



**Τμήμα
Μηχανικών
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΠΑΙΧΝΙΔΙ ΜΑΘΗΜΑΤΙΚΩΝ ΓΙΑ ΤΑ ΠΑΙΔΙΑ
ΔΗΜΟΤΙΚΟΥ – ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ
ΣΕ ANDROID**

ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΙΝΙΔΗΣ ΑΜ 1303

ΘΕΟΓΝΩΣΙΑ ΝΙΚΟΛΑΟΥ ΑΜ 1249

**Επιβλέπων
ΓΙΑΝΝΗΣ ΤΖΗΜΑΣ**

2015

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Αφιέρωση

Στη κορούλα μας Μαχηάελια που την λατρεύομαι

Ευχαριστίες

Η ολοκλήρωση αυτής της πτυχιακής υλοποιήθηκε με την υποστήριξη ενός αριθμού ανθρώπων στους οποίους θα θέλαμε να εκφράσουμε τις θερμότερες ευχαριστίες μας. Πρώτα από όλους θα ήθελα να ευχαριστήσω τον κύριο Γιάννη Τζήμα για την πολύτιμη βοήθεια του καθ' όλη την διάρκεια της πτυχιακής μας εργασίας.

Συνεχίζοντας θα θέλαμε να ευχαριστήσουμε όλους τους καθηγητές μας που μας έδωσαν τις πολύτιμες βάσεις όλα αυτά τα χρόνια για να συνεχίσουμε το έργο τους και εμείς με την σειρά μας στην επομένη γενιά.

Εν τέλη θα θέλαμε να ευχαριστήσουμε θερμά τις οικογένειες μας που ήταν πάντα δίπλα μας και μας στήριζαν για να φτάσουμε στο τελειωτικό αυτό στάδιο της φοιτητικής μας ζωής.

Κωνσταντίνος Κωνσταντινίδης

Θεογνωσία Νικολάου

Σεπτέμβριος 2015

Περιεχόμενα

ΑΦΙΕΡΩΣΗ.....	III
ΕΥΧΑΡΙΣΤΙΕΣ	IV
1 Η JAVA	14
1.1 Εισαγωγή.....	14
1.2 Τι είναι η Java?	15
1.2.1 Απλή	15
1.2.2 Αντικειμενοστραφής.....	17
1.2.3 Συμβατή με Δίκτυα.....	17
1.2.4 Σταθερή	17
1.2.5 Ασφαλής	18
1.2.6 Ουδέτερη της Υποκείμενης Αρχιτεκτονικής.....	18
1.2.7 Φορητή	19
1.2.8 Interpreted.....	19
1.2.9 Υψηλής Απόδοσης	20
1.2.10 Πολυνηματική	20
1.2.11 Δυναμική	20
1.3 Η JAVA πλατφόρμα	21
1.3.1 Η εικονική μηχανή της Java (<i>Java Virtual Machine</i>).....	21
1.3.2 Συμπεράσματα.....	22
2 ANDROID	23
2.1 Τι είναι το Android;	23

2.2	Αρχιτεκτονική του Android	24
2.3	Android Linux Kernel.....	25
2.3.1	Τι είναι ο πυρήνας λειτουργικού συστήματος.....	26
2.3.2	Τι είναι ο πυρήνας Linux.....	26
2.4	Βιβλιοθήκες (Libraries)	27
2.5	Η εικονική μηχανή Dalvik (Dalvik Virtual Machine).....	28
2.6	Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime).....	29
2.7	Πλαίσιο Εφαρμογής (Application Framework).....	30
3	ΕΦΑΡΜΟΓΕΣ ANDROID	33
3.1	Το αρχείο AndroidManifest.xml.....	33
3.2	Αρχείο build.gradle.....	34
3.3	Οι φάκελοι src & res	37
3.4	Δομικά Μέρη μιας Εφαρμογής.....	37
3.5	Εξέλιξη του Android.....	40
3.5.1	Cupcake 1.5	40
3.5.2	Donut 1.6	41
3.5.3	Éclair 2.0-2.1	43
3.5.4	Froyo 2.2.x.....	44
3.5.5	Gingerbread 2.3.x	45
3.5.6	Honeycomb 3.x.....	46
3.5.7	Ice Cream Sandwich 4.x.....	48

3.5.8	Jelly Bean 4.1.x – 4.2.x	49
3.5.9	Lollipop 5.x – 5.1.x	50
3.5.10	Mashmallow 6.x	51
3.6	Γιατί Ανάπτυξη σε Android;.....	52
3.6.1	Λειτουργικότητα και ευελιξία.....	52
3.6.2	Πλήρης παραμετροποίηση	53
3.6.3	Διαδραστικότητα	53
3.6.4	Εύκολη Ανάπτυξη Εφαρμογών για Κινητά.....	54
3.6.5	Συμπεράσματα.....	54
4	ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ANDROID	55
4.1	Κύκλος ανάπτυξης μιας εφαρμογής	55
4.2	Εγκατάσταση Λογισμικού	55
4.3	Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής	56
4.4	Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής	57
4.4.1	Εργαλείο καταγραφής συμβάντων – LogCat	58
4.4.2	Android Device Monitor ή Dalvik Debug Monitor Server (DDMS).....	62
4.5	Τελική έκδοση και δημοσίευση της εφαρμογής στο ευρύ κοινό.....	65
4.6	Android SDK	66
4.7	Εικονικές Συσκευές Android (Android Virtual Devices – AVD)	68
4.8	Android Studio IDE και ADT (Android Development Tools)	69

4.9	Προκλήσεις ανάπτυξης εφαρμογών στο Android	70
4.9.1	Android Design Guidelines	71
4.9.2	Υποστήριξη πολλαπλών συσκευών.....	74
4.9.3	Υποστήριξη παλαιότερων εκδόσεων του Android.....	74
4.9.4	Υποστήριξη πολλαπλών διαστάσεων οθόνης και πυκνότητας pixel (ppi)	78
5	ΤΟ ΠΑΙΧΝΙΔΙ – MATH GAME.....	82
5.1	Εισαγωγή στο παιχνίδι Android.....	82
5.2	Το Math Game	83
5.3	Δομή προγράμματος	87
5.4	Λειτουργία παιχνιδιού.....	90
5.5	Πως λειτουργεί το παιχνίδι	95
5.6	Συμπέρασμα και Μελλοντικά σχέδια	96

Λίστα εικόνων

ΕΙΚΟΝΑ 1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ANDROID	24
ΕΙΚΟΝΑ 2 ΚΥΚΛΟΣ ΖΩΗΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ (ACTIVITY LIFECYCLE)	32
ΕΙΚΟΝΑ 3 ΤΟ ΑΝΑΝΕΩΜΕΝΟ ANDROID MARKET	42
ΕΙΚΟΝΑ 4 Η ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ ΤΗΣ ΕΚΔΟΣΗΣ GINGERBREAD	46
ΕΙΚΟΝΑ 5 Η ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ ΤΗΣ ΕΚΔΟΣΗΣ HONEYCOMB	47
ΕΙΚΟΝΑ 6 ΒΗΜΑ ΠΡΩΤΟ – ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ	56
ΕΙΚΟΝΑ 7 ΒΗΜΑ ΔΕΥΤΕΡΟ – ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ	56
ΕΙΚΟΝΑ 8 ΒΗΜΑ ΤΡΙΤΟ – DEBUGGING ΚΑΙ ΔΟΚΙΜΑΣΤΙΚΗ ΦΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	58
ΕΙΚΟΝΑ 9 ΠΑΡΑΔΕΙΓΜΑ FORCE CLOSE	59
ΕΙΚΟΝΑ 10 LOGCAT ΣΤΟ ANDROID STUDIO IDE	60
ΕΙΚΟΝΑ 11 DDMS	63
ΕΙΚΟΝΑ 12 ΔΗΜΟΣΙΕΥΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	65
ΕΙΚΟΝΑ 13 ΟΙ ΕΙΚΟΝΙΚΕΣ ΣΥΣΚΕΥΕΣ	69
ΕΙΚΟΝΑ 14 - ANDROID STUDIO IDE	70
ΕΙΚΟΝΑ 15 ANDROID DESIGN	72
ΕΙΚΟΝΑ 16 ΣΤΑΤΙΣΤΙΚΑ ΣΤΟΙΧΕΙΑ ΕΝΕΡΓΩΝ ΣΥΣΚΕΥΩΝ ΜΕΧΡΙ ΤΙΣ 3 ΤΟΥ ΑΥΓΟΥΣΤΟΥ 2015	76
ΕΙΚΟΝΑ 17 ΑΝΑΛΥΣΕΙΣ ΟΘΟΝΩΝ ΒΑΣΕΙ DPI	79
ΕΙΚΟΝΑ 18 ΤΥΠΟΣ ΟΘΟΝΩΝ ΚΑΙ ΜΕΓΕΘΩΝ ΤΩΝ ΕΝΕΡΓΟΠΟΙΗΜΕΝΩΝ ΣΥΣΚΕΥΩΝ ΜΕΧΡΙ 3 ΑΥΓΟΥΣΤΟΥ 2015	80
ΕΙΚΟΝΑ 19 ΤΟ ΠΑΙΧΝΙΔΙ	82
ΕΙΚΟΝΑ 20 ΤΟ ΚΕΝΤΡΙΚΟ ΜΕΝΟΥ	83
ΕΙΚΟΝΑ 21 Η ΕΦΑΡΜΟΓΗ ΣΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΣΥΣΚΕΥΕΣ	86
ΕΙΚΟΝΑ 22 - ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ	87
ΕΙΚΟΝΑ 23 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	88
ΕΙΚΟΝΑ 24 ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ	90
ΕΙΚΟΝΑ 25 ΔΗΜΙΟΥΡΓΙΑ ΠΑΙΧΤΗ	91
ΕΙΚΟΝΑ 26 ΟΘΟΝΗ ΕΠΙΛΟΓΗΣ ΠΑΙΧΤΗ	92
ΕΙΚΟΝΑ 27 ΤΥΠΟΣ ΠΑΙΧΝΙΔΙΟΥ	93
ΕΙΚΟΝΑ 28 ΠΡΩΤΟ ΣΤΑΔΙΟ, ΕΠΙΛΟΓΗ ΕΠΙΠΕΔΩΝ	94
ΕΙΚΟΝΑ 29 ΜΕΤΑΒΑΣΗ ΣΤΟ ΕΠΟΜΕΝΟ ΕΠΙΠΕΔΟ	94

Πρόλογος

Με την εξέλιξη της τεχνολογίας όλο και περισσότεροι άνθρωποι αποκτούν έξυπνα κινητά και όλο και περισσότερα παιδιά έρχονται σε επαφή με συσκευές που έχουν εγκατεστημένο ένα λειτουργικό όπως είναι τα έξυπνα κινητά και τα tablets. Στα σχολεία οι μαθητές μαθαίνουν με διάφορα προγράμματα και παιχνίδια μαθηματικά, χημεία και αλλά μαθήματα και τα ίδια αρέσκονται στο να παίζουν διάφορα παιχνίδια επιμορφωτικά και μη σε έξυπνες συσκευές.

Στόχος της πτυχιακής αυτής είναι να δημιουργήσουμε ένα διασκεδαστικό επιμορφωτικό παιχνίδι μαθηματικών για το λειτουργικό σύστημα Android. Με την εφαρμογή αυτή θα μπορούν οι μαθητές όλων των τάξεων του δημοτικού να μάθουν μαθηματικά μέσα από ένα διασκεδαστικό παιχνίδι και να λύσουν διάφορες ασκήσεις αλλά και να συναγωνιστούν για να γίνουν καλύτεροι.

Ο κυριότερος στόχος μας είναι να την κάνουμε όσο πιο γρήγορη και απλή γίνεται για τον χρήστη καθώς και πιο διασκεδαστική.

Στα κεφάλαια πιο κάτω θα δείξουμε πως το επιτύχαμε αυτό αφού δείξουμε διάφορα αποσπάσματα του κώδικα και screenshots από το πρόγραμμα.

Περίληψη

Αρχικά στην πτυχιακή αυτή θα ασχοληθούμε με την Java ως κυρίως γλωσσά προγραμματισμού, τι είναι που χρησιμοποιείται, πως λειτουργεί και τους λόγους που επιλέχτηκε για το λειτουργικό σύστημα Android. Στην συνέχεια θα δούμε τι είναι το λειτουργικό σύστημα Android, πως αναπτύχθηκε, που χρησιμοποιείται και τις διαφορές του από τα υπόλοιπα λειτουργικά συστήματα για έξυπνα κινητά. Για την κατανόηση του application θα δούμε λίγα πράγματα για τις εφαρμογές Android και το Android SDK.

Αφού αναλύσουμε τα βασικά σημεία για την κατανόηση του παιχνιδιού θα δούμε πως μπορούν οι χρήστες να χρησιμοποιήσουν το παιχνίδι που φτιάξαμε.

Στο τέλος θα δούμε πως μπορεί το πρόγραμμα να γίνει ακόμη καλύτερο και ποια είναι τα σχέδια μας για το μέλλον.

Abstract

This thesis will begin with the analysis and information about Java, the programming language that is used to build Android applications. It will be discussed how it is working and the reasons it is chosen for the operating system Android. Next we are going to analyze the operating system Android, how it is developed and the main differences from other mobile operating systems. For better understanding about Android applications we are going to describe a few things about the structure of an Android application and how to build an Android application with the use of Android SDK.

After analyzing the basic steps about Android Application we are going to describe the application we created for this thesis and how the users can use it.

At the end we will describe our problems and experience we had during programming our game and what are our plans for the future.

1 Η Java

1.1 Εισαγωγή

Η Java είναι μιας γενικής χρήσης, αντικειμενοστραφής γλωσσά υψηλού επίπεδου προγραμματισμού που αναπτύχτηκε από την Sun Microsystems. Μια μικρή ομάδα μηχανικών, γνωστοί ως Green Team ξεκίνησε την γλωσσά το 1991. Η Java αρχικά ονομαζόταν Oak και αφορούσε την ανάπτυξη λογισμικού για καταναλωτικά ηλεκτρονικά. Πρόκειται για μικρές, αξιόπιστες, φορητές, πραγματικού χρόνου συσκευές που στην αρχή βασίζονταν στην C++. Το όνομα Oak ήταν ανεπιτυχές, έτσι ώστε το 1995 η Sun άλλαξε το όνομα σε Java. Σκοπός της ήταν η επίλυση μερικών προβλημάτων του σημερινού προγραμματισμού, όπως animation, την αλληλεπίδραση πραγματικού χρόνου) και την εξερεύνηση του Web.

Ας πάρουμε το Παγκόσμιος Ιστό για παράδειγμα. Οι φυλλομετρητές περιορίζονται σε ότι μπορεί να προσφέρει η HTML γλώσσα: σελίδες δυο διαστάσεων με κείμενο, φωτογραφίες ή γραφικά και συνδέσμους. Κάποια μορφή επεκτασιμότητας προσέφερε η JavaScript από την Netscape Corporation, αλλά οι σελίδες παραμένουν βαρετές, στατικές και ελλιπείς σε περιεχόμενο. Οι φυλλομετρητές δεν διαθέτουν την ιδιότητα της αλληλεπίδρασης πραγματικού χρόνου. Χρησιμοποιώντας την Java γλώσσα προγραμματισμού, η Sun κατασκεύασε τον νέο φυλλομετρητή Παγκόσμιος Ιστού, τον HotJava. Πλέον, η σχέση μεταξύ εξυπηρετητή και πελάτη αλλάζει. Ο νέος φυλλομετρητής εκμεταλλεύεται την υπολογιστική δύναμη του πελάτη και αντί να κατεβάζει απλές στατικές σελίδες κειμένου και εικόνων, ο HotJava κατεβάζει μικρά προγράμματα σε Java που τρέχουν στον πελάτη. Ο φυλλομετρητής είναι γρήγορος, ευέλικτος, πολύπλευρος και εύκολα προσαρμόσιμος.

Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL – General Public License) όσον αφορά το μεταγλωττιστή (javac) και το

πακέτο ανάπτυξης JDK – Java Development Kit. Στις 27 Απριλίου 2010 η εταιρία λογισμικού Oracle Corporation ανακοίνωσε ότι μετά από πολύμηνες συζητήσεις ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών που είχε στην κατοχή της. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της Java.

Από την ημέρα της δημοσιοποίησης της τον Μάιο του 1995, η Java έχει εξαπλωθεί σε όλο το διαδίκτυο και σε εκατομμύρια συσκευές παντού. Η Java έχει κιόλας λύσει τα περισσότερα προβλήματα στο μοντέλο πελάτη/εξυπηρετητή και έχει προάγει την χρήση του Παγκόσμιος Ιστού.

1.2 Τι είναι η Java?

Η Java είναι συγχρόνως γλώσσα προγραμματισμού και πλατφόρμα έχοντας τα εξής χαρακτηριστικά: απλή, αντικειμενοστραφής, συμβατή με δικτυακά πρωτόκολλα, ουδέτερη της υποκείμενης αρχιτεκτονικής, φορητή, ασφαλής, υψηλής απόδοσης, δυναμική, σταθερή, interpreted και πολυνηματική. Στις ακόλουθες παραγράφους θα αναλύσουμε καθεμία από αυτές τις έννοιες.

1.2.1 Απλή

Στόχος της ομάδας της Sun που ανέπτυξε την Java, ήταν να δημιουργήσει μια γλώσσα προγραμματισμού εύκολη στην χρήση, που δεν θα απαιτούσε πολλή εξάσκηση και εκπαίδευση. Οι περισσότεροι προγραμματιστές σήμερα δουλεύουν είτε με την C είτε με την C++. Έτσι, παρόλο που η C++ δεν ήταν η κατάλληλη για το αρχικό σχέδιο, η Java χτίστηκε βάσει της C++ με σκοπό να γίνει όσο το δυνατόν περισσότερο κατανοητή.

Η Java παραλείπει πολλά από τα μη χρησιμοποιούμενα και δυσκολονόητα χαρακτηριστικά της C++. Προστέθηκαν διεργασίες, όπως η αυτόματη συλλογή των "σκουπιδιών" (automatic garbage collection), διευκολύνοντας τον προγραμματισμό σε Java. Με την διεργασία της αυτόματης συλλογής "σκουπιδιών", που συνιστάται από την περιοδική αποδέσμευση της μνήμης

που δεν χρησιμοποιείται, μεγάλο μέρος από την χρονοβόρα δουλεία των προγραμματιστών αυτοματοποιείται και μειώνονται τα bugs της εφαρμογής.

Ένα πλεονέκτημα της Java που οφείλεται στην απλότητα της είναι το μέγεθος των απαραίτητων εργαλείων. Ο μεταφραστής της *Java* και οι βασικές βιβλιοθήκες είναι μικρές και ο κώδικάς της Java είναι τόσο μικρός σε μέγεθος που μπορεί άνετα να τρέξει σε οποιαδήποτε μικρή μηχανή ή να κατέβει από το δίκτυο.

1.2.2 Αντικειμενοστραφής

Όταν λέμε ότι μία γλώσσα προγραμματισμού είναι αντικειμενοστραφής, εννοούμε όταν η τεχνική σχεδιασμού ενός προγράμματος να συγκεντρώνεται σε αντικείμενα. Ένα αντικείμενο είναι ο συνδυασμός δεδομένων, διαδικασιών και λειτουργιών με βασική ιδιότητα την απόκρυψη του συνδυασμού αυτού. Τα αντικείμενα δεν είναι ανεξάρτητα μεταξύ τους, αλλά βρίσκονται σε σχέση αλληλεξάρτησης με τα υπόλοιπα. Ακόμη υπάρχει η έννοια της κληρονομικότητας μεταξύ των αντικειμένων, δηλαδή ένα αντικείμενο μπορεί να κληρονομήσει δεδομένα από άλλα αντικείμενα.

Οι γλώσσες αντικειμενοστραφή προγραμματισμού είναι γλώσσες υψηλού επιπέδου, αφαιρετικές, αποτελεσματικές, γρήγορες και χρησιμοποιούνται για την δημιουργία μεγάλων και σημαντικών εφαρμογών.

1.2.3 Συμβατή με Δίκτυα

Η Java έχει μια μεγάλη βιβλιοθήκη από ρουτίνες για την επιτυχημένη συνεργασία με τα πρωτόκολλα HTTP και FTP. Κατ'αυτόν τον τρόπο, οι δικτυακές συνδέσεις δημιουργούνται ευκολότερα από ότι με την C ή την C++. Τα προγράμματα σε Java μπορούν να έχουν πρόσβαση μέσω δικτύου σε αντικείμενα, με την ίδια άνεση που ένας χρήστης προσπελάζει ένα τοπικό σύστημα αρχείων.

1.2.4 Σταθερή

Η Java προορίζεται για την σύνταξη προγραμμάτων που θα είναι αξιόπιστα από όλες τις πλευρές. Δίνεται έμφαση στον έλεγχο για πιθανά προβλήματα και στον έλεγχο σε πραγματικό χρόνο και στην εξάλειψη καταστάσεων που προκαλούν λάθη.

Η μεγαλύτερη διαφορά μεταξύ της Java και C/C++ είναι ότι η Java έχει ένα μοντέλο δεικτών που εξαφανίζει την πιθανότητα της επαναχρησιμοποίησης της μνήμης και την καταστροφή των δεδομένων. Αντί για αριθμητικούς δείκτες (pointer arithmetic), η Java έχει πραγματικούς πίνακες (true arrays).

Οι προγραμματιστές της Java δεν έχουν να φοβηθούν την ακούσια (ή μη) τροποποίηση της μνήμης, γιατί δεν υπάρχουν δείκτες (pointers). Εξάλλου, τα προγράμματα σε Java δεν μπορούν να αποκοτήσουν μη εγκεκριμένη πρόσβαση στην μνήμη.

1.2.5 Ασφαλής

Η Java προορίζεται για χρήση σε ανοικτά, δικτυωμένα περιβάλλοντα. Γι' αυτό το λόγο, ιδιαίτερη προσοχή έχει δοθεί στην ασφάλεια που παρέχει η γλώσσα. Η Java επιτρέπει την κατασκευή προγραμμάτων ελεύθερων από ιούς και η τροποποίηση τους είναι αδύνατη. Οι τεχνικές πιστοποίησης ταυτότητας βασίζονται στην ασύμμετρη κρυπτογραφία.

Υπάρχει μεγάλη σχέση μεταξύ του τρόπου διαχείρισης της μνήμης και της παρεχόμενης ασφάλειας. Αλλαγές στην σημασιολογία των δεικτών της μνήμης κάνουν αδύνατη την μη έγκυρη πρόσβαση στα δεδομένα της μνήμης ή της πρόσβασης των δεδομένων των αντικειμένων. Με αυτόν τον τρόπο καταπολεμούνται οι περισσότεροι ιοί.

1.2.6 Ουδέτερη της Υποκείμενης Αρχιτεκτονικής

Η Java έχει σχεδιαστεί για να υποστηρίζει δικτυακές εφαρμογές. Ένα δίκτυο, όμως, αποτελείται από ποικιλία διαφορετικών συστημάτων, με διαφορετικά CPU και λειτουργικά συστήματα. Για να μπορούν οι Java εφαρμογές να εκτελούνται παντού στο δίκτυο, το πρόγραμμα Java πρέπει να περάσει από δύο διαδικασίες ώστε να καταλήξει σε εκτελέσιμη μορφή. Πρώτα ο μεταγλωττιστής, μετατρέπει τον πηγαίο κώδικα του προγράμματος σε μία ενδιάμεση γλώσσα που καλείται Java bytecodes. Τα Java bytecodes είναι ανεξάρτητα της πλατφόρμας και με τη χρήση του ερμηνευτή (interpreter) κάθε bytecode εντολή μετατρέπεται σε κατάλληλη δυαδική μορφή για να τρέξει στον εκάστοτε υπολογιστή. Η μεταγλώττιση (compilation) συμβαίνει μόνο μια φορά για κάθε Java πρόγραμμα, η ερμηνεία (interpretation) γίνεται κάθε φορά που το πρόγραμμα εκτελείται.

Τα Java bytecodes μπορούμε να τα φανταστούμε σαν την γλώσσα μηχανής για την Java Virtual Machine (JVM). Κάθε Java ερμηνευτής (π.χ. ένας Web browser που μπορεί να τρέχει applets) είναι μια λογισμική εφαρμογή της Java Virtual Machine. Η JVM αναλαμβάνει να μετατρέψει τα bytecodes σε κατάλληλη εκτελέσιμη μορφή, ανάλογα με το υποκείμενο software και hardware.

Η τεχνική που περιγράφηκε παραπάνω καλείται "write once, run anywhere". Το Java πρόγραμμα μεταγλωττίζεται μία φορά σε Java bytecodes με τον μεταγλωττιστή της Java. Έπειτα, τα bytecodes μπορούν να τρέξουν σε οποιαδήποτε μηχανή που έχει μία εφαρμοσμένη JVM (ο ερμηνευτής).

1.2.7 Φορητή

Το γεγονός ότι είναι ανεξάρτητη της υποκείμενης πλατφόρμας αποτελεί μεγάλο μέρος του ότι είναι φορητή, άλλα υπάρχουν και άλλα σημεία που χαρακτηρίζουν την φορητότητα της.

Σε αντίθεση με την C/C++ δεν υπάρχουν καθόλου χαρακτηριστικά που εξαρτούνται από το CPU του υπολογιστή. Έτσι, τα μεγέθη των πρωταρχικών τύπων δεδομένων είναι καθορισμένα και η συμπεριφορά τους είναι παντού η ίδια. Για παράδειγμα, "int" σημαίνει πάντα ένας 32-bit ακέραιος και το "float" αντιπροσωπεύει πάντα έναν 32-bit floating αριθμό.

1.2.8 Interpreted

Τα Java bytecodes μεταφράζονται σε πραγματικό χρόνο σε εντολές μηχανής που εξαρτώνται από την πλατφόρμα και δεν αποθηκεύονται πουθενά. Η διαδικασία είναι πιο αποτελεσματική και γρήγορη. Μαζί με τα bytecodes μεταφέρονται πληροφορίες που μπορούν να χρησιμοποιηθούν κατά την εκτέλεση και παρέχουν την βάση για τους ελέγχους που πραγματοποιεί ο συνδετής (linker). Επίσης τα προγράμματα γίνονται πιο επιδεκτικά σε debugging διαδικασίες.

1.2.9 Υψηλής Απόδοσης

Η διαδικασία παραγωγής των εντολών μηχανής είναι απλή και γρήγορη. Ο κώδικας που προκύπτει είναι αποτελεσματικός. Ο μεταγλωττιστής από την μεριά του εφαρμόζει αυτόματη κατανομή των καταχωρητών (automatic register allocation) όταν παράγει τα bytcodes. Η τελική μορφή του κώδικα (εκτελέσιμη δυαδική μορφή) είναι μικρή σε μέγεθος και ταχύτατη στην εκτέλεση.

1.2.10 Πολυνηματική

Τα προγράμματα σε Java έχουν την δυνατότητα να αντιμετωπίζουν πολλές καταστάσεις-διαδικασίες ταυτόχρονα. Σε αντίθεση, η C και C++ είναι μονονηματικές γλώσσες. Τα πλεονεκτήματα της πολυνημάτωσης είναι η καλύτερη συμπεριφορά σε πραγματικό χρόνο και η καλύτερη αλληλεπιδραστική ανταπόκριση.

1.2.11 Δυναμική

Η Java είναι πιο δυναμική γλώσσα από την C ή C++. Έχει αναπτυχθεί για να προσαρμοστεί σε ένα εξελισσόμενο περιβάλλον. Οι βιβλιοθήκες εργαλείων αναπτύσσονται ελεύθερα με την πρόσθεση νέων μεθόδων και μεταβλητών, χωρίς να επηρεάζονται οι ήδη υπάρχουσες εφαρμογές.

1.3 Η JAVA πλατφόρμα

Πλατφόρμα είναι το hardware ή software περιβάλλον όπου τρέχει ένα πρόγραμμα. Η Java πλατφόρμα διαφέρει από τις άλλες πλατφόρμες, γιατί είναι μία software-only πλατφόρμα που τρέχει πάνω από άλλες hardware πλατφόρμες. Οι υπόλοιπες πλατφόρμες περιγράφονται σαν συνδυασμός hardware και software.

1.3.1 Η εικονική μηχανή της Java (*Java Virtual Machine*)

Η Java παρέχει την δυνατότητα "write once, run everywhere" μέσω της JVM. Η JVM εφαρμόζεται πάνω από το λειτουργικό σύστημα της μηχανής και τα προγράμματα σε Java τρέχουν πάνω από την virtual machine. Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή javac, ο οποίος παράγει έναν αριθμό από αρχεία .class (κώδικας byte ή bytecode). Ο κώδικας byte είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττίζεται. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (να σημειωθεί εδώ ότι αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine). Πιο σύγχρονες εφαρμογές της Εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα bytecode απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Πρέπει να σημειωθεί ότι η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

1.3.2 Συμπεράσματα

Η Java είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα, εύκολη στην εκμάθηση, γρήγορη, ασφαλής, χωρίς πολλές απαιτήσεις και με πολλές δυνατότητες. Το κυριότερο πλεονέκτημα της που την κάνει να ξεχωρίζει από τις υπόλοιπες γλώσσες προγραμματισμού είναι ότι τρέχει παντού. Τα προγράμματα που είναι γραμμένα σε *Java* τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh καθώς και σε κονσόλες παιχνιδιών χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Ακόμη όλα τα εργαλεία που χρειάζεται κάποιος για να γράψει Java προγράμματα έρχονται δωρεάν, από το περιβάλλον ανάπτυξης μέχρι εργαλεία build. Γι' αυτούς τους λόγους χρησιμοποιείται άλλωστε και στο λειτουργικό σύστημα Android.

2 Android

2.1 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα για κινητά τηλέφωνα και όχι μόνο (βασισμένο στο Linux) το οποίο αναπτύσσεται από την Google. Είναι ανοιχτού κώδικα λογισμικό, το οποίο σημαίνει ότι οποιοσδήποτε μπορεί να δει τον πηγαίο κώδικα και να τον χρησιμοποιήσει ή παραμετροποιήσει.

Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Δίνει την δυνατότητα στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java που προαναφέρθηκε πιο πάνω, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android πραγματοποιήθηκε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού όπως επίσης και κατασκευής hardware. Οι εταιρίες αυτές ασχολούνται με την ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google έχει δημοσιεύσει το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

Το πρώτο κινητό με Android κυκλοφόρησε από την HTC με την έκδοση 1.0 Android και ονομάζονταν HTC Dream τον Οκτώβριο του 2008. Αργότερα ακλούθησαν κι άλλοι κατασκευαστές που το υιοθέτησαν στα κινητά τους και αργότερα η Google παρουσίασε το πρώτο επίσημο έξυπνο κινητό της Nexus One. Η συσκευή κατασκευάστηκε από την HTC και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010. Μέχρι το τέλος του 2010 το Android κατάκτησε την αγορά ξεπερνώντας τις υπόλοιπες πλατφόρμες για έξυπνα κινητά παγκοσμίως.

2.2 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, μιας εφαρμογής 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 (Εικόνα 1) θα δούμε οπτικά την αρχιτεκτονική αυτή.



Εικόνα 1 Αρχιτεκτονική 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)

2.3 Android Linux Kernel

Όπως αναφέραμε και πιο πάνω η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.4.0 για το Android 5.0.2) του Linux Kernel, η οποία υποστηρίζει όλες τις βασικές λειτουργίες του λειτουργικού συστήματος Android. Οι λειτουργίες αυτές αφορούν διαχείριση διεργασιών, διαχείριση μνήμης, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και γενικά ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του λογισμικού με του υλικού της συσκευής. Ενδεικτικά ο πυρήνας (kernel) του Android περιέχει:

- Οδηγό κάμερας
- Οδηγό Wifi και Bluetooth
- Οδηγό προβολής οθόνης
- και οτιδήποτε άλλο hardware υπάρχει στην συσκευή

Μπορεί ο πυρήνας του Android να βασίζεται στον πυρήνα του Linux, αλλά έχει αρκετά διαφορετικά στοιχεία. Ο κυριότερος λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και γρηγορότερος για χρήση σε κινητές συσκευές. Αυτό στην πράξη σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds που δημιούργησε τα Linux, έχει αναφέρει ότι σταδιακά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει για λίγα χρόνια ακόμα.

2.3.1 Τι είναι ο πυρήνας λειτουργικού συστήματος

Ο όρος πυρήνας (kernel) στους υπολογιστές αναφέρεται στο βασικό, θεμέλιο τμήμα ενός λειτουργικού συστήματος, το οποίο αλληλεπιδρά στενά με το υλικό(hardware) του υπολογιστή. Ο πυρήνας (kernel) είναι υπεύθυνος για την διαχείριση των διαθέσιμων πόρων του συστήματος, τον χωρισμό τους στις εκτελούμενες εφαρμογές και την επικοινωνία μεταξύ υλικού και λογισμικού.

Δεν υπάρχει γενική σύμφωνη γνώμη σχετικά με το θέμα διαχωρισμού λειτουργικού συστήματος και πυρήνα στους υπολογιστές και δεν είναι σπάνιο το Λειτουργικό Σύστημα να ταυτίζεται με τον πυρήνα του, θεωρώντας οτιδήποτε εκτελείται στον χώρο χρήστη ως συνοδευτικό λογισμικό συστήματος ή λογισμικό εφαρμογών.

2.3.2 Τι είναι ο πυρήνας Linux

Σε κάθε πολυεπεξεργαστικό λειτουργικό σύστημα, υπάρχει ένα πολύ σημαντικό τμήμα κώδικα που λέγεται πυρήνας και αναλαμβάνει την διανομή του χρόνου μεταξύ των διεργασιών και την διανομή της μνήμης μεταξύ των εφαρμογών . Προαιρετικά μπορεί να περιλαμβάνει οδηγούς συσκευών για κάρτες δικτύου, δίσκους, σειριακές θύρες κλπ, τα διάφορα πρωτόκολλα επικοινωνίας και λοιπό κώδικα για την διαχείριση τους. Ο πυρήνας στο Λειτουργικό Σύστημα Linux είναι το σημαντικότερο γνώρισμα του, καθώς εκτός από το λειτουργικό σύστημα εμπεριέχει και τους οδηγούς συσκευών και πλήθος υλοποιήσεις πρωτοκόλλων και συστημάτων αρχείων.

Ο πυρήνας Linux δημιουργήθηκε ως πρώτο στάδιο από τον Φιλανδό φοιτητή πληροφορικής Linus Torvalds το 1991. Το Λειτουργικό Σύστημα Linux από την αρχή της δημιουργίας του συγκέντρωσε προγραμματιστές και χρήστες οι οποίοι προσάρμοσαν κώδικα από άλλα προγράμματα ελεύθερου λογισμικού για χρήση με το καινούριο λειτουργικό σύστημα. Ο πυρήνας έχει μαζέψει πολλές συμμετοχές από χιλιάδες προγραμματιστές, και πάνω σε αυτόν βασίζονται πολλές διανομές Linux.

Σήμερα το Linux παρέχει όλα τα χαρακτηριστικά όσα θεωρούνται αναγκαία για ένα σύγχρονο πυρήνα λειτουργικού, όπως:

- διαμοιραζόμενες βιβλιοθήκες
- υποστήριξη πολυεπεξεργαστικών συστημάτων (SMP)
- εικονική μνήμη
- σωστή διαχείριση μνήμης
- πραγματική πολυδιεργασία
- δικτύωση μέσω TCP/IP κ.α.

2.4 Βιβλιοθήκες (Libraries)

Στο 2ο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των προγραμμάτων. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά αφομοιώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Επί της ουσίας αποτελούν ένα από τα δομικά υλικά των προγραμμάτων και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Λειτουργικού Συστήματος Android γίνονται αντιληπτές στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής.

Σχεδόν όλο το σύνολο των βιβλιοθηκών είναι γραμμένο στις γλώσσες προγραμματισμού 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 αρκετών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG

- Surface Manager – το οποίο διαχειρίζεται την πρόσβαση στο σύστημα προβολής και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία κατάγονται από πολλαπλές εφαρμογές.
- LibWebCore – μια μοντέρνα μηχανή υποστήριξης πλοήγηση στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- 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 εφαρμογής.

2.5 Η εικονική μηχανή Dalvik (Dalvik Virtual Machine)

Σχεδόν το γενικό 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 δική 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 αυτό το καθιστά δυνατό να τρέχει τις εφαρμογές του σε πολλά νήματα ταυτόχρονα και να απασχολεί 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. Εξαιτίας του απλού της σχεδιασμού, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

2.6 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)

Ο χρόνος εκτέλεσης των εφαρμογών του Android, ανήκει στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Εδώ υπάρχει ένα κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της εικονικής μηχανής Dalvik τις λειτουργίες της οποίας, περιγράψαμε παραπάνω.

2.7 Πλαίσιο Εφαρμογής (Application Framework)

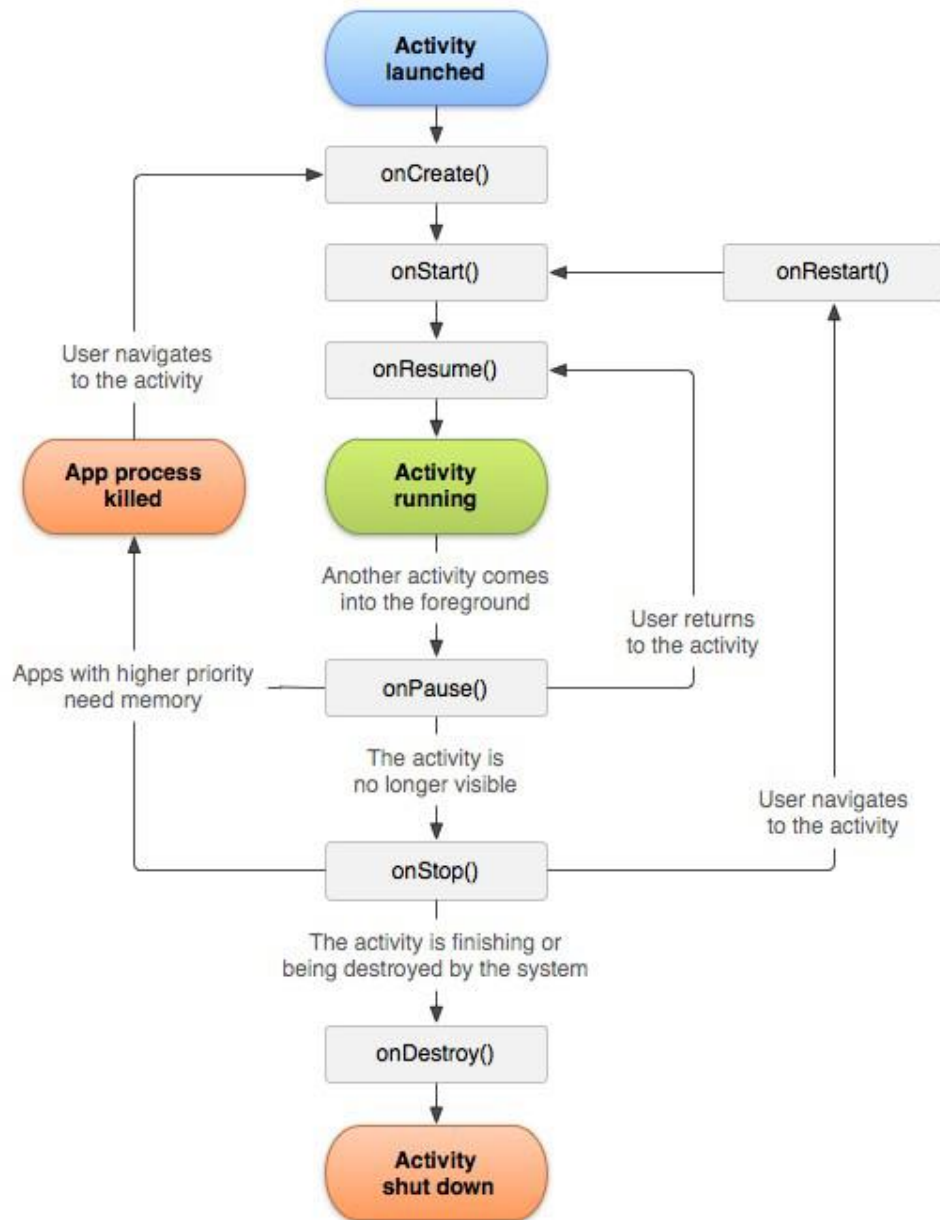
Το ΛΣ Android παρέχει στους προγραμματιστές μια πλατφόρμα ανάπτυξης ανοιχτού κώδικα και τη δυνατότητα να αναπτύξουν με αυτή καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι προγραμματιστές έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να έχουν πρόσβαση σε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

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

- Σύστημα προβολών (View System) – αποτελεί ένα σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό ενός προγράμματος.
- Παραδείγματα προβολών είναι οι λίστες (ListView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ
- Πάροχος Περιεχομένου (Content Provider) – δίνει τη δυνατότητα στις εφαρμογές να ανταλλάσσουν ή να μοιράζονται δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο του κινητού. Μερικά παραδείγματα δεδομένων, είναι οι βάσεις δεδομένων των εφαρμογών και οι επαφές του χρήστη.
- Διαχειριστής Πόρων (Resource Manager) – δίνει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.
- Διαχειριστής Ειδοποιήσεων (Notification Manager) – παρέχει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Για

παράδειγμα όπως είναι είναι οι ειδοποιήσεις στην μπάρα ειδοποιήσεων, τα toast μηνύματα στο κάτω μέρος της οθόνης της εφαρμογής, η δόνηση του κινητού, η ενεργοποίηση της οθόνης, κλπ.

- Διαχειριστής Δραστηριοτήτων (Activity Manager) – ο οποίος διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει ικανότητα πλοήγησης από δραστηριότητα σε δραστηριότητα αποθηκεύοντας στη μνήμη τη σειρά εκτέλεσης αυτών. Στο σχεδιάγραμμα (Εικόνα 2) φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας.



Εικόνα 2 Κύκλος ζωής δραστηριότητας(Activity lifecycle)

3 Εφαρμογές Android

Κάθε εφαρμογή αποτελείτε 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 αποτελείτε 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 αποτελείτε από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας της εφαρμογής, αλλά μόνο από ένα κύριο.

3.1 Το αρχείο AndroidManifest.xml

Κάθε project του Android 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. Ακόμη βάσει αυτού του αρχείου το Google Play φιλτράρει τις συσκευές οι οποίες είναι κατάλληλες και συμβατές με την εφαρμογή μας. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό όνομα της εφαρμογής που φαίνεται στον χρήστη
- Οι δραστηριότητες της εφαρμογής

- Οι εκδόσεις των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, παρόχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.
- Τους τύπους οθονών που υποστηρίζει
- Τα μεγέθη των οθονών που υποστηρίζει
- Αν έχει την δυνατότητα να μετακινηθεί στην κάρτα μνήμης ή όχι

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

3.2 Αρχείο build.gradle

Εδώ ορίζονται διάφορα πράγματα που αφορούν το project. Το Gradle είναι ένα εργαλείο κατασκευής γενικής χρήσης. Χτίζει λίγο πολύ οτιδήποτε για να εφαρμοστεί στο σενάριο δημιουργίας. Από μόνο του, εντούτοις, δεν χτίζει τίποτα εκτός κι αν προσθέσετε κώδικα δέσμης ενεργειών κατασκευής.

Τα περισσότερα αρχεία Java είναι αρκετά παρόμοια όσον αφορά τα βασικά: θα πρέπει να οριστούν Java αρχεία πηγαίου κώδικα, να εκτελεστούν κάποιες δοκιμές, και να δημιουργήσει ένα αρχείο JAR που περιέχει τα Classes. Θα ήταν ωραίο αν δεν χρειαζόταν να κωδικοποιήσει όλα αυτά για κάθε project. Ευτυχώς, δεν χρειάζεται με το Gradle. Το Gradle λύνει αυτό το πρόβλημα με τη χρήση των πρόσθετων. Ένα πρόσθετο είναι μια επέκταση για Gradle που διαμορφώνει το project με κάποιο τρόπο, συνήθως με την προσθήκη ορισμένων προ-ρυθμισμένων καθηκόντων που μαζί κάνουν κάτι χρήσιμο. Το Gradle έρχεται με μια σειρά από πρόσθετα, και μπορείτε εύκολα να γράψετε το δικό σας και να τα μοιραστείτε με άλλους. Ένα τέτοιο πρόγραμμα είναι το πρόσθετο του Android. Αυτό το πρόσθετο

προσθέτει κάποιες αυτοματοποιημένες εργασίες στο project με σκοπό να γίνει συμβατό και να μπορεί να τρέξει η εφαρμογή ως εφαρμογή Android .

Με λίγα λόγια το Gradle μας βοηθά να κάνουμε αυτόματα κάποιες ενέργειες μετατροπής του κώδικα για να το τρέξουμε στην συσκευή μας ή στο emulator. Μέσα στο αρχείο Gradle μπορούμε να ορίσουμε:

- Κατασκευαστικές παραλλαγές. Το σύστημα κατασκευής μπορεί να δημιουργήσει πολλαπλά αρχεία APK με διαφορετικό προϊόν και να οικοδομήσει διαμορφώσεις για την ίδια εφαρμογή. Αυτό είναι χρήσιμο όταν θέλετε να δημιουργήσετε διαφορετικές εκδόσεις της εφαρμογής σας χωρίς να χρειάζεται να δημιουργήσετε ξεχωριστό πρόγραμμα ή ενότητες για κάθε έκδοση.
- Εξαρτήσεις. Το σύστημα διαχειρίζεται τις εξαρτήσεις κατασκευής του προγράμματος και υποστηρίζει εξαρτήσεις από το τοπικό σύστημα αρχείων και από απομακρυσμένα σημεία. Αυτό σας αποτρέπει από το να χρειάζεται να αναζητήσετε, να κατεβάσετε και να αντιγράψετε πακέτα για τις εξαρτήσεις σας στον φάκελο του προγράμματος σας.
- Manifest καταχωρήσεις. Το σύστημα κατασκευής σας δίνει τη δυνατότητα να καθορίσετε τιμές για ορισμένα στοιχεία του δηλωτικού αρχείου στη διαμόρφωση εναλλακτικής κατασκευής. Αυτές οι τιμές κατασκευής αντικαθιστούν τις υπάρχουσες τιμές στο αρχείο Manifest που αναλύσαμε πιο πάνω. Αυτό είναι χρήσιμο αν θέλετε να δημιουργήσετε πολλαπλά αρχεία APK για τις ενότητες σας, όπου κάθε ένα από τα αρχεία APK έχει στόχο διαφορετικό όνομα της εφαρμογής, ελάχιστη έκδοση SDK, ή έκδοση SDK.
- Υπογραφή. Το σύστημα κατασκευής σας δίνει τη δυνατότητα να καθορίσετε τις ρυθμίσεις υπογραφής στη διαμόρφωση κατασκευής,

και μπορούν να υπογραφτούν αρχεία APK σας κατά τη διάρκεια της διαδικασίας κατασκευής.

- Proguard. Το σύστημα κατασκευής σας δίνει τη δυνατότητα να καθορίσετε ένα διαφορετικό αρχείο κανόνων Proguard για κάθε παραλλαγή κατασκευής. Το σύστημα κατασκευής μπορεί να τρέξει το Proguard για να «προστατέψει» τα Classes κατά τη διάρκεια της διαδικασίας κατασκευής.
- Δοκιμές. Για τα περισσότερα πρότυπα, το σύστημα κατασκευής δημιουργεί έναν κατάλογο δοκιμών, androidTest και δημιουργεί μια δοκιμή, έτσι ώστε να μην χρειαστεί να δημιουργήσετε μια ξεχωριστή. Το σύστημα κατασκευής μπορεί επίσης να τρέξει τις δοκιμές σας κατά τη διάρκεια της διαδικασίας κατασκευής.

3.3 Οι φάκελοι src & res

Στον φάκελο src (από την λέξη 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.

Ο φάκελος res (από την λέξη resources) περιέχει όλα τα αρχεία εικόνας, κειμένου, ήχου, xml layout τα οποία χρησιμοποιούνται από τα 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 τα οποία χρησιμοποιεί η εφαρμογή, 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.

3.4 Δομικά Μέρη μιας Εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο AndroidManifest.xml. Ποιά 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 του 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. Παραδείγματος χάρη, 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 Περιεχομένου (Content Providers) - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε πιο πάνω 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 είδος υπηρεσίας η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα η τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά

και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής. Ένα παράδειγμα είναι το ποσοστό της μπαταρίας του κινητού.

3.5 Εξέλιξη του Android

Όπως αναφέραμε παραπάνω, το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα. Η εξέλιξη του λόγω της open source φύσης του είναι ραγδαία και αυτό αντικατοπτρίζεται στο γεγονός ότι από τον Σεπτέμβριο του 2008 μέχρι τον Ιούνιο του 2014 έχουν κυκλοφορήσει 17 εκδόσεις. Μέχρι σήμερα υπάρχουν περισσότερο από 1 δισεκατομμύρια ενεργοποιημένες συσκευές με Android σε 190 μέρες και κάθε μέρα ενεργοποιούνται 1.5 εκατομμύριο συσκευές. Το Android πλέον κατέχει περισσότερο από το 82,8% μερίδιο παγκοσμίως από τις πωλήσεις έξυπνων κινητών με την δεύτερη θέση να κατέχει το iOS της Apple. Στην παγκοσμία αγορά των tablets το Android κατέχει το 62% .

Για να δούμε όμως λίγο πως εξελίχτηκε το Android όλα αυτά τα χρόνια. 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 για παράδειγμα ονόματα πόλεων (Windows Vienna, Chicago, Detroit), ονόματα ζώων (OS X Leopard, Lion, Tiger, Puma), στην περίπτωση όμως του Android τα κώδικα ονόματα έρχονται στη μορφή επιδόρπιου!

3.5.1 Cupcake 1.5

Κυκλοφόρησε 30 Απριλίου το 2009 και βασιζόταν στο 2.6.27 Linux kernel.

Πρόσφερε αρκετά καινούργια χαρακτηριστικά και αλλαγές στο γραφικό περιβάλλον του χρήστη.

Μερικά από τα χαρακτηριστικά ήταν:

- Δυνατότητα καταγραφής κινούμενης εικόνας με την χρήση της αντίστοιχης λειτουργίας του τηλεφώνου.
- Μεταφόρτωση αρχείων βίντεο στο YouTube και εικόνων στο Picasa κατευθείαν από το τηλέφωνο.

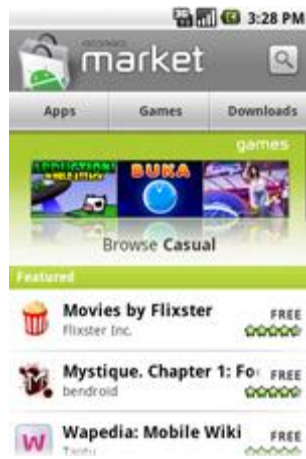
- Επανασχεδιασμένο λογισμικό πληκτρολογίου με λειτουργία αυτόματης συμπλήρωσης κειμένου.
- Δυνατότητα αυτόματης σύνδεσης ασύρματης συσκευής ακουστικού Bluetooth εφόσον εντοπιστεί σε μια συγκεκριμένη απόσταση.
- Νέα widgets και φάκελοι που μπορούσαν να τοποθετηθούν στην επιφάνεια εργασίας.
- Εφέ αλλαγής οθονών και μενού.
- Διευρυμένη λειτουργία αντιγραφής/επικόλλησης για να περιλαμβάνει δικτυακές διευθύνσεις.

3.5.2 Donut 1.6

Κυκλοφόρησε τον Σεπτέμβρη του 2009 και ήταν βασισμένο στην έκδοση 2.6.29 του Linux Kernel.

Προστεθήκαν χαρακτηριστικά όπως:

- Αναζήτηση βάσει φωνής ή κειμένου στο ιστορικό αναζήτησης, επαφών αλλά και στο web.
- Υποστήριξη για τους developers να εμφανίζουν το περιεχόμενο τους στα αποτελέσματα αναζήτησης.
- Εισαγωγή σύνθεσης ομιλίας με δυνατότητα να μιλάει σε διάφορες γλώσσες ένα κείμενο.
- Καλύτερη αναζήτηση και προβολή εικόνων στο Android Market.
- Καλύτερα και πιο γρήγορα προγράμματα για την κάμερα του κινητού.
- Πολλαπλή επιλογή φωτογραφιών για διαγραφή.
- Υποστήριξη τεχνολογιών όπως CDMA, 802.1X, VPN.
- Υποστήριξη μεγαλύτερων οθονών.
- Περισσότερα «εργαλεία» για προγραμματιστές για gestures.



Εικόνα 3 Το ανανεωμένο Android Market

3.5.3 Éclair 2.0-2.1

Κυκλοφόρησε 26 Οκτωβρίου 2009 και βασιζόταν στην έκδοση 2.6.29

Υπήρξαν σημαντικές αλλαγές σε σχέση με τα προηγούμενα:

- Εκτεταμένη δυνατότητα συγχρονισμού, επιτρέποντας στους χρήστες την πρόσθεση πολλαπλών λογαριασμών για συγχρονισμό των email και επαφών τους.
- Συγκεντρωμένη προβολή πολλαπλών λογαριασμών email σε μια σελίδα.
- Υποστήριξη της έκδοσης 2.1 του Bluetooth.
- Καλύτερη προβολή, διαχείριση επαφών.
- Δυνατότητα αναζήτησης των μηνυμάτων.
- Καινούργια χαρακτηριστικά για την κάμερα όπως υποστήριξη zoom, flash, scene modes αλλά και διαφόρων άλλων εφέ.
- Καλύτερη εισαγωγή, γρηγορότερη αλλά και καλύτερη αναγνώριση λεξικού του εικονικού πληκτρολογίου.
- Καλύτερο και γρηγορότερο πρόγραμμα περιήγησης με υποστήριξη του HTML5.
- Καλύτερο πρόγραμμα ημερολογίου.
- Γρηγορότερο και πιο όμορφο γραφικό περιβάλλον.
- Υποστήριξη περισσότερων μεγεθών και τύπων οθονών.
- Καλύτερους χάρτες.
- Καλύτερη υποστήριξη multi-touch.

3.5.4 Froyo 2.2.x

Κυκλοφόρησε 20 Μαΐου 2010 και ήταν βασισμένο στην έκδοση 2.6.32 του Linux Kernel. Τα κυριότερα χαρακτηριστικά της αναβάθμισης όσο αφορά την διεπαφή του λειτουργικού με τον χρήστη είναι:

- Νέο widget στη home screen για να βοηθήσει τους νέους χρήστες να ρυθμίσουν εύκολα και γρήγορα τη συσκευή τους.
- Η λειτουργία τηλεφώνου, ο browser και το μενού εκκίνησης εφαρμογών αποκτούν συντομεύσεις στην αρχική οθόνη.
- Νέες ρυθμίσεις ασφαλείας με δυνατότητα ρύθμισης αριθμητικού κωδικού ή και ανάμεικτου κωδικού με νούμερα και αλφαβητικούς χαρακτήρες για ξεκλείδωμα του πληκτρολογίου. Οι απομακρυσμένοι διαχειριστές μπορούν να «επιβάλλουν» ένα κωδικό σε πολλές συσκευές ασύρματα.
- Οι απομακρυσμένοι διαχειριστές μπορούν επίσης ασύρματα να επαναφέρουν μια συσκευή στις εργοστασιακές τις ρυθμίσεις ή να σβήσουν όλα τα προσωπικά δεδομένα μιας συσκευής σε περίπτωση απώλειας ή κλοπής.
- Υποστήριξη Exchange Calendars μέσα από το ημερολόγιο της συσκευής.
- Εύκολη ρύθμιση ενός Exchange account αφού ο χρήστης χρειάζεται μόνο να ξέρει το user name και το password του.
- Αυτοσυμπλήρωση email διευθύνσεων από τον κατάλογο.
- Αλλαγές στη Gallery με τις φωτογραφίες όπου πλέον μπορεί ο χρήστης να χρησιμοποιήσει «χειρονομίες» επί της οθόνης.
- Νέα κουμπιά στην οθόνη για έλεγχο της φωτογραφικής μηχανής και συγκεκριμένα των zoom, flash, white balance, geo-tagging, focus και exposure.
- Ενεργοποίηση του LED Flash στη λειτουργία βιντεοκάμερας.
- Χρήση της συσκευής (του Nexus One) σαν Wi-Fi hotspot με ταυτόχρονη σύνδεση μέχρι και 8 συσκευών.

- Διαμοιρασμός της 3G σύνδεσης δεδομένων του κινητού σε υπολογιστή με Windows ή Linux συνδεδεμένο μέσω USB.
- Προσθήκη και άλλων γλωσσών στο πληκτρολόγιο της συσκευής με ευκολία εναλλαγής τους με μια κίνηση του χρήστη πάνω στο πληκτρολόγιο.
- «Εγκατάσταση JavaScript μηχανής V8» για γρηγορότερη φόρτωση εξαιρετικά φορτωμένων σελίδων με κωδικοποίηση JavaScript.
- Καλύτερες αποδόσεις στο CPU και στην διαχείριση της μνήμης με τις εφαρμογές να φορτώνουν γρηγορότερα.

3.5.5 Gingerbread 2.3.x

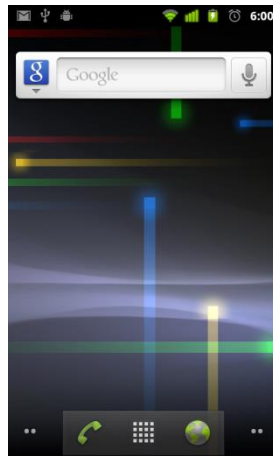
Κυκλοφόρησε 6 του Δεκέμβρη 2010 και ήταν βασισμένη στην έκδοση 2.6.35 του Linux kernel.

Έφερε αρκετές σημαντικές αλλαγές στην ταχύτητα και καινούργια χαρακτηριστικά.

Μερικά από τα χαρακτηριστικά είναι:

- Υποστήριξη SIP VOIP για κλήσεις.
- Καλύτερη και γρηγορότερη εισαγωγή στο πληκτρολόγιο και υποστήριξη φωνητικής εισαγωγής.
- Καλύτερη υποστήριξη του copy-paste.
- Υποστήριξη της τεχνολογίας NFC για ασύρματη μετάδοση δεδομένων και όχι μόνο.
- Καλύτερο ήχο με περισσότερα εφέ.
- Υποστήριξη περισσότερων καμερών.
- Υποστήριξη περισσότερων πρωτοκόλλων ταινιών και ήχων.
- Καλύτερο performance μέσω του garbage collector.
- Υποστήριξη περισσότερων αισθητήρων όπως βαρόμετρο και γυροσκόπιο.

- Επικοινωνία με βίντεο και ήχο μέσω Google Talk.
- Καλύτερα προγράμματα για email, camera.
- Καλύτερα εφέ στο γραφικό περιβάλλον.
- Καλύτερη διαχείριση μπαταρίας.
- Δημιουργία νέου ψηφιακού πορτοφολιού για αγορές-πληρωμές.



Εικόνα 4 Η κεντρική οθόνη της έκδοσης Gingerbread

3.5.6 Honeycomb 3.x

Στις 22 Φεβρουαρίου 2011 κυκλοφόρησε η έκδοση 3.0 που ήταν βασισμένη στην έκδοση 2.6.36 του Linux Kernel και ήταν αποκλειστικά για tablets. Διαθέτει ανανεωμένο γραφικό περιβάλλον με αρκετά 3D στοιχεία, καλύτερο web browser με tabs και ανανεωμένο εικονικό πληκτρολόγιο, βελτιστοποιημένη εφαρμογή για ανάγνωση Google e-books, ειδική έκδοση του Gmail για tablets, δυνατότητα βιντεοκλήσεων μέσω εφαρμογής Google Talk και ανανεωμένη έκδοση Google Maps.

Ακόμη για πρώτη φορά έγινε υποστήριξη για πολυπύρηνους επεξεργαστές και συντονισμός της κάρτας γραφικών με το λειτουργικό σύστημα για να μην επιβαρύνεται ο επεξεργαστής. Επίσης ήταν η πρώτη έκδοση που υποστήριξε την δυνατότητα σύνδεσης περιφερειακών και κατάργησε εντελώς τα φυσικά κουμπιά αφού ενσωμάτωσαν εικονικά στο λειτουργικό

επιτρέποντας στους κατασκευαστές να κάνουν καλύτερα κινητά με μεγαλύτερες οθόνες αλλά με παρόμοιες διαστάσεις.



Εικόνα 5 Η κεντρική οθόνη της έκδοσης Honeycomb

3.5.7 Ice Cream Sandwich 4.x

Κυκλοφόρησε 19 του Οκτώβρη 2011 βασισμένη στον ανανεωμένο Linux kernel 3.0.1. Ίσως μια από τις καλύτερες εκδόσεις Android αφού έφερε τεράστιες αλλαγές στο γραφικό περιβάλλον, υποστήριξε πολυπύρηνους επεξεργαστές και ήταν το πρώτο ενιαίο λειτουργικό σύστημα για tablets και smartphones.

Μερικά από τα σημαντικότερα χαρακτηριστικά είναι:

- Καινούριο, πιο όμορφο χρηστικό γραφικό περιβάλλον με καινούρια animations, fonts αλλά και εικονικά κουμπιά.
- Εύκολο και πιο γρήγορο multitasking.
- Πλήρες σύστημα ειδοποιήσεων με gestures.
- Καλύτερη διαχείριση των widget και αλλαγή μεγέθους.
- Ανασχεδιασμένο πληκτρολόγιο ώστε να παρέχει πιο άμεση διόρθωση λέξεων και καλύτερο Copy, Cut και Paste.
- Καλύτερο και γρηγορότερο πρόγραμμα περιήγησης με δυνατότητα πολλαπλών καρτελών.
- Face unlock, ξεκλείδωμα του κινητού με το πρόσωπο του χρήστη.
- Καλύτερη διαχείριση data.
- Καλύτερα προγράμματα για email, ημερολόγιο και επαφές.
- Καλύτερος διαμοιρασμός αρχείων με το Android Beam μέσω NFC, με μια απλή επαφή των δύο συσκευών.
- Πιο εύκολη δημιουργία φακέλων.
- Υποστήριξη του High Definition (HD) format για ταινίες αλλά και οθόνες κινητών.
- Υποστήριξη της τεχνολογίας WiFi-DIRECT για γρήγορη ανταλλαγή αρχείων μέσω WiFi.
- Γρηγορότερο, πιο σταθερό με καλύτερη διαχείριση πόρων και του hardware.

3.5.8 Jelly Bean 4.1.x – 4.2.x

Μια από τις σημαντικές εκδόσεις του Android που κυκλοφόρησε αρχικά 9 του Ιουλίου 2012 και είναι βασισμένη στην έκδοση 3.1.10 του Linux kernel. Ίσως η καλύτερη έκδοση μέχρι σήμερα γιατί είναι η γρηγορότερη και πιο όμορφη έκδοση μέχρι σήμερα. Είναι η μονή έκδοση που κατάφεραν να συνεργαστούν αρμονικά το software με το hardware. Υλοποίησαν πιο «smooth» γραφικό περιβάλλον με το Project Butter όπου χρησιμοποιώντας triple buffering, extended vsync timing και με ρυθμό 60 καρτέ (frames) ανά δευτερόλεπτο (fps) επετεύχθηκε αυτή η σταθερότητα και ταχύτητα αυτής της έκδοσης.

Τα σημαντικότερα χαρακτηριστικά που παρουσιάστηκαν είναι:

- Εκτεταμένο vsync timing και triple buffering.
- Πρόβλεψη θέσης δαχτύλων του χρήστη.
- CPU input boost.
- Ενισχυμένη προσιτότητα, η οποία εστιάζει το πού τοποθετεί ο χρήστης το χέρι του καθώς εκτελεί εντολές στην οθόνη αφής.
- Αυξημένη υποστήριξη πολλών γλωσσών.
- Νέοι τρόποι εμφάνισης της διεπαφής χρήστη, με μεγαλύτερες ειδοποιήσεις, υποστήριξη περισσότερων μορφών περιεχομένου στη μπάρα ειδοποιήσεων, με τις εφαρμογές να μπορούν να συνοδεύουν από μία έως και τρεις διαφορετικές ειδοποιήσεις.
- App widgets με ικανότητα αλλαγής μεγέθους.
- Φωτογραφίες επαφών υψηλότερης ανάλυσης.
- Καλύτερη εισαγωγή κειμένου και πληκτρολόγιο με gestures.
- Γρήγορη εύρεση συσκευών που προστίθενται ή αφαιρούνται και εντοπισμός των δυνατοτήτων της κάθε συσκευής.
- Android Beam, βασισμένο στο NFC.
- Εύρεση δικτύου WiFi, με υποστήριξη multicast DNS-based υπηρεσιών.
- Υπηρεσία εύρεσης υπηρεσιών WiFi-Direct.
- Διαχείριση εύρους δικτύου.
- Έξοδος USB Audio.
- Πολυκάναλος ήχος και υποστήριξη κωδικοποίησης/ αποκωδικοποίησης ήχου AAC 5.1.
- Καλύτερη εμπειρία browsing με HTML5 video, μεγαλύτερες ταχύτητες rendering και μειωμένη χρήση μνήμης, καλύτερη απόδοση στο HTML5/CSS3/Canvas animation και ανανεωμένη JavaScript Engine.

- Βελτιωμένη εισαγωγή κειμένου.
- Google Cloud Messaging for Android
- Κρυπτογράφηση εφαρμογών.
- Έξυπνες ενημερώσεις εφαρμογής.
- Google Play services (για αγορά μουσικής, ταινιών, βιβλίων).
- Καλύτερες φωτογραφίες με περισσότερα εφέ.
- Καλύτερο lockscreen με δυνατότητα διαφόρων screensaver.
- Υποστήριξη πολλαπλών λογαριασμών σε ένα tablet, έτσι ώστε να είναι δυνατή η χρήση του από διαφορετικούς χρήστες με τις δικές τους προσωπικές ρυθμίσεις (homescreen, widgets, εφαρμογές κλπ.).
- Ο χρήστης μπορεί να στέλνει το περιεχόμενο της Android συσκευής του σε οποιαδήποτε συμβατή συσκευή (τηλεόραση, οθόνη κλπ.).
- Η προσωπική βοήθεια του Google Now γίνεται ακόμα καλύτερη, τώρα με δυνατότητα παρακολούθησης κρατήσεων αεροπορικών εισιτηρίων, ξενοδοχείων, κατάσταση παραγγελιών, ώρες προβολής σε κοντινούς κινηματογράφους, πληροφορίες για συναυλίες κλπ.

3.5.9 Lollipop 5.x – 5.1.x

Η τελευταία έκδοση του Android που υπάρχει μέχρι σήμερα. Κυκλοφόρησε αρχικά 12 του Νοέμβρη 2014 και είναι βασισμένη στην έκδοση 3.1.10 του Linux kernel.

Η έκδοση αυτή έφερε κυρίως αλλαγές στη διεπαφή χρήστη (User Interface) με την παρουσίαση του Material Design ως νέα γλώσσα σχεδίασης. Τα χρώματα και τα σχήματα έγιναν πιο φλάτ, πιο μινιμαλιστικά και παρουσιάστηκαν οι κάρτες για την προβολή και εμφάνιση στοιχείων. Ακόμη προστέθηκε μια νέα εικονική μηχανή στον πυρήνα του Android , η οποία είναι γρηγορότερη και περισσότερο συμβατή με νέους επεξεργαστές , η ART(Android Runtime) με σκοπό να αντικαταστήσει μελλοντικά την Dalvik. Επίσης παρουσιάστηκε το Project Volta όπου προσφέρει καλύτερη αυτονομία και λιγότερη κατανάλωση μπαταρίας.

Αξίζει να αναφερθεί ότι παρουσιάστηκαν και άλλες πλατφόρμες που υποστηρίζει η συγκεκριμένη έκδοση, όπως το Android Wear για έξυπνα ρολόγια, το Android Auto για το σύστημα του αυτοκινήτου, το Android TV

για την τηλεόραση και την πλατφόρμα παρακολούθησης της υγείας Google Fit.

Τα σημαντικότερα χαρακτηριστικά που παρουσιάστηκαν είναι:

- Android Runtime (ART).
- Υποστήριξη για 64-bit CPUs.
- OpenGL ES 3.1 για γραφικά.
- Γραφικά τύπου Vector , για μεγέθυνση χωρίς να αλλοιώνετε η ποιότητα.
- Material design.
- Νέα lock screen.
- Project Volta, για καλύτερη μπαταρία.
- Η Lock screen που υποστηρίζει συντομεύσεις σε προγράμματα και ειδοποιήσεις.
- Πολλαπλοί χρηστές στην ίδια συσκευή.
- Είσοδος και έξοδος ήχου από συσκευές USB.
- Προσθήκη 15 νέων γλωσσών.
- Δυνατότητα έλεγχου συσκευών Bluetooth από το Wi-Fi.
- Υποστήριξη πολλαπλών καρτών SIM.
- Προστασία συσκευής: αν η συσκευή χαθεί ή κλαπεί, παραμένει κλειδωμένη μέχρι να γίνει είσοδος στο λογαριασμό της Google, ακόμη και όταν γίνει επαναφορά των εργοστασιακών ρυθμίσεων της συσκευής.
- High-definition κλήσεις.

3.5.10 Marshmallow 6.x

Η τελευταία αυτή έκδοση του Android έχει κυκλοφορήσει μόνο για developers και αναμένετε να κυκλοφορήσει στα τέλη του Σεπτεμβρίου.

Θα φέρει αλλαγές κυρίως στα προγράμματα του λειτουργικού, υποστήριξη για συσκευές δακτυλικών αποτυπωμάτων, καλύτερη διαχείριση μπαταρίας καθώς και υποστήριξη του νέου τύπου USB type C.

3.6 Γιατί Ανάπτυξη σε Android;

Είναι η πιο πολυσυζητημένη πλατφόρμα τα τελευταία χρόνια με ραγδαία ανάπτυξη και ζήτηση σε ολόκληρο τον κόσμο. Σήμερα το Android κατέχει πλέον περισσότερο από 82.8 % του market share παγκοσμίως που το κάνει νούμερο ένα λειτουργικό σύστημα για έξυπνα κινητά.

Το Android Software Development Kit (SDK) παρέχει στους προγραμματιστές όλα τα εργαλεία και APIs για να αρχίσουν να αναπτύσσουν λογισμικό για την πλατφόρμα Android χρησιμοποιώντας τη γλώσσα προγραμματισμού Java .Ο λόγος που έχει αναπτυχθεί και αγαπηθεί τόσο πολύ είναι επειδή είναι ανοικτού κώδικα λειτουργικό σύστημα και πολλοί προγραμματιστές ασχολούνται μαζί του καθημερινά. Ακόμη τα εργαλεία ανάπτυξης καθώς και οι βιβλιοθήκες και τα βοηθήματα είναι εντελώς δωρεάν σε σχέση με άλλα λειτουργικά συστήματα που έχουν περιορισμένο αριθμό βοηθημάτων και υπάρχει πληρωμή συνδρομής ή ένα συγκεκριμένο ποσό για την ανάπτυξη προγραμμάτων σε αυτά. Ας δούμε όμως μερικά από τα ισχυρά χαρακτηριστικά του Android.

3.6.1 Λειτουργικότητα και ευελιξία.

Το Android είναι μια μοναδική πλατφόρμα που επιτρέπει την ανάπτυξη εφαρμογών λογισμικού το οποίο εκμεταλλεύεται πλήρως τις δυνατότητες μιας συμβατής συσκευής. Για παράδειγμα, οι προγραμματιστές εφαρμογών είναι ελεύθεροι να δημιουργήσουν εφαρμογές που χρησιμοποιούν οποιαδήποτε από τις βασικές λειτουργίες του τηλεφώνου όπως η αποστολή SMS, τηλεφωνικές κλήσεις, τη λήψη φωτογραφιών, το GPS κτλ. Έτσι διευκολύνονται στην ανάπτυξη πιο περίπλοκου και πιο πλούσιου λειτουργικού λογισμικού. Επιπλέον, η πλατφόρμα ανάπτυξης Android είναι μια πλατφόρμα multi-tasking, πράγμα που σημαίνει ότι κάθε εφαρμογή μπορεί να τρέξει στο τηλέφωνο ταυτόχρονα με κάποια άλλη χωρίς να επηρεαστεί η απόδοσή του και αυτό είναι καλύτερο από το να περιορίζεται σε μία εφαρμογή κάθε φορά. Το Android είναι μια πλατφόρμα ανοικτού

κώδικα, πράγμα που σημαίνει ότι μπορεί εύκολα να επεκταθεί και να τροποποιηθεί για να συμβαδίζει και να υιοθετεί τις τελευταίες τεχνολογίες και εξελίξεις. Το γεγονός ότι και η πηγή της πλατφόρμας είναι ανοικτή διασφαλίζει ότι η ανάπτυξη του Android θα έχει συνεχή πρόοδο και θα εξελίσσεται αφού ένας μεγάλος αριθμός ικανών Android προγραμματιστών εργάζεται για τη δημιουργία ελεύθερων για χρήση προηγμένων εργαλείων λογισμικού.

3.6.2 Πλήρης παραμετροποίηση

Δεν υπάρχει διαφορά μεταξύ των λειτουργιών / εφαρμογών οι οποίες είναι ενσωματωμένες στο τηλέφωνο από τις εφαρμογές που δημιουργούνται και προστίθενται από τρίτους προγραμματιστές Android. Οι τελευταίες μπορούν και έχουν την ίδια πρόσβαση σε όλες τις κύριες λειτουργίες της συσκευής, κάτι που επιτρέπει στους τελικούς χρήστες να απολαμβάνουν ένα ευρύ φάσμα εφαρμογών Android που μπορούν να χρησιμοποιηθούν για απεριόριστους σκοπούς. Με συσκευές χτισμένες στην πλατφόρμα Android, οι χρήστες έχουν τη δυνατότητα να προσαρμόσουν πλήρως τη συσκευή τους ανάλογα με τις ανάγκες και τις απαιτήσεις τους. Τυχόν εφαρμογές ακόμα και οι βασικές λειτουργίες μπορεί να τροποποιηθούν ή να αντικατασταθούν πλήρως από άλλες. Για παράδειγμα, ο χρήστης μπορεί να χρησιμοποιήσει την επιθυμητή του εφαρμογή για να εμφανίσει τις φωτογραφίες που είναι αποθηκευμένες στο τηλέφωνό του, ή για να έχει πρόσβαση στην αλληλογραφία του.

3.6.3 Διαδραστικότητα

Οι προγραμματιστές Android μπορούν να δημιουργήσουν πολύπλοκες καινοτόμες εφαρμογές με σχεδόν απεριόριστη λειτουργικότητα. Για παράδειγμα, μια εφαρμογή μπορεί να μεταδώσει τα δεδομένα από το κινητό σας με το διαδίκτυο (κάτι που μπορεί να περιλαμβάνει το ημερολόγιο σας και τις προγραμματισμένες εκδηλώσεις, λίστα με τις επαφές, τις

φωτογραφίες σας και ακόμη και την τρέχουσα θέση σας, αλλά και παραγγελίες, τιμολόγια κτλ) και να λάβει όλα όσα μπορεί να χρειαστεί online και να εμφανίζονται στην οθόνη της συσκευής.

3.6.4 Εύκολη Ανάπτυξη Εφαρμογών για Κινητά

Η πλατφόρμα παρέχει στο καθένα που ασχολείται με την ανάπτυξη εφαρμογών τη δυνατότητα χρησιμοποίησης μια μεγάλης ποικιλίας από βιβλιοθήκες και τα χρήσιμα εκείνα εργαλεία που μπορούν να χρησιμοποιηθούν για τη δημιουργία του πιο εξελιγμένου λογισμικού. Αυτή η ολοκληρωμένη δέσμη από έτοιμα εργαλεία αυξάνει σημαντικά την παραγωγικότητα των προγραμματιστών Android εφαρμογών και τους βοηθά να δημιουργήσουν εκπληκτικά πλούσιο λογισμικό γρηγορότερα και με λιγότερα λάθη.

3.6.5 Συμπεράσματα

Γνωρίζουμε ότι το Android είναι το πρώτο λειτουργικό σύστημα σε χρήση αλλά και σε ανάπτυξη με τεράστια διαφορά παγκοσμίως. Είναι παραμετροποιήσιμο με απεριόριστες δυνατότητες και τα προσφέρει όλα αυτά δωρεάν αφού είναι ανοικτού κώδικα και τα εργαλεία του μπορεί να τα κατεβάσει ο οποιοσδήποτε. Δεν υπάρχει αμφιβολία ότι αξίζει και είναι το λειτουργικό που θα παραμείνει και θα συζητιέται για πολλά χρόνια ακόμη. Οπότε δεν υπάρχει λόγος για κάποιον προγραμματιστή να μην πειραματιστεί έστω μια φορά με το Android και τότε σίγουρα θα τον εκπλήξει το τι μπορεί να καταφέρει.

4 Ανάπτυξη εφαρμογών σε Android

4.1 Κύκλος ανάπτυξης μιας εφαρμογής

Πριν προχωρήσουμε στην υλοποίηση της εφαρμογής μας και επεξήγηση αυτής θα αναφερθούμε σε μερικά πράγματα σχετικά με την ανάπτυξη και την ετοιμασία μιας εφαρμογής για το λειτουργικό σύστημα Android. 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.

4.2 Εγκατάσταση Λογισμικού

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείτε να οργανώσει 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. Συνήθως χρησιμοποιείται και συνιστάτε από την Google το Android Studio αλλά μπορεί να χρησιμοποιηθεί και το Eclipse IDE ή Netbeans IDE.

Στη συνέχεια θα πρέπει να δημιουργηθεί ένας αριθμός από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμαστεί η λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας. Ιδανικά ο developer θα πρέπει να 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 6 Βήμα πρώτο – Εγκατάσταση λογισμικού

4.3 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 εις πέρας 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 7 Βήμα δεύτερο – Ανάπτυξη κώδικα εφαρμογής

4.4 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής

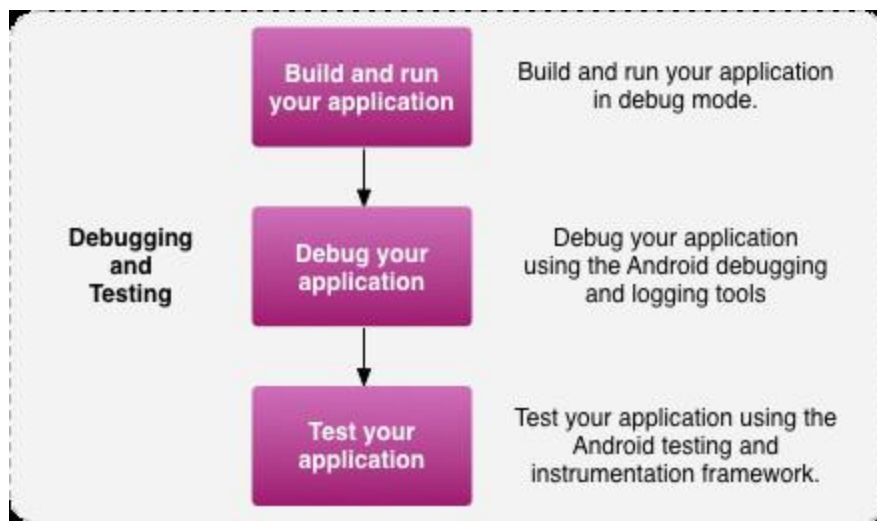
Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείτε από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω. Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug 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.

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 2ου 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. Είναι όμως το σημαντικότερο στάδιο και κάθε σωστός developer πρέπει να το κάνει συνεχώς μέχρι να διορθώσει όλα τα προβλήματα της εφαρμογής πριν να την διαθέσει στο ευρύ κοινό.



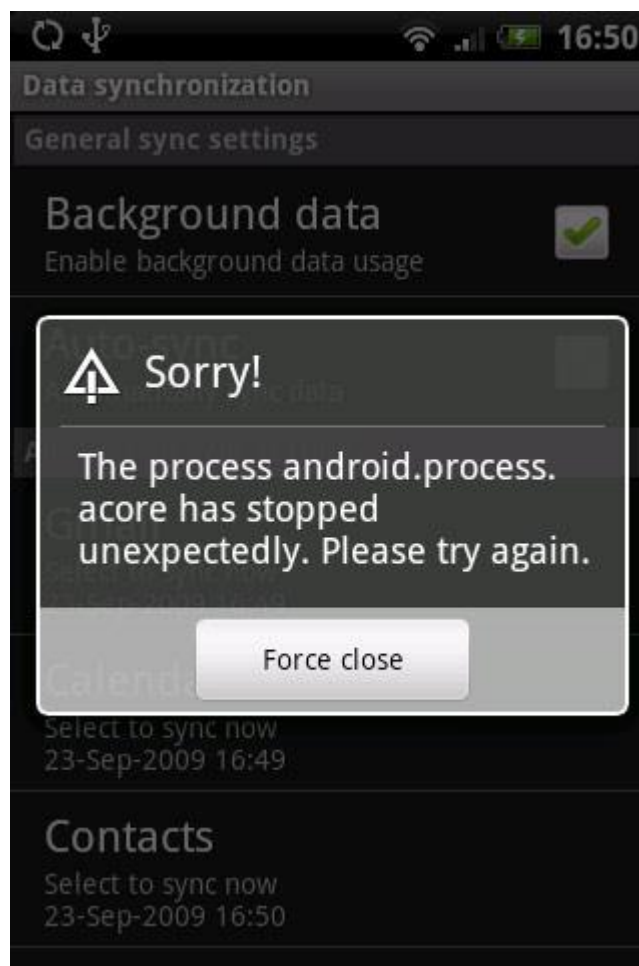
Εικόνα 8 Βήμα τρίτο – Debugging και δοκιμαστική φάση της εφαρμογής

4.4.1 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”.

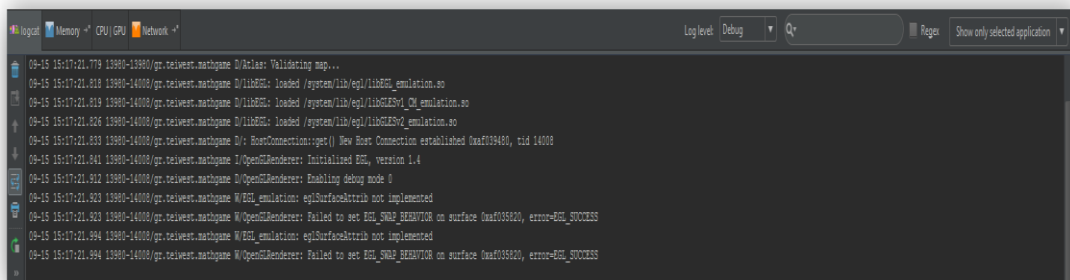
Στον προγραμματισμό υπάρχουν οι λεγόμενες “εξαιρέσεις” (exceptions), καταστάσεις δηλαδή που προκύπτουν όταν κάτι δεν πάει καλά, και αυτό έχει σαν αποτέλεσμα την διακοπή λειτουργίας του προγράμματος σε περίπτωση που δεν έχουμε φροντίσει να “χειριστούμε” την εξαίρεση. Συνηθισμένο παράδειγμα εξαίρεσης λειτουργίας 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 9), δίνοντας μας την “επιλογή” να πατήσουμε “Force Close”.



Εικόνα 9 Παράδειγμα 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 Android Studio ενσωματώνει μία GUI έκδοση του LogCat (Εικόνα 10) για αποτελεσματικότερη αποσφαλμάτωση του κώδικα μας.



Εικόνα 10 Logcat στο Android Studio IDE

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 καταγράφει 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 αυτό 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`.

4.4.2 Android Device Monitor ή Dalvik Debug Monitor Server (DDMS)

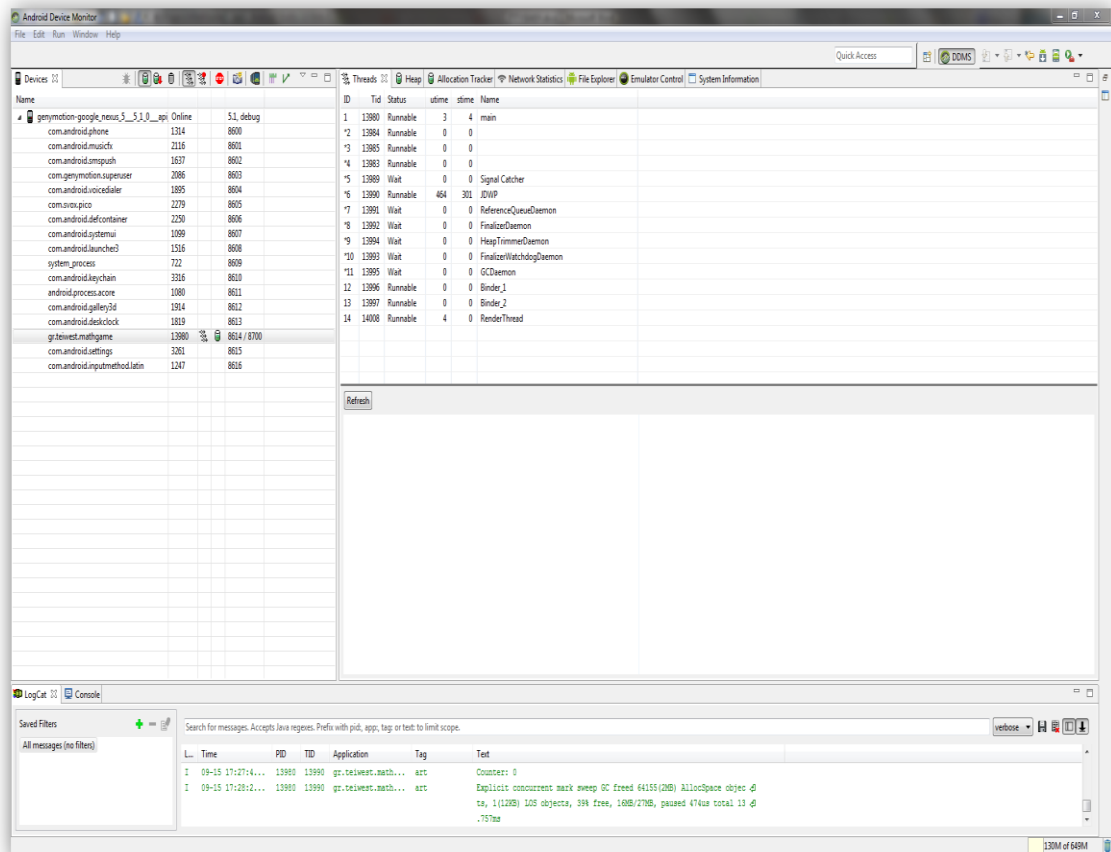
Το Android συνοδεύεται μεταξύ των άλλων, από ένα εργαλείο αποσφαλμάτωσης το οποίο ονομάζεται Android Device Monitor ή Dalvik Debug Monitor Server (DDMS), και το οποίο παρέχει:

- Υπηρεσίες προώθησης θυρών (port forwarding).
- Λήψη εικόνας τις επιφάνειας εργασίας τις συσκευής (screenshot).
- Πληροφορίες για τις διεργασίες και τα νήματα (threads) τις συσκευής.
- Το εργαλείο logcat.
- Πληροφορίες δικτύου και εισερχομένων κλήσεων.
- Δημιουργία ψευδών SMS και πληροφοριών τοποθεσίας.
- Πληροφορίες χρήσης data από την εφαρμογή μας.
- Πληροφορίες για το CPU Usage.

Το DDMS είναι ενσωματωμένο στο Android Studio 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. 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. 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 11 DDMS

Στην παραπάνω εικόνα (Εικόνα 11) βλέπουμε ένα τυπικό παράθυρο του DDMS όπως αυτό εμφανίζεται στο Android Studio IDE. Αριστερά φαίνονται οι συνδεδεμένες συσκευές (μία εικονική μέσω emulator) και από κάτω (κέντρο) υπάρχει η διαχείριση του emulator για να δημιουργήσουμε ψευδή κλήσεις, μηνύματα αλλά και για να αλλάξουμε το τύπο δικτύου.

Δεξιά βλέπουμε ένα παράθυρο με αρκετές καρτέλες. Εκεί βρίσκονται οι πληροφορίες για τα νήματα (threads) του συγκεκριμένου emulator, ποσό heap καταλαμβάνει αλλά και μνήμη και η διαχείριση των αρχείων του. Τέλος στο κάτω μέρος υπάρχει το logcat που αναλύσαμε πιο πάνω και εμφανίζει τα «μηνύματα» της εφαρμογής που επιλέξαμε. Το παράθυρο του DDMS μπορούμε να το αλλάξουμε όπως μας βολεύει εμάς καλύτερα. Μπορούμε να μετακινήσουμε τα «παραθυράκια», να προσθέσουμε ή και να αφαιρέσουμε μερικά.

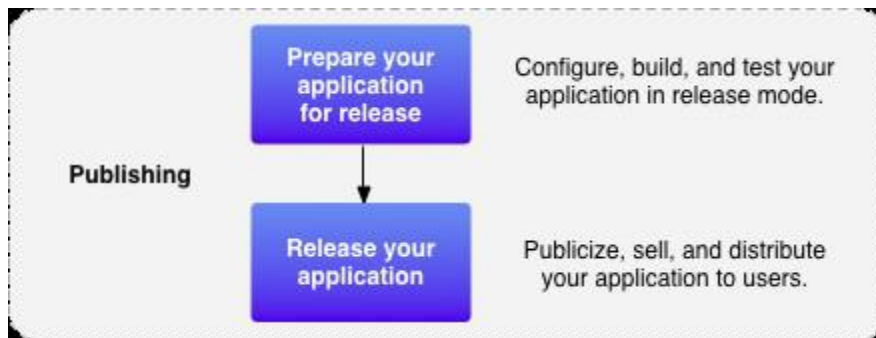
Το DDMS είναι ίσως το χρησιμότερο από όλα τα debugging tools μιας και ενσωματώνει από τα πιο απλά (πχ logcat) μέχρι εξειδικευμένα εργαλεία debugging του Android το network statistics, και απευθύνεται κυρίως σε πιο έμπειρους developers οι οποίοι μπορούν να αξιολογήσουν αποτελεσματικότερα τις ενδείξεις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

4.5 Τελική έκδοση και δημοσίευση της εφαρμογής στο ευρύ 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.

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 12 Δημοσίευση της εφαρμογής

4.6 Android SDK

Το Android SDK (Software Developers Kit) αποτελεί μια συλλογή εργαλείων και βιβλιοθηκών που καθιστούν εφικτή την ανάπτυξη εφαρμογών στο Android. Τη στιγμή που γράφτηκε η παρούσα πτυχιακή, το SDK έχει φτάσει στην έκδοση 24.3.4 η οποία υποστηρίζει το Android 5.1.1(API 23). 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.

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).

4.7 Εικονικές Συσκευές Android (Android Virtual Devices – AVD)

Όπως γράψαμε και πιο πάνω, 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 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 αποτελείται από:

- Το προφίλ του υλικού: Σε αυτό προσδιορίζονται οι ιδιότητες και τα χαρακτηριστικά της εικονικής συσκευής. Μπορούμε παραδείγματος χάρη να ορίσουμε την ανάλυση της οθόνης και την πυκνότητα σε pixel (dpi), το μέγεθος της μνήμης RAM, αν η συσκευή θα έχει κάμερα, υποστήριξη GPS, Wi-Fi 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.

AVD Name	Target Name	Platform	API Level	CPU/ABI
NexusS	Android 2.3.1	2.3.1	9	ARM (armeabi)
QHD	Android 2.3.3	2.3.3	10	ARM (armeabi)
QVGA	Android 2.3.3	2.3.3	10	ARM (armeabi)
APIS-MAPS	Google APIs (Google Inc.)	2.3.3	10	ARM (armeabi)
x86	Intel Atom x86 System Image (Intel Corporation)	2.3.3	10	Intel Atom (x86)
Droid4	Droid4 (Motorola Mobility, Inc.)	2.3.3	10	ARM (armeabi)
Motorola-RAZR	DroidRAZR (Motorola Mobility, Inc.)	2.3.3	10	ARM (armeabi)
Honeycomb	Android 3.2	3.2	13	ARM (armeabi)
Motorola-Xoom2	XOOM2 (Motorola Mobility, Inc.)	3.2	13	ARM (armeabi)
Motorola-Xoom-ME	XOOMME (Motorola Mobility, Inc.)	3.2	13	ARM (armeabi)
Galaxy-Nexus	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
ICS-TABLET-WSVGA	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
ICS-TABLET-WXGA720	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
ICS-TABLET-WXGA800	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
GalaxyCore	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
LG-L4I	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
Netel	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
Slimini	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
GalaxyNexus	Android 4.2.2	4.2.2	17	ARM (armeabi-v7a)
JB-NexusS	Android 4.3.1	4.3.1	18	ARM (armeabi-v7a)
Nexus4	Android 4.3.1	4.3.1	18	ARM (armeabi-v7a)
GalaxyS5	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
LatestPhone	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
Nexus10	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
Nexus7	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
Note2	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)

Εικόνα 13 Οι εικονικές συσκευές

4.8 Android Studio IDE και ADT (Android Development Tools)

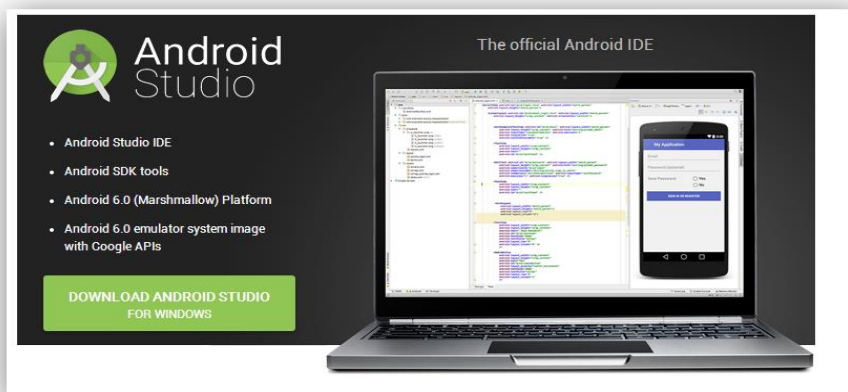
Ο προγραμματισμός στο Android βασίζεται στην γλώσσα Java όπως αναφέραμε πιο πάνω και ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει κώδικα και να επεξεργαστεί τα αρχεία *.Java και *.XML και μετέπειτα να τα κάνει compile μέσω γραμμής εντολών χρησιμοποιώντας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

Η Google δημιούργησε το Android Studio που είναι βασισμένο στο Netbeans και 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 δοκιμάσει 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 πλέον χρησιμοποιεί το Android Studio μαζί με τα SDK Tools οπότε ξεκινάμε με αυτό σαν δεδομένο.



Εικόνα 14 - Android Studio IDE

4.9 Προκλήσεις ανάπτυξης εφαρμογών στο Android

Λόγο της ανοικτού κώδικα φύσεως του Android, υπάρχει μεγάλη υιοθέτηση από πολλές εταιρίες έτσι η ανάπτυξη εφαρμογών στο Android είναι μια

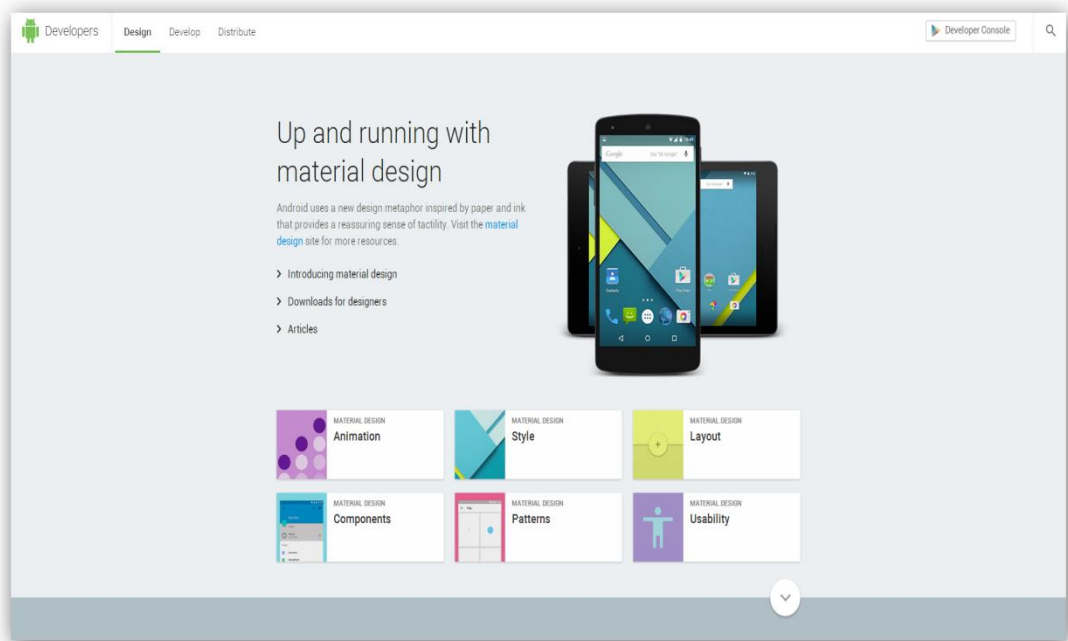
αρκετά απαιτητική διαδικασία, διότι απαιτεί την υποστήριξη εκατοντάδων συσκευών, την υλοποίηση και υποστήριξη ενός λειτουργικού περιβάλλοντος διεπαφής χρήστη, και άλλα πολλά τα οποία θα αναλύσουμε παρακάτω.

4.9.1 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 γι' αυτό 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. 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 (Εικόνα 14) για να καθοδηγήσει τους developers σε μία κοινή γραμμή ανάπτυξης εφαρμογών ώστε να πετύχει αύξηση λειτουργικότητας. Η σελίδα ανανεώθηκε με νέο υλικό όταν παρουσιάστηκε το Material Design που είναι αυτό που χρησιμοποιείται σήμερα ως βάση και στα περισσότερα προγράμματα της Google.



Εικόνα 15 Android Design

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 πώς να επιστρέψει στην αρχική οθόνη ή που βρίσκεται το μενού των επιλογών.

Στο site υπάρχει πληθώρα παραδειγμάτων “καλού σχεδιασμού” τα οποία αφορούν την χρήση του νέου Material Design, της Action Bar, την χρήση των tabs με gestures, την δυνατότητα δηλαδή να αλλάζουμε οθόνες σέρνοντας αριστερά ή δεξιά το δάχτυλό μας στην οθόνη της συσκευής. Επίσης υπάρχει και η ενότητα “Resources” στην οποία υπάρχει ένα πλήρες πακέτο εικονιδίων, χρωμάτων 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 δώσει κάποιο εντελώς διαφορετικό UI (User Interface)

το οποίο όμως θα εξυπηρετεί καλύτερα τους δικούς του σκοπούς, μπορεί να το κάνει ελεύθερα. Αυτό άλλωστε είναι και η ομορφιά του Android!

4.9.2 Υποστήριξη πολλαπλών συσκευών

Μια σημαντική πρόκληση που πρέπει να αντιμετωπίσει κάθε Android Developer είναι η υποστήριξη και σωστή λειτουργία της εφαρμογής του σε πολλαπλές συσκευές.

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 6 κύριες διαθέσιμες εκδόσεις (2.1-5.1.1) με κυρίαρχη έκδοση την 4.4 Kit Kat.
- 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 2GB μνήμης RAM. Εκτός από τις διαφορές σε επίπεδο SOC 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 υπόψη 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.

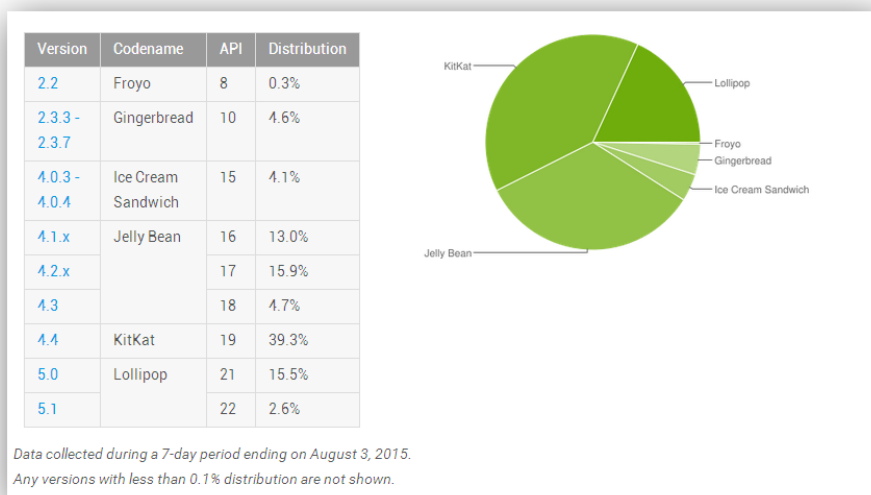
4.9.3 Υποστήριξη παλαιότερων εκδόσεων του Android

Όπως γράψαμε παραπάνω αλλά και στο εισαγωγικό κομμάτι της εργασίας, αυτή τη στιγμή υπάρχουν 6 κύριες διαθέσιμες εκδόσεις του Android με πιο πρόσφατη την έκδοση 5.1.x με κωδική ονομασία “Lollipop” και κυρίαρχη έκδοση την 4.4 “Kit Kat”.

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 τις 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 διάσημες βιβλιοθήκες που έχουν αναπτυχθεί από προγραμματιστές του Android, και έχει καλυφθεί το κενό που άφησε η Google.



Εικόνα 16 Στατιστικά στοιχεία ενεργών συσκευών μέχρι τις 3 του Αυγούστου 2015

Όπως βλέπουμε από τα παραπάνω στατιστικά στοιχεία (Εικόνα 15) που ενημερώνονται αυτόματα κάθε 2 εβδομάδες από την Google, τη μερίδα του λέοντος κατέχουν οι εκδόσεις 4.4 (KitKat) με 39,3% και 4.1.X-4.3(Jelly Bean) με ποσοστό 33.6%. Η νεότερη έκδοση του Android, κατέχει μόλις το 18,1% των συσκευών παρότι κυκλοφόρησε από τον Ιούνιο του 2014, δηλαδή 1 χρόνο και 2 μήνες μετά 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.

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` ή του `build.gradle` και εφόσον έχει καθοριστεί ένα κατώτατο στοχευόμενο 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 αφού σιγουρευτούμε ότι λειτουργεί σε όλες τις εκδόσεις σωστά, ώστε να εξασφαλίσουμε υποστήριξη και στις νεότερες συσκευές.

4.9.4 Υποστήριξη πολλαπλών διαστάσεων οθόνης και πυκνότητας pixel (ppi)

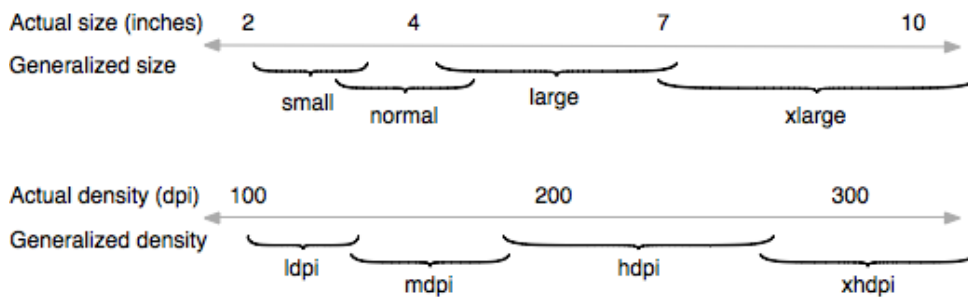
Η ταυτόχρονη υποστήριξη των πολλών εκδόσεων του Android είναι η πρώτη πρόκληση του προγραμματιστή. Η δεύτερη μεγάλη πρόκληση είναι η ταυτόχρονη υποστήριξη των πολλαπλών διαστάσεων οθόνης που διαθέτουν οι εκατοντάδες συσκευές που κυκλοφορούν στην αγορά, και η διαφορετική πυκνότητα pixel που διαθέτει η 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 υπόψη 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 1920 x 1200 pixels (Full HD) για τα νεότερα tablets.

Δημοφιλείς αναλύσεις είναι η 240x320(QVGA), 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).

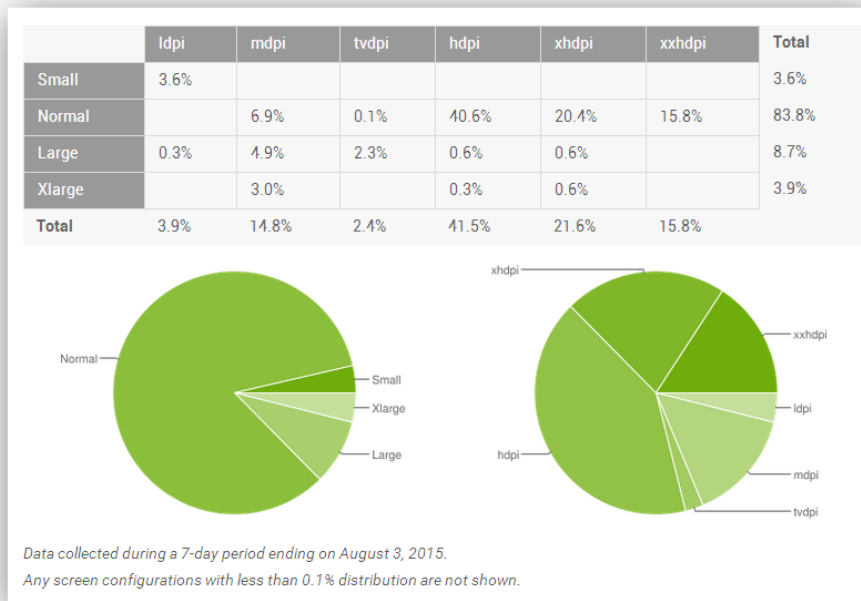
Όπως βλέπουμε υπάρχει μια πληθώρα αναλύσεων για να υποστηρίξει ο προγραμματιστής η κάθε μία με περισσότερο η λιγότερο διαθέσιμο χώρο στην οθόνη. Το Android για πρακτικούς λόγους έχει χωρίσει τις διαφορετικές αναλύσεις οθονών σε τέσσερις κατηγορίες οθονών οι οποίες συσχετίζονται άμεσα με τέσσερις κατηγορίες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



Εικόνα 17 Αναλύσεις οθονών βάσει dpi

Όπως βλέπουμε από το παραπάνω σχεδιάγραμμα (Εικόνα 16), το Android χωρίζει το μέγεθος τις 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 18 Τύπος οθονών και μεγεθών των ενεργοποιημένων συσκευών μέχρι 3 Αυγούστου 2015

Από ότι βλέπουμε από το παραπάνω σχεδιάγραμμα (Εικόνα 17), τα πράγματα εν τέλη δεν είναι και τόσο τραγικά για τον developer! Οι normal μεγέθους οθόνες κατέχουν τη μερίδα του λέοντος σε διαφορετικό μέγεθος DPI.

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 υποστηρίζει η εφαρμογή του με τον ίδιο τρόπο που επιλέγει πια API θα υποστηρίζονται, δηλαδή μέσω του `AndroidManifest.xml` ή του `build.gradle`. 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 οθονών. Δεν πρέπει να ξεχνάμε άλλωστε ότι οι μεγαλύτερες οθόνες παρέχουν και περισσότερο χώρο στον developer για να προβάλλει επιπλέον υλικό, με διαφορετικό ίσως τρόπο από ότι θα το προέβαλε σε μια μικρότερη οθόνη.

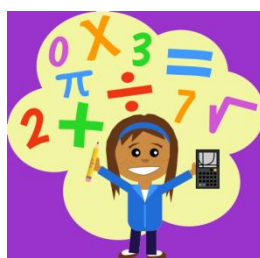
Η δεύτερη δυνατότητα που παρέχει είναι αυτή της χρήσης πολλών γραφικών, διαφορετικών διαστάσεων, ώστε να εξυπηρετούνται σωστά όλες οι αναλύσεις οθόνης. Η υλοποίηση αυτής της δυνατότητας είναι εξαιρετικά απλή. Ο developer δημιουργεί φακέλους στο project του, με το όνομα της διάταξης που θέλει να παρέχει γραφικά η layout (π.χ 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.

5 Το παιχνίδι – Math Game

5.1 Εισαγωγή στο παιχνίδι Android

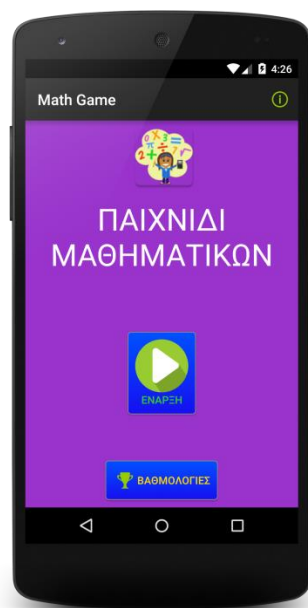
Όπως αναφέραμε πιο πάνω σκοπός της πτυχιακής ήταν να αναπτύξουμε ένα παιχνίδι διασκεδαστικό για τα παιδιά όλων των τάξεων του δημοτικού.

Το πρόγραμμα αυτό είναι μεγέθους ~6,5MB(στο κινητό) και αποτελείται από 5,000 γραμμές κώδικα περίπου (έξω τα library που χρησιμοποιήθηκαν και τα λεκτικά ,3800 περίπου κώδικας και 1000 περίπου το γραφικό περιβάλλον).



Εικόνα 19 Το παιχνίδι

5.2 To Math Game



Εικόνα 20 Το κεντρικό μενού

Το παιχνίδι που δημιουργήσαμε είναι ένα διασκεδαστικό παιχνίδι μαθηματικών για τα παιδιά του δημοτικού.

Υπάρχουν διαφορετικά στάδια με διαφορετική δυσκολία ανάλογα με την τάξη του μαθητή και με διασκεδαστικό τρόπο ο χρήστης θα καλείτε να λύσει μια μαθηματική πράξη. Υπάρχουν δυο τύποι παιχνιδιού, το ελεύθερο χωρίς κανένα περιορισμό και με χρόνο όπου πρέπει να λύσει την πράξη σε ένα συγκεκριμένο χρονικό όριο. Αν ο χρήστης κάνει σωστά την μαθηματική πράξη κερδίζει πόντους και πηγαίνει στο επόμενο επίπεδο μέχρι να τελειώσει το στάδιο-τάξη. Αν κάνει λάθος την μαθηματική πράξη χάνει μια ζωή από τις 3 που έχει διαθέσιμες. Τους πόντους που κερδίζει μπορεί να τους εξαργυρώσει για να δει την λύση ενός επίπεδου σε περίπτωση που δυσκολεύεται. Ακόμη μπορεί να αποθηκεύσει και να δει το σκορ μετά ή να το μοιραστεί με φίλους του ή να συνεχίσει από εκεί που έμεινε την τελευταία φορά.

Υπάρχουν 30 επίπεδα για κάθε τάξη του δημοτικού χωρισμένα σε 3 στάδια, 9 επίπεδα για κάθε επίπεδο. Τα τελευταία 3 στάδια θα είναι μόνους

και θα δίνουν περισσότερους πόντους αλλά στο παιχνίδι με τον χρόνο έχουν λιγότερο χρόνο από όλα τα προηγούμενα στάδια.

Συγκεκριμένα:

- 1 στάδιο 5 πόντους κάθε επίπεδο
- 2 στάδιο 10 πόντους κάθε επίπεδο
- 3 στάδιο 15 πόντους κάθε επίπεδο
- Bonus στάδιο 30 πόντους κάθε επίπεδο

Στο παιχνίδι με τον χρόνο:

- 1 στάδιο 1 λεπτό κάθε επίπεδο
- 2 στάδιο 45 δευτερόλεπτα κάθε επίπεδο
- 3 στάδιο 30 δευτερόλεπτα κάθε επίπεδο
- Bonus στάδιο 15 δευτερόλεπτα κάθε επίπεδο

Χαρακτηριστικά :

1. Actionbar με εικονίδια για όλα τα κινητά και tablets για γρήγορη και αποτελεσματική λειτουργία της εφαρμογής
2. Material Design σύμφωνα με τα πρότυπα και τα στάνταρτ της Google
3. SQLite database με όλες τις πληροφορίες μαζεμένες η οποία μπορεί να γίνει εύκολα ανανέωση
4. Φύλαξη των σκορ των παιχτών
5. Δυνατότητα κοινοποίησης των σκορ των παιχτών
6. Δυνατότητα διαγράψης παιχτών
7. Δυνατότητα πολλαπλών παιχτών

8. Δυνατότητα συνέχισης παιχνιδιού
9. Δυνατότητα εξαργύρωσης πόντων για προβολή απάντησης σε επίπεδα

Ακόμη η εφαρμογή υποστηρίζει τα περισσότερα κινητά και όλα τα tablets (7-10 ιντσών). Συγκεκριμένα υποστηρίζει συσκευές από την έκδοση 3.X (API 11) έως την 5.1.1, οι οποίες έχουν οθόνη με πυκνότητες ldpi, mdpi, hdpi, xdpi, xxdpi, xxxdpi.

Με αριθμούς αυτό σημαίνει ότι η εφαρμογή υποστηρίζει το 95.1% των κινητών και tablets που κυκλοφορούν αυτή την στιγμή.

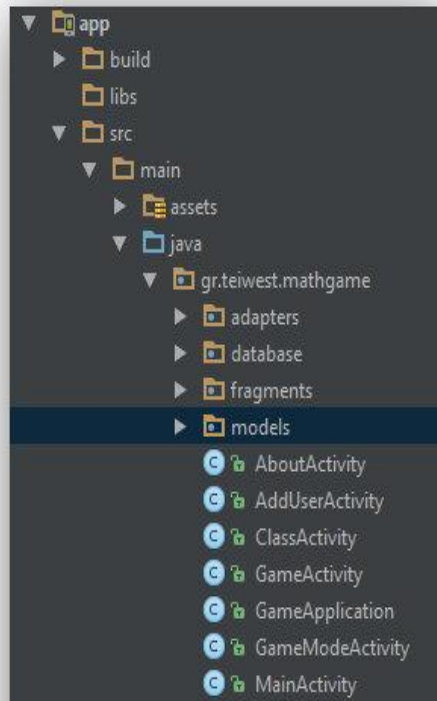
Για το testing χρησιμοποιήθηκαν διάφορα AVD(Android emulator) αλλά και εικονικές συσκευές από το πρόγραμμα Genymotion



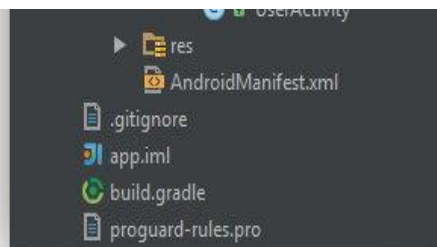
Εικόνα 21 Η εφαρμογή σε διαφορετικές συσκευές

5.3 Δομή προγράμματος

Η δομή του προγράμματος είναι όπως φαίνεται στα αριστερά (Εικόνα 22).



Εικόνα 22 - Δομή προγράμματος



Αποτελείται από 4 πακέτα.

Το `gr.teiwest.mathgame.adapters` το οποίο περιέχει όλα τα custom classes για list-gridview και tabs Adapters που βοηθούν στην γρήγορη φόρτωση και προβολή εξατομικευμένων λιστών του προγράμματος αλλά και των δικών μας tab στην ActionBar.

Στο 2^ο πακέτο `gr.teiwest.mathgame.database` που έχει την κυρίως κλάση της βάσης και όλες τις βοηθητικές μεθόδους για δημιουργία, διάβασμα και εγγραφή ή διαγραφή στην βάση δεδομένων μας.

Το 3^ο πακέτο `gr.teiwest.mathgame.fragments` που περιέχονται όλα τα Fragments για τα tabs των σταδίων.

Το 4^ο `gr.teiwest.mathgame.models` περιέχονται όλα τα κύρια μοντέλα κλάσεις του προγράμματος μας που θα μας βοηθήσουν να επεξεργαζόμαστε και να

χειριζόμαστε κάθε μοντέλο πριν την εγγραφή του ή κατά την ανάγνωση του από την βάση δεδομένων.

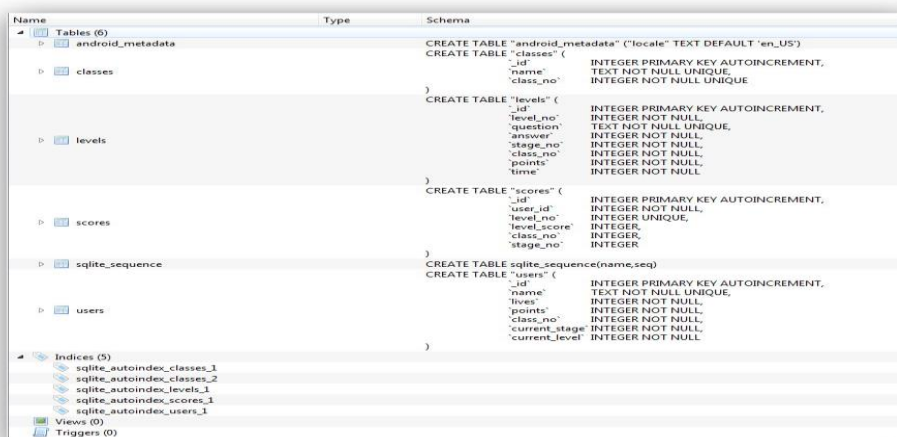
Τέλος το κυρίως package του προγράμματος μας `gr.teiwest.mathgame` περιέχει όλα τα Activities του προγράμματος και όλες τις μεθόδους για την

λειτουργία του παιχνιδιού μας. Με λίγα λόγια ο κώδικας που δρα πίσω από κάθε layout που βλέπει ο χρήστης.

Εκτός των βοηθητικών πακέτων από άλλους συνάδελφους που ευχαριστούμε θερμά που μοιράζονται την πολύτιμη δουλειά τους, χρησιμοποιήθηκαν Support Libraries που βοηθούν στην ομαλότερη και καλύτερη συμβατότητα της εφαρμογής με παλαιότερες εκδόσεις του Android τόσο στο UI αλλά και στην χρήση των API που είναι διαθέσιμα στις νεότερες εκδόσεις μόνο .

Εκτός των αρχείων στον φάκελο src παρατηρούμε ότι υπάρχει και φάκελος res ο οποίος περιέχει πολλούς υποφακέλους. Ο λόγος είναι ότι η εφαρμογή μας θέλουμε να υποστηρίζει όλα τα μεγέθη αλλά και πυκνότητας pixel οθόνες αλλά και τις περισσότερες εκδόσεις κινητών και tablets.

Επίσης στον φάκελο assets βρίσκεται η database που συμπληρώσαμε και θα γίνει αντιγραφή στη συσκευή αργότερα.



Εικόνα 23 Η βάση δεδομένων

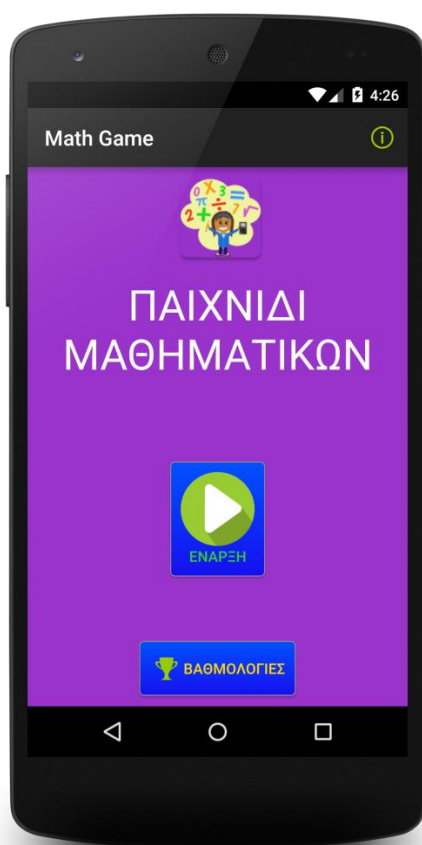
Λόγω το ότι το πρόγραμμα είναι αρκετά μεγάλο (5.000 γραμμές κώδικα) και χρειάζεται πολύ χρόνο για να εξηγήσουμε τι κάνει η κάθε κλάση στο πρόγραμμα μας, δεν θα μπούμε στην διαδικασία ανάλυσης κάθε κλάσης.

Στο τέλος της πτυχιακής υπάρχει ολόκληρο το πρόγραμμα για να το μελετήσετε με σχόλια παντού για την καλύτερη κατανόηση του.

5.4 Λειτουργιά παιχνιδιού

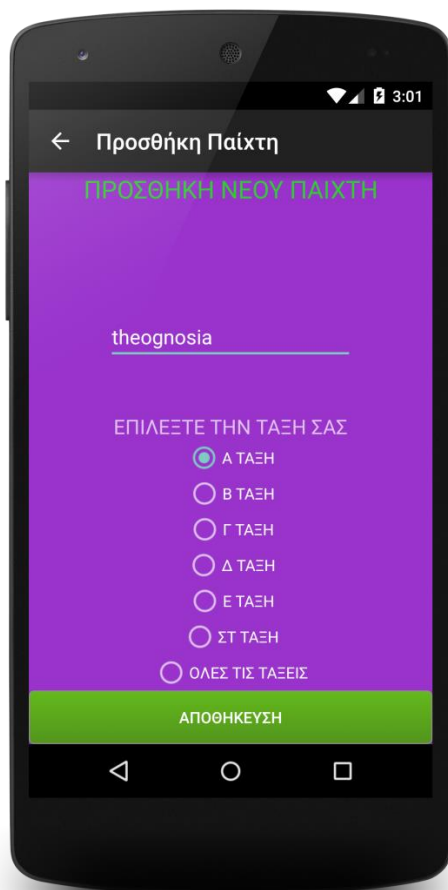
Η εμπειρία της χρήσης της εφαρμογής για τον απλό μέσο χρήστη έχει σχεδιαστεί έτσι ώστε να είναι η καλύτερη και να είναι εύκολη ακόμη και για τα μικρά παιδιά.

Αρχικά όταν ο χρήστης ανοίξει την εφαρμογή βλέπει την πιο κάτω οθόνη (Εικόνα 24). Οπού μπορεί να επιλέξει να δει τις βαθμολογίες ή να παίξει το παιχνίδι πατώντας τα ανάλογα κουμπιά.



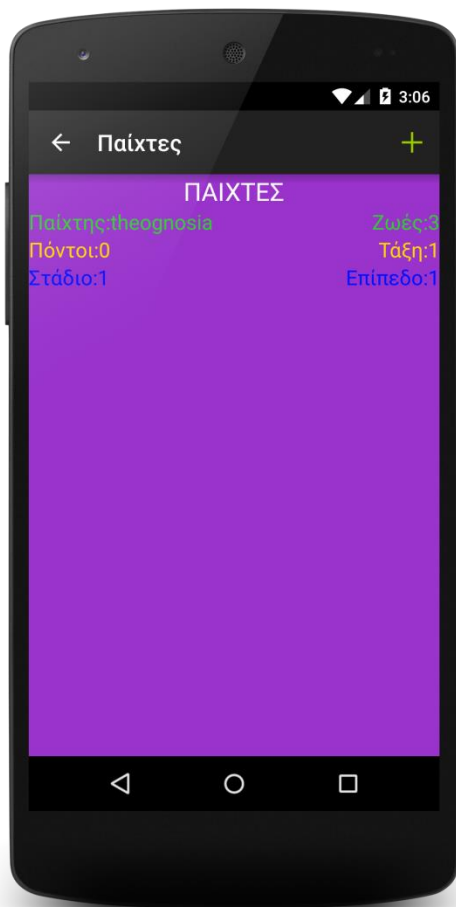
Εικόνα 24 Είσοδος χρήστη

Ο χρήστης για να χρησιμοποιήσει την εφαρμογή πρέπει να κάνει έναν απλό «λογαριασμό» και να καταχωρήσει το όνομα και να επιλέξει την τάξη του ή και ακόμη να επιλέξει όλες οι τάξεις για να παίξει όλα τα στάδια για όλες τις τάξεις (Εικόνα 25).



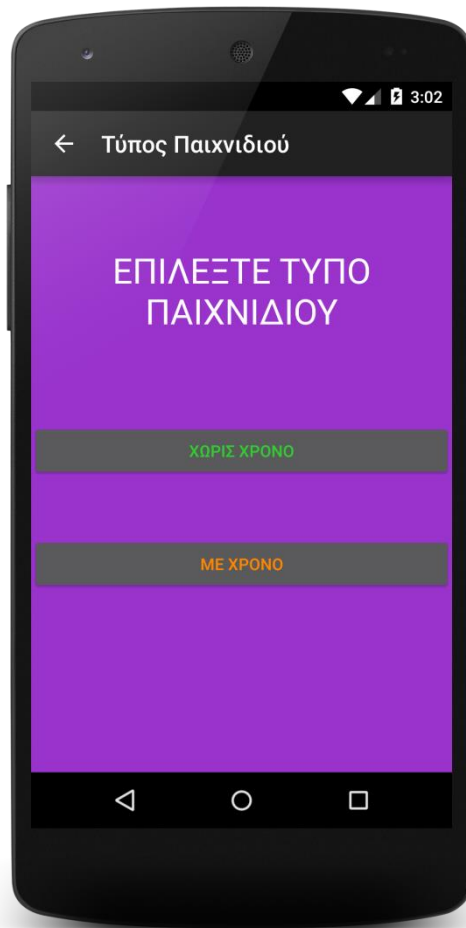
Εικόνα 25 Δημιουργία παίχτη

Την επομένη φορά που θα ξεκινήσει το πρόγραμμα αφού συμπληρώσει τα στοιχεία του θα δει την πιο κάτω οθόνη για να επιλέξει τον χρήστη για να παίξει (Εικόνα 26). Πάνω πάνω βρίσκεται η Action Bar με κουμπί για γρήγορη προσθήκη κάλλου παίχτη.



Εικόνα 26 Οθόνη επιλογής παίχτη

Η επομένη οθόνη που θα δει ο χρήστης είναι αυτή της επιλογής τύπου παιχνιδιού. Αν θέλει να παίξει με χρόνο ή χωρίς. (Εικόνα 27)



Εικόνα 27 Τύπος παιχνιδιού

Αφού ο χρήστης επιλέξει τον τύπο του παιχνιδιού που επιθυμεί είναι έτοιμος να ξεκινήσει. Επομένως η τελευταία οθόνη πριν την αρχή του παιχνιδιού μας είναι αυτή της επιλογής των σταδίων και των επιπέδων για κάθε τάξη (Εικόνα 28). Για κάθε επίπεδο που περνά ξεκλειδώνετε αυτόματα το επόμενο και χρωματίζονται τα ολοκληρωμένα έτσι ώστε να ξέρει που έχει μείνει (Εικόνα 29). Αφού περάσει τα 9 επίπεδα του κάθε σταδίου αυτομάτως πηγαίνει στο επόμενο στάδιο μέχρι και την ολοκλήρωση όλων των σταδίων, συμπεριλαμβανόμενου και του μόνου.



Εικόνα 28 Πρώτο στάδιο, επιλογή επιπέδων



Εικόνα 29 Μετάβαση στο επόμενο επίπεδο

5.5 Πως λειτουργεί το παιχνίδι

Το Παιχνίδι είναι πολύ απλό στην χρήση. Ο παίχτης επιλέγει το επίπεδο που επιθυμεί και μεταφέρετε στην οθόνη με την μαθηματική πράξη που πρέπει να λύσει.

Στη περίπτωση που διάλεξε χωρίς χρόνο στον τύπο του παιχνιδιού έχει όσο χρόνο χρειαστεί να γράψει την λύση της μαθηματικής πράξης στο κουτί και να πατήσει το κουμπί «το βρήκα». Αν δυσκολευτεί και χρειαστεί κάποια βοήθεια μπορεί να πατήσει το κουμπί «βοήθεια» και με την εξαργύρωση 100 πόντων από τους πόντους του θα εμφανιστεί η λύση.

Αν η απάντηση είναι σωστή ακούγεται εύθυμος ήχος και παίρνει στο επόμενο επίπεδο με το κατάλληλο μήνυμα. Αν η απάντηση είναι λάθος ακούγεται άλλος ήχος, χάνει μια ζωή από τις 3 διαθέσιμες και συνεχίζει την προσπάθεια. Αν τελειώσουν οι ζωές αυτομάτως ο παίχτης μηδενίζεται και ξανά ξεκάνει από την αρχή.

Αν βρει όλες τις απαντήσεις (9) του συγκεκριμένου σταδίου μεταφέρετε αυτόματα στο επόμενο στάδιο μέχρι και το μόνους στάδιο. Αν βρει και τις ερωτήσεις του μόνους σταδίου τότε εμφανίζονται τα κατάλληλα μηνύματα και μαρκάρονται ως τελειωμένα όλα τα στάδια. Ο παίχτης είναι ελεύθερος να παίξει ξανά όποιο στάδιο θέλει ή να διαγράψει το σκορ του και να ξανά ξεκινήσει από την αρχή.

Στην περίπτωση που επέλεξε τον τύπο παιχνιδιού με χρόνο, ισχύουν τα πιο πάνω με την μόνη διαφορά ότι για κάθε ερώτηση έχει συγκεκριμένο χρόνο να απαντήσει. Στην περίπτωση που τελειώσει ο χρόνος και δεν έχει δώσει μια απάντηση τότε χάνει μια ζωή.

Τώρα στο προγραμματιστικό μέρος...

Με το άνοιγμα του παιχνιδιού γίνεται αυτόματα αντιγραφή της βάσης μας από τον φάκελο assets στην συσκευή του χρήστη, έτσι ώστε να μπορούμε να δουλέψουμε με την βάση αργότερα. Όταν ο χρήστης δημιουργήσει καινούργιο παίχτη γίνεται αυτόματα η δημιουργία στην βάση του παίχτη με

αρχικές τιμές. Όταν ξεκινήσει το παιχνίδι βάσει με το που πάτησε ο χρήστης φορτώνετε το ανάλογο στάδιο και οι ιδιότητες του. Όταν ο χρήστης περάσει το στάδιο γίνεται ενημέρωση της εγγραφής του χρήστη και δημιουργία νέας εγγραφής για σκορ. Αν ο χρήστης δεν βρει την απάντηση τότε αφαιρείτε μια ζωή και ενημερώνετε η εγγραφή του χρήστη. Με λίγα λόγια όλες οι ενέργειες του χρήστη καλούν μεθόδους από και προς την βάση δεδομένων του παιχνιδιού μας αλλά και μεθόδους για ενημέρωση του γραφικού περιβάλλοντος όπως του χρονομέτρου, των ζώων κτλ.

5.6 Συμπέρασμα και Μελλοντικά σχέδια

Η ανάπτυξη εφαρμογής στο Android είναι μια απαιτητική εργασία η οποία απαιτεί σημαντικές γνώσεις Java και δεν είναι τόσο εύκολο όσο φαίνεται στον απλό χρήστη. Χρειάζεται πάνω από όλα επίμονη και υπομονή εκτός από το πολύ διάβασμα που έπρεπε να κάνουμε.

Μας δυσκόλεψαν πολύ οι μέθοδοι για τις εγγραφές στη βάση δεδομένων αλλά και για την αποθήκευση σε αυτή από διάφορους τόπους του προγράμματός μας. Ακόμη είχαμε αρκετά προβλήματα με τα σκορ και την φύλαξη αυτών αλλά και με την βάση που δεν έκλεινε σωστά με αποτέλεσμα να αποτυγχάνει το πρόγραμμα στην πορεία. Ένα ακόμη πρόβλημα που συναντήσαμε ήταν με την δημιουργία των δικών μας adapter για τα στάδια και την επισήμανση των συμπληρωμένων καθώς και του επόμενου που ξεκλείδωσε ο χρήστης...

Ευτυχώς για εμάς υπήρχαν άπειρα παραδείγματα στο Ιντερνέτ για όλες τις απορίες μας και βρήκαμε αρκετούς κώδικες από συνάδελφους που μας γλίτωσαν πολύ χρόνο και κόπο. Με λίγη υπομονή και επίμονη βγάλαμε εις πέρας το πρόγραμμα.

Η εφαρμογή μας βγήκε αρκετά καλή για πρώτη έκδοση. Υπάρχουν βεβαίως αρκετές αλλαγές και καλύτερεύσεις που μπορούν να γίνουν στο μέλλον αλλά ήταν μια καλή εμπειρία για εμάς. Μερικά από τα σχέδια μας είναι:

- Περισσότερα επίπεδα
- Ανανέωση επιπέδων από το διαδίκτυο
- Καλύτερα γραφικά και animations στο παιχνίδι
- Ενσωμάτωση του Google Online Multiplayer με achievements και rewards

Σίγουρα θα μας μείνει αξέχαστη η εμπειρία αυτή και αποκομίσαμε αρκετά πράγματα που σίγουρα μερικά από αυτά θα μας χρησιμέψουν σύντομα στην ζωή μας.

Βιβλιογραφία

BIBΛΙΑ

1. J. Friesen, 2010, "Learn Java for Android Development", Apress
2. M. Murphy, 2011, "Beginning Android Development 2", Apress
3. M. Murphy, 2012, "Beginning Android Development 3", Apress
4. M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition", CommonsWare
5. M. Murphy, 2008, "The busy coders guide to Android Development, 1.1 version", CommonsWare
6. M. Murphy, 2008, "The busy coders guide to Android Development, 2 version", CommonsWare
7. M. Murphy, 2008, "The busy coders guide to Android Development, 3.6 version", CommonsWare
8. J. Morris, 2011, "Android User Interface Development Beginner's Guide", Pakt
9. C. Hasenan, 2008, "Android Essentials", Firstpress
10. J. Steele, 2010, "The Android Developer's Cookbook", Addison & Wesley
11. R. Meier, 2010, "Professional Android 2 Application Development", Wrox
12. Sun Microsystems, 1997, "Java Conventions"
13. J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley, 2012, "The Java Language Specification, 06-02-2012 version", Oracle America
14. I. Darwin, 2001, "Java Cookbook, First Edition", O Reilly

INTERNET

1. Τι είναι το Android
<http://el.wikipedia.org/wiki/Android>
[http://en.wikipedia.org/wiki/Android %28operating system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)
2. Android 1.6 Donut version
<http://developer.android.com/about/versions/android-1.6-highlights.html>
3. Android 2.2 Froyo version
<http://www.myphone.gr/forum/showthread.php?t=277505>
4. Android 2.3 Gingerbread version
<http://developer.android.com/about/versions/android-2.3-highlights.html>
5. Android 3.0 Honeycomb version
<http://developer.android.com/about/versions/android-3.0-highlights.html>
6. Android 4.1 Jellybean
<http://techblog.gr/software/android-4-1-jelly-bean-official-010101/>
7. Android 4.2 Jellybean
<http://www.pinnokio.gr/arthro/ta-xarakthristika-toy-neoy-android-42-jelly-bean>
<http://www.android.com/whatsnew/>
8. Android 4.4 KitKat
https://en.wikipedia.org/wiki/Android_KitKat
9. Android 5.0 Lollipop
https://en.wikipedia.org/wiki/Android_Lollipop
10. Android Mobile Market Share
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
11. Android version history
http://en.wikipedia.org/wiki/Android_version_history
12. Ενεργοποιημένες Android συσκευές
<http://thenextweb.com/google/2014/06/25/google-activations-io-2014/>

- <http://www.engadget.com/2013/04/16/eric-schmidt-google-now-at-1-5-million-android-activations-per/>
13. Android Debugging
<http://developer.android.com/tools/debugging/index.html>
 14. Η ιστορία του Android
<http://www.allaboutandroid.gr/?p=6362>
 15. Ανάλυση της αρχιτεκτονικής του Android
<http://developer.android.com/guide/basics/what-is-android.html>
 16. Πληροφορίες σχετικά με τη δομή και τη χρήση του αρχείου AndroidManifest.xml
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
 17. Οι φάσεις του σχεδιασμού τις εφαρμογής περιλαμβάνονται στην παρακάτω πηγή
<http://developer.android.com/guide/developing/index.html>
 18. Οδηγίες του Android Studio
<http://developer.android.com/tools/studio/index.html>
 19. Gradle Plugin
<https://developer.android.com/tools/building/plugin-for-gradle.html>
 20. Οδηγίες εγκατάστασης του Android SDK
<http://developer.android.com/sdk/installing.html>
 21. 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>
 22. 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>
 23. Ο καταμερισμός των διαστάσεων οθόνης προς την πυκνότητα pixel των συσκευών που χρησιμοποιούν το Play Store
<http://developer.android.com/resources/dashboard/screens.html>
 24. Οδηγίες υποστήριξης πολλαπλών αναλύσεων στις εφαρμογές

- http://developer.android.com/guide/practices/screens_support.html
25. Ανάλυση και οδηγίες χρήσης του Android Debug Bridge
<http://developer.android.com/guide/developing/tools/adb.html>
26. Σχετικά με την διαχείριση εικονικών συσκευών
<http://developer.android.com/guide/developing/devices/index.html>
27. Πληροφορίες σχετικά με το εργαλείο LogCat
<http://developer.android.com/guide/developing/tools/logcat.html>
28. Χρησιμοποίηση του DDMS
<http://developer.android.com/tools/debugging/ddms.html>
29. Τι είναι η Java
<http://el.wikipedia.org/wiki/Java>
http://www.islab.demokritos.gr/gr/html/ptixiakos/kostas-aris_ptyxiakh/Phtml/java.htm
30. Τι είναι ο πυρήνας
http://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%AE%CE%BD%CE%B1%CF%82_%28%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AD%CF%82%29
31. Τι είναι ο πυρήνας Linux
http://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%AE%CE%BD%CE%B1%CF%82_Linux
32. ListView & Context menu (για τις λίστες των βαθμολογιών και των παιχτών)
<http://stackoverflow.com/questions/3771568/showing-empty-view-when-listview-is-empty>
<http://stackoverflow.com/questions/18632331/using-contextmenu-with-listview-in-android>
<http://www.vogella.com/tutorials/AndroidRecyclerView/article.html>
http://www.vogella.com/tutorials/AndroidListView/article.html#adapterperformance_holder
33. GridView (για τα επίπεδα)
<http://www.learn2crack.com/2014/01/android-custom-gridview.html>
34. Tabs (για τα στάδια)

- <http://www.androidhive.info/2013/10/android-tab-layout-with-swipeable-views-1/>
- <http://developer.android.com/training/implementing-navigation/lateral.html>
35. Countdown Timer (για το χρονόμετρο)
- <http://www.theappguruz.com/blog/android-count-timer>
- <http://developer.android.com/reference/android/os/CountDownTimer.html>
- <http://www.sanfoundry.com/java-android-program-demonstrate-count-down-timer/>
36. Media Playback (για την αναπαραγωγή των ήχων)
- <http://developer.android.com/guide/topics/media/mediaplayer.html>
37. SQLite
- <http://www.vogella.com/tutorials/AndroidSQLite/article.html>
- <http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>
- <http://blog.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>
38. Genymotion emulator
- <https://www.genymotion.com/>
39. Android SQLiteAssetHelper (για την αντιγραφή της βάσης στο κινητό)
- <https://github.com/jgilfelt/android-sqlite-asset-helper>
40. Compatibility Action Bar source code
- <http://developer.android.com/tools/samples/index.html>
41. Android Support library
- <http://developer.android.com/tools/extras/support-library.html#Downloading>
42. Ήχοι που χρησιμοποιήθηκαν (CC0 1.0 Universal License)
- <https://www.freesound.org/people/jbeetle/sounds/274510/>
- <https://www.freesound.org/people/AlaskaRobotics/sounds/221567/>
- <https://www.freesound.org/people/TheDweebMan/sounds/278164/>
- <https://www.freesound.org/people/notchfilter/sounds/43696/>

43. Εικόνες που χρησιμοποιήθηκαν (CC0 1.0 Universal License)

<https://openclipart.org>

44. Οι ερωτήσεις και οι απαντήσεις είναι παρμένες από το Υπουργείου Παιδείας και Πολιτισμού της Κύπρου

<http://www.schools.ac.cy/klimakio/Themata/mathimatika>