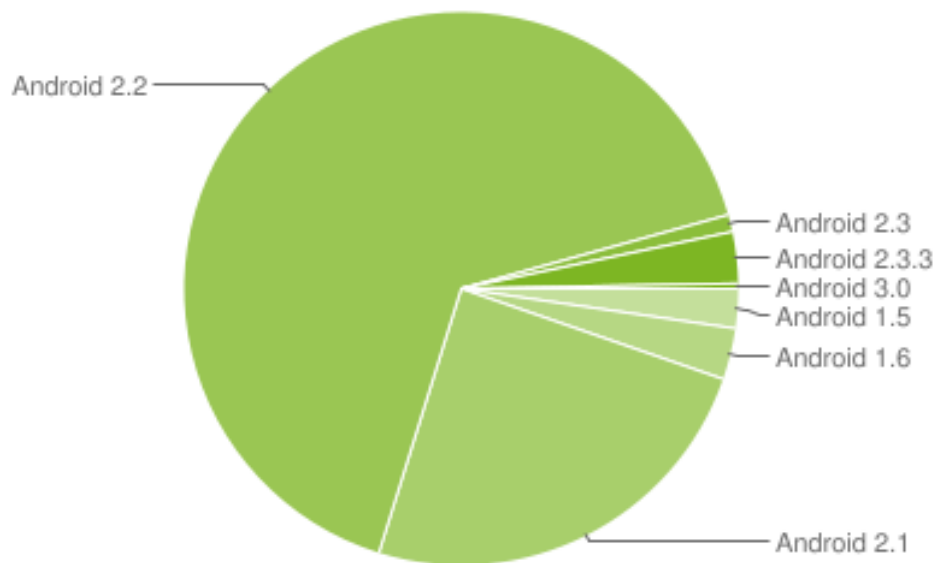
Η επόμενη έκδοση όπως ανακοινώθηκε στο Google I/O 2011 στο San Francisco, πρόκειται να είναι το "Ice Cream Sandwich", το οποίο αποτελεί την προσπάθεια της εταιρίας για ενιαίο λειτουργικό σύστημα για όλες τις συσκευές.



Εικόνα 1.10: Λογότυπο Android Ice Cream Sandwich

Το οποίο ήταν διαθέσιμο στο τελευταίο τρίμηνο του 2011 και φιλοδοξία της Google εκαι έχει χρησιμοποιηθεί σε smartphones, tablets, laptops, netbooks κλπ. Το Android Ice Cream Sandwich φέρει μαζί του “ολογραφικό” 3D UI, πλουσιότερα widgets, ενισχυμένο multitasking, νέα APIs και φυσικά είναι πλήρως open source. Για να πάρουμε μια καλή γεύση του Android Ice Cream Sandwich, η Google παρουσίασε τη λειτουργία face tracking, η οποία δεν περιορίζεται μόνο στις κινήσεις του κεφαλιού, αλλά ακολουθεί τα μάτια, τη μύτη και το στόμα. Στην περίπτωση που πραγματοποιείται video κλήση, “ζουμάρει” αυτόματα προς αυτόν που μιλά εκείνη τη στιγμή.

Ένα ακόμα ενδιαφέρον στατιστικό που πρέπει να δούμε είναι τα ποσοστά των android εκδόσεων που είναι εγκατεστημένες σε όλες τις android συσκευές όπως αυτά ανακοινώθηκαν για τον Απρίλιο του 2011 (Σχήμα 1.2).



Σχήμα 1.2: Ποσοστά των εγκατεστημένων εκδόσεων στις android συσκευές

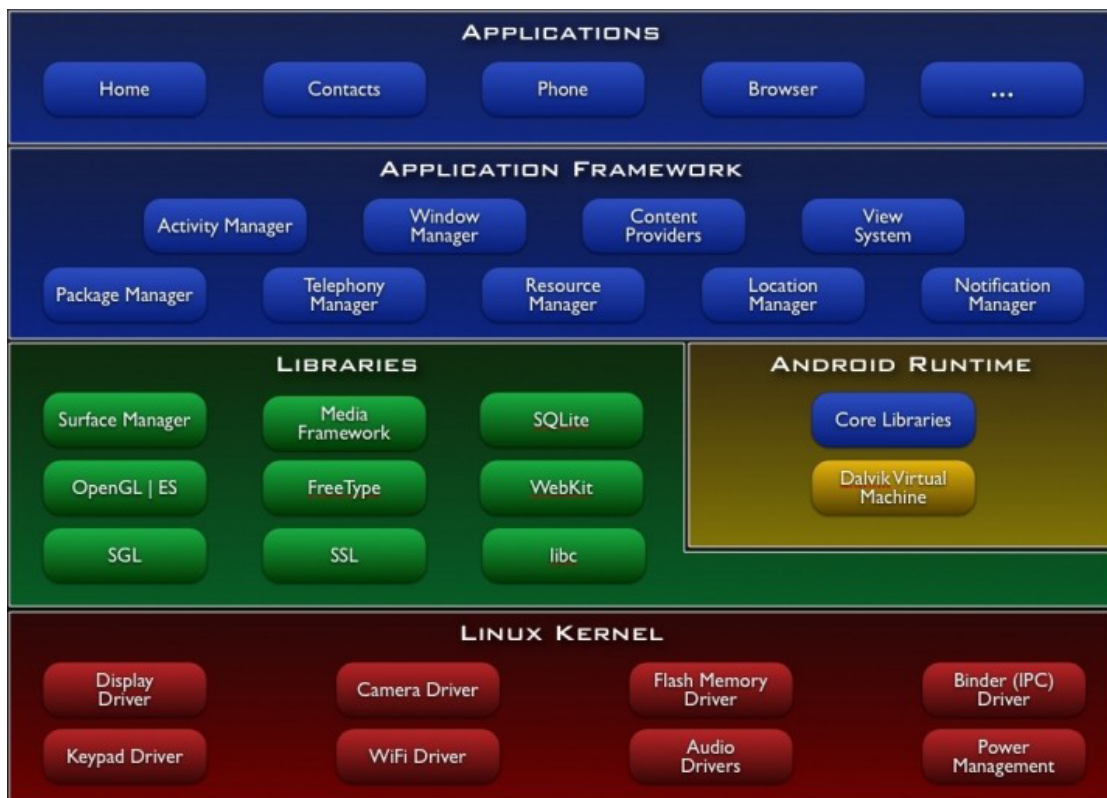
Όπως φαίνεται στην (Σχήμα 1.2) πιο πάνω, το Froyo είναι εγκατεστημένο στο μεγαλύτερο μέρος των συσκευών συγκεντρώνοντας το 65.9% αυτών. Το νεότερο Gingerbread συγκεντώνει μόλις το 3% των συσκευών λόγω του ότι δεν έχει περάσει πολύς καιρός από την επίσημη κυκλοφορία του. Το Honeycomb ως μια έκδοση αποκλειστικά για ταμπλέτες και ουσιαστικά με μηδαμινό χρόνο κυκλοφορίας βρίσκεται στο 0.3%.

Τέλος οι παλαιότερες εκδόσεις του Android (Cupcake, Donut, Eclair) με τον καιρό θα συρρικνώνονται όπως είναι λογικό, παρόλο που το Eclair εξακολουθεί να είναι εγκατεστημένο στο 24.5% των Android συσκευών.

Το Android έχει καταπληκτικά χαρακτηριστικά και πολλαπλές δυνατότητες, ενώ συνεχίζει να εκσυγχρονίζεται. Επιπρόσθετα παρέχει καταπληκτικά εργαλεία για την ανάπτυξη εφαρμογών που κάνουν την ζωή του κατασκευαστή πραγματικά πολύ πιο εύκολη. Για να κατανοήσουμε την πρωτοτυπία αυτής της πλατφόρμας, θα δούμε παρακάτω την βασική της αρχιτεκτονική και θα συζητήσουμε γιατί το Android ευκολύνει την ζωή του προγραμματιστή ενώ ταυτόχρονα του παρέχει πάμπολλες επιλογές και δυνατότητες για την δημιουργία πρωτοποριακών εφαρμογών.

1.2.3 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 (core) εφαρμογές, μεταξύ αυτών, ενός email client, μια εφαρμογή διαχείρισης SMS, ενός ημερολογίου, ενός browser, εφαρμογή διαχείρισης επαφών, και άλλες οι οποίες έρχονται δεμένες με την υπόλοιπη στοιβάδα λογισμικού του Android. Στο επίσημο σχεδιάγραμμα που ακολουθεί θα δούμε οπτικά την αρχιτεκτονική αυτή. (Σχήμα 1.3).



Σχήμα 1.3: Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android

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

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

1.2.3.1 Πυρήνας Linux (Linux kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

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

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 4-5 χρόνια ακόμα.

1.2.3.2 Εγγενής Βιβλιοθήκες – *Native Libraries* και εικονική μηχανή *Dalvik*

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

- *System C library* – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (*libc*) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
- Βιβλιοθήκες Πολυμέσων – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG
- *Surface Manager* – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- *LibWebCore* – μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (*browser engine*) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις *WebViews* που ενσωματώνονται στις εφαρμογές.
- *SGL* – η γνωστή μηχανή δισδιάστατων γραφικών
- Βιβλιοθήκες 3D – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 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

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

Η Dalvik λοιπόν είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android. Η κάθε εφαρμογή τρέχει μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία .dex ώστε να τρέξουν στην VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση.

Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο. Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

1.2.3.3 Χρόνος Εκτέλεσης – Android Runtime

Στο ίδιο επίπεδο με τις εγγενής βιβλιοθήκες, βρίσκουμε και τον χρόνο εκτέλεσης Android. Εδώ ζουν βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για

φορητές συσκευές από την Google. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από αρχεία .class και .jar. Εν αντιθέσει όμως με τα .class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία.

Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java. Κάποια πακέτα και κλάσεις υπάρχουν και στο Android κάποια άλλα δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα το Android παρέχει και επιπρόσθετα προσαρμοσμένα στις δικές του ανάγκες.

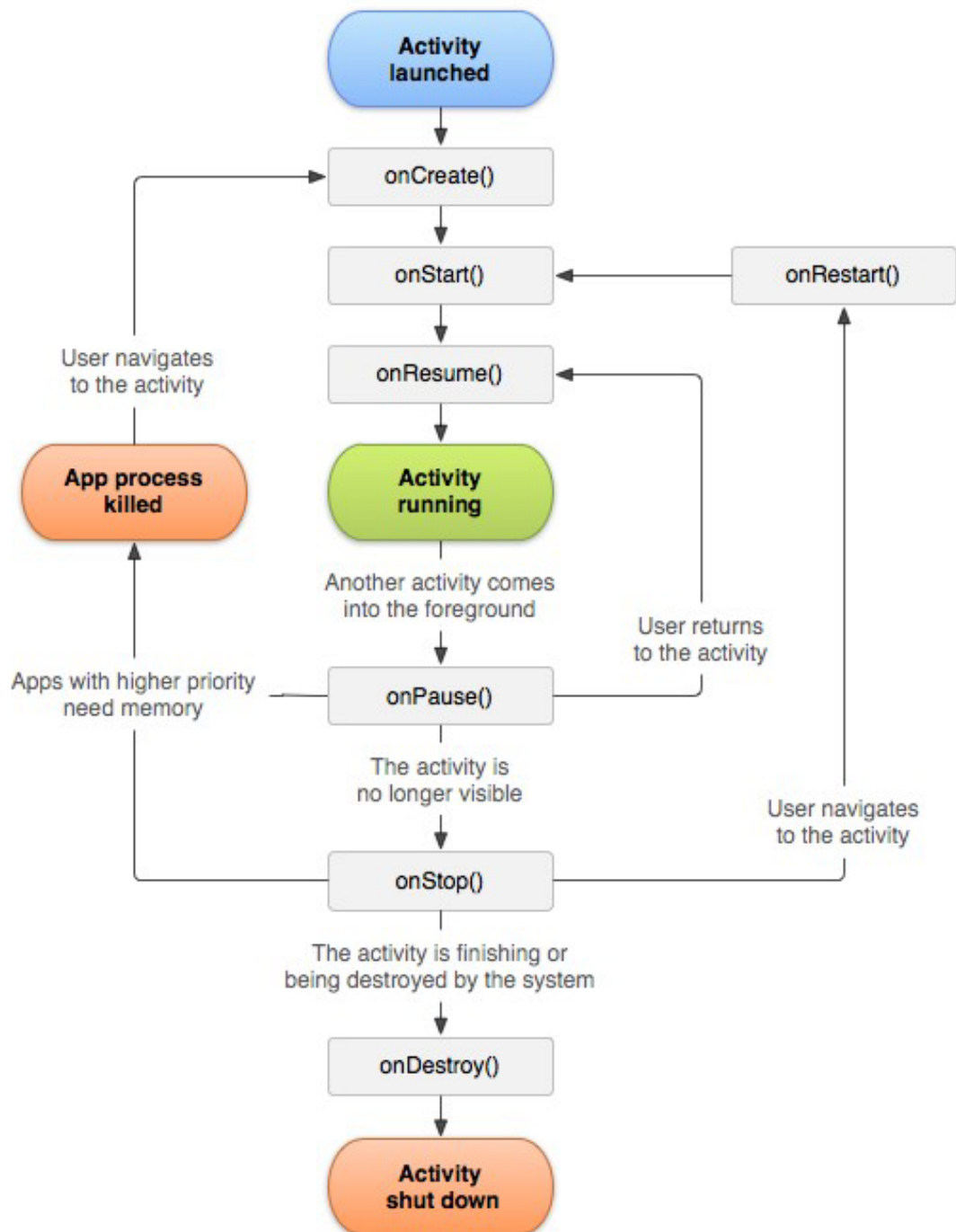
1.2.4 Πλαίσιο Εφαρμογής – Application Framework

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

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- Σύστημα προβολών (View System) – αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ.

- Πάροχος Περιεχομένου (Content Provider) – δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- Διαχειριστής Πόρων (Resource Manager) – παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ
- Διαχειριστής Ειδοποιήσεων (Notification Manager) – δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ
- Διαχειριστής Δραστηριοτήτων (Activity Manager) – διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών. Στο σχεδιάγραμμα (Εικόνα 1.4) φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας.



Σχήμα 1.4: Ο κύκλος ζωής μιας Δραστηριότητας Android

1.2.5 Εφαρμογές και Widgets

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορεί κανείς να κατεβάσει από το online κατάστημα 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 εκτιμάτε ότι ο αριθμός των downloads από το Android Market μέχρι το Δεκέμβριο του 2011 είχε υπερβεί 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.

1.2.6 Στο εσωτερικό μιας εφαρμογής του Android

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

Ξεκινώντας, η κάθε εφαρμογή αποτελείται όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής.

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

1.2.7 Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται

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

Κάποιες από αυτές τις πληροφορίες είναι:

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

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

1.2.8 Οι φάκελοι `src` & `res`

Στον φάκελο `src` (εκ του `source`) περιέχονται τα αρχεία κλάσης τις 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 τον μοναδικό φάκελο στο `project` στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας.

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

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

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

1.2.9 Οι υπόλοιποι φάκελοι του project

Ένα `project` αποτελείται από περισσότερους από τους 3 βασικούς φακέλους, κάποιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο `project` λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο `build path` του `project` μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου `res`, όπως είναι οι φάκελοι `drawable-hdpi`, `drawable-mdpi`, `layout-port`, `menu`, κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίπτωση.

1.2.10 Δομικά Μέρη μιας Εφαρμογής

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

- Δραστηριότητες (Activities) – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και

οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

- Προθέσεις (Intents) – Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.

- Υπηρεσίες (Services) – Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.

- Πάροχος Περιεχόμενου (Content Providers) - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.

- Δέκτες Μετάδοσης (Broadcast Receivers) – Πρόκειται για ένα είδους υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο

συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε.

Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

1.2.11 Ασφάλεια στο Android

Τη στιγμή που μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στο δικό της εικονική μηχανή η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

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

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη

τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να τις παρέχει πρόσβαση στους πόρους που ζητάει.

1.3 Κινητά τηλέφωνα Android

1.3.1. Η Πρώτη Συσκευή Android

Η πρώτη συσκευή που «έτρεχε» το λειτουργικό σύστημα Android ήταν το HTC Dream, γνωστό και ως T-Mobile G1, που έκανε την εμφάνιση του στα ράφια των καταστημάτων προς το τέλος του 2008.

Στη συνέχεια, πολλά νέα Android κινητά έκαναν την εμφάνιση τους με “ναυαρχίδα” αυτών, το Motorola DROID (έκδοση CDMA), το οποίο από το Νοέμβρη του 2009 διατίθεντο αποκλειστικά από την Αμερικανική εταιρία τηλεπικοινωνιών Verizon, ενώ η GSM έκδοση του το Motorola Milestone, από διάφορους παρόχους κινητής τηλεφωνίας ανά τον κόσμο.

Στην αυγή του 2010, η Google τάρaxε τα νερά στον χώρο των τηλεπικοινωνιών με την ανακοίνωση του Google Nexus One, μιας καινοτόμου συσκευής που έτρεχε την νεότερη έκδοση του Android, 2.1 (που θεωρείται κυρίως βελτίωση της 2.0 ή Eclair). Κατασκευασμένο από την HTC, το Nexus One ξεχωρίζει για τα Live Wallpapers (κινούμενα backgrounds νέου σχεδιασμού που αλληλεπιδρούν με τις κινήσεις δαχτύλου του χρήστη) και τη νέα αναβαθμισμένη Έκθεση εικόνων και βίντεο (gallery). Η συσκευή έχει επίσης, το πλεονέκτημα της άμεσης και απροβλημάτιστης συνεργασίας με όλες τις υπηρεσίες της Google (Gmail, Google Contacts, Google Calendar κ.α.).

Στην [Εικόνα 1.11] μπορούμε να δούμε μερικές τυπικές συσκευές Android . Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματα των Android κινητών τηλεφώνων



Εικόνα 1.11: Τυπικές Συσκευές Android

1.3.2 Εναλλακτικά Πληκτρολόγια

Από πληκτρολόγια που προβλέπουν λέξεις όπως το SwiftKey, μέχρι το καινοτόμο Swype και το 8pen, υπάρχουν πολλές διαφορετικές επιλογές πληκτρολογίου για το Android.

Πληκτρολογώντας σε ένα μικροσκοπικό πληκτρολόγιο κινητού δεν αποτελεί διασκέδαση για κανένα, οπότε είναι σπουδαίο που το Android προσφέρει τόσες πολλές επιλογές για να το κάνει όσο πιο ανώδυνο γίνεται και πολύ εύκολο στην εγκατάσταση.

1.3.3 Αυτοματοποίηση

Μια από τις πιο ισχυρές και χρήσιμες εφαρμογές στο Android είναι το Tasker, το

πρόγραμμα αυτοματοποίησης που επιτρέπει τη μετατροπή του κινητού σε σούπερ κινητό. Επιτρέπει την ενεργοποίηση και την απενεργοποίηση των ρυθμίσεων για συγκεκριμένες εφαρμογές, ανάλογα την τοποθεσία, την ώρα και σχεδόν οποιαδήποτε άλλη πιθανή κατάσταση. Με τις σωστές εντολές, το Tasker μπορεί να έχει πρόσβαση ακόμα και στις πιο βαθιές και σκοτεινές ρυθμίσεις του κινητού, κάτι το οποίο δεν μπορούν να το κάνουν τα άλλα λειτουργικά.

1.3.4 Προσαρμοσμένο Γραφικό Περιβάλλον

Οι χρήστες του Android έχουν την δυνατότητα με τα Home Launchers να μπορούν να τροποποιήσουν την αρχική τους οθόνη. Οι Home Launchers προσθέτουν όλα τα είδη επιπλέον χαρακτηριστικών για τις αρχικές οθόνες του κινητού όπως χειρονομίες, διαφόρων ειδών συντομεύσεις, ακόμα και χαμηλού επιπέδου ρυθμίσεις που μπορούν να βοηθήσουν στην αύξηση της ταχύτητας ενός παλιού κινητού.

Είτε χρησιμοποιείται το πολύ γρήγορο LauncherPro, είτε το απίστευτα παραμετροποιήσιμο ADWLauncher, είτε το GoLauncher, οι Home Launchers προσφέρουν ένα σωρό πρόσθετες επιλογές στην συσκευή.

1.3.5 Αναιρούμενα Μέσα Αποθήκευσης και Μπαταρία

Δεν είναι μέρος του Android λογισμικού απαραίτητα, αλλά η ανοιχτού λογισμικού φύση του δίνει αρκετά πλεονεκτήματα όπως η αφαίρεση και αναβάθμιση της μπαταρίας και της κάρτας μνήμης.

Αν διαπιστωθεί ότι έχει φτάσει στο όριο του χώρου αποθήκευσης στο iPhone, δεν υπάρχει άλλη δυνατότητα εκτός της διαγραφής, ενώ στο Android γίνεται επέκταση του χώρου αποθήκευσης με την εισαγωγή μιας μεγαλύτερης κάρτας μνήμης.

Παρόμοια, η αντικατάσταση της μπαταρίας με μια μεγαλύτερης διάρκειας, θα

κάνει το κινητό να κρατήσει ακόμα περισσότερο χωρίς να χρειαστεί φόρτιση [Εικόνα 1.12].



Εικόνα 1.12: Φωτογραφία μιας καθημερινής κάρτας μνήμης

1.3.6 Ασύρματη Εγκατάσταση Εφαρμογών

Η αναζήτηση και η ανακάλυψη νέων εφαρμογών πρέπει να είναι κάτι το διασκευστικό και όχι να αποτελεί πρόκληση λόγω ενός μικροσκοπικού App Store. Η περιήγηση στο App Store και το Cydia App Store δεν είναι και τόσο διασκεδαστική μέσω του iPhone και είτε θα πρέπει να γίνει κατέβασμα (download) των εφαρμογών στο κινητό, είτε να γίνει σύνδεση με το iTunes για τον συγχρονισμό.

Με το νέο Android Market ή σελίδες τρίτων όπως το AppBrain, είναι δυνατή η αναζήτηση μιας εφαρμογής και με το πάτημα ενός κουμπιού γίνεται αυτόματα η εγκατάσταση στο κινητό, ώστε να είναι έτοιμη προς χρήση την επόμενη φορά. Τι πιο βολικό από αυτό;

1.3.7 Προσαρμοσμένες ROM

Αν και υπάρχουν πολλές εφαρμογές τρίτων στο Android που δίνουν προηγμένα χαρακτηριστικά, ένα από τα πιο ενδιαφέροντα πράγματα σχετικά με το λειτουργικό αυτό και το ότι είναι ανοιχτού κώδικα, είναι το γεγονός ότι οι άνθρωποι μπορούν να το πάρουν, να το «πειράξουν» και να εγκαταστήσουν την δικιά τους έκδοση αντί για αυτή που είχε το κινητό εξαρχής. Είτε πρόκειται για την γεμάτη χαρακτηριστικά CyanogenMod, είτε για την γεμάτη γραφικά MIUI ROM, δεν υπάρχει όριο στο πόσο μπορεί να «πειραχτεί» το λειτουργικό Android. Όπως και με τους Launchers, αυτό δίνει την δυνατότητα για πολλές έξτρα ρυθμίσεις συστήματος που δεν θα μπορούσαν να γίνουν σε άλλα λειτουργικά και τα κάνει εύκολα προσβάσιμα σε όλους. Είτε «πειράζουν» την ταχύτητα του κινητού, είτε χαρακτηριστικά όπως το Radio FM, οι προσαρμοσμένες ROM είναι ένα από τα μεγαλύτερα πλεονεκτήματα του Android.

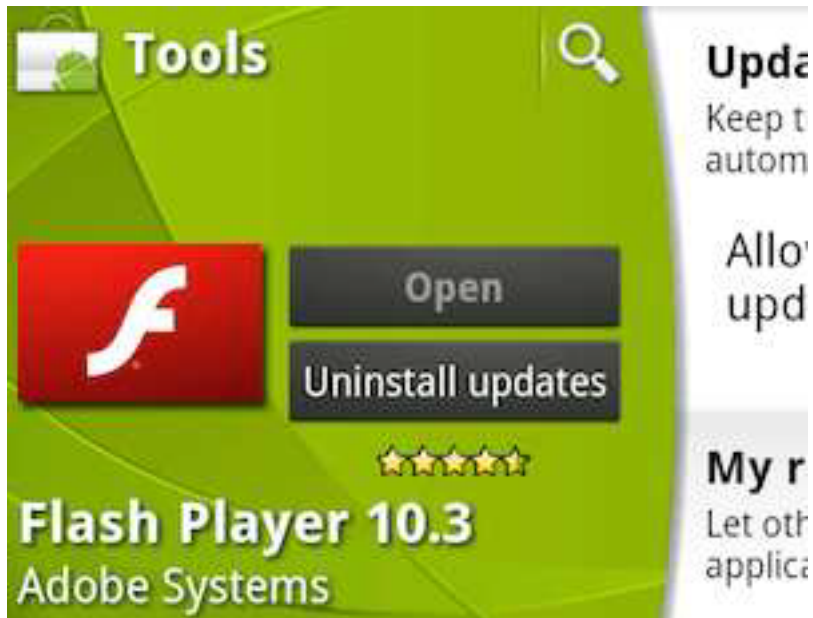
1.3.8 Έλεγχος του Τηλεφώνου από τον Υπολογιστή σου

Υπάρχουν αρκετές εφαρμογές που επιτρέπουν τον έλεγχο ενός Android κινητού από τον υπολογιστή, είτε απλά πρόκειται για αποστολή κάποιου μηνύματος κειμένου από τον Chrome, είτε πρόσβασης σε άλλες λειτουργίες απευθείας από έναν web browser. Μια από τις καλύτερες εφαρμογές είναι το LazyDroid. Με αυτόν τον τρόπο δίνεται η δυνατότητα χρήσης μια ξεχωριστής εφαρμογής, η οποία έχει πρόσβαση σε όλες τις βασικές λειτουργίες.

1.3.9 Flash Player

Για την πλοήγηση στο Internet είναι περισσότερο από απαραίτητη η χρήση του Flash. Ουσιαστικά, αποτελεί τη βάση της πλοήγησης. Είτε πρόκειται για μια

σελίδα εξ' ολοκλήρου φτιαγμένη σε Flash, είτε για βίντεο, είτε για παιχνίδια, η εγκατάσταση του Flash στο κινητό ή το Tablet δίνει πρόσβαση σε πολλά πράγματα που διαφορετικά θα ήταν αδύνατη. Το Flash είναι τόσο διαδεδομένο, που η έλλειψη του ισοδυναμεί ουσιαστικά με απουσία Internet [Εικόνα 1.13].



Εικόνα 1.13: Πώς εμφανίζεται το uninstall στο Android

1.3.10 Πραγματική Ενσωμάτωση Εφαρμογών

Το Google Voice μπορεί να είναι διαθέσιμο και για το iPhone, αλλά η εμπειρία δεν θα είναι ποτέ ίδια με αυτήν σε ένα κινητό Android. Άλλες εφαρμογές για το iPhone, κάνουν ανακατεύθυνση πάντα στον προεπιλεγμένο dialer και εφαρμογή email. Ακόμα και αν ο χρήστης επιθυμεί να χρησιμοποιήσει το Google Voice σαν κύριο, θα πρέπει να ανοίγει μόνος του την εφαρμογή κάθε φορά. Στο Android, εφαρμογές όπως το Google Voice ενσωματώνονται απευθείας με το λειτουργικό. Έτσι, κάθε φορά που πραγματοποιείται κλήση, περνάει από το Google Voices. Η πραγματική ενσωμάτωση, όπως αυτή, κάνει τις αρχικές εφαρμογές του κινητού απολύτως απρόσκοπτες, κάτι το οποίο δεν υπάρχει στο κλειδωμένο iPhone.

1.4 Ταμπλέτες Android



Εικόνα 1.14: Ταμπλέτες Android

1.4.1 Tablet PC: Με δυο Λόγια

Μια ταμπλέτα (Tablet PC) είναι ένας ηλεκτρονικός υπολογιστής μεγέθους συνήθως 6 έως 10 ιντσών, η οποία έχει ως βασικά χαρακτηριστικά αυτά ενός κανονικού Η/Υ και συνήθως περιλαμβάνει ένα πλήρες λειτουργικό σύστημα. Ένα φορητό Tablet PC περιλαμβάνει μια οθόνη αφής ως κύρια μέθοδο εισόδου δεδομένων και είναι σχεδιασμένο έτσι ώστε να χρησιμοποιείται για προσωπική χρήση.

Ο όρος Tablet PC έγινε γνωστός από μια παρουσίαση της Microsoft το 2001 όπου και χρησιμοποιήθηκε για πρώτη φορά. Οι ταμπλέτες χρησιμοποιούν εικονικά πληκτρολόγια και αναγνώριση γραφής, έτσι ώστε να είναι δυνατή η εισαγωγή κειμένου μέσω της οθόνης αφής. Όλες οι ταμπλέτες έχουν δυνατότητα ασύρματης σύνδεσης μέσω Wi-Fi, αλλά και ενσύρματα στο διαδίκτυο.

Το λογισμικό τους περιλαμβάνει εφαρμογές γραφείου, προγράμματα περιήγησης στο διαδίκτυο, παιχνίδια, αλλά από την στιγμή που τρέχουν ολοκληρωμένα λειτουργικά συστήματα, μπορούν ουσιαστικά να χρησιμοποιήσουν οποιοδήποτε πρόγραμμα υποστηρίζει στο λειτουργικό τους. Συνήθως βέβαια, η υπολογιστική δύναμη αυτών των συστημάτων είναι περιορισμένη, οπότε δεν είναι πάντα ικανά να ανταπεξέλθουν στους πόρους που χρειάζονται μερικές πιο απαιτητικές

εφαρμογές, με αποτέλεσμα να μην προσφέρουν την καλύτερη εμπειρία χρήσης. Σύμφωνα με έρευνες που πραγματοποιήθηκαν στις αρχές του 2011, οι ταμπλέτες βρίσκονται ακόμα στο στάδιο εισαγωγής τους στην αγορά, αφού μόνο το 5% των Αμερικάνων έχει στην κατοχή του μια, εκ των οποίων το 3% είναι οι συσκευές iPad της Apple.

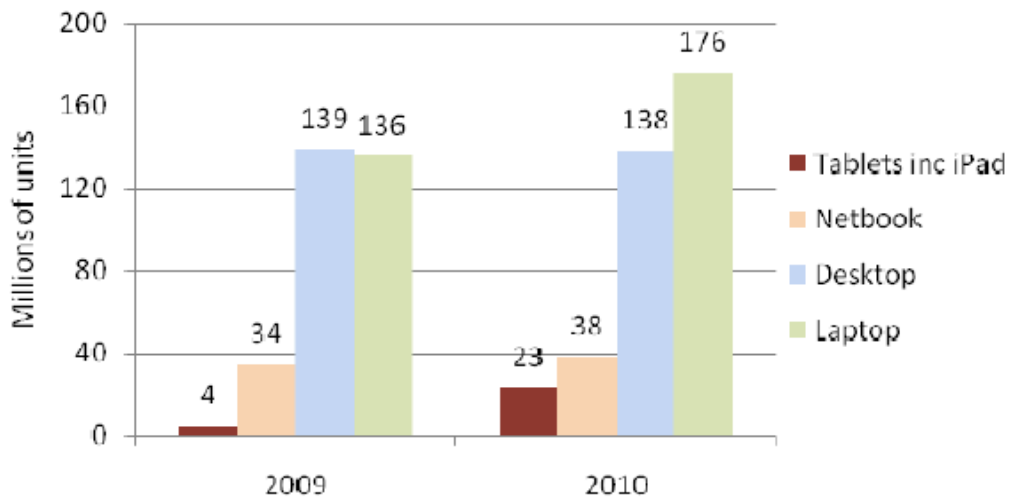
1.4.2 Ιστορική αναδρομή

Οι ταμπλέτες υπάρχουν από το 2001, αλλά έγιναν δημοφιλέστερες μετά την έλευση του Apple iPad, κερδίζοντας ένα σημαντικό μερίδιο του ενδιαφέροντος του κοινού. Ειδικά με την έλευση των Android ταμπλετών που πλέον κυριαρχούν στην αγορά, αναμένεται στο άμεσο μέλλον να αντικαταστήσουν πλήρως τα netbooks, μπαίνοντας για τα καλά στην καθημερινότητα των χρηστών.

Έρευνα της εταιρίας Strategy Analytics έδειξε πως ως το 2015 η αγορά θα έχει κέρδος από τις πωλήσεις των tablets 49 δισ. δολάρια [6]. Η αγορά ταμπλετών τελευταίας τεχνολογίας θα είναι η τρίτη σε δύναμη αγορά, μετά από αυτή της τηλεόρασης και των προσωπικών ηλεκτρονικών υπολογιστών. Η εταιρία ερευνών ανέφερε πως, σύμφωνα με τις προβλέψεις, θα έχουν πουληθεί 149 εκατομμύρια μονάδες ως το 2015, δηλαδή οι πωλήσεις θα αυξηθούν κατά 6,5 φορές από 2010.

Παρακάτω φαίνεται καθαρά ο ρυθμός αύξησης πωλήσεων των ταμπλετών, σε σύγκριση με όλα τα άλλα είδη προσωπικών υπολογιστών από το 2009 [Εικόνα 1.15].

PC and Tablet sales volumes



Εικόνα 1.15: Αριθμός πωλήσεων Η/Υ και ταμπλετών

1.4.2.1 Πώς εξελίσσεται έως σήμερα

Η απόλυτη κυριαρχία του iPad ίσως έχει αρχίσει να κλονίζεται. Όταν είχε πρωτοκυκλοφορήσει η ταμπλέτα, ήταν και η μοναδική στην αγορά και σχεδόν άμεσα εδραιώθηκε. Μετά από σύντομο χρονικό διάστημα, άρχισαν οι κατασκευαστές τηλεφώνων Android να κυκλοφορούν στην αγορά υπερμεγέθη smartphones και να τα παρουσιάζουν ως ταμπλέτες με λειτουργικό Froyo ή Gingerbread. Η πιο δημοφιλής συσκευή από αυτές ήταν η Samsung Galaxy Tab, η οποία κατάφερε και πήρε κάποιο μέρος της πίτας από την αγορά. Με την ταμπλέτα της HP πρακτικά να έχει φύγει από το σκηνικό και το Playbook της BlackBerry να αποτελεί μία μάλλον αποτυχία της εταιρείας, το iPad εξακολουθούσε να κρατάει τα πρωτεία.

Και παρόλο που στην πλειονότητά τους οι ταμπλέτες Android με το Honeycomb, δεν ανταποκρίθηκαν στις απαιτήσεις, οι φθηνές Gingerbread απέκτησαν μία θέση στην αγορά. Έτσι ενώ, ακόμα και αν δεν είναι οι απόλυτα κερδοφόρες, υπάρχουν όμως εκεί και κάνουν αισθητή την παρουσία της Google στην αγορά.

Οι ταμπλέτες Android, αποτέλεσαν το 27% της παγκόσμιας αγοράς για το τέταρτο τρίμηνο του '11, παρουσιάζοντας μία άνοδο της τάξεως του 2,3% από το 2010. 4,5 εκατομμύρια ταμπλέτες Android πουλήθηκαν κατά το ίδιο τρίμηνο, ενώ σε σύνολο πουλήθηκαν 16,7 εκατομμύρια ταμπλέτες.

Στις πωλήσεις αυτές σημαντική παρουσία είχε η Samsung, με πωλήσεις που άγγιξαν το 1,5 εκατομμύριο. Ένας ακόμα ανταγωνιστής που παρουσιάστηκε, ήταν και η ταμπλέτα της Amazon Kindle Fire, η οποία τάραξε αρκετά τα νερά. Τέλος και με την πρόσφατη παρουσίαση του «ενωπικού» λογισμικού για ταμπλέτες και smartphones, Android ICS, τα πράγματα έχουν αρχίσει και θερμαίνονται.

Η Apple έχει αρχίσει να νιώθει έντονα την παρουσία των αντιπάλων της, ενώ και οι δικαστικές διαμάχες με την Samsung καλά κρατούν. Ομολογουμένως, τα μέτωπα που έχουν ανοίξει είναι πολλά.

1.4.3 Η Adobe φέρνει τις Touch Apps στις ταμπλέτες Android



Εικόνα 1.16: Το Touch Apps της Android

Τον Οκτώβριο 2011 η Adobe είχε παρουσιάσει την οικογένεια εφαρμογών Touch Apps. Αυτή περιλάμβανε μία συλλογή 6 εργαλείων σχεδιασμένων έτσι ώστε να κάνουν την χρήση αυτών ευκολότερη και χρηστικότερη σε ταμπλέτες.

Αυτές οι εφαρμογές είναι διαθέσιμες στο Android Market, για ταμπλέτες που τρέχουν Android 3.1 και πάνω. Το πακέτο περιλαμβάνει μεταξύ άλλων και τα Photoshop Touch, Collage and Proto, καθένα από τα οποία κοστολογείται στα 9,99 δολάρια. Οι εφαρμογές αυτές θα διαδραματίσουν και κεντρικό ρόλο στην επικείμενη πρωτοβουλία της Adobe, Creative Cloud, η οποία θα επιτρέπει στους χρήστες να μοιράζονται, να προβάλλουν και να μεταφέρουν αρχεία σε πολλαπλές συσκευές.

Αυτό είναι κάτι το οποίο δεν αναμένεται να ξεκινήσει μέχρι το πρώτο εξάμηνο του επόμενου έτους, ενώ η πλήρης σουίτα των εφαρμογών αφής για iOS χρήστες αναμένεται να κυκλοφορήσει στις αρχές του επόμενου έτους.

Οι χρήστες ταμπλετών Android θα μπορούν πλέον να έχουν πρόσβαση σε όλες τις Apps Touch, με την Adobe, όμως, να υποστηρίζει ότι για καλύτερα αποτελέσματα χρειάζεται τουλάχιστον μία οθόνη 8,9 ιντσών με ανάλυση 1280x800.

1.4.4 Η εξέλιξη των Tablets

Καθώς η αγορά των tablet pc (ή «ταμπλέτες» όπως είναι αλλιώς γνωστές) ωριμάζει, είναι αναμενόμενο να υπάρχουν σημαντικές αλλαγές, τόσο στην εμφάνιση όσο και στο «περιεχόμενό» τους. Παράλληλα, θα υπάρχουν και αναβαθμίσεις στο λειτουργικό τους σύστημα, με το Android 4.0

Οι νέες ταμπλέτες που αναμένονται με το νέο έτος, προβλέπεται να είναι ακόμα πιο λεπτές και ελαφριές. Όσον αφορά τα μοντέλα των 10 ιντσών, το βάρος θα κυμαίνεται στα 540-590gr και το πάχος τους δε θα ξεπερνάει το 1 εκατοστό. Τώρα, μάλιστα, που έχει περάσει η επιτακτική ανάγκη της γρήγορης εμφάνισης στην αγορά, οι κατασκευαστές των tablet pc αναμένεται να δώσουν μεγαλύτερη έμφαση στις λεπτομέρειες, τις οποίες υπόσχονταν από τα πρώτα κιόλας μοντέλα τους, όπως η ποιοτική απεικόνιση, η καθαρότητα του κειμένου στην οθόνη, τα ηχεία, η θύρα υπέρυθρων (για χρήση της ταμπλέτας ως Τηλε-κοντρόλ).

Η ποικιλία στα μεγέθη θα συνεχίζει να υφίσταται, από 7 έως 10.1 ίντσες, καθώς δεν έχει διαμορφωθεί ακόμα σαφής προτίμηση στους χρήστες. Επίσης αναμένονται ακόμα χαμηλότερες τιμές, χάρη στο Amazon Kindle Fire των 199\$, το οποίο πωλείται ελαφρώς χαμηλότερα του κόστους του. Σύμφωνα με το διευθύνοντα της Nvidia οι τιμές των tablet pc που χρησιμοποιούν τον επεξεργαστή Tegra θα έχουν πέσει στα 299\$ ως τα μέσα του 2012.

1.4.4.1 Χρήση 4-πύρηνων επεξεργαστών

Η Nvidia έβγαλε τον περασμένο Νοέμβριο στην αγορά την πλατφόρμα Tegra 3. Γνωστή ως «project Kal-EI», η πλατφόρμα Tegra 3 περιέχει ένα 4πύρηνο επεξεργαστή ARM Cortex A9, ένα πέμπτο πυρήνα «χαμηλής κατανάλωσης» για να ασχολείται με «δευτερεύοντα» καθήκοντα, όπως η μουσική και ένα 12πύρηνο επεξεργαστή GeForce για τα γραφικά. Αυτός ο «κινητήρας» είναι παραπάνω από επαρκής για να φέρει τις ταμπλέτες στην ίδια γραμμή εκκίνησης με τα laptop.

Απ' ότι φαίνεται το Asus Transformer Prime είναι το πρώτο tablet pc που θα τρέχει με τον Tegra 3 της Nvidia. Αυτή η ταμπλέτα-laptop (εξ ου και η ονομασία Transformer) κυκλοφορεί στην τιμή των 500 δολαρίων με μνήμη 32GB. Τον Ιανουάριο του 2012 στη διεθνή έκθεση CES, αναμένεται να ανακοινωθούν και άλλες ταμπλέτες που θα χρησιμοποιούν τον Tegra 3.

1.4.4.2 Υψηλότερη ανάλυση

Απέναντι στα 1024 x 768 pixel του iPad 2, το οποίο δίνει 132 pixels ανά ίντσα, άλλα μοντέλα, όπως το Toshiba Thrive 7" tablet προσφέρουν απεικόνιση στα 1280 x 800, που εκτοξεύει την ανάλυση στα 255 pixel ανά ίντσα. Οργιάζουν οι φήμες, ωστόσο, ότι το iPad 3 θα έχει οθόνη με απεικόνιση αντίστοιχη εκείνης του iPhone 4s, γεγονός το οποίο -αν επαληθευθεί- θα δώσει τη ρεβάνς στην Apple.

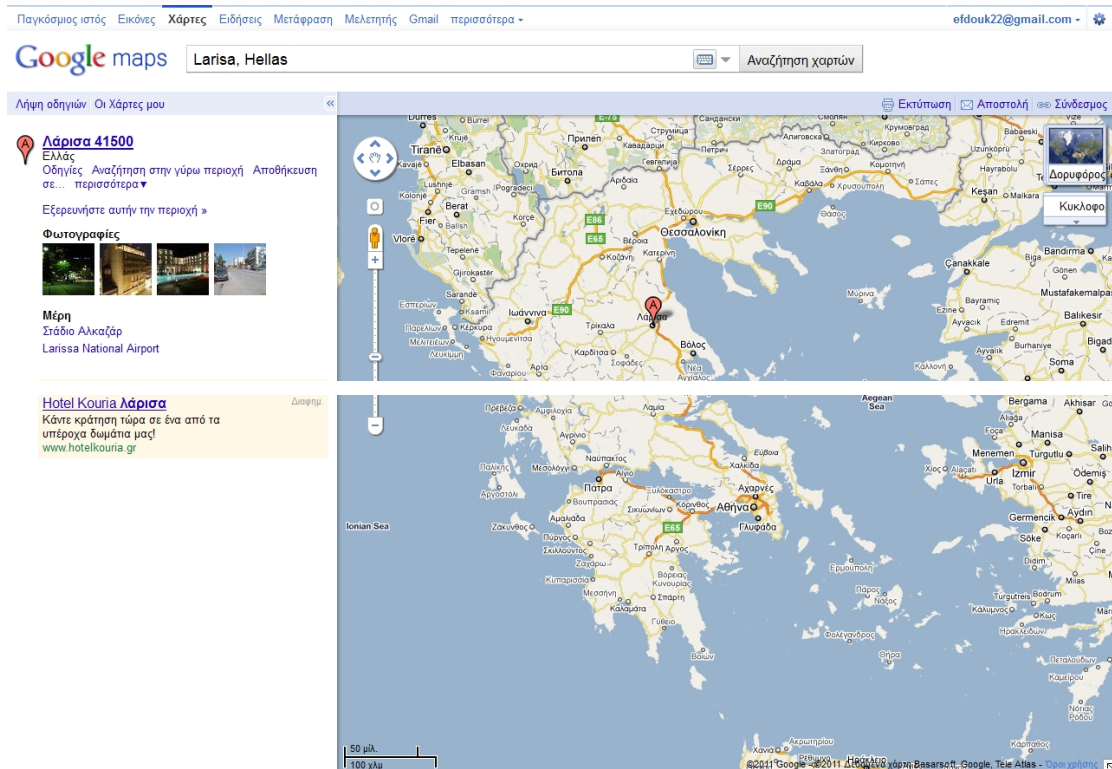
Τα επιπλέον pixels είναι σημαντικά, καθώς απαλύνουν τις ακμές στα γράμματα με αποτέλεσμα καθαρότερο κείμενο στην οθόνη. Μέσα στο 2012 αναμένεται σημαντική εξέλιξη στα λειτουργικά συστήματα των tablet pc. Ήδη έχουν ακουστεί πολλά για το Android 4.0 το επονομαζόμενο Icecream Sandwich- με το οποίο η Google επιχειρεί να γεφυρώσει το κενό μεταξύ των smartphones που τρέχουν Android 2.x, και των tablets με το Android 3.x. Η Asus έχει ανακοινώσει ότι θα παρέχει ένα firmware upgrade για το Transformer Prime, το οποίο αυτή τη στιγμή πωλείται με Android 3.2.

1.5 Χάρτες Google - Google Maps

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

Η τεχνολογία των χαρτών Google (εικόνα 1.17) χρησιμοποιεί μια στενή παραλλαγή της προβολής Mercator (ένας κυλινδρικής προβολής χάρτης του κόσμου), με αποτέλεσμα να μην μπορεί να δείξει περιοχές γύρω από τους πόλους. Ένα σχετικό προϊόν είναι το Google Earth, ένα αυτόνομο πρόγραμμα το

οποίο προσφέρει περισσότερα χαρακτηριστικά ως προς την παγκόσμια προβολή, συμπεριλαμβανομένων των πολικών περιοχών.



Εικόνα 1.17: Χάρτες Google

1.5.1 Υλοποίηση

Η τεχνολογία αυτή χρησιμοποιεί εκτενώς την γλώσσα Javascript. Όταν ο χρήστης σύρει τον χάρτη τότε φορτώνονται από τον εξυπηρετητή τα τετράγωνα του πλέγματος, τα οποία ακολούθως εισάγονται στην σελίδα. Όταν ο χρήστης ψάχνει για μια επιχείρηση τα αποτελέσματα φορτώνονται στο παρασκήνιο για εισαγωγή στο πλαίσιο πλαίσιο και τον χάρτη, με αποτέλεσμα να αποφεύγεται η επαναφόρτωση της σελίδας. Οι τοποθεσίες καθορίζονται δυναμικά, με την τοποθέτηση μιας κουκκίδας (pin) πάνω από τις εικόνες του χάρτη. Επιπρόσθετα, η σελίδα χρησιμοποιεί JSON για μεταφορά δεδομένων, αντί XML, για σκοπούς απόδοσης.

1.5.2 Επεκτασιμότητα

Το γεγονός ότι οι χάρτες Google είναι τεχνολογία γραμμένη σχεδόν εξολοκλήρου σε Javascript και XML, επιτρέπει σε τελικούς χρήστες να παράγουν client-side scripts και server-side hooks, τα οποία επιτρέπουν με την σειρά τους σε ένα χρήστη ή μια ιστοσελίδα να εισάγουν επεκταμένες ή τροποποιημένες λειτουργίες στην διεπαφή των χαρτών Google.

1.5.3 Προγραμματιζόμενη διεπαφή εφαρμογής χαρτών Google (Google Maps API)

Τον Ιούνιο του 2005, η Google διέθεσε το Google Maps API, με σκοπό να επιτρέψει στους προγραμματιστές να ενσωματώσουν τους Χάρτες Google στις ιστοσελίδες τους.

Το API αυτό είναι δωρεάν για εμπορική χρήση, με την προϋπόθεση ότι η ιστοσελίδα για την οποία θα χρησιμοποιηθεί, δεν χρεώνει τους χρήστες της για πρόσβαση σε αυτήν. Ιστοσελίδες που χρεώνουν και θέλουν να χρησιμοποιήσουν το Google Maps API, μπορούν να αγοράσουν το Google Maps Premier. Με την χρήση της διεπαφής, μπορεί κάποιος να ενσωματώσει την ιστοσελίδα Χαρτών Google σε εξωτερική ιστοσελίδα με την δυνατότητα τοποθέτησης δικών του δεδομένων πάνω από τους χάρτες.

Αν και αρχικά το Google Maps API ήταν ένα API για Javascript, μετέπειτα επεκτάθηκε ώστε να συμπεριλάβει ένα API για Adobe Flash και μια υπηρεσία για ανάκτηση στατικών εικόνων χαρτών. Επίσης συμπεριέλαβε δικτυακές υπηρεσίες για την εκτέλεση γεωκωδικοποιήσεων και δημιουργία κατευθύνσεων οδήγησης.

1.5.4 Χάρτες Google για κινητά

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

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

- Android (η σχέση του Android με την υπηρεσία αυτή μελετάται στην συνέχεια αναλυτικότερα).
- iOS (iPhone/iPod Touch/iPad)
- Windows Mobile
- Nokia/Symbian (σε συγκεκριμένα μόνο μοντέλα)
- Symbian OS (UIQ v3)
- BlackBerry
- κ.α

1.5.5 Λοιπές χρήσεις των χαρτών Google από την Google

- Google Ditu
 - Η κινέζικη έκδοση των Χαρτών Google.
- Google Moon
 - Ένα εργαλείο με εικόνες της σελήνης ενσωματωμένες στην διεπαφή των Χαρτών Google.
- Google Mars
 - Με στοιχεία που συνέλλεξε η NASA στις δύο αποστολές της στον πλανήτη Άρη.
- Google Sky
 - Online εργαλείο χαρτογράφησης του ορατού σύμπαντος χρησιμοποιώντας φωτογραφίες που λήφθηκαν από το διαστημικό τηλεσκόπιο Hubble.
- Google Ride Finder
 - Εργαλείο που ενσωματώνεται σε GPS συστήματα αυτοκινήτων για υπηρεσίες ταξί και λιμουζίνων.
- Google Transit
 - Υπηρεσία που υπολογίζει διαδρομές, χρόνους διελεύσεως και κόστη.
- Google Biking directions

- Εργαλείο που αναζητάει κατευθύνσεις και μονοπάτια για οδούς ποδηλάτων.
- Google My Maps
 - Εργαλείο για δημιουργία προσωπικών και επαγγελματικών χαρτών.
- Google Street View
 - Πανοραμική άποψη των οδών.

1.5.6 Οι χάρτες Google και το Android

Το Android προσφέρει στους προγραμματιστές την δυνατότητα να ενσωματώσουν κάποιο χάρτη στην εφαρμογή τους. Αυτό μπορεί να γίνει με δύο τρόπους. Σύμφωνα με τον πρώτο τρόπο, που είναι και ο πιο απλός, ο προγραμματιστής μπορεί να ενσωματώσει στην οθόνη ένα στοιχείο το οποίο καλείται 'WebView'. Στο 'WebView', ο προγραμματιστής μπορεί να απεικονίσει οποιαδήποτε διαδικτυακή εφαρμογή με αντίστοιχο τρόπο που αυτή απεικονίζεται σε έναν φυλλομετρητή και κατ' επέκταση τους Χάρτες Google.

Η Google όμως θέλοντας να δώσει περισσότερη ελευθερία στους προγραμματιστές, δημιούργησε ένα άλλο στοιχείο διεπιφάνειας χρήστη, το 'MapView' γεγονός που μας οδηγεί στην δεύτερη εναλλακτική. Το MapView σε συνδυασμό με την κλάση MapActivity και τις διεπαφές mapping APIs, είναι ένα ισχυρό εργαλείο στα χέρια οποιουδήποτε προγραμματιστή που θέλει να δημιουργήσει καινοτόμες και πρωτοποριακές εφαρμογές με την χρήση των χαρτών. Με αυτό τον τρόπο ο κατασκευαστής μπορεί να χρησιμοποιήσει υπάρχων χάρτες και να τους τροποποιήσει όπως αυτός θέλει, με την δυνατότητα να χειριστεί τις αλληλεπιδράσεις του χρήστη με τον χάρτη, να τοποθετήσει δικά του δεδομένα πάνω σε αυτόν κ.τ.λ.

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

η Google απαιτεί την απόκτηση από τον εκάστοτε προγραμματιστή ενός map-api key, κλειδί το οποίο προσφέρει η ίδια (<http://code.google.com/android/maps-api-signup.html>). Για την απόκτηση του κλειδιού ο ενδιαφερόμενος πρέπει να δώσει το αποτύπωμα MD5 του ψηφιακού πιστοποιητικού το οποίο θα χρησιμοποιηθεί για την υπογραφή της εφαρμογής.

Κεφάλαιο 2 : Ανάπτυξη Εφαρμογών στο Android

2.1 Κύκλος Ανάπτυξης Εφαρμογής

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

2.1.1 Εγκατάσταση Λογισμικού

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

Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει την λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας. Ιδανικά ο developer θα διαθέτει έναν αριθμό διαφορετικών φυσικών συσκευών ώστε να δοκιμάσει ο ίδιος πως συμπεριφέρεται η εφαρμογή του σε κάθε περίπτωση, όμως αυτή η πρακτική μπορεί να αποδειχθεί πολυδάπανη και χρονοβόρα. Εδώ αναλαμβάνουν δράση η ευελιξία των AVDs, για τις οποίες θα γράψουμε περισσότερα παρακάτω.

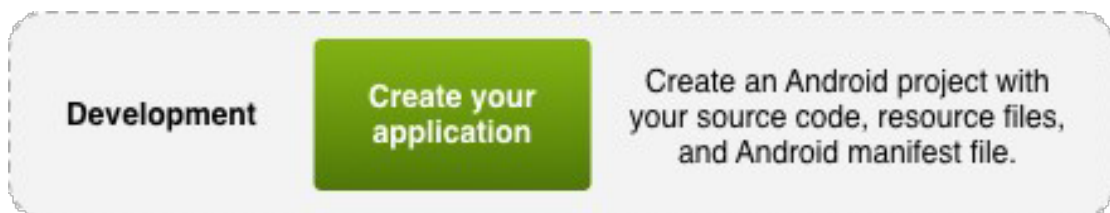


Εικόνα 2.1: Πρώτο βήμα - Εγκατάσταση Λογισμικού

2.1.2 Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής

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

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



Εικόνα 2.2: Δεύτερο βήμα - Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής

2.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής

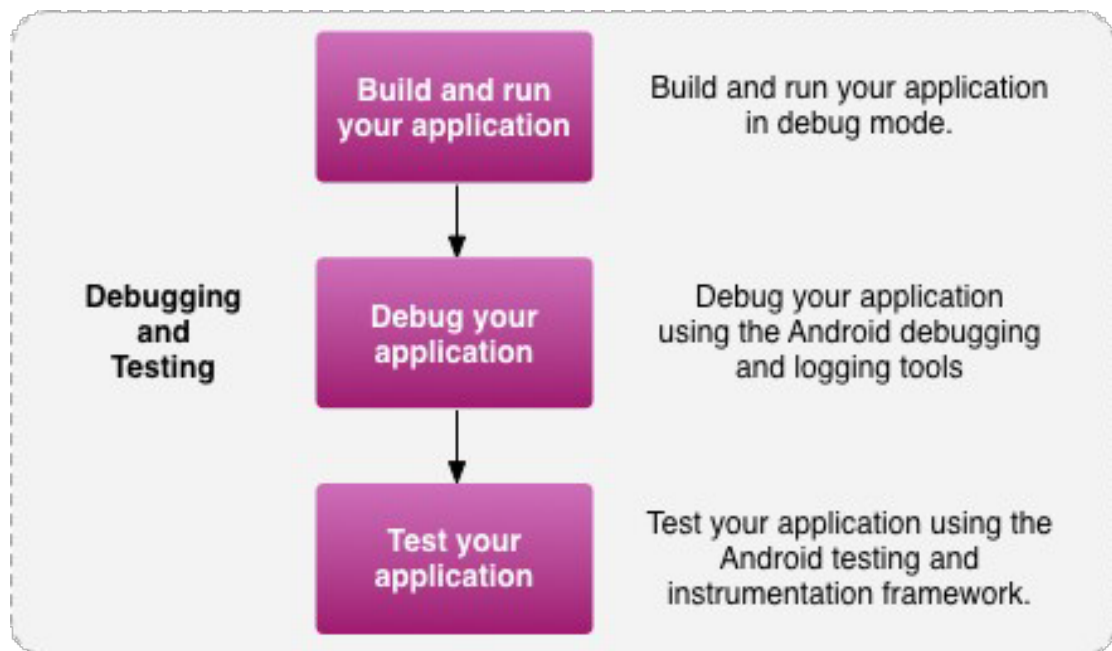
Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείτε από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω.

Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode. Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης (IDE) προϋποθέτουν ότι ο κώδικας δεν έχει κανένα συντακτικό λάθος, αλλιώς ειδοποιούν τον χρήστη να τα διορθώσει. Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB push εντολής. Για το ADB θα μιλήσουμε εκτενώς παρακάτω.

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

Υπάρχουν φυσικά και άλλα εργαλεία τα οποία θα αναλυθούν εκτενώς παρακάτω. 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 2 βήματος, της αποσφαλμάτωσης. Ένα προαιρετικό στάδιο είναι η “Δημόσια δοκιμαστική φάση” της εφαρμογής. Σε αυτή τη φάση εθελοντές προσφέρονται να δοκιμάσουν τις λειτουργίες της εφαρμογής στις συσκευές τους και να αναφέρουν προβλήματα, παρατηρήσεις, προτάσεις και άλλα σχόλια που μπορεί προκύψουν από τη χρήση της εφαρμογής.

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



Εικόνα 2.3: Τρίτο βήμα - Δοκιμή και Debugging της εφαρμογής

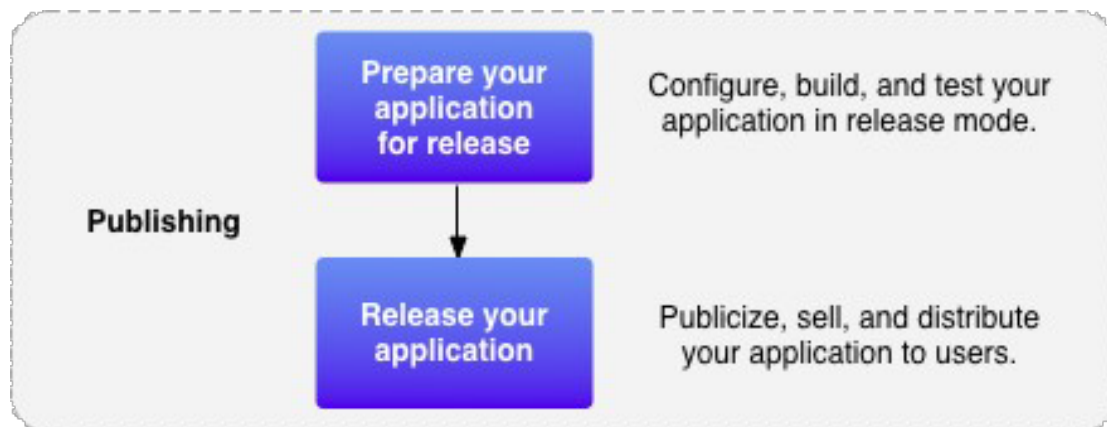
2.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό

Στο τέταρτο και τελευταίο στάδιο της ανάπτυξης ο προγραμματιστής έχει να κάνει μερικές τελευταίες κινήσεις. Πρώτον πρέπει να έχει διορθώσει όλα τα σφάλματα που προέκυψαν από την διαδικασία αποσφαλμάτωσης, να κάνει τις τελευταίες ρυθμίσεις και tweaks της εφαρμογής, 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 developer, 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 διαθέσει την εφαρμογή δωρεάν. Τη στιγμή που γράφονται αυτές οι γραμμές δεν είναι εφικτό ακόμη για τους Έλληνες Developers να

διαθέσουν εφαρμογές επί πληρωμή, με την Google να μην έχει ανακοινώσει ακόμη πότε αυτό θα γίνει εφικτό.

Επίσης άλλο ένα μέσο διάθεσης μπορεί να είναι η προσωπική η εταιρική ιστοσελίδα του δημιουργού. Το μειονέκτημα φυσικά σε αυτή τη περίπτωση είναι η έλλειψη ελέγχου για updates της εφαρμογής από έναν αυτόματο μηχανισμό ελέγχου και λήψης όπως είναι τα διάφορα marketplaces.



Εικόνα 2.4: Τέταρτο βήμα - Δημοσίευση της Εφαρμογής

2.2 Android SDK

Το Android SDK (Software Developers Kit) αποτελεί μια συλλογή εργαλείων και βιβλιοθηκών που καθιστούν εφικτή την ανάπτυξη εφαρμογών στο Android. Τι στιγμή που γράφονται αυτές οι γραμμές, το SDK έχει φτάσει στην έκδοση r19 η οποία υποστηρίζει το Android 4.0.3. Το λογισμικό ανάπτυξης λοιπόν περιλαμβάνει μια μεγάλη λίστα με εργαλεία ανάπτυξης. Σε αυτά περιλαμβάνονται:

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

Το SDK υποστηρίζει πολλά δημοφιλή λειτουργικά συστήματα συμπεριλαμβανομένων όλων των σύγχρονων διανομών Linux, το MAC OS X 10.4.9 και μεταγενέστερα, και τα Windows XP και τις μεταγενέστερες εκδόσεις.

Το λογισμικό ανάπτυξης αποτελείται από πακέτα τα οποία βρίσκονται αποθηκευμένα σε ένα επίσημο repo της Google, και ο προγραμματιστής μπορεί να κατεβάσει πέραν των βασικών πακέτων, και άλλα τα οποία υποστηρίζουν παλαιότερες εκδόσεις του Android, ή άλλες συσκευές εκτός κινητών συσκευών (πχ Google TV Addon).

Όσον αφορά την υποστήριξη παλαιότερων εκδόσεων του Android, το SDK κάνει εφικτή την υποστήριξη σε αυτές δίνοντας στον προγραμματιστή την δυνατότητα να στοχεύσει αυτός σε πια APIs θα απευθύνεται η εφαρμογή του. Αυτό είναι αναγκαίο λόγω του ότι πολλοί χρήστες έχουν παλαιότερες λειτουργικές συσκευές οι οποίες κυκλοφορήσαν με παλαιότερες εκδόσεις του Android (πχ 1.6 ή 2.1), και ο κατασκευαστής της συσκευής δεν έχει ή δεν πρόκειται να βγάλει αναβάθμιση για την συσκευή τους. Το πρόβλημα αυτό είναι γνωστό σαν διάσπαση του Android (Android Fragmentation) και θα αναλυθεί εκτενώς παρακάτω.

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

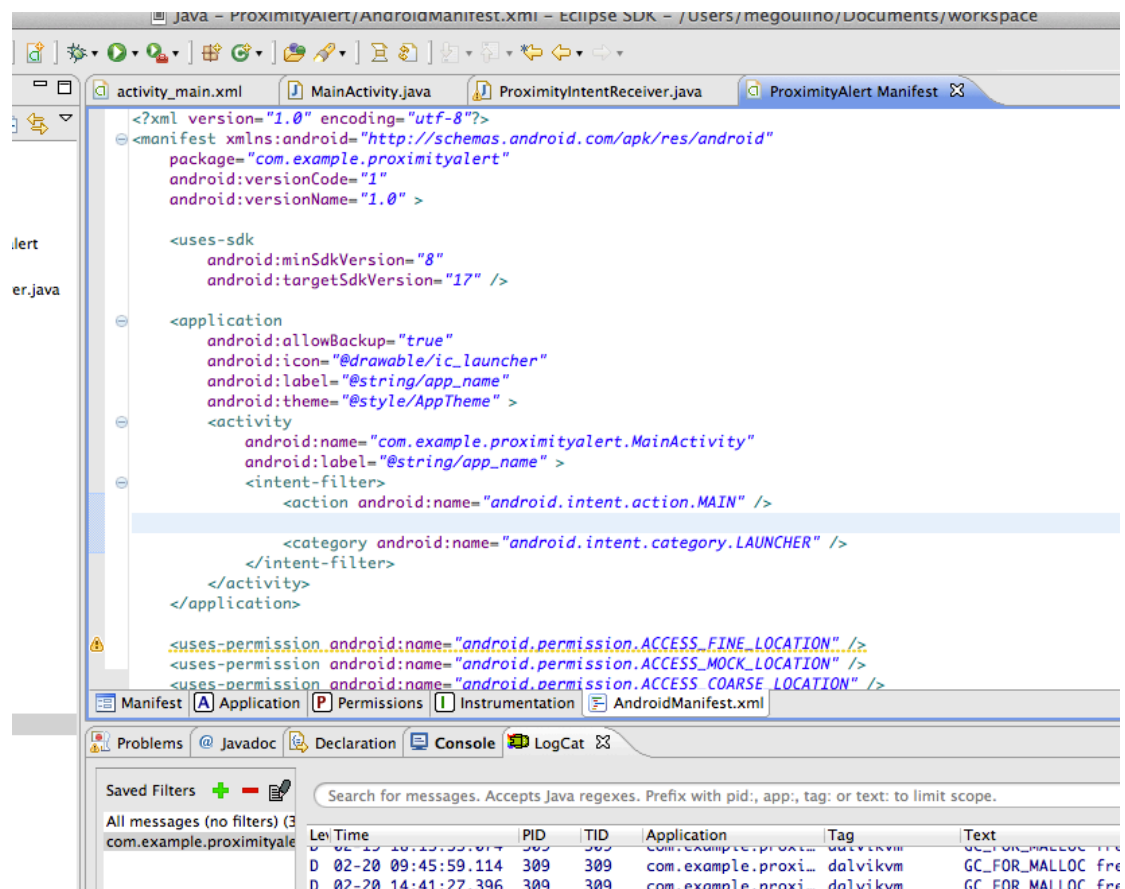
Ο προγραμματισμός στο Android βασίζεται στην γλώσσα Java και ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει κώδικα για να επεξεργαστεί τα αρχεία *.Java και *.XML και μετέπειτα να τα κάνει compile μέσω γραμμής εντολών χρησιμοποιώντας το JDK (Java Development Kit). Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός στον χρήστη γιαυτό συνίσταται η χρήση ενός IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse ή το Netbeans.

Η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό το

ADT plugin, το οποίο παρέχει σύνδεση με το Android SDK με όλες τις δυνατότητες που περιλαμβάνει αυτό. Επίσης το plugin παρέχει σύνδεση με τον AVD Manager, για διαχείριση και εκκίνηση από το GUI του, εικονικών συσκευών Android για δοκιμές και debugging των εφαρμογών.

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

Η επιλογή ενός IDE που κάνει όλη την πολύπλοκη δουλειά για μας είναι προφανής λοιπόν. Επίσης τα περισσότερα παραδείγματα και άρθρα για το Android στηρίζονται στο γεγονός ότι η πλειονότητα των developers χρησιμοποιεί το Eclipse μαζί με το ADT plugin οπότε ξεκινάμε με αυτό σαν δεδομένο.



Εικόνα 2.5: Προεπιλεγμένη διάταξη του Eclipse IDE

2.4 Προκλήσεις ανάπτυξης εφαρμογών στο Android

Η ανάπτυξη εφαρμογών στο προγραμματιστικό περιβάλλον του Android είναι μια αρκετά απαιτητική διαδικασία, διότι απαιτεί την υποστήριξη εκατοντάδων συσκευών, την υλοποίηση και υποστήριξη ενός λειτουργικού περιβάλλοντος διεπαφής χρήστη, και άλλα πολλά τα οποία θα αναλύσουμε παρακάτω.

2.4.1 Android Design Guidelines

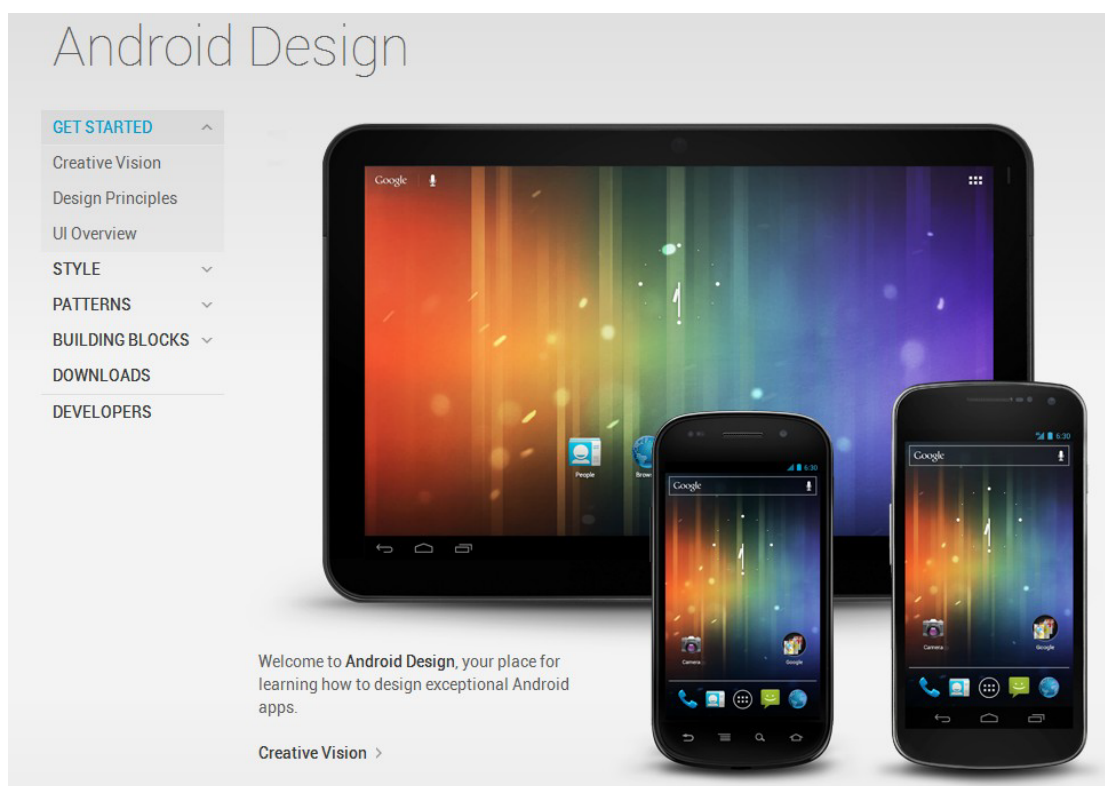
Η υλοποίηση μιας νέας εφαρμογής στο Android αλλά και στα υπόλοιπα λειτουργικά συστήματα, ξεκινάει από τις λειτουργικές απαιτήσεις, δηλαδή τις δυνατότητες και λειτουργίες που θα υποστηρίζει η εφαρμογή, και συνεχίζει με τον σχεδιασμό του UI layout που θα δίνει πρόσβαση στον χρήστη στις παραπάνω λειτουργίες. Ο σχεδιασμός λοιπόν έχει μεγαλύτερη σημασία από τις ίδιες τις λειτουργίες της εφαρμογής μιας και είναι το σημείο πρόσβασης προς αυτές.

Μια κακοσχεδιασμένη εφαρμογή η οποία κρύβει τις λειτουργίες τις πίσω από πολλά κουμπιά και μενού ενδέχεται να μπερδέψει τον χρήστη και ίσως και να τον αποτρέψει από το να την χρησιμοποιήσει. Αυτό φυσικά δεν είναι επιθυμητό γιαυτό και υπάρχουν κάποια ενδεικτικά επίσημα Guidelines (οδηγίες) τα οποία υποδεικνύουν στους developers τις ιδεατές και μη συμπεριφορές κατά τον σχεδιασμό της εφαρμογής τους.

Στα πρώτα χρόνια του Android ο σχεδιασμός των εφαρμογών ήταν είτε καθαρή μεταφορά (port) από άλλο OS (πχ iOS, Symbian) και συνήθως δεν ακολουθούσε καμία κοινή γραμμή σχεδιασμού με τις υπόλοιπες εφαρμογές κάτι που πολλές φορές μπερδευε τον χρήστη που είχε συνηθίσει το κοινό layout πολλών εφαρμογών των υπολοίπων mobile OSes. Αυτό η Google προσπάθησε να το αλλάξει με την έλευση του Android 4.0 (Ice Cream Sandwich) όποτε και

δημοσίευσε στο Ίντερνετ τη σελίδα Android Design για να καθοδηγήσει τους developers σε μία κοινή γραμμή ανάπτυξης εφαρμογών ώστε να πετύχει αύξηση λειτουργικότητας.

Η αύξηση λειτουργικότητας επιτυγχάνεται λόγω του ότι ο χρήστης θα έχει γνωστά σημεία επαφής σε κάθε εφαρμογή οπότε δεν θα χρειάζεται να ψάχνει εκ νέου πως να επιστρέψει στην αρχική οθόνη ή που βρίσκεται το μενού των επιλογών, κλπ.



Εικόνα 2.6: Αρχική Σελίδα Android Design

Στο site υπάρχει πληθώρα παραδειγμάτων “καλού σχεδιασμού” τα οποία αφορούν την χρήση της Action Bar, την χρήση των swappable tabs, την δυνατότητα δηλαδή να αλλάζουμε οθόνες σέρνοντας αριστερά η δεξιά το δάχτυλό μας στην οθόνη της συσκευής, κλπ. Επίσης υπάρχει και η ενότητα “Downloads” στην οποία υπάρχει ένα πλήρες πακέτο εικονιδίων τα οποία μπορούμε να χρησιμοποιήσουμε στα μενού των εφαρμογών μας η όπου αλλού θέλουμε.

Οι οδηγίες φυσικά είναι ενδεικτικές και όχι αναγκαστικές. Αν κάποιος developer θέλει να “παρεκτραπεί” και να δώσει κάποιο εντελώς διαφορετικό interface το οποίο όμως θα εξυπηρετεί καλύτερα τους δικούς του σκοπούς, μπορεί να το κάνει ελεύθερα. Αυτή άλλωστε είναι και η ομορφιά του Android!

2.4.2 Υποστήριξη πολλαπλών συσκευών

Το λειτουργικό σύστημα Android τρέχει σε μια πληθώρα συσκευών οι οποίες μπορεί να έχουν πολύ διαφορετικές προδιαγραφές η μία από την άλλη. Η διαφοροποίηση των συσκευών εντοπίζεται:

- Στις πολλές εκδόσεις του Android που υπάρχουν. Αυτή τη στιγμή που γράφονται αυτές οι γραμμές υπάρχουν 9 διαθέσιμες εκδόσεις (1.5-4) με κυρίαρχη έκδοση την 2.3 Gingerbread.
- Στην μεγάλη ποικιλία hardware που κυκλοφορεί στην αγορά.

Υπάρχουν συσκευές με επεξεργαστή στα 600Mhz και 256MB διαθέσιμης μνήμης RAM και υπάρχουν και συσκευές με επεξεργαστή 4 πυρήνων (Quad Core) στα 1.5Mhz και 1GB μνήμης RAM. Εκτός από τις διαφορές σε επίπεδο microchip η κύρια διαφορά μεταξύ των συσκευών εντοπίζεται στην μεγάλη ποικιλία διαστάσεων οθόνης και πυκνότητας pixel.

Ο developer λοιπόν για να κάνει την εφαρμογή του προσβάσιμη σε όσο τον δυνατόν περισσότερες συσκευές χρηστών, πρέπει να λάβει σοβαρά υπόψιν του τις 2 παραπάνω παραμέτρους και να σχεδιάσει την εφαρμογή του έτσι ώστε αυτή να υποστηρίζει την πλειονότητα των συσκευών. Βέβαια αυτό σημαίνει συνεχή προσαρμογή της εφαρμογής στις νέες συνθήκες που μπορεί να προκύψουν, και χρήση των νέων δυνατοτήτων που ενδεχομένως θα παρέχει μια νέα έκδοση του λειτουργικού, χωρίς να επηρεάζεται η υποστήριξη στις παλαιότερες συσκευές.

Αυτό είναι μεγάλο ζήτημα το οποίο θα αναλυθεί εκτενώς παρακάτω.

2.4.2.1 Υποστήριξη παλαιότερων εκδόσεων του Android

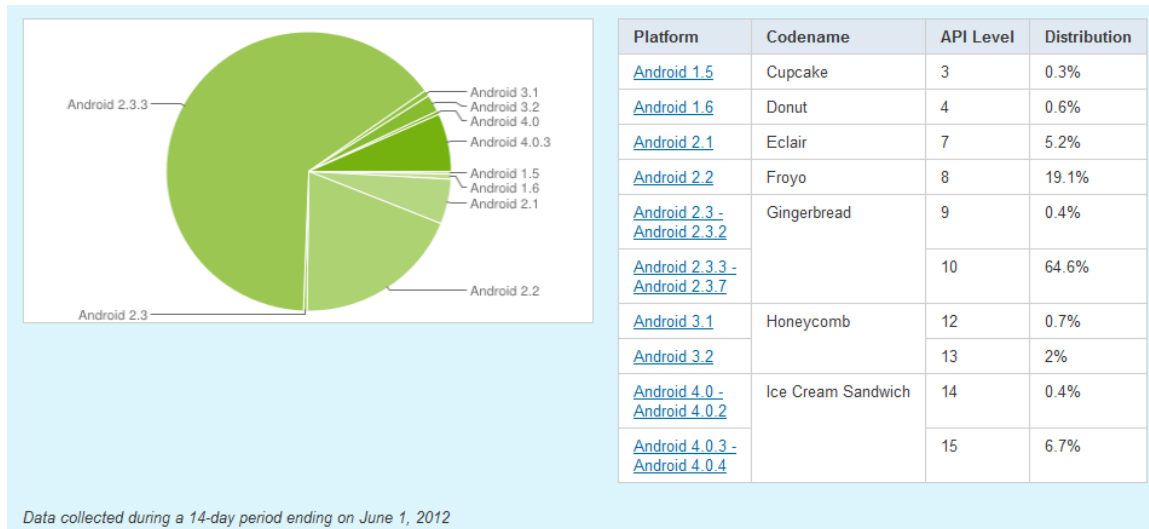
Όπως γράψαμε παραπάνω αλλά και στο εισαγωγικό κομμάτι της εργασίας, αυτή τη στιγμή υπάρχουν 9 διαθέσιμες εκδόσεις του Android με πιο πρόσφατη την έκδοση 4.0.3 με κωδική ονομασία “Ice Cream Sandwich” ή “ICS” και κυρίαρχη έκδοση την 2.3.3 “Gingerbread”.

Αυτή η συνεχής εξέλιξη της πλατφόρμας αποτελεί πλεονέκτημα αλλά και πρόκληση για τον προγραμματιστή ο οποίος θα πρέπει να ακολουθεί τις εξελίξεις και να χρησιμοποιεί τις νέες δυνατότητες που του προσφέρει η κάθε έκδοση, χωρίς να παραγκωνίζει την υποστήριξη στις παλαιότερες εκδόσεις του Android. Αυτό είναι ένα σημαντικό πρόβλημα καθότι κάποια νέα features δεν υποστηρίζονται στα παλιότερα APIs και άρα καθιστούν αδύνατη τη χρήση τους σε κάποια παλαιότερη έκδοση του Android.

Η Google έχει προσπαθήσει να λύσει αυτό το πρόβλημα βγάζοντας μαζί με κάθε νέα έκδοση του λειτουργικού της, και μιας “βιβλιοθήκης συμβατότητας” η οποία αναλαμβάνει να κάνει διαθέσιμα τα νέα εργαλεία στα παλιότερα APIs.

Σε γενικές γραμμές η συγκεκριμένη λύση λειτουργεί αρκετά ικανοποιητικά αλλά δυστυχώς δεν μεταφέρονται πάντα όλες οι νέες δυνατότητες στις παλαιότερες εκδόσεις του λειτουργικού συστήματος με αυτό τον τρόπο.

Τρανό παράδειγμα αυτού είναι η υποστήριξη της ActionBar η οποία προστέθηκε στην έκδοση 3.0 του Android και πλέον ανήκει στις βασικές οδηγίες σχεδιασμού εφαρμογών (android design guidelines). Η βιβλιοθήκη υποστήριξης την κάνει διαθέσιμη στις παλαιότερες εκδόσεις, αλλά με σαφείς περιορισμούς στην χρήση της! Ευτυχώς το συγκεκριμένο feature το υποστηρίζουν άψογα 2 βιβλιοθήκες που έχουν αναπτυχθεί από προγραμματιστές του Android, και έχει καλυφτεί το κενό που άφησε η Google.



Εικόνα 2.7: Στατιστικά στοιχεία ενεργών συσκευών Android – Ιούνιος 2012

Όπως βλέπουμε από τα παραπάνω στατιστικά στοιχεία που ενημερώνονται αυτόματα κάθε 2 εβδομάδες από την Google, τη μερίδα του λέοντος κατέχουν οι εκδόσεις 2.3 με 65% και η 2.2 με ποσοστό 19,1%. Η νεότερη έκδοση του Android, 4.0 κατέχει μόλις το 7,1% παρότι κυκλοφόρησε τον Οκτώβριο του 2011, δηλαδή 7 μήνες από τη στιγμή που γράφονται αυτές οι γραμμές. Αυτό δυστυχώς πρόκειται για το φαινόμενο διάσπασης (fragmentation) του Android το οποίο θα αναλυθεί εκτενέστερα παρακάτω.

Γενικά μια καλή προγραμματιστική συμπεριφορά σύμφωνα με την κοινότητα, είναι η εκάστοτε εφαρμογή μας να υποστηρίζει τουλάχιστον το 90% των ενεργών συσκευών όσον αφορά την έκδοση Android που φοράνε. Αυτό από ότι βλέπουμε από το παραπάνω γράφημα είναι αρκετά εύκολο καθώς υποστηρίζοντας μόνο τις εκδόσεις 2.2 και 2.3 αυτή τη στιγμή, έχουμε υποστηρίξει το ~85% των ενεργών συσκευών! Παρόλα αυτά έχουμε όμως αποκλείσει έναν σημαντικό αριθμό χρηστών οι οποίοι χρησιμοποιούν παλαιότερες (1.6, 2.1) αλλά και νεότερες (4.0) συσκευές.

Οι developers πρέπει να βλέπουν και να αξιολογούν το μερίδιο των συσκευών με τέτοιο τρόπο ώστε να υποστηρίζουν όσο το δυνατό περισσότερες συσκευές χρηστών χωρίς όμως να υποβαθμίζουν την ποιότητα και λειτουργικότητα της

συσκευής τους. Πρόκειται για μια λεπτή ισορροπία που επιτυγχάνεται μετά από αρκετή προσπάθεια από μέρος των developers.

Η υποστήριξη των διαφορετικών εκδόσεων ορίζεται στο αρχείο `AndroidManifest.xml` και εφόσον έχει καθοριστεί ένα κατώτατο στοχευμένο API, η εφαρμογή μας δεν μπορεί να εγκατασταθεί σε συσκευή που φοράει παλαιότερη έκδοση από αυτή που υποστηρίζει το API.

Καλή πρακτική σχεδιασμού λοιπόν είναι να στοχεύσουμε ένα αρκετά χαμηλό API το οποίο όμως δεν θα μας αναγκάσει να κάνουμε συμβιβασμούς στις λειτουργίες της εφαρμογής, και στο τέλος να κάνουμε `compile` την τελική έκδοση με τη νεότερη έκδοση του SDK ώστε να εξασφαλίσουμε υποστήριξη και στις νεότερες συσκευές.

2.4.2.2 Υποστήριξη πολλαπλών διαστάσεων οθόνης και πυκνότητας pixel

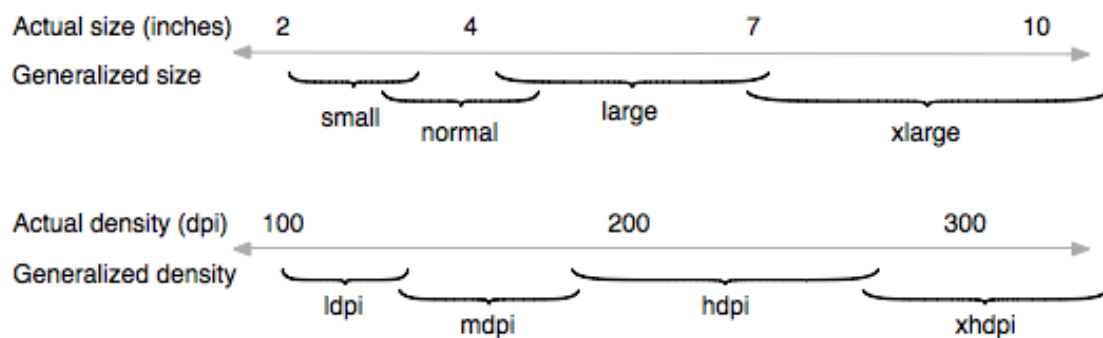
Η ταυτόχρονη υποστήριξη των πολλών εκδόσεων του Android είναι η πρώτη πρόκληση του προγραμματιστή του. Η δεύτερη μεγάλη πρόκληση είναι η ταυτόχρονη υποστήριξη των πολλαπλών διαστάσεων οθόνης που διαθέτουν οι εκατοντάδες συσκευές που κυκλοφορούν στην αγορά, και η διαφορετική πυκνότητα pixel που διαθέτει η κάθε μία από αυτές.

Όπως γράψαμε και παραπάνω το γραφικό περιβάλλον μιας εφαρμογής είναι ίσως σημαντικότερο και από τις δυνατότητες που παρέχει, καθότι ένα κακοσχεδιασμένο layout μπορεί να κάνει την εφαρμογή δύσχρηστη ή ακόμη και άχρηστη! Άρα είναι πολύ σημαντικό για τον προγραμματιστή να λάβει υπόψιν του την πληθώρα αναλύσεων και διαφορετικών διαστάσεων οθόνης που διαθέτουν οι συσκευές.

Όσον αφορά το εύρος των συσκευών μπορεί να φτάσει από τις 2.4' (μικρά smartphones) έως τις 13' (μεγάλα tablets), και οι ανάλυση αυτών των συσκευών

ξεκινάει από τα 240x320 pixels (QVGA) και φτάνει μέχρι τα 1920x1080 (Full HD) για τα νεότερα tablets! Δημοφιλείς αναλύσεις είναι η 240x320(QVGA), η 240x400 (WQVGA), η 320x480 (HVGA), και φυσικά η πιο δημοφιλής όλων η 480x800 (WVGA). Νεότερες και μεγαλύτερης οθόνης συσκευής υποστηρίζουν και μεγαλύτερες αναλύσεις όπως η 540x960 (qHD) και 720x1280(WXGA).

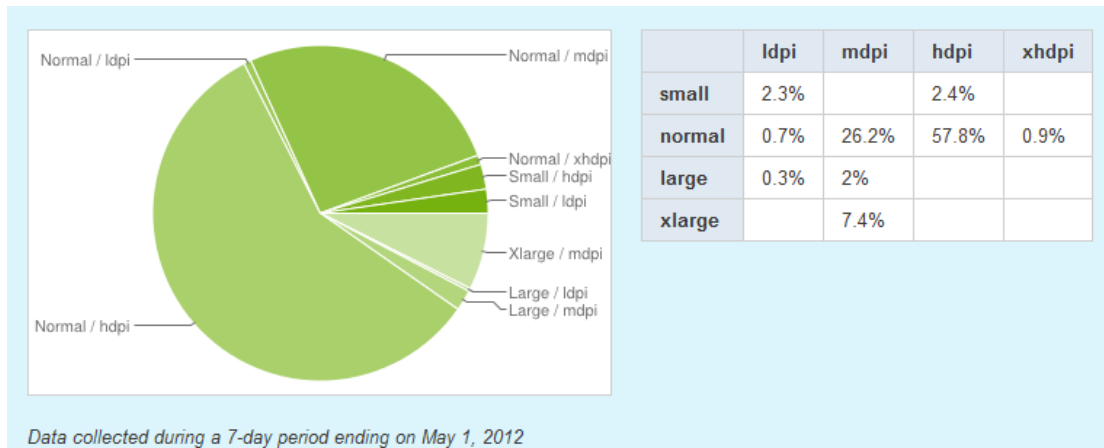
Όπως βλέπουμε υπάρχει μια πληθώρα αναλύσεων για να υποστηρίξει ο προγραμματιστής η κάθε μία με περισσότερο η λιγότερο διαθέσιμο χώρο στην οθόνη. Το android για πρακτικούς λόγους έχει χωρίσει τις διαφορετικές αναλύσεις οθονών σε τέσσερις κατηγορίες οθονών οι οποίες συσχετίζονται άμεσα με τέσσερις κατηγορίες πυκνότητας pixel ανά ίντσα.



Εικόνα 2.8: Διαχωρισμός Αναλύσεων και Πυκνότητας σε Κατηγορίες.

Όπως βλέπουμε από το παραπάνω σχεδιάγραμμα, το Android χωρίζει το μέγεθος της οθόνης σε 4 επιμέρους κατηγορίες αναλόγως το μέγεθος της σε ίντσες, και οι αναλύσεις χωρίζονται επίσης σε 4 επιμέρους κατηγορίες DPI (Dots Per Inch). Αυτό γίνεται ώστε να διευκολύνει όσο το δυνατόν περισσότερο τους προγραμματιστές, να βελτιώσουν την εμφάνιση των εφαρμογών τους με όσο το δυνατόν λιγότερο κόπο.

Όπως και στην περίπτωση των πολλών εκδόσεων του Android που έχουν πρόσβαση στο μάρκετ, η Google παρέχει στους developers την κατανομή μεγέθους οθόνης προς DPI για να βοηθήσει τους προγραμματιστές να σχεδιάσουν τις εφαρμογές τους αποδοτικότερα



Εικόνα 2.9

Από ότι βλέπουμε από το παραπάνω σχεδιάγραμμα, τα πράγματα εν τέλη δεν είναι και τόσο τραγικά για τον developer! Όπως και στις εκδόσεις των API 2 μεγέθη κατέχουν τη μερίδα του λέοντος, και τα 2 μάλιστα στις ίδιες αναλύσεις (normal) αλλά σε διαφορετικό μέγεθος DPI. Σε γενικές γραμμές οι πιο συνηθισμένες διατάξεις που πρέπει να δοκιμάσει ο developer την εφαρμογή του συνοψίζονται στον παρακάτω πίνακα.

	ldpi	mdpi	hdpi
Small	240x320 pixels 120 dpi		240x400 pixels 240 dpi
Normal		480x800 pixels 160 dpi	480x800 pixels 240 dpi

Πίνακας 2.1: Συνηθισμένες Αναλύσεις/Κατανομή pixels

Φυσικά δεν πρέπει να αγνοηθούν οι υπόλοιπες διατάξεις μεγέθους οθόνης προς dpi, αλλά όπως και στην περίπτωση της υποστήριξης των API, αυτό είναι μια λεπτή ισορροπία η οποία είναι στο χέρι του προγραμματιστή να την κρατήσει. Η πλατφόρμα του Android φυσικά εκτός από το να μας δημιουργεί πολύπλοκες

καταστάσεις προς διαχείριση, μας δίνει επιλογές και δυνατότητες ώστε να αντεπεξέλθουμε στο απαιτητικό έργο της ανάπτυξης εφαρμογής στο περιβάλλον του.

Οι δυνατότητες συνοψίζονται ως εξής:

- Ο developer μπορεί να επιλέξει ποιες οθόνες θα υποστηρίξει η εφαρμογή του με τον ίδιο τρόπο που επιλέγει πια API θα υποστηρίζονται, δηλαδή μέσω του `AndroidManifest.xml`. Αυτό σημαίνει ότι η υποστήριξη διαφορετικών οθονών είναι καθαρά επιλογή του προγραμματιστή, ο οποίος αν κρίνει αναγκαίο μπορεί να αποκλείσει την λειτουργία της εφαρμογής στις οθόνες που δεν επιθυμεί να υποστηρίξει.
- Στην συνηθέστερη περίπτωση που ο προγραμματιστής θέλει να υποστηρίξει επιπλέον αναλύσεις και διαστάσεις οθόνης πέρα από τις συνηθισμένες, το SDK του δίνει τη δυνατότητα παρέχοντας του δύο εξαιρετικές δυνατότητες:

Η πρώτη είναι η δυνατότητα χρήσης διαφορετικών layout ανά διαφορετικό μέγεθος οθόνης. Αυτό σημαίνει ότι ο developer δεν χρειάζεται να συμβιβαστεί σε ένα layout xml ώστε να καλύψει όλες τις οθόνες. Μπορεί να χρησιμοποιήσει όσα θέλει, για να καλυφθούν όσο το δυνατόν περισσότερα μεγέθη και αναλύσεις οθόνης. Δεν πρέπει να ξεχνάμε άλλωστε ότι οι μεγαλύτερες οθόνες παρέχουν και περισσότερο χώρο στον developer για να προβάλει επιπλέον υλικό, με διαφορετικό ίσως τρόπο από ότι θα το προέβλεπε σε μια μικρότερη οθόνη

Η δεύτερη δυνατότητα που παρέχει είναι αυτή της χρήσης πολλών γραφικών, διαφορετικών διαστάσεων, ώστε να εξυπηρετούνται σωστά όλες οι αναλύσεις οθόνης. Η υλοποίηση αυτής της δυνατότητας είναι εξαιρετικά απλή. Ο developer δημιουργεί φακέλους στο project του, με το όνομα της διάταξης που θέλει να παρέχει γραφικά η layout (πχ `drawable – large – hdpi`) ή χρησιμοποιεί τους υπάρχοντες φακέλους, και αποθηκεύει στον καθένα το ίδιο γραφικό (*.png, *.jpg, ή

*.gif) αλλά στην ανάλυση που επιθυμεί να προβληθεί αυτό στην εκάστοτε διαφορετική διάταξη μεγέθους οθόνης προς DPI.

2.5 Δοκιμή και Αποσφαλμάτωση (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 της θα έχει όσο το δυνατόν καλύτερη εμπειρία χρήσης! Καλύτερη εμπειρία χρήσης φυσικά οδηγεί σε καλές εντυπώσεις, και περισσότερους χρήστες με την πάροδο του χρόνου. Μια κακή εμπειρία χρήσης, οδηγεί σε δυσαρεστημένους χρήστες και άρα αποτυχία.

Ο αυτοσκοπός της ανάπτυξης εφαρμογών είναι η ικανοποίηση των αναγκών όσο το δυνατόν μεγαλύτερου αριθμού χρηστών! Αυτό φυσικά δεν επιτυγχάνεται αμέσως. Πρέπει να ακολουθηθεί 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; Μιας εφαρμογή μπορεί να αποτελείτε από μερικές απλές κλάσεις των 50 γραμμών κώδικα, ή μπορεί να αποτελείτε από δεκάδες κλάσεις χωρισμένες σε πακέτα, αποτελούμενες από δεκάδες χιλιάδες γραμμές κώδικα η καθεμία! Η έρευνα για σφάλματά στην δεύτερη περίπτωση, έχει πολύ χειρότερη αναλογία πιθανοτήτων, από την έρευνα για “ψύλλους στα άχυρα”.

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.

2.5.1 Android Debug Bridge (ADB)

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. Τον ρόλο αυτό αναλαμβάνει το Android Debug Bridge (ADB).

Πρόκειται για ένα εργαλείο γραμμής εντολών που έρχεται μαζί με το Android SDK και το οποίο αποτελείτε από 3 μέρη:

- 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 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 εργαλείου “δαίμονα” (daemon) που τρέχει στη συσκευή.
- Ο “δαίμονας” (daemon) που τρέχει σαν διεργασία παρασκηνίου στην εικονική η πραγματική συσκευή που χρησιμοποιείτε για εξομίωση.

Όταν ξεκινάει το ADB, ο client ελέγχει αν υπάρχει κάποια υπάρχουσα διεργασία του server που να εκτελείτε 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 εξομίωσης, 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 δημιουργείτε σύνδεση adb με την συσκευή.

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

`adb [-d|-e|-s <serialNumber>] <command>` όπου `-d` είναι η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη συσκευή, `-e` η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη εικονική συσκευή, `-s <serialNumber>` η απευθείας εντολή στην 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” στην συσκευή εξομοίωσης που ακούει στη θύρα 5556.

2.5.2 Εικονικές Συσκευές Android (Android Virtual Devices – AVD)

Όπως γράψαμε και παραπάνω, ο developer πρέπει πριν να εκδώσει την εφαρμογή του να την δοκιμάσει σε ένα αριθμό συσκευών για να εξασφαλίσει την ομαλή λειτουργία της σε όλες τις συνθήκες. Φυσικά το κόστος των συσκευών είναι αρκετά μεγάλο για να αποθαρρύνει τον προγραμματιστή να έχει στην κατοχή του 10-20 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 του Android.

Πρόκειται για μια συσκευή εξομοίωσης η οποία μας επιτρέπει να εξομοιώσουμε την λειτουργία και συμπεριφορά μιας κανονικής συσκευής, ορίζοντας τις επιλογές υλικού και λογισμικού που θέλουμε στον εξομοιωτή του Android. Με αυτό τον τρόπο 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 αποτελείτε 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 (dpi), 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. Μπορούμε επίσης να επιλέξουμε και μεταξύ ειδικών εκδόσεων της πλατφόρμας, μεταξύ των οποίων τις Google TV, και άλλων.
- 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 και σε μια πραγματική συσκευή.

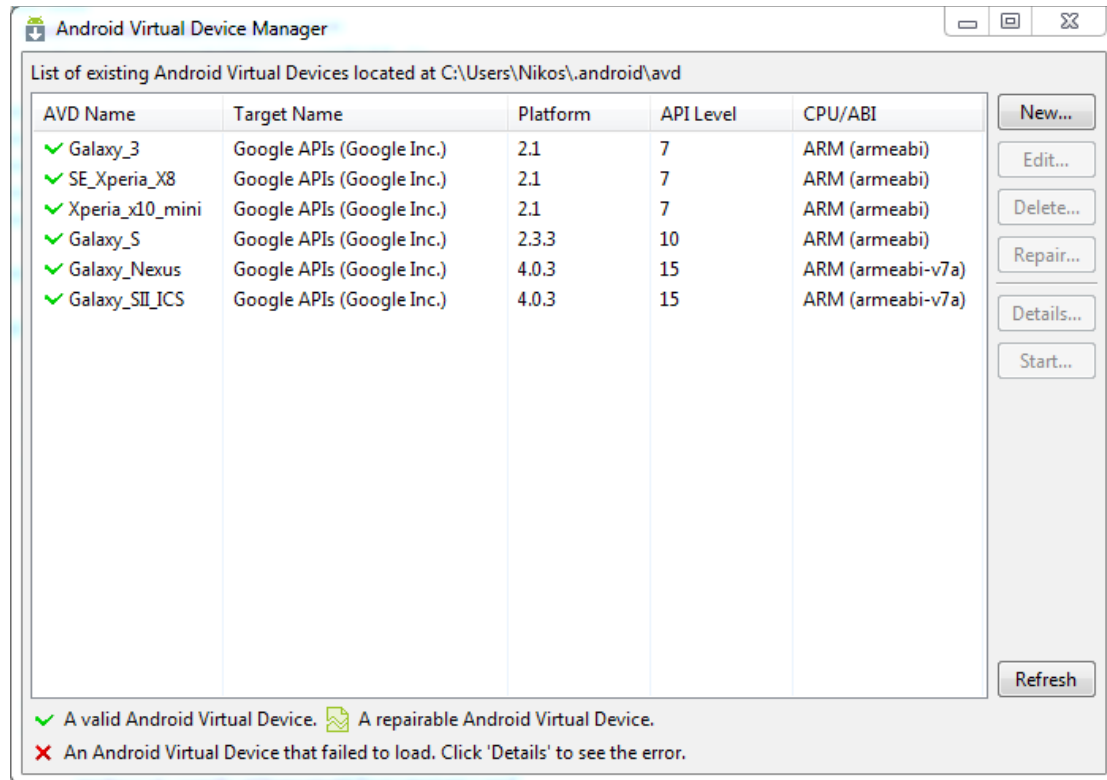
2.5.2.1 Δημιουργία διαφορετικών εικονικών συσκευών

Η δημιουργία εικονικών μηχανών είναι μια ιδιαίτερα εύκολη και γρήγορη διαδικασία. Η διαχείριση αυτών των συσκευών γίνεται από το γραφικό περιβάλλον της εφαρμογής AVD Manager, η οποία έρχεται μαζί με το SDK και στην περίπτωση που χρησιμοποιούμε το Eclipse μαζί με το ADT plugin, αυτή ενσωματώνεται στο γραφικό περιβάλλον του Eclipse.

Το γραφικό περιβάλλον του AVD Manager είναι πολύ λιτό. Αποτελείτε από μια λίστα με τις εικονικές συσκευές που έχουμε δημιουργήσει και στη δεξιά πλευρά υπάρχουν 7 κουμπιά διαχείρισης των συσκευών μας.

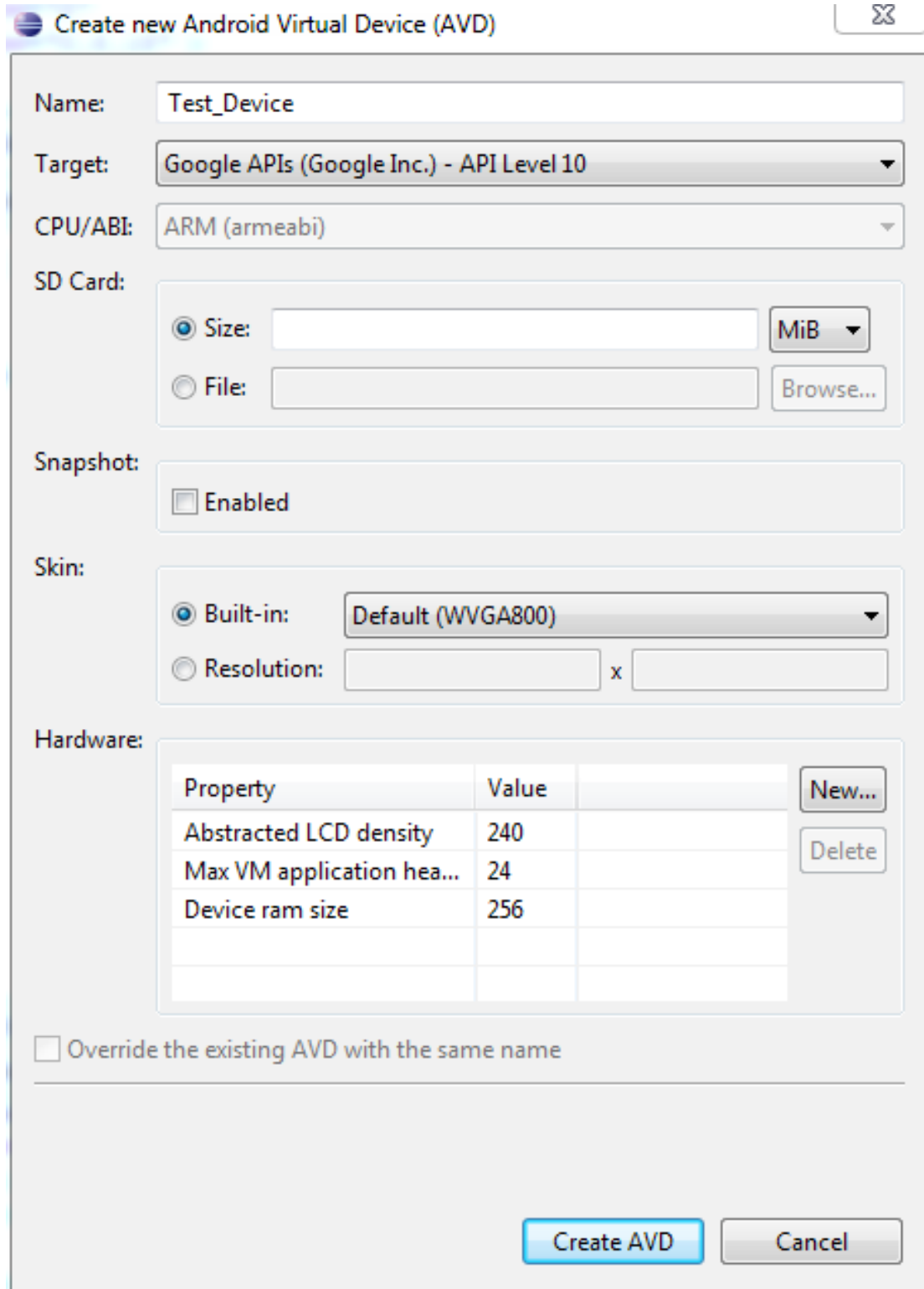
Παρακάτω (Εικόνα 2.10) εμφανίζεται η λίστα με τις εικονικές συσκευές που δοκιμάστηκε η εφαρμογή CST Connect, κατά τη διάρκεια της δοκιμαστικής της

φάσης. Φυσικά το προφίλ υλικού δεν είναι ακριβώς αυτό που διαθέτουν οι συσκευές που κατονομάζονται στην λίστα, αλλά οι διαστάσεις οθόνης και η έκδοση API, είναι ακριβής.



Εικόνα 2.10: AVD Manager

Πατώντας το κουμπί New εμφανίζεται ένα νέο παράθυρο δημιουργίας εικονικής συσκευής (Εικόνα 2.11), στο οποίο μπορούμε να ορίσουμε το όνομα, την έκδοση του Android που θέλουμε να τρέξει η συσκευή, την ανάλυση οθόνης, και τα υπόλοιπα χαρακτηριστικά του hardware που επιθυμούμε. Αφού πατήσουμε το κουμπί create AVD, η συσκευή μας είναι έτοιμη προς χρήση και προστίθεται στη λίστα μαζί με τις υπόλοιπες εικονικές συσκευές.



Εικόνα 2.11 – Παράθυρο δημιουργίας νέας Εικονικής Μηχανής

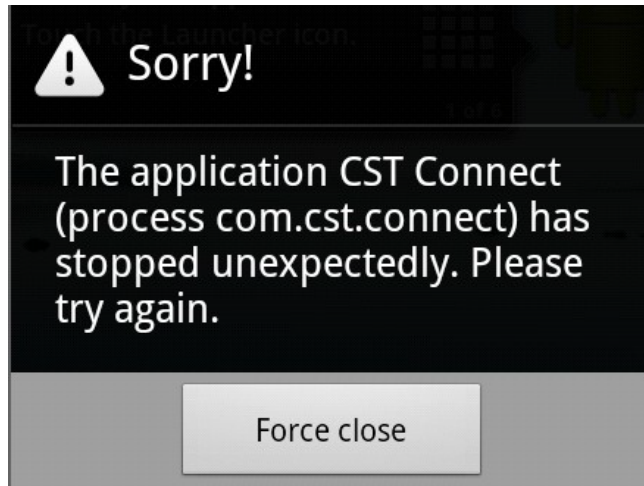
2.5.3 Εργαλείο καταγραφής συμβάντων – LogCat

Το Android διαθέτει ένα μηχανισμό καταγραφής συμβάντων, σκοπός του οποίου είναι η συλλογή και προβολή των αρχείων αποσφαλμάτωσης του συστήματος. Τα δεδομένα των διάφορων εφαρμογών αλλά και του λειτουργικού συστήματος συγκεντρώνονται σε μια σειρά από buffers, τους οποίους μετά μπορούμε να προβάλουμε και να φιλτράρουμε με την εντολή “logcat”.

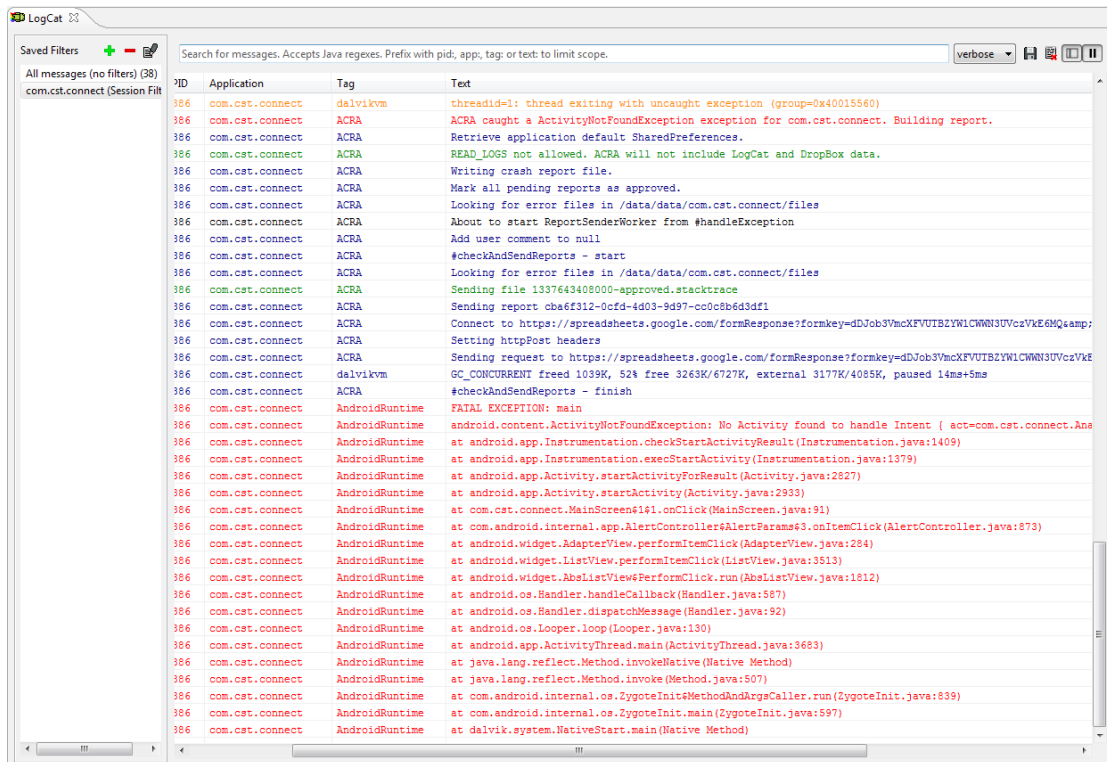
Στον προγραμματισμό υπάρχουν οι λεγόμενες “εξαιρέσεις” (exceptions), καταστάσεις δηλαδή που προκύπτουν όταν κάτι δεν πάει καλά, και αυτό έχει σαν αποτέλεσμα την διακοπή λειτουργίας του προγράμματος σε περίπτωση που δεν έχουμε φροντίσει να “χειριστούμε” την εξαίρεση. Συνηθισμένο παράδειγμα εξαίρεσης λειτουργίας του Android είναι η “NullPointerException”, η οποία μας εμφανίζεται όταν προσπαθούμε να προσπελάσουμε κάποια μεταβλητή η αντικείμενο που έχει μηδενική (Null) τιμή. Προγραμματιστικά βέβαια υπάρχει η δυνατότητα να βάλουμε δικλείδα ασφαλείας σε μερικά επίφοβα σημεία του κώδικα μας, και να σταματήσουμε την απότομη διακοπή λειτουργίας της εφαρμογής, εμφανίζοντας έναντι μόνο το μήνυμα σφάλματος στα logs του συστήματος.

Τα σφάλματα λειτουργίας μιας εφαρμογής στο Android, που προκύπτουν από εξαιρέσεις λειτουργίας, συνήθως προκαλούν τον άμεσο τερματισμό της εφαρμογής προβάλλοντας ένα παράθυρο με το όνομα της εφαρμογής που τερματίστηκε, και ένα απλό μήνυμα σφάλματος (Εικόνα 2.12), δίνοντας μας την “επιλογή” να πατήσουμε “Force Close”.

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 ως συνεπακόλουθο, της εφαρμογής που αναπτύσσουμε ώστε να εντοπίσουμε τις πηγές των σφαλμάτων, οι οποίες συνήθως αν όχι πάντα, είναι exceptions στον κώδικα μας. Το eclipse ενσωματώνει μία GUI έκδοση του LogCat (Εικόνα 2.13) για αποτελεσματικότερη αποσφαλμάτωση του κώδικα μας.



Εικόνα 2.12: Παράθυρο Force Close



Εικόνα 2.13: Η γραφική απεικόνιση του Logcat όπως εμφανίζεται στο Eclipse IDE

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, και για τη γραμμή του κώδικα αλλά και την κλάση στην οποία συνέβη το σφάλμα! Στην συγκεκριμένη περίπτωση είχαμε μια “ActivityNotFoundException” η οποία συνέβη στην κλάση “MainScreen” στη γραμμή 91.

Με το παραπάνω απλό παράδειγμα μπορούμε να αντιληφθούμε την χρησιμότητα ύπαρξης του logcat. Άμεσα μας έκανε γνωστό το ακριβές σημείο του κώδικα μας που προκάλεσε το σφάλμα και μάλιστα λόγω της φύσης της εξαίρεσης, μπορούμε να το διορθώσουμε σε χρόνο μηδέν. Η συγκεκριμένη εξαίρεση προκύπτει όταν μέσω ενός Intent κάνουμε μετάβαση από την μία Activity σε μία άλλη, και είτε συνήθως λόγω ορθογραφικού σφάλματος είτε πχ λόγω παράλειψης ενσωμάτωσης της Activity στο αρχείο AndroidManifest, η Activity που ορίζουμε στο Intent, δεν είναι διαθέσιμη.

Βέβαια υπάρχουν πολλές άλλες εξαιρέσεις (πάνω από 200 διαφορετικές) 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 κώδικα μας.

Φυσικά υπάρχουν και τα σφάλματα στον κώδικα τα οποία δεν προκαλούν αναγκαστικό κλείσιμο της εφαρμογής, αλλά παρόλα αυτά συμβάλουν στην μη σωστή λειτουργία της. Αυτού του είδους τα σφάλματα φυσικά δεν τα πιάνει ο compiler, ούτε εμφανίζονται με την 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;

Για το σκοπό μπορούμε να χρησιμοποιήσουμε το σύστημα καταγραφής του

Android, για να πάρουμε τις πληροφορίες που θέλουμε. Η κλάση Log υπάρχει για αυτό ακριβώς το σκοπό. Περιλαμβάνει έναν αριθμό από διαφορετικές μεθόδους αναλόγως με τον τύπο του σφάλματος που ψάχνουμε. Η πιο συνηθισμένη μέθοδος είναι η Log.d(), όπου το d συμβολίζει τη λέξη Debug.

2.5.4 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 παρέχει:

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

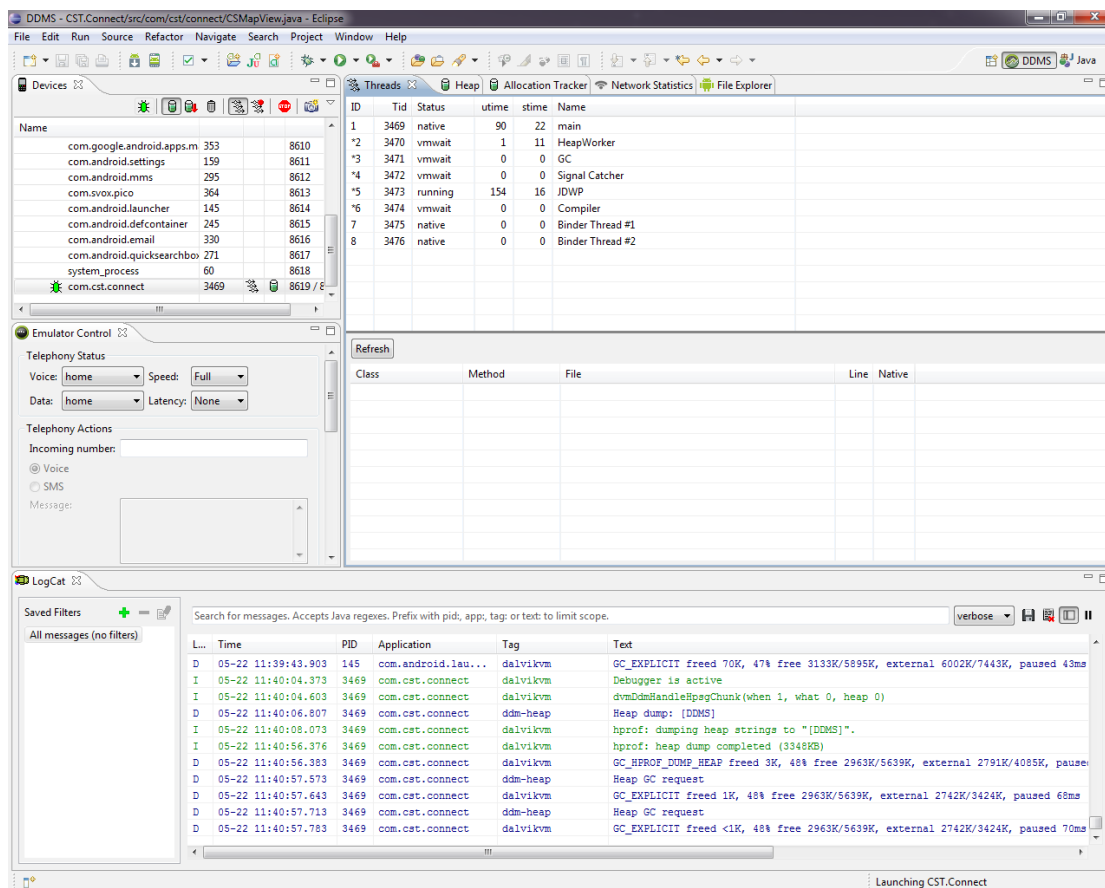
Το DDMS είναι ενσωματωμένο στο Eclipse και επίσης συμπεριλαμβάνεται στο Android SDK. Δουλεύει κανονικά είτε είναι συνδεδεμένο σε εικονική συσκευή μέσω του emulator είτε είναι συνδεδεμένο σε κανονική φυσική συσκευή. Από προεπιλογή αν είναι συνδεδεμένες δύο συσκευές ταυτόχρονα και η μία είναι εικονική,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

Σε προηγούμενο κεφάλαιο μιλώντας για την ασφάλεια στο Android, αναφέραμε ότι κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM). Κάθε μηχανή έχει και μία μοναδική θύρα επικοινωνίας στην οποία μπορεί να συνδεθεί κάποιο εργαλείο αποσφαλμάτωσης.

Κατά την εκκίνηση του το DDMS συνδέεται στο ADB. Αφού συνδεθεί κάποια συσκευή στο υπολογιστή μέσω ADB, αυτόματα δημιουργείτε μια υπηρεσία

παρακολούθησης μεταξύ του DDMS και του ADB η οποία ειδοποιεί το DDMS πότε ξεκινάει και πότε σταματάει η λειτουργία μιας εικονικής μηχανής. Όταν η VM βρίσκεται σε λειτουργία, το DDMS παίρνει το process ID της VM 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 ξεκινάει από την θύρα 8600 για την πρώτη VM και συνεχίζει για όσες εικονικές μηχανές, τρέχουν ταυτόχρονα στην συσκευή μας. Ισχύει ότι η κάθε VM έχει μία θύρα επικοινωνίας για debugging, αλλά το DDMS μπορεί να ακούσει σε πολλές θύρες ταυτόχρονα ώστε να λάβει δεδομένα από παντού.



Εικόνα 2.14: Παράθυρο DDMS

Στην παραπάνω εικόνα (Εικόνα 2.14) βλέπουμε ένα τυπικό παράθυρο του

DDMS όπως αυτό εμφανίζεται στο Eclipse. Αριστερά φαίνονται οι συνδεδεμένες συσκευές (μία εικονική μέσω emulator) και από κάτω υπάρχει η λίστα με τις διαθέσιμες εικονικές μηχανές που τρέχουν στην συσκευή. Έχοντας επιλέξει την VM της εφαρμογής com.cst.connect μας έχουν γίνει διαθέσιμα το logcat output στο κάτω μέρος της οθόνης και στο δεξιό κομμάτι της οθόνης οι υπόλοιπες πληροφορίες που συλλέγει ο debugger από την εφαρμογή.

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.

2.5.5 Application Crash Reporter for Android (ACRA)

Η συλλογή των debugging logs λοιπόν, είναι μια πολύ απλή διαδικασία η οποία πραγματοποιείται πολύ γρήγορα εφόσον συνδέσουμε την συσκευή μας μέσω του Android Debug Bridge. Τι γίνεται όμως στην περίπτωση που θέλουμε να συλλέξουμε τα logs μιας συσκευής χρήστη η οποία εμφάνισε κάποιο σφάλμα; Όπως αντιλαμβανόμαστε δεν είναι εφικτό να του ζητήσουμε να συνδέσει τη συσκευή του μέσω ADB και αφού τα πάρει να μας τα στείλει με email! Και στην περίπτωση που τυχαίναμε σε περίπτωση συνεργάσιμου και γνώστη χρήστη, πόσο εύκολο θα ήταν να του ζητάγαμε να μας στέλνει συνεχώς το stack trace του logcat ώστε να εντοπίσουμε το σφάλμα και να το διορθώσουμε;

Οι μηχανικοί του Android έχουν μεριμνήσει για αυτό και γιαυτό 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 στον 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; Η πιο συνηθέστερα, πως γίνεται να συλλέξει ο developers τα logs των χρηστών στην beta testing φάση της εφαρμογής του; Όπως είπαμε παραπάνω είναι εξαιρετικά ενοχλητικό και μη βολικό από την πλευρά μας να ζητάμε συνεχώς από τον χρήστη να μας τα στέλνει με email, και δυστυχώς το Android δεν έχει μεριμνήσει για αυτή την περίπτωση.

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 περεταίρω ανάλυση! Ένα παράδειγμα τέτοια βιβλιοθήκης η οποία χρησιμοποιείτε από πολλές δημοφιλείς εφαρμογές και χρησιμοποιήθηκε κατά κόρον και στο beta testing της εφαρμογής CST Connect, είναι η βιβλιοθήκη ACRA.

Η ACRA (αρχικά για “Application Crash Report for Android”) είναι μια βιβλιοθήκη του Android η οποία αυτόματα συλλέγει τα δεδομένα των αναφορών σφάλματος και τα μας τα αποστέλλει σε μορφή εγγράφου GoogleDoc. Η βιβλιοθήκη είναι open source (Apache License 2.0) και φιλοξενείτε στην σελίδα code.google.com/p/acra/.

Τα χαρακτηριστικά της βιβλιοθήκης είναι τα εξής:

- Δυνατότητα παραμετροποίησης ειδοποιήσεων στην συσκευή του χρήστη (silent report, ειδοποίηση toast, ειδοποίηση στην Notification Bar + παράθυρο διαλόγου.
- Είναι δυνατόν να χρησιμοποιηθεί σε όλες τις εκδόσεις του Android και όχι μόνο από την 2.2 και πάνω όπως η επίσημη βιβλιοθήκη!

- Λεπτομερείς πληροφορίες για την συσκευή στην οποία 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 φάση 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.

Όπως βλέπουμε η ACRA έχει πολλές εξαιρετικές δυνατότητες και πραγματικά λύνει τα χέρια του developer όσον αφορά το κομμάτι του remote debugging. Η πιο συνηθισμένη χρήση της βιβλιοθήκης είναι τα silent reports ώστε ο χρήστης πέρα από το παράθυρο του Force Close (Εικόνα 2.12) δεν παρατηρεί κάτι άλλο. Μετά το silent report ενημερώνεται άμεσα το έγγραφο GoogleDoc που έχουμε δημιουργήσει και συνδέσει με τη βιβλιοθήκη και μετέπειτα μπορούμε να επιλέξουμε να ενημερωνόμαστε άμεσα με email κάθε φορά που γίνεται αλλαγή στο έγγραφό μας. Με λίγα λόγια ένα σύστημα άμεσου bug reporting! Φυσικά μπορεί να χρησιμοποιηθεί ταυτόχρονα με την επίσημη βιβλιοθήκη της Google, αλλά μάλλον η δεύτερη είναι περιττή.

2.6 Κατακερματισμός του Android

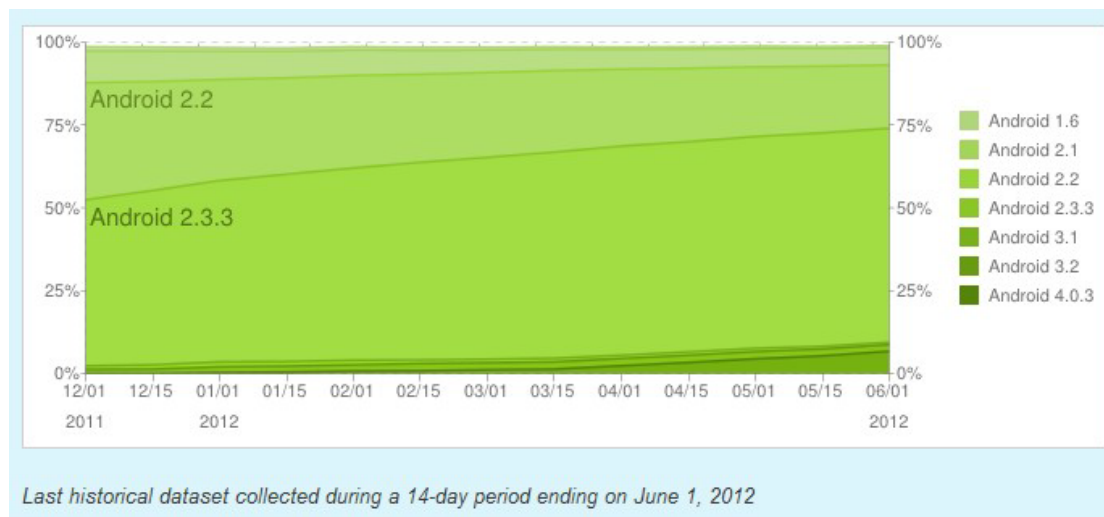
Στις προηγούμενες σελίδες έγινε λεπτομερή αναφορά σε όλες τις προκλήσεις που έχουν να αντιμετωπίσουν οι developers του Android, και επίσης έγινε λεπτομερής ανάλυση τις διαδικασίας ανάπτυξης εφαρμογών αλλά και των διαθέσιμων εργαλείων ανάπτυξης, δοκιμής και αποσφαλμάτωσης που παρέχει το Android και η κοινότητα του. Σε πολλά σημεία έγινε αναφορά στο πρόβλημα κατακερματισμού του Android. Τι είναι αυτό όμως;

Όπως γνωρίζουμε το Android ανήκει και αναπτύσσεται από την Google η οποία ανά τακτά διαστήματα χρονικά διαστήματα κάνει διαθέσιμη την νέα έκδοση με τη μορφή πηγαίου κώδικα στην κοινότητα. Ο κώδικας είναι open source φυσικά και οποιοσδήποτε μπορεί να τον κατεβάσει τοπικά και είτε να τον χρησιμοποιήσει για να φτιάξει μια έκδοση του λειτουργικού για λειτουργία σε κάποια συσκευή, είτε να συνεισφέρει στην περεταίρω ανάπτυξη του. Αυτή είναι και η δύναμη του λειτουργικού συστήματος! Η open source φύση του έχει προσελκύσει δεκάδες χιλιάδες ταλαντούχους developers ανά τον πλανήτη όπως και μεγάλες εταιρίες τηλεπικοινωνιών, οι οποίες επενδύουν εκατομμύρια δολάρια για ανάπτυξη συσκευών με λειτουργικό σύστημα Android.

Το Android λοιπόν χάρη στη δυναμική του έχει κατακτήσει ένα μεγάλο μερίδιο της αγοράς και είναι διαθέσιμο σε δεκάδες συσκευές κινητής τηλεφωνίας. Συνήθως οι κατασκευαστές κυκλοφορούν τις συσκευές τους είτε με την νεότερη έκδοση του λειτουργικού συστήματος, είτε με την αμέσως προηγούμενη μαζί με την υπόσχεση της αναβάθμισης στην νεότερη όταν αυτή έχει δοκιμαστεί επαρκώς από την εταιρία. Δυστυχώς όμως το θέμα των αναβαθμίσεων είναι αρκετά “ευαίσθητο”, διότι οι εταιρίες με στόχο την ευκολία και το κέρδος δεν κυκλοφορούν επαρκείς αναβαθμίσεις. Αντί αυτού προτιμούν να διαθέτουν τις νέες εκδόσεις λειτουργικού στις νεότερες συσκευές τους, με σκοπό να τις προτιμήσουν οι χρήστες αφήνοντας τις παλιές συσκευές με την υπόσχεση τις αναβάθμισης. Το κενό των εταιριών έρχεται να καλύψει η τεράστια κοινότητα developers του

Android οι οποίοι όντας κάτοχοι “παραμελημένων” συσκευών φροντίζουν να παρέχουν οι ίδιοι τις πολυπόθητες αναβαθμίσεις στα κινητά τους και άρα και συνήθως και των υπόλοιπων χρηστών! Αυτές οι εκδόσεις του Android είναι γνωστές ως “Custom Roms” και φυσικά υποστηρίζονται αποκλειστικά από τους developers τους και όχι από τις κατασκευάστριες εταιρίες των κινητών.

Παρά την τεράστια υποστήριξη της κοινότητας ο κατακερματισμός στο Android παραμένει υψηλός και εκτός και αν αλλάξει κάτι στον τρόπο των αναβαθμίσεων, τότε ίσως στο μέλλον να διογκωθεί ακόμη περισσότερο. Στο παρακάτω γράφημα φαίνεται καλύτερα το πρόβλημα.



Εικόνα 2.15: Χρονικό πλαίσιο υιοθέτησης των εκδόσεων του Android

Όπως βλέπουμε το μοτίβο υιοθέτησης των νέων εκδόσεων του Android είναι αρκετά αργό, και αυτό οφείλετε κυρίως στην αδιαφορία των κατασκευαστών να παρέχουν στις παλαιότερες συσκευές υποστήριξη. Οι περισσότερες συσκευές υποστηρίζουν τις νεότερες εκδόσεις του Android χάρη στη συνεισφορά της κοινότητας, αλλά οι περισσότεροι χρήστες δεν έχουν τις απαραίτητες γνώσεις ή εμπειρία για να χρησιμοποιήσουν κάποια από αυτές τις custom εκδόσεις.

Φυσικά όλες αυτές οι συσκευές δεν γίνεται να αγνοηθούν από τους προγραμματιστές εφαρμογών του Android γιατί αποτελούν μέρος τις πύκας και άρα πιθανοί χρήστες των εφαρμογών τους. Παραπάνω αναφερθήκαμε

λεπτομερώς στις προκλήσεις προγραμματισμού σε διαφορετικά API και τα προβλήματα συμβατότητας που προκαλούν στον προγραμματιστή αυτά. Είναι συχνό φαινόμενο πολλές εταιρίες να επιλέγουν πρώτα το iOS για να αναπτύξουν εφαρμογές, με το Android να ακολουθεί δεύτερο και ο κύριο λόγος φυσικά είναι η υποστήριξη των εκατοντάδων συσκευών που τρέχουν τις διάφορες εκδόσεις του λειτουργικού συστήματος.

Μεγάλες εταιρίες ανάπτυξης mobile εφαρμογών μάλιστα αποσύρουν την υποστήριξη τους στο Android γιατί το θεωρούν ασύμφορο! Παράδειγμα αυτής της περίπτωσης είναι το παιχνίδι “Battleheart” της εταιρίας Mika Mobile που είναι διαθέσιμο για Android και iOS.

Κάποιος μπορεί εύκολα αν αντιληφθεί ότι εφόσον μια μεγάλη εταιρία με πολλούς developers βρίσκει ασύμφορη για ανάπτυξη, την πλατφόρμα του Android, ο απλός developer που προσπαθεί να βγάλει ένα μικρό εισόδημα από εκεί, βρίσκεται σε ακόμη δυσκολότερη θέση.

Κεφάλαιο 3: Ανάλυση απαιτήσεων εφαρμογής

3.1 Ανάπτυξη UI (User Interface) σε smartphones συσκευές

Ο σχεδιασμός Διεπαφής (User Interface) ειδικά για κινητές συσκευές (mobile devices) διαφέρει σε σχέση με τις desktop εφαρμογές. Κύρια διαφορά τους όσον αφορά τις κινητές συσκευές, είναι η περιορισμένη επιφάνεια και οι μικρές αναλύσεις. Επίσης υπάρχει διαφορά όσον αφορά την πλοήγηση. Στις desktop εφαρμογές χρησιμοποιούμε πληκτρολόγιο και ποντίκι, ενώ στις κινητές συσκευές τον αντίχειρα στην οθόνη του κινητού (touchscreen).

Συνεπώς οι τεχνικές σχεδιασμού εφαρμογής για κινητά τηλέφωνα λόγω της ιδιομορφίας τους αποτελεί πρόκληση για μελετημένη και ορθή χωροταξική χρήση της οθόνης του κινητού τηλεφώνου.

3.2 Δημιουργία φόρμας

Χρησιμοποίηση στοιχείων φόρμας για κινητά τηλέφωνα. Στην συγκεκριμένη εφαρμογή θα χρησιμοποιηθούν textboxes και buttons.

3.3 Χρησιμοποίηση GPS δεδομένων

Οι Android συσκευές διαθέτουν δυνατότητα σύνδεσης GPS λαμβάνοντας ανα πάσα στιγμή τις συντεταγμένες της τοποθεσίας του κινητού τηλεφώνου. Στη συγκεκριμένη εφαρμογή γίνεται εκτενής χρήση του GPS λαμβάνοντας περιοδικά τις συντεταγμένες του κινητού με σκοπό την επεξεργασία των δεδομένων.

3.4 Σύγκριση γεωγραφικών σημείων

Σκοπός της εφαρμογής είναι η σύγκριση του γεωγραφικού σημείου που έχει θέσει ο χρήστης σε σχέση με το γεωγραφικό σημείο που βρίσκεται τη δεδομένη στιγμή. Σε περίπτωση που ο χρήστης βρίσκεται σε κοντινή απόσταση από το σημείο που έχει θέσει, τότε θα λαμβάνει μια ειδοποίηση.

Η συγκεκριμένη υλοποίηση μπορεί να χρησιμοποιηθεί ως γεωγραφική υπενθύμιση σε αντίθεση με τη χρονική. Συγκεκριμένα όταν ο χρήστης περνάει από ένα συγκεκριμένο σημείο να λαμβάνει την ειδοποίηση.

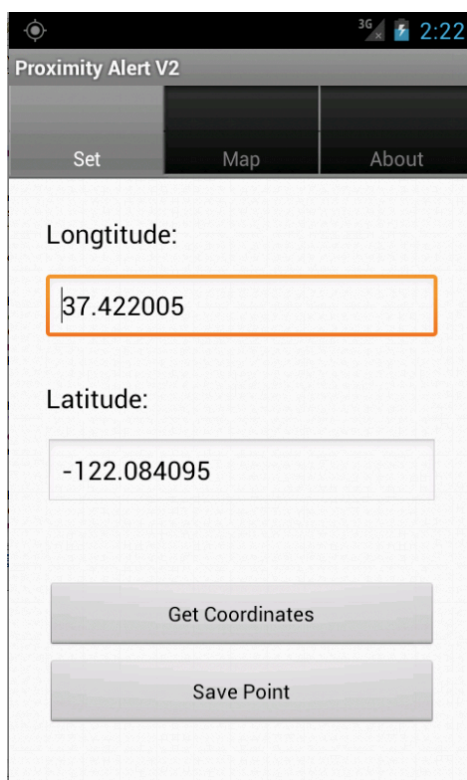
3.5 Χρησιμοποίηση του emulator

Σημαντικό εργαλείο για την ανάπτυξη mobile εφαρμογών είναι η χρησιμοποίηση του emulator. Ο emulator συνοδεύεται από εργαλεία όπως το DDMS τα οποία θα χρησιμοποιηθούν κατά την εισαγωγή συντεταγμένων στο σύστημα.

Κεφάλαιο 4: Σχεδιασμός Εφαρμογής

Η πλοήγηση της εφαρμογής πραγματοποιείται με tab menus. Επιλέχθηκε ο συγκεκριμένος τρόπος εξαιτίας της άμεσης εναλλαγής των οθονών αποφεύγοντας τα περιττά «clicks». Επιπλέον η ιδιομορφία της συγκεκριμένης εφαρμογής, μας επιτρέπει την «σπατάλη» του σημείου της οθόνης που βρίσκεται μόνιμα το tab menu. Αξίζει να σημειωθεί, λόγω του περιορισμένου χώρου της οθόνης των κινητών συσκευών σε ορισμένες περιπτώσεις η χρησιμοποίηση των tab menus είναι ασύμφορη εξαιτίας της μόνιμης τοποθέτησης στο πάνω μέρος της οθόνης.

Η εφαρμογή χωρίζεται σε τρεις ενότητες (tab menus), Set, Map και About.



Στο tab **Set** υπάρχει η φόρμα δυο πεδίων (longitude, latitude) και δυο buttons (Get Coordinations, Save Point). Στο tab **Map** θα εμφανίζεται το σημείο στον χάρτη (Googlemaps) που έχει ορίσει ο χρήστης. Στο **About**, γενικές πληροφορίες της εφαρμογής.

Κεφάλαιο 5: Υλοποίηση Εφαρμογής

5.1 Ιδιομορφία hardware κινητών συσκευών

Η ανάπτυξη εφαρμογών για κινητά τηλέφωνα διαφέρει από αυτόν των desktop εφαρμογών. Συγκεκριμένα, τα κινητά τηλέφωνα διαθέτουν:

- Χαμηλή επεξεργαστική ισχύς
- Περιορισμένη μνήμη RAM
- Περιορισμένο χώρο δίσκου μόνιμης αποθήκευσης
- Μικρές οθόνες με χαμηλή ανάλυση
- Υψηλά κόστη μεταφοράς δεδομένων (data transfer)
- Χαμηλές ταχύτητες σύνδεσης
- Αναξιόπιστες συνδέσεις μεταφοράς δεδομένων
- Περιορισμένη μπαταρία

Συνεπώς, κατά την υλοποίηση της εφαρμογής, θα πρέπει να ληφθούν υπόψιν οι παραπάνω περιορισμοί οι οποίοι θα συνδράμουν σημαντικά στη σωστή ανάπτυξη της εφαρμογής.

5.2 Σχεδιασμός UI

Στο Android SDK το UI (User Interface) ορίζεται υπο μορφή XML. Καταρχήν στο XML ορίζουμε την διάταξη που θα έχουν τα στοιχεία UI (textboxes, buttons κλπ.) κατά την εισαγωγή τους. Στη συνέχεια τοποθετούμε τα στοιχεία στα σημεία που επιθυμούμε. Αυτό μπορεί να πραγματοποιηθεί είτε γράφοντας απευθείας κώδικα σε XML, είτε από το διαθέσιμο Graphical Layout του Eclipse.

Στο συγκεκριμένο project έχουν υλοποιηθεί τέσσερα layouts. Τα `activity_main.xml`, `set_layout.xml`, `map_layout.xml`, `about_layout.xml`.

5.2.1 Activity_main.xml

Το activity_main.xml είναι το κύριο layout το οποίο ενεργοποιείται κατά την εκκίνηση της εφαρμογής. Σε αυτό το layout δηλώνουμε ότι θα χρησιμοποιήσουμε tabs menu για την πλοήγηση (navigation) της εφαρμογής.

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content" />
        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"/>
    </LinearLayout>
</TabHost>
```

5.2.2 Set_layout.xml

Το set_layout.xml τοποθετείται σε ένα από τα τρία tabs menu. Συγκεκριμένα εμφανίζεται στην επιλογή **Set** του tab menu. Περιέχει δύο textboxes στα οποία εμφανίζονται το longitude και το latitude αντίστοιχα. Επίσης περιέχει δύο buttons **Get Coordinations** και **Save Position**. Ένα δείγμα του XML παρουσιάζεται παρακάτω. Στο TextView tag εμφανίζεται το text, στο EditText το textbox, ενώ το Button tag είναι το κουμπί του layout. Το tag LinearLayout είναι μια από τις μορφές στοίχισης (χωροταξικά) στην οθόνη του κινητού τηλεφώνου.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Longitude:"
    android:layout_marginTop="25dip"
    android:layout_marginLeft="25dip"
    android:layout_marginRight="25dip"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/point_latitude"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dip"
    android:layout_marginLeft="25dip"
    android:layout_marginRight="25dip"
/>

. . .

<Button
    android:id="@+id/save_point_button"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="25dip"
    android:layout_marginLeft="25dip"
    android:layout_marginRight="25dip"
    android:layout_marginTop="0dip"
    android:text="Save Point" />

</LinearLayout>

```

5.3 Activities

Τα Activities στο Android SDK είναι οθόνες στις οποίες ορίζεται το layout που θα χρησιμοποιηθεί, καθώς και οι event handlers ανάλογα με τις ενέργειες που έχουν οριστεί να εκτελούνται στο συγκεκριμένο activity.

```

public class SetActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.set_layout);
    }
}

```

```
}
```

Στον παραπάνω κώδικα, παρουσιάζεται το activity της επιλογής **Set** απο το tab menu. Αρχικά ορίζεται το όνομα της κλάσης SetActivity η οποία είναι προέκταση της κλάσης Activity. Η Activity είναι κλάση του Android SDK και ελέγχει όλες τις κλάσεις που λειτουργούν ως Activities.

Στη συνέχεια ορίζουμε το layout το οποίο θα χρησιμοποιηθεί με την εντολή setContentView. Εκεί μέσα ορίζουμε ότι το layout που θα χρησιμοποιηθεί είναι το set_layout (set_layout.xml).

Με παρόμοιο τρόπο λειτουργούν όλα τα υπόλοιπα activities που έχουν οριστεί στην εφαρμογή.

5.4 Navigation

Η Πλοήγηση (navigation) στην εφαρμογή γίνεται μέσω του tabs menu. Το tab menu ορίζεται αρχικά ως layout. Αυτό είναι το κύριο layout το οποίο φορτώνεται αυτόματα κατά την εκκίνηση της εφαρμογής. Συγκεκριμένα είναι το activity_main.xml το οποίο είδαμε παραπάνω. Ουσιαστικά το μόνο που δηλώνουμε είναι ότι θα χρησιμοποιήσουμε tab menus απο εδώ και στο εξής. Η τοποθέτηση των περιεχομένων των tabs είναι Activities τα οποία εκτελούνται μέσα στο κυρίως activity των tab menus. Στον παρακάτω κώδικα παρουσιάζεται ο τρόπος που ορίζονται αυτά.

```
public class MainActivity extends TabActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TabHost tabHost = getTabHost();

        // Tab for Set
        TabSpec setspec = tabHost.newTabSpec("Set");
        // setting Title and Icon for the Tab
        setspec.setIndicator("Set");
        Intent setIntent = new Intent(this, SetActivity.class);
```

```

    setspec.setContent(setIntent);

    // Tab for Map
    TabSpec mapspec = tabHost.newTabSpec("Map");
    mapspec.setIndicator("Map");
    Intent mapIntent = new Intent(this, MapActivity.class);
    mapspec.setContent(mapIntent);

    // Tab for About
    TabSpec aboutspec = tabHost.newTabSpec("About");
    aboutspec.setIndicator("About");
    Intent aboutIntent = new Intent(this, AboutActivity.class);
    aboutspec.setContent(aboutIntent);

    // Adding all TabSpec to TabHost
    tabHost.addTab(setspec); // Adding set tab
    tabHost.addTab(mapspec); // Adding map tab
    tabHost.addTab(aboutspec); // Adding about tab
}
}

```

Όπως φαίνεται στον παραπάνω κώδικα, αρχικά ορίζουμε το layout που θα χρησιμοποιήσουμε, το οποίο είναι το `activity_main` (`activity_main.xml`). Στη συνέχεια δημιουργούμε τα tabs δημιουργώντας αντικείμενα απο την `TabSpec`. Στην method `newTabSpec()` ορίζουμε το όνομα του tab που θα εμφανίζεται στο μενου. Ενώ στην `setIndicator()` ορίζουμε το label του tab που θα αλληλοσχετίσουμε σε άλλα σημεία του κώδικα.

Στη συνέχεια χρησιμοποιώντας αντικείμενο της κλάσης `Intent` ορίζουμε ποιιά κλάση (η οποία είναι `activity`) θα κληθεί οταν το συγκεκριμένο tab θα ενεργοποιηθεί. Μέσα αυτό το `activity` περιέχει όρισμα του layout που θα χρησιμοποιήσει όπως αναφέρθηκε προηγουμένως.

5.5 Λειτουργία GPS

Όσον αφορά την λειτουργία του GPS υπάρχουν δυο ειδών κλάσεων. Η μια είναι η **BroadcastReceiver** η οποία εκτελείται ως `background process` στο Android καθώς αναμένει δεδομένα απο το GPS.

```

public class ProximityIntentReceiver extends BroadcastReceiver {

```

```

private static final int NOTIFICATION_ID = 1000;

@Override
public void onReceive(Context context, Intent intent) {

    String key = LocationManager.KEY_PROXIMITY_ENTERING;

    Boolean entering = intent.getBooleanExtra(key, false);

    if (entering) {
        Log.d(getClass().getSimpleName(), "entering");
    }
    else {
        Log.d(getClass().getSimpleName(), "exiting");
    }

    NotificationManager notificationManager =
        (NotificationManager)
        context.getSystemService(Context.NOTIFICATION_SERVICE);

    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
                                                            null, 0);

    Notification notification = createNotification();
    notification.setLatestEventInfo(context,
        "Proximity Alert!", "You are near your point of interest.",
        pendingIntent);

    notificationManager.notify(NOTIFICATION_ID, notification);
}
}

```

Ο παραπάνω κώδικας εκτελείται ως background process στην συσκευή Android. Συγκεκριμένα η κλάση **ProximityIntentReceiver** η οποία κληρονομεί χαρακτηριστικά από την **BroadcastReceiver** αναμένει να παραλάβει συντεταγμένες από την συσκευή GPS του κινητού. Η μέθοδος **onReceive()** εκτελείται κατά τη λήψη σήματος από το GPS. Στη συνέχεια γίνεται έλεγχος εάν ο χρήστης πλησιάζει στο σημείο ενδιαφέροντος ή απομακρύνεται αυτό. Ο έλεγχος δηλώνεται στο ιστορικό της εφαρμογής μέσω της **Log.d**. Στη συνέχεια ο **NotificationManager** αναλαμβάνει να ειδοποιήσει τον χρήστη καθώς πλησιάζει στο σημείο ενδιαφέροντος.

Η άλλη είναι η **SetActivity** η οποία αναλαμβάνει να διαχειριστεί τα δεδομένα από το GPS μέσω του UI που είναι στη διάθεση του χρήστη. Ορισμένες ενέργειες της

SetActivity είναι η λήψη των γεωγραφικών συντεταγμένων που βρίσκεται ο χρήστης πατώντας το κουμπί **Get Coordinations**.

```
public class SetActivity extends Activity {

    // in Meters
    private static final long MINIMUM_DISTANCECHANGE_FOR_UPDATE = 1;
    // in Milliseconds
    private static final long MINIMUM_TIME_BETWEEN_UPDATE = 1000;

    private static final long POINT_RADIUS = 1000; // in Meters
    private static final long PROX_ALERT_EXPIRATION = -1;

    private static final String POINT_LATITUDE_KEY = "POINT_LATITUDE_KEY";
    private static final String POINT_LONGITUDE_KEY =
        "POINT_LONGITUDE_KEY";

    private static final String PROX_ALERT_INTENT =
        "com.example.ProximityAlertV2";

    private static final NumberFormat nf = new
        DecimalFormat("##.#####");

    private LocationManager locationManager;

    private EditText latitudeEditText;
    private EditText longitudeEditText;
    private Button findCoordinatesButton;
    private Button savePointButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.set_layout);

        locationManager = (LocationManager)
            getSystemService(Context.LOCATION_SERVICE);

        locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER,
            MINIMUM_TIME_BETWEEN_UPDATE,
            MINIMUM_DISTANCECHANGE_FOR_UPDATE,
            new MyLocationListener()
        );

        latitudeEditText = (EditText) findViewById(R.id.point_latitude);
        longitudeEditText = (EditText) findViewById(R.id.point_longitude);
        findCoordinatesButton = (Button)
            findViewById(R.id.find_coordinates_button);
        savePointButton = (Button) findViewById(R.id.save_point_button);

        findCoordinatesButton.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
```



```

        populateCoordinatesFromLastKnownLocation();
    }
});

savePointButton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        saveProximityAlertPoint();
    }
});
}
}

```

Στο **SetActivity** αρχικά δηλώνονται οι σταθερές οι οποίες θα χρησιμοποιηθούν στην εφαρμογή. Ενδεικτικά ορισμένες απο αυτές είναι οι `MINIMUM_DISTANCE_FOR_UPDATE` στην οποία δηλώνεται η απόσταση στην οποία θα ενημερώνεται η εφαρμογή για τις συντεταγμένες. Επίσης είναι η `POINT_RADIUS` στην οποία ορίζουμε την ακτίνα στην οποία θα λάβει ειδοποίηση ο χρήστης μόλις εισέλθει σε αυτήν.

Στην συνέχεια ορίζεται ο **LocationManager** ο οποίος θα είναι ενήμερος για τις συντεταγμένες του χρήστη. Στην **SetActivity** είναι δηλωμένα και τα στοιχεία UI (User Interface) της εφαρμογής ως αντικείμενα. Για παράδειγμα το κουμπί έχει οριστεί ως αντικείμενο **savePointButton** με σκοπό την περαιτέρω αναφορά του στον κώδικα. Η αντιστοιχία μεταξύ layout και κώδικα γίνεται μέσω της:

```
findViewById(R.id.save_point_button);
```

Στην οποία γίνεται η αντιστοιχία μεταξύ την ονομασίας του button στο XML layout αρχείο **save_point_button** με το αντικείμενο στον κώδικα **savePointButton** όπως είδαμε νωρίτερα.

Στον παραπάνω κώδικα υπάρχει επίσης ο event handler του button ο οποίος είναι αυτός:

```

savePointButton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        saveProximityAlertPoint();
    }
});
}

```

```
    }  
});
```

Ο **setOnClickListener** αναμένει να πατηθεί το button **savePointButton**, μόλις πατηθεί τότε καλεί την μέθοδο **saveProximityAlertPoint()**.

5.6 Googlemaps

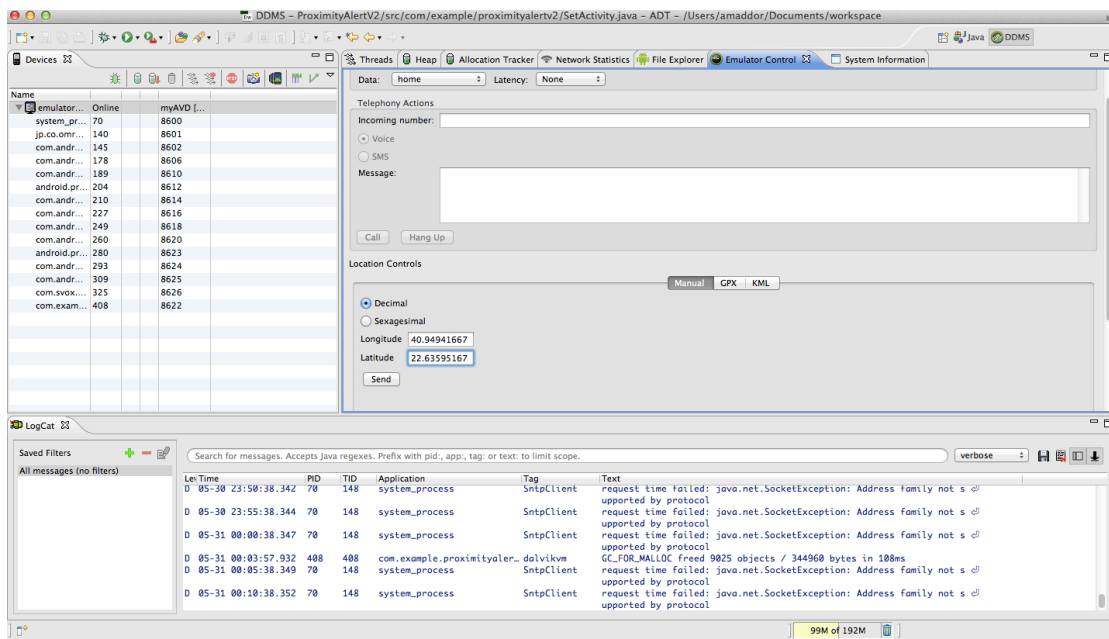
Στο πλάνο υλοποίησης της εφαρμογής ήταν η ενσωμάτωση googlemaps στο οποίο ο χρήστης επιλέγοντας το tab Map να εμφανιζόταν στον χρήστη το σημείο υπενθύμισης που είχε ορίσει.

Η ανάπτυξη της συγκεκριμένης εφαρμογής υλοποιήθηκε για την έκδοση Android 2.2. Η Google πρόσφατα απέσυρε την υποστήριξη του GoogleMaps V1 και την αντικατέστησε με την GoogleMaps V2 στην οποία έχουν γίνει αλλαγές οι οποίες δεν ήταν δυνατόν να ενσωματωθούν στην εφαρμογή δεδομένης της ημερομηνίας παράδοσης.

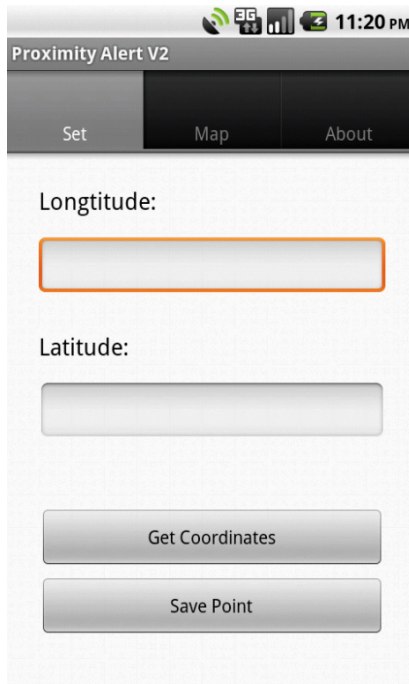
Κεφάλαιο 6: Δοκιμές Λειτουργίας

Κατά το compilation ενεργοποιείται ο emulator και αυτόματα εγκαθιστά και εκτελεί την εφαρμογή. Στο bin υποκατάλογο της εφαρμογής δημιουργείται ένα αρχείο με επέκταση .apk. Αυτό είναι το αρχείο εγκατάστασης το οποίο εάν το μεταφέρουμε σε ένα Android κινητό τηλέφωνο μπορούμε να εγκαταστήσουμε την εφαρμογή και να την δοκιμάσουμε σε πραγματικό περιβάλλον.

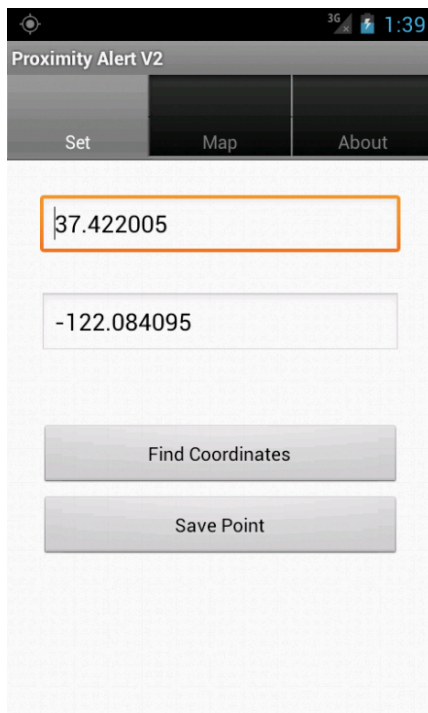
Για να δοκιμάσουμε την εφαρμογή στον emulator χρησιμοποιήθηκε το εργαλείο DDMS (Dalvik Debug Monitoring Service). Απο το DDMS γίνεται η εισαγωγή των γεωγραφικών συντεταγμένων και απο εκεί το DDMS τις στέλνει στον emulator. Στην παρακάτω εικόνα είναι το περιβάλλον του DDMS.



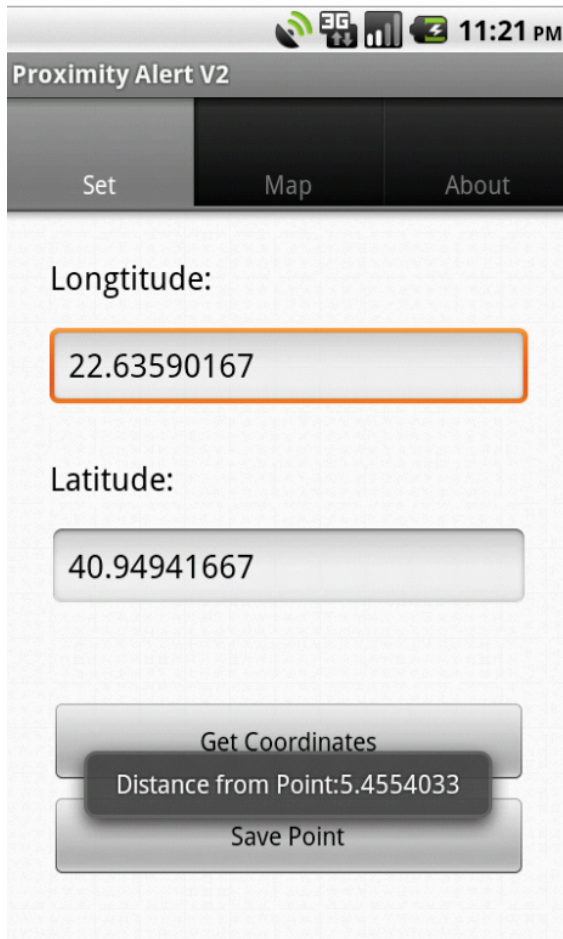
Κατα την πρώτη εκτέλεση της εφαρμογής, η πρώτη οθόνη που εμφανίζεται είναι η παρακάτω.



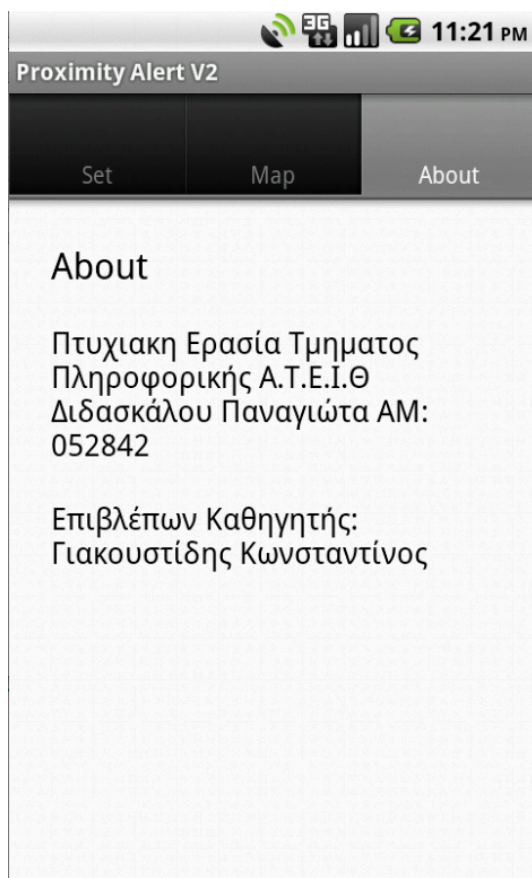
Στη συνέχεια ενεργοποιώντας το κουμπί **Get Coordinates** φορτώνονται οι συντεταγμένες στα textboxes των Longitude και Latitude όπως φαίνεται στην παρακάτω εικόνα:



Στη συνέχεια πατώντας το κουμπί Save Point, σώζονται οι συγκεκριμένες συντεταγμένες στην μνήμη. Καθώς πλησιάζει ο χρήστης στο συγκεκριμένο σημείο, του εμφανίζεται το παρακάτω μήνυμα ενημερώνοντάς τον ότι πλησιάζει στο σημείο που έχει ορίσει καθώς και την απόσταση που απομένει μέχρι τον προορισμό σε μέτρα.



Πατώντας το tab menu **About** εμφανίζεται η παρακάτω οθόνη



Κεφάλαιο 7 : Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης

Η ραγδαία ανάπτυξη των έξυπνων κινητών τηλεφώνων (smartphones) έχουν ανοίξει τα τελευταία χρόνια νέα δυναμική αγορά. Αυτή η δυναμική φέρνει ανάπτυξη των νέων τεχνολογιών προσφέροντας σημαντικά εργαλεία στους developers με σκοπό την δημιουργία ποιοτικών εφαρμογών. Μια από αυτές τις πλατφόρμες είναι και αυτή του Android.

Το θέμα της συγκεκριμένης πτυχιακής ήταν η αφορμή να πραγματοποιηθεί έρευνα όσον αφορά την πλατφόρμα του Android γενικότερα, την αρχιτεκτονική του και τα συνοδευτικά εργαλεία ανάπτυξης εφαρμογών. Με την εμπειρία που αποκτήθηκε όλων των ανωτέρω, αναπτύχθηκε εφαρμογή για κινητά τηλέφωνα Android.

Η λογική της αρχιτεκτονικής του Android είναι ιδιαίτερα προχωρημένη καθώς βασίζεται πλήρως στην εικονική μηχανή (Virtual Machine) Dalvik. Με αυτό τον τρόπο οι Android εφαρμογές μπορούν να εκτελούνται σε ποικίλες συσκευές χωρίς να απασχολεί τον προγραμματιστή συμβατότητα των συσκευών που θα εκτελεστεί η εφαρμογή του.

Η εμπειρία μου όσον αφορά τον προγραμματισμό περιοριζόταν στις desktop εφαρμογές. Η ενασχόλησή μου με την συγκεκριμένη πτυχιακή, με βοήθησε να διαπιστώσω ότι υπάρχουν τεχνολογίες οι οποίες είναι δομημένες με τέτοιο τρόπο ώστε να δημιουργούν νέες δυνατότητες εξέλιξης της πληροφορικής γενικότερα.

Η είσοδος των smartphones στον τομέα της Πληροφορικής οδήγησε ώστε να επαναγραφεί με νέα λειτουργικά συστήματα τα οποία είναι ελαφρύτερα, πιο σωστά δομημένα με ιδιαίτερη ευελιξία και σημαντικά περιθώρια εξέλιξης. Η κύρια αιτία αυτής της εξέλιξης ήταν οι περιορισμένοι πόροι του συστήματος των κινητών συσκευών.

Στις μέρες μας, οι εφαρμογές κινητών τηλεφώνων δεν έχουν να ζηλέψουν σε τίποτα τις εφαρμογές desktop. Αυτό συμβαίνει γιατί μια εφαρμογή για κινητό τηλέφωνο μπορεί να εκτελεστεί και σε μια συσκευή με μεγαλύτερη ανάλυση ενός tablet.

Όσον αφορά τη μελλοντική εξέλιξη της συγκεκριμένης εφαρμογής, ιδιαίτερο κίνητρο αποτελεί η μετάβασή της σε νεότερες εκδόσεις Android με σκοπό την ενσωμάτωση των Googlemaps V2. Επίσης, θα μπορούσε να δέχεται πολλαπλά σημεία υπενθύμισης παρέχοντας τη δυνατότητα στον χρήστη να εισάγει δικό του μήνυμα ή ήχο. Τέλος, θα ήταν ιδιαίτερα ενδιαφέρουσα η έρευνα λειτουργίας της συγκεκριμένης εφαρμογής μέσω web services και caching.

Συντομογραφίες

OS: Operating System

OHA: Open Handset Alliance

SDK: Software Development Kit

UI: User Interface

A2DP: Advanced Audio Distribution Profile

AVRCP: Audio/Video Remote Control Profile

CDMA: Code Division Multiple Access

EVDO: Evolution-Data Optimized

VPN: Virtual Private Network

WVGA: Wide Video Graphics Array

HTML: HyperText Markup Language

GAL: Global Address List

USB: Universal Serial Bus

WiFi: Wireless Fidelity

GIF: Graphics Interchange Format

NFC: Near Field Communication

2D: two-dimensional

3D: three-dimensional

EBook: Electronic Book

API: Application Programming Interface

NDK: Native Development Kit

XML: Extensible Markup Language

SQL: Structured Query Language

Gmail: Google Mail

QR-Code: Quick Response Code

ISO: International Organization for Standardization

ZXing: Zebra Crossing

RFID: Radio Frequency Identification

JSON: JavaScript Object Notation

MD5: Message-Digest Algorithm

GPS: Global Positioning System

WLAN: Wireless Local Area Network

PNG: Portable Network Graphics

OpenGL ES: Open Graphics Library for Embedded Systems

ADT: Android Development Tool

SD: Secure Digital

AVD: Android Virtual Device

UTF-8: 8-bit Unicode Transformation Format

SAX: Simple API for XML

DOM: Document Object Model

URI: Uniform Resource Identifier

W3C: World Wide Web Consortium

Βιβλιογραφία

- J. Friesen, 2010, “Learn Java for Android Development”, Apress
- W.M. Lee, 2011, “Beginning Android Application Development”, Wrox
- M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition", CommonsWare
- J. Morris, 2011, “Android User Interface Development Beginner's Guide”, Pakt Publishing
- C. Hasenan, 2008, “Android Essentials”, Firstpress
- J. Steele, 2010, “The Android Developer's Cookbook”, Addison & Wesley
- R. Meier, 2010, “Professional Android 2 Application Development”, Wrox

Ιστότοποι

- 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>
- Πληροφορίες σχετικά με τη δομή και τη χρήση του του αρχείου AndroidManifest.xml
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- Οι φάσεις του σχεδιασμού τις εφαρμογής περιλαμβάνονται στην παρακάτω πηγή

<http://developer.android.com/guide/developing/index.html>

- Οδηγίες εγκατάστασης του Android SDK

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

- Η επίσημη ιστοσελίδα του Eclipse IDE, το οποίο χρησιμοποιήθηκε κατά την υλοποίηση του project και είναι και προτεινόμενο από την Google.

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

- Οδηγίες εγκατάστασης του ADT plugin

<http://developer.android.com/sdk/eclipse-adt.html#installing>

- Η ιστοσελίδα “Design Guidelines”, η οποία περιέχει πληροφορίες, παραδείγματα, και οδηγίες ώστε να χρησιμοποιηθούν από τους developers για τον σχεδιασμό των εφαρμογών τους.

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

- Ο καταμερισμός των εκδόσεων του Android όπως αυτά καταγράφονται από την πρόσβαση των συσκευών στο Google Play Store.

<http://developer.android.com/resources/dashboard/platform-versions.html>

- Ο καταμερισμός των διαστάσεων οθόνης προς την πυκνότητα pixel των συσκευών που χρησιμοποιούν το Play Store

<http://developer.android.com/resources/dashboard/screens.html>

- Οδηγίες υποστήριξης πολλαπλών αναλύσεων στις εφαρμογές

http://developer.android.com/guide/practices/screens_support.html

- Ανάλυση και οδηγίες χρήσης του Android Debug Bridge

<http://developer.android.com/guide/developing/tools/adb.html>

- Σχετικά με την διαχείριση εικονικών συσκευών

<http://developer.android.com/guide/developing/devices/index.html>

- Πληροφορίες σχετικά με το εργαλείο LogCat
<http://developer.android.com/guide/developing/tools/logcat.html>
- Η βιβλιοθήκη ανοιχτού κώδικα ACRA η οποία χρησιμοποιήθηκε στο project για να παρέχει απομακρυσμένη αναφορά σφαλμάτων.
<http://code.google.com/p/acra/>
- Έρευνα της δημιουργού εταιρίας της εφαρμογής “OpenSignalMaps” σχετικά με το fragmentation του Android, από την οποία χρησιμοποιήθηκαν δεδομένα και εικόνες.
<http://opensignalmaps.com/reports/fragmentation.php>
- Steven J. Vaughan-Nichols, August 18, 2011, “Linus Torvalds on Android, the Linux fork”, Zdnet.com
<http://www.zdnet.com/blog/open-source/linus-torvalds-on-android-the-linuxfork/9426>
- Casey Johnston, March 12 2012, “Instagram comes to Google Play as big developer leaves”, arstechna.com
<http://arstechnica.com/gadgets/2012/03/instagram-comes-to-google-play-as-bigdeveloper-leaves/>
- Η κοινότητα προγραμματιστών “Stack Overflow” παρέχει ένα μέρος για ερωτήσεις και απαντήσεις στα περισσότερα προβλήματα που μπορεί να συναντήσει ένας developer του Android, και η συνεισφορά της ήταν ανεκτίμητη.
www.stackoverflow.com