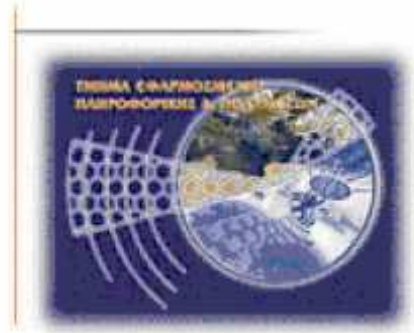




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων**



Πτυχιακή Εργασία

**Τίτλος: ΑΝΑΛΥΣΗ ΚΑΙ ΧΡΗΣΗ ΕΡΓΑΛΕΙΩΝ ΓΙΑ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΕΦΑΡΜΟΓΩΝ ΣΕ ANDROID**

Δασκαλάκης Ελευθέριος (ΑΜ:1672)

Επιβλέπων Καθηγητής: Τριανταφυλλίδης Γιώργος

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω όλους όσους με υποστήριξαν. Επιπλέον οφείλω ένα μεγάλο ευχαριστώ στον επόπτη καθηγητή κύριο Τριανταφυλλίδη Γεώργιο για τις πολύτιμες συμβουλές και την καθοδήγησή του κατά την εκπόνηση της συγκεκριμένης εργασίας, σε όλα τα στάδια της προσπάθειας αυτής.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο τη μελέτη, κατανόηση και ανάπτυξη ενός λειτουργικού προγράμματος για έξυπνα τηλέφωνα ή smartphones, όπως είναι ευρέως γνωστά στην αγορά. Το λειτουργικό πρόγραμμα που έχει μελετηθεί και αναπτυχθεί στο παρόν πόνημα είναι το ονομαζόμενο Android. Για τις ανάγκες της παρουσίασης, δημιουργήθηκε μια εφαρμογή που τρέχει σε smartphone με λειτουργικό Android.

Αρχικά, γίνεται επεξήγηση των λόγων επιλογής του συγκεκριμένου λειτουργικού συστήματος και κατόπιν ανάλυση των ενδότερων επιμέρους συστατικών στοιχείων του. Στην συνέχεια, ακολουθεί μια ιστορική αναφορά σχετικά με τις εκδόσεις και τα χαρακτηριστικά του Android, καθώς παρουσιάζονται και μελετούνται το κάθε ένα ξεχωριστά. Επίσης, αναφέρονται οι συσκευές (κινητά τηλέφωνα και ταμπλέτες – Tablets) που χρησιμοποιούν το Android και γίνεται ανάλυση των λειτουργιών τους. Εν συνεχεία, γίνεται μια εκτενής παρουσίαση των εργαλείων ανάπτυξης λογισμικού και εφαρμογών στην πλατφόρμα Android, καθώς παρατίθενται και τα ανάλογα παραδείγματα.

Τέλος, παρουσιάζεται βήμα προς βήμα η ανάπτυξη μιας Android Application με την χρήση του λειτουργικού SDK στο εργαλείο Eclipse.

Abstract

The present thesis aims to study, understand and develop software for smartphones, which are widespread in the market. The software that has been studied and developed in the thesis is the so-called Android. For the purposes of the related presentation, an application for Android smartphones has been developed.

Initially, the reasons of choosing this specific software are given, followed by an analysis of its inner counterparts. Afterwards, there is a historical report of the Android versions and characteristics, each one of them being separately presented and examined. Furthermore, Android devices (mobile phones and tablets) are referred to and their functions are analyzed. Thereafter, an extensive presentation of the software development tools and applications over Android platform is been done. Then, under each tool analysis, the necessary examples are following.

Finally, the development of an Android application, using the SDK software and the Eclipse tool, is presented step by step.

ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες.....	i
Περίληψη.....	ii
Abstract.....	iii
1. Εισαγωγή.....	1
1.1 Επιλογή του Android.....	1
1.2 Τι είναι το Android.....	3
1.3 Αρχιτεκτονική του Android.....	5
1.4 Η εικονική μηχανή Dalvik.....	6
1.5 Πυρήνας (Linux kernel).....	7
1.6 Εγγενείς Βιβλιοθήκες (Native Libraries).....	7
1.6.1 Βιβλιοθήκες Android.....	7
1.6.2 Προηγμένες Βιβλιοθήκες Android.....	8
1.7 Χρόνος Εκτέλεσης (Android Runtime).....	9
1.8 Πλαίσιο Εφαρμογής (Application Framework).....	9
1.9 Βασικά Συστατικά Εφαρμογών του Android	11
2. Ιστορικά (Εκδόσεις και Χαρακτηριστικά).....	14
2.1 Android 1.5 CUPCAKE.....	14
2.2 Android 1.6 DONUT.....	15
2.3 Android 2.0 ECLAIR.....	16
2.4 Android 2.2 FROYO.....	17
2.5 Android 2.3 GINGERBREAD.....	18
2.6 Android 3.0 HONEYCOMB.....	18
2.7 Android Cream Sandwich.....	19
2.8 Εφαρμογές και Widgets.....	21

2.9 Γιατί Android.....	21
3. Κινητά Τηλέφωνα Android.....	22
3.1 Η Πρώτη Συσκευή με Android.....	22
3.2 Εναλλακτικά Πληκτρολόγια.....	22
3.3 Αυτοματοποιήσεις.....	23
3.4 Προσαρμοσμένο Γραφικό Περιβάλλον.....	23
3.5 Widgets.....	23
3.6 Αναιρούμενα Μέσα Αποθήκευσης και Μπαταρία.....	24
3.7 Ασύρματη Εγκατάσταση Εφαρμογών.....	25
3.8 Προσαρμοσμένες ROM.....	25
3.9 Έλεγχος του Τηλεφώνου από τον Υπολογιστή σου.....	25
3.10 Flash Player.....	25
3.11 Πραγματική Ενσωμάτωση Εφαρμογών.....	26
4. Ταμπλέτες Android.....	27
4.1 Tablet PC: Με Δυο Λόγια.....	27
4.2 Ιστορική Αναδρομή.....	15
4.2.1 Πώς εξελίσσεται έως σήμερα.....	28
4.3 Η Adobe φέρνει τις Touch Apps στις ταμπλέτες Android.....	29
4.4 Η Εξέλιξη των Tablets.....	29
4.4.1 Χρήση 4-πύρινων επεξεργαστών.....	30
4.4.2 Υψηλότερη ανάλυση.....	30
4.5 Μια απλή εφαρμογή στις ταμπλέτες Android.....	31
5 Εργαλεία ανάπτυξης λογισμικού.....	32
5.1 Android SDK.....	32
5.1.1 Μερικά από τα χαρακτηριστικά του SDK.....	33
5.1.2 Ο εξομοιωτής (emulator).....	33
5.1.3 Άλλα εργαλεία του Android SDK	33

5.1.4	Μια γρήγορη εφαρμογή για το SDK.....	34
5.2	Android NDK (Native Development Kit).....	40
5.2.1	Τα κύρια χαρακτηριστικά του NDK.....	40
5.2.2	Τι παρέχει το Android NDK.....	41
5.2.3	Πότε γίνεται ανάπτυξη σε ατόφιο κώδικα.....	41
5.2.4	Περιεχόμενα του NDK.....	42
5.2.5	Εργαλεία Ανάπτυξης.....	42
5.2.6	Τεκμηρίωση.....	46
5.2.7	Ανάπτυξη εφαρμογής για το Android NDK.....	44
5.3	App Inventor for Android.....	52
5.3.1	Η ανάπτυξη της εφαρμογής.....	52
5.3.2	Τα αρχικά πλεονεκτήματα του App Inventor.....	53
5.3.3	Τι είναι το App Inventor.....	53
5.3.4	Μια γρήγορη εφαρμογή για το App Inventor.....	56
5.4	Hyper Next Android Creator.....	63
5.4.1	HyperNext.....	65
5.4.2	Περιβάλλον προγραμματισμού.....	65
5.4.3	Εφαρμογή για το HyperNext Android.....	66
6	Android Application.....	71
6.1	Μια Android application με την χρήση των SDK και Eclipse.....	71
6.2	Προετοιμασία.....	71
6.2.1	Εγκατάσταση του Eclipse IDE.....	71
6.2.2	Εγκατάσταση του ADT plugin για το Eclipse.....	72
6.2.3	Εγκατάσταση του Android SDK.....	73
6.3	Δημιουργία Android-εφαρμογής.....	74
6.4	Συμπέρασμα.....	81
BIBΛΙΟΓΡΑΦΙΑ.....	82	

Πίνακας Εικόνων

Εικόνα 1 : Λογότυπο του Android

Εικόνα 2 : Η μάχη μεταξύ Android και Iphone.

Εικόνα 3 : Πωλήσεις Λειτουργικών Συστημάτων για τα Smartphones.

Εικόνα 4 : Εταιρίες λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας.

Εικόνα 5 : Λογότυπο πλατφόρμας Android.

Εικόνα 6 : Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android.

Εικόνα 7 : Ο κύκλος ζωής μιας Δραστηριότητας Android

Εικόνα 8 : Λογότυπο Android 1.5 CUPCAKE

Εικόνα 9 : Λογότυπο Android 1.6 DONUT

Εικόνα 10 : Λογότυπο Android 2.0 ECLAIR

Εικόνα 11 : Λογότυπο Android 2.2 FROYO

Εικόνα 12 : Λογότυπο Android 2.3 GINGERBREAD

Εικόνα 13 : Λογότυπο Android 3.0 HONEYCOMB

Εικόνα 14 : Λογότυπο Android Ice Cream Sandwich

Εικόνα 15 : Ποσοστά των εγκατεστημένων εκδόσεων στις Android συσκευές

Εικόνα 16 : Τυπικές Συσκευές Android

Εικόνα 17 : Όπως παρουσιάζεται η αρχική οθόνη μιας εφαρμογής Widgets

Εικόνα 18 : Φωτογραφία μιας καθημερινής κάρτας μνήμης

Εικόνα 19 : Όπως ακριβώς εμφανίζεται το uninstall στο Android.

Εικόνα 20 : Ταμπλέτες Android

Εικόνα 21 : Αριθμός πωλήσεων PC και Tablet

Εικόνα 22 : Το Touch Apps της Android

Εικόνα 23 : Χειρόγραφες σημειώσεις για ταμπλέτες Android

Εικόνα 24 : Μια μικρή ματιά στο γραφικό περιβάλλον του Antipaper

Εικόνα 25 : Η εγκατάσταση του Android SDK

Εικόνα 26 : Ο αρχικός βασικός κώδικας που περιέχει το πρόγραμμα

Εικόνα 27 : Τροποποίηση του κώδικα για την εφαρμογή

Εικόνα 28 : Επιλέγοντας RUN, γίνεται η εκτέλεση του κώδικα

Εικόνα 29 : Η εφαρμογή φορτώνει κανονικά

Εικόνα 30 : Όπως έχει αναφερθεί, το γραφικό περιβάλλον είναι ίδιο με αυτό μιας Android συσκευής

Εικόνα 31 : Ξεκλείδωμα της συσκευής για να τρέξει η εφαρμογή. Ο προσομοιωτής Android στην αρχική οθόνη

Εικόνα 32 : Το Text κείμενο (που προαναφέρθηκε) “Efarmogi gia Android SDK”

Εικόνα 33 : Ο χειρισμός της εφαρμογής καταδεικνύει την ομοιότητα του προσομοιωτή με ένα πραγματικό κινητό τηλέφωνο Android

Εικόνα 34 : Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή

Εικόνα 35 : Λογότυπο εφαρμογής Android NDK

Εικόνα 36 : Εφαρμογή για το Android NDK

Εικόνα 37 : Στο Help, εγκαθιστούμε το New Software στο μενού. Επιλέγουμε Galileo ως το χώρο ενημέρωσης ("Work with"). Αφήνουμε τα στοιχεία να τρέξουν στο Eclipse C / C ++

Εικόνα 38 : Επιλέγουμε 'NAI' σε όλα και περιμένουμε το Eclipse για να κάνει επανεκκίνηση. Έχουμε στο C / C ++ την υποστήριξη στο Eclipse IDE τώρα

Εικόνα 39 : Cygwin: ένα σύνολο λογισμικού που προσομοιώνει Unix περιβάλλον των Windows

Εικόνα 40 : Με την επιλογή του mirror, κάνουμε κλικ στο Next και στην συνέχεια το Cygwin θα κατεβάσει και θα παρουσιάσει τη λίστα των διαθέσιμων πακέτων

Εικόνα 41 : Ολοκλήρωση εγκατάστασης

Εικόνα 42 : Για τον έλεγχο των εργαλείων που είναι σημαντικά για το Android NDK πληκτρολογούμε make-n στην κονσόλα

Εικόνα 43 : Εγκατάσταση του Android NDK

Εικόνα 44 : Το γραφικό περιβάλλον του Eclipse

Εικόνα 45 : Δημιουργία νέου project

Εικόνα 46 : Δημιουργία του φακέλου JNI στη ρίζα του νέου project

Εικόνα 47 : Ο βασικός κώδικας που χρησιμοποιήθηκε για την εφαρμογή

Εικόνα 48 : Screenshot τρόπου ελέγχου της εφαρμογής

Εικόνα 49 : Πρέπει να αλλάξετε τον κώδικα της Java της τάξης NdkFooActivity για να χρησιμοποιήσετε τον κώδικα NDK

Εικόνα 50 : Σε αυτό το σημείο μπορούμε να διακρίνουμε τον εξομοιωτή (emulator) του eclipse με την εφαρμογή μας. Στην περίπτωση μας το (Hello from native code)

Εικόνα 51 : Λογότυπο εφαρμογής App Inventor

Εικόνα 52 : Γραφικό περιβάλλον App Inventor

Εικόνα 53 : Κύρια οθόνη του App Inventor

Εικόνα 54 : Έχουμε επιλέξει το learn σε δεύτερο παράθυρο

Εικόνα 55 : Εγκατάσταση λογισμικού που χρειαζόμαστε. Στην περίπτωση μας instructions for Windows

Εικόνα 56 : Εγκατάσταση και εκτέλεση εφαρμογής

Εικόνα 57 : Προσθέτουμε το πρώτο Button

Εικόνα 58 : Σε αυτό το σημείο έχουμε προσθέσει στο components την φωτογραφία και τον ήχο που θα χρησιμοποιήσουμε

Εικόνα 59 : Εδώ διακρίνουμε καθαρά τον ήχο που έχουμε προσθέσει

Εικόνα 60 : Τρέχουμε την εφαρμογή μας και ταυτόχρονα εγκαθιστούμε τον κώδικα που χρειαζόμαστε

Εικόνα 61 : Εδώ μπορούμε να διακρίνουμε το block editor του App Inventor

Εικόνα 62 : Έχουμε ανοίξει το My block, προσθέτουμε το button και τον ήχο μέσα σε αυτό

Εικόνα 63 : Επιλέγουμε το Connect Device και στην συνέχεια τρέχουμε το new emulator

Εικόνα 64 : Με τον εξομοιωτή (emulator), όπως μπορούμε να διακρίνουμε, εκτελούμε την εφαρμογή μας

Εικόνα 65 : Ο κώδικας μας στην προκειμένη περίπτωση

Εικόνα 66 : Λογότυπο εφαρμογής HyperNext Android

Εικόνα 67 : Μόλις έχουμε ανοίξει την ηλεκτρονική σελίδα Hypernext Android και κάνουμε την εγγραφή μας

Εικόνα 68 : Εικόνα κατά την έναρξη του HyperNext Android Creator

Εικόνα 69 : Σε αυτό το σημείο μπορούμε να διακρίνουμε το γραφικό περιβάλλον του HAC

Εικόνα 70 : Δημιουργούμε ένα Demo Project με το όνομα Hello Word

Εικόνα 71 : Έχουμε προσθέσει ένα **Button** με την επιλογή **New Button** και επιλέγουμε την ιδιότητα **Script** έτσι ώστε να γράψουμε αυτό που θα εμφανίζεται πατώντας το **Button**.

Εικόνα 72 : Έχουμε πατήσει το **Run** και στην συνέχεια πάνω στο **Button** της εφαρμογής μας για να εμφανίσει αυτό που γράψαμε

Εικόνα 73 : Στο **C Mode** επιλέγουμε **Design**. Στην συνέχεια, επιλέγουμε το **Manager Devices** από την γραμμή εργαλείων **Android**

Εικόνα 74 : Από την γραμμή εργαλείων επιλέγω το **Go** και στην συνέχεια το **Run** για να τρέξει ο **emulator**

Εικόνα 75 : Έχει ανοίξει ο **emulator** του **Android**. Πατώντας πάνω στο **Button** που έχουμε δημιουργήσει και εμφανίζεται η εφαρμογή μας. Στην περίπτωση μας, ένας κειμενογράφος που γραφεί “**Hello world**”.

Εικόνα 76 : **Java Platform**

Εικόνα 77 : Εγκατάσταση του **Eclipse**

Εικόνα 78 : Εγκατάσταση του **Eclipse**

Εικόνα 79 : Εγκατάσταση του **SDK**

Εικόνα 80: **Android Virtual Device (AVD)**

Εικόνα 81 : **Android Virtual Device (AVD)**

Εικόνα 82 : **Android Virtual Device (AVD)**

Εικόνα 83 : Κώδικας στο **main.xml** για να δηλώσουμε τα 3 στοιχεία μας

Εικόνα 84 : Κώδικας στο **AndroidManifest.xml**

Εικόνα 85 : Συνολικός κώδικας

Εικόνα 1: Ολοκλήρωση εφαρμογής

Κίνητρο για την Διεξαγωγή της Εργασίας

1.Εισαγωγή

Στην παρούσα πτυχιακή εργασία, περιγράφονται και μελετούνται τα εργαλεία για Android Development. Μελετάται επίσης, η αρχιτεκτονική του συστήματος και αναλύονται τα σημαντικότερα τμήματά της. Κύριος σκοπός της εργασίας αυτής είναι να υλοποιηθεί μια εφαρμογή, η οποία θα διέπεται από τους κανόνες του προαναφερθέντος λειτουργικού συστήματος, με γενικότερο στόχο την ομαλή λειτουργία του σε συσκευές που υποστηρίζουν το λειτουργικό Android. Ο όρος Android έχει ελληνική προέλευση, καθώς προέρχεται από την λέξη Andro-«ανθρώπινη» + eides-«μορφή, σχήμα». Συνεπώς, η έννοια που δίνεται στην λέξη Android είναι το «Ανδροειδές» και συμβολίζεται από το ρομπότ με την μορφή ανθρώπου, όπως εμφανίζεται και στην [Εικόνα 1].



Εικόνα 2: Λογότυπο του Android

1.1 Επιλογή του Android

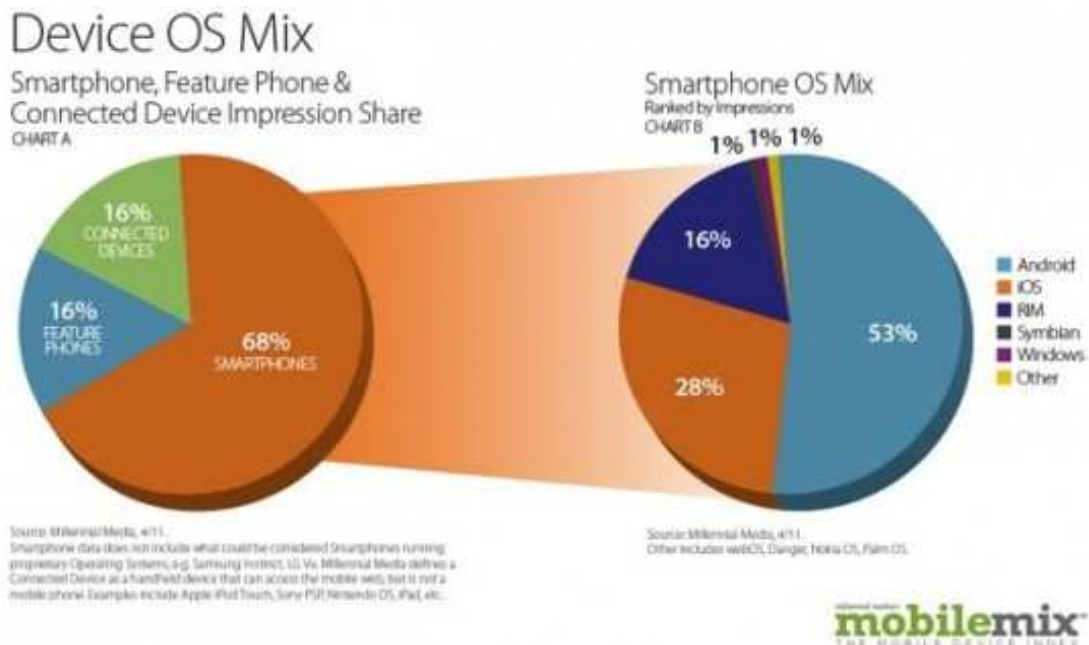
Ο πρωταρχικός λόγος της επιλογής για αυτήν την εργασία, ήταν η επιλογή μιας πλατφόρμας, η οποία θα μπορούσε να υποστηρίξει τον σχεδιασμό και την υλοποίηση μιας εφαρμογής. Παράλληλα η αναζήτηση για κάτι το καινούργιο στον χώρο που θα άξιζε την μελέτη του και θα ήταν πολλά υποσχόμενο, οδήγησε σε δύο υποψήφιες πλατφόρμες, το Android και το iPhone. Οι δύο αυτές πλατφόρμες φαίνεται ότι θα πρωταγωνιστήσουν στον χώρο των έξυπνων τηλεφώνων (smart phones), αφού διαθέτουν εξαιρετικά χαρακτηριστικά και απίστευτες δυνατότητες [Εικόνα 2].



Εικόνα 3: Η μάχη μεταξύ Android και iPhone

Στις αρχές Οκτωβρίου 2009, το iPhone κατείχε τα σκήπτρα στις πωλήσεις. Παρόλα αυτά, με τη ραγδαία ανάπτυξη που είχε το Android, οι ερευνητές υποστήριζαν ότι τα δεδομένα θα έχουν ανατραπεί μέχρι το 2012 το αργότερο. Προς έκπληξη αρκετών, κατά την έναρξη συγγραφής της παρούσας πτυχιακής εργασίας (Νοέμβριος 2011), το Android έχει ήδη ξεπεράσει προ πολλού σε πωλήσεις το iPhone και συνεχίζει να επεκτείνεται. Ωστόσο, είναι μια μάχη η οποία αναμένεται να συνεχιστεί, με τις δύο πλατφόρμες να εκσυγχρονίζουν ανά τακτά χρονικά διαστήματα τα χαρακτηριστικά τους.

Χαρακτηριστική είναι η έρευνα που ανακοίνωσε η Millennial Media[1]. Τα αποτελέσματα είναι αναμενόμενα, δηλαδή το Android OS κυριαρχεί με παρουσία στο 53% των smartphones παγκοσμίως, με το iOS να ακολουθεί με 28% και το BlackBerry OS με 16% [Εικόνα 3].



Εικόνα 4: Πωλήσεις Λειτουργικών Συστημάτων για τα Smartphones

Επίσης, ο Hugo Barra (Product Management Director του Android) ανακοίνωσε στο πρόσφατο Google I/O που έγινε στις 10 Μαΐου 2011 στο San Francisco, ορισμένα εντυπωσιακά στατιστικά για την πλατφόρμα. Η Google δεν έχασε την ευκαιρία να ρίξει τα «καρφιά» της σε οποιοδήποτε αμφισβητεί την κυριαρχία του Android OS και ανακοίνωσε τα ιλιγγιώδη στατιστικά στοιχεία που συνοδεύουν το υπέρ-επιτυχημένο λειτουργικό σύστημα, τα οποία παρατίθενται παρακάτω:

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

Οι δυνατότητες και οι προοπτικές του Android ήταν το έναυσμα, ώστε να γίνει ξεκάθαρη η επιλογή της πλατφόρμας ανάπτυξης της εφαρμογής, έναντι του λειτουργικού iOS. Μεγάλο πλεονέκτημα του λειτουργικού Android είναι ότι παρέχει τη δυνατότητα στον καθένα να γράψει τη δική του εφαρμογή, να μάθει, να πειραματιστεί με το κινητό του και γιατί όχι, να παίξει! Είναι ένα πρόγραμμα ανοικτού κώδικα.

1.2 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Οι συνιδρυτές της Android πήγαν να εργαστούν στην Google συμπεριλαμβανομένων των Andy Rubin (συν-ιδρυτής της Danger), Rich Miner (συν-ιδρυτής της Wildfire Communications, Inc), Nick Sears (πρώην αντιπρόεδρος της T-Mobile), και Chris White (επικεφαλής σχεδιασμού και ανάπτυξης interface στο WebTV). Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Στην Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google έχει ήδη συγκεντρώσει μια σειρά από εταίρους hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Περισσότερες εικασίες ότι η Google θα εισέλθει στην αγορά κινητής τηλεφωνίας άρχισαν τον Δεκέμβριο του 2006. Δημοσιεύσεις από το BBC και τη The Wall Street Journal πληροφορούσαν ότι η Google ήθελε την έρευνα και τις εφαρμογές σε κινητά τηλέφωνα και εργάζονται σκληρά για να τις προωθήσουν στην αγορά. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυξε μια Google-branded συσκευή. Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητών τηλεφώνων και τους φορείς δικτύων.

Τον Σεπτέμβριο του 2007, η InformationWeek κάλυψε μια μελέτη αξιολόγησης αναφέροντας ότι η Google έχει καταθέσει αρκετές πατέντες στον τομέα της κινητής τηλεφωνίας. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC Corporation της Ταϊβάν, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας.[2][3]. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. [4]

Ενδεικτικά, αναφέρονται μερικά μέλη του οργανισμού αυτού [Εικόνα 4], για να γίνει ορατή η τεράστια προοπτική που δημιουργείται:



Εικόνα 5: Εταιρίες ανάπτυξης λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας

Μερικές από τις εταιρίες αυτές αναγράφονται παρακάτω:

- Intel
- Vodafone
- Motorola
- Samsung
- Sony Ericsson
- ebay
- Google
- Verizon
- Texas Instruments
- Htc

Κατά τα λεγόμενα τους, όπως αναφέρεται και στο επίσημο site, η OHA (Open Handset Alliance) αντιπροσωπεύει: «Μια δέσμευση για την ειλικρίνεια, ένα κοινό όραμα για το μέλλον και συγκεκριμένα σχέδια για να κάνει το όραμα μια πραγματικότητα. Για να επιταχύνει την καινοτομία στο κινητό και να προσφέρει στους καταναλωτές μια πλουσιότερη, λιγότερο ακριβή και καλύτερη εμπειρία κινητών τηλεφώνων.»

http://www.openhandsetalliance.com/oha_faq.html

Η OHA ελπίζει να παραδώσει μια καλύτερη εμπειρία λογισμικού κινητών στους καταναλωτές, παρέχοντας μια πλατφόρμα για την ανάπτυξη καινοτόμων κινητών εφαρμογών γρηγορότερα και με υψηλότερη ποιότητα, χωρίς τέλη αδείας για τους προγραμματιστές λογισμικού ή τους κατασκευαστές κινητών τηλεφώνων. Η επιτυχία του Android ως πλατφόρμα κινητών τηλεφώνων θα εξαρτηθεί κατά ένα μεγάλο μέρος από την επιτυχία των συνεργατών της OHA στην κυκλοφορία επιθυμητών κινητών τηλεφώνων και κινητών υπηρεσιών που θα ενθαρρύνουν την υιοθέτηση των Android τηλεφώνων. Οι προγραμματιστές έχουν την ευκαιρία να δημιουργήσουν καινοτόμες, νέες

εφαρμογές κινητών για Android ώστε να ενθαρρυνθούν περισσότερες επιχειρήσεις κινητής τεχνολογίας να γίνουν μέλη της ΟΗΑ.

Στην [Εικόνα 5] φαίνεται το λογότυπο της Android.



Εικόνα 6: Λογότυπο πλατφόρμας Android

1.3 Αρχιτεκτονική του Android

Όπως αναφέραμε πιο πριν, το Android είναι μια στοίβα λογισμικού. Η λογική πίσω από αυτήν την έκφραση και σε όλη την φιλοσοφία του Android, κρύβεται στο ακόλουθο διάγραμμα με τα βασικά συστατικά του [Εικόνα 6].

Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα, που επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές με πρόσβαση που ελέγχεται από τους περιορισμούς ασφάλειας που έχουν τεθεί. Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση υπάρχοντων εφαρμογών ή δημιουργίας καινούργιων, χρησιμοποιώντας κώδικα από αυτές.

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

- Διαχειριστής Δραστηριοτήτων (Activity Manager) - Ελέγχει τον κύκλο ζωής των δραστηριοτήτων, συμπεριλαμβανομένης της διαχείρισης του σωρού δραστηριοτήτων.
- Όψεις (Views) - Χρησιμοποιούνται για να κατασκευάσουν διαφανές χρήστη (user interface) για τις δραστηριότητες (activities).
- Διαχειριστή Ειδοποιήσεων (Notification Manager) - Παρέχει ένα συνεπή και αποτελεσματικό μηχανισμό για να προειδοποιεί τον χρήστη

- Παροχέας Περιεχομένου (Content Providers) - Επιτρέπουν στις εφαρμογές να μοιραστούν δεδομένα μεταξύ τους.
- Διαχειριστής Πόρων (Resource Manager) - Υποστηρίζει πόρους εκτός κώδικα όπως συμβολοσειρές και γραφικά.



Εικόνα 7: Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android

Στην στοίβα του Android [Εικόνα 6], παρατηρούμε 4 επίπεδα. Ακολούθως θα περιγράψουμε συνοπτικά τα βασικά αυτά επίπεδα χωρίς να μπούμε σε λεπτομέρειες για όλα τα περιεχόμενα του κάθε επιπέδου. Αν ο αναγνώστης επιθυμεί να μάθει περισσότερα, μπορεί να επισκεφθεί την επίσημη ιστοσελίδα του Android για κατασκευαστές (<http://developer.android.com>). Κάθε επίπεδο στην αρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο πάνω επίπεδα. Ας δούμε τώρα αυτά τα επίπεδα ξεκινώντας από το πιο χαμηλό.

1.4 Η εικονική μηχανή Dalvik

Ένα από τα στοιχεία κλειδιά του Android είναι η εικονική μηχανή Dalvik. Το Android χρησιμοποιεί την δικιά του εικονική μηχανή και όχι μια παραδοσιακή, με σκοπό να εξασφαλίσει ότι πολλαπλά στιγμιότυπα τρέχουν αποτελεσματικά σε μια ενιαία συσκευή.

Η Dalvik VM (Virtual Machine) χρησιμοποιεί τον πυρήνα Linux της συσκευής για να χειριστεί τις χαμηλού επιπέδου λειτουργίες που περιλαμβάνουν την ασφάλεια, τον πολυνηματισμό και τη διαχείριση διαδικασιών και μνήμης. Είναι επίσης δυνατό να γραφτούν εφαρμογές C/C++ που τρέχουν άμεσα στο εσωτερικό του λειτουργικού Linux. Αν και μπορεί να γίνει αυτό, στις περισσότερες περιπτώσεις δεν υπάρχει κανένας λόγος.

Μέσω της Dalvik VM επιτυγχάνεται η ρύθμιση της πρόσβασης στο υλικό και στις υπηρεσίες του συστήματος. Με τη χρησιμοποίηση αυτής της εικονικής μηχανής στην εκτέλεση εφαρμογής, η οποία προσφέρει ένα αφαιρετικό στρώμα, οι κατασκευαστές δεν χρειάζεται να ανησυχήσουν για κάποια υλοποίηση υλικού (hardware implementation).

Η Dalvik VM εκτελεί τα Dalvik εκτελέσιμα αρχεία, ένα format βελτιστοποιημένο έτσι ώστε να καταλαμβάνει την ελάχιστη μνήμη. Τα .dex εκτελέσιμα αρχεία δημιουργούνται μετασχηματίζοντας κλάσεις που έχουν μεταγλωττιστεί από Java χρησιμοποιώντας εργαλεία που παρέχονται μέσα στο SDK.

Μια απλή Java VM είναι μια εικονική μηχανή βασισμένη σε στοίβα (stack-based). Η Dalvik VM από την άλλη είναι μια εικονική μηχανή βασισμένη σε μητρώα (registerbased). Με τον τρόπο αυτό αυξάνεται η αποδοτικότητα του επεξεργαστή του κινητού. Επίσης, οι εικονικές μηχανές που είναι βασισμένες σε καταχωρητές (registers) επιτρέπουν ταχύτερους χρόνους εκτέλεσης των μεγάλων προγραμμάτων.

1.5 Πυρήνας Linux (Linux kernel)

Το Android είναι βασισμένο στα γερά θεμέλια του Linux. Ο πυρήνας Linux είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να βρεθεί παντού, από ρολόγια χειρός μέχρι υπερυπολογιστές. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντας του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφορμών στο μέλλον. Ειδικότερα, το Android χρησιμοποιεί τον πυρήνα Linux για την διαχείριση μνήμης, την διαχείριση διεργασιών, την δικτύωση και άλλες υπηρεσίες του λειτουργικού συστήματος.

1.6 Εγγενείς Βιβλιοθήκες (Native Libraries)

Στο αμέσως υψηλότερο επίπεδο βρίσκουμε τις Native Libraries – Εγγενείς Βιβλιοθήκες. Όλες αυτές είναι γραμμένες στην γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το τηλέφωνο. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να στηθούν από μόνες τους. Υπάρχουν για να μπορούν να κληθούν από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και μετά, οι κατασκευαστές μπορούν να γράφουν τις δικές τους τέτοιες βιβλιοθήκες με την χρήση της Εργαλειοθήκης NDK (Native Development Kit).

1.6.1 Βιβλιοθήκες Android

Το Android προσφέρει διάφορα APIs για την ανάπτυξη εφαρμογών. Ο ακόλουθος κατάλογος APIs πυρήνα δίνει μια ιδέα για το τι είναι διαθέσιμο. Όλες οι συσκευές Android υποστηρίζουν το λιγότερο αυτά τα APIs :

- android.util - Το πακέτο βοηθήματος (utility) πυρήνων περιέχει χαμηλού επιπέδου κλάσεις, όπως τα εξειδικευμένα containers, μορφοποιητές (formatters) συμβολοσειρών, και βοηθήματα για parsing XML αρχείων.
- android.os - Το πακέτο λειτουργικού συστήματος παρέχει την πρόσβαση στις βασικές υπηρεσίες του λειτουργικού συστήματος όπως τη διαβίβαση μηνυμάτων, την επικοινωνία μεταξύ των επικοινωνιών, τις λειτουργίες ρολογιών, και την αποσφαλμάτωση (debugging).

- android.graphics - Το API γραφικών παρέχει τις χαμηλού επιπέδου κλάσεις γραφικών που υποστηρίζουν τους καμβάδες (canvases), τα χρώματα, και τα βασικά στοιχεία ζωγραφικής, και παρέχει τη δυνατότητα να ζωγραφίσει κανείς πάνω σε καμβά (canvas).
 - android.text - Τα εργαλεία επεξεργασίας κειμένων για αναπαράσταση και ανάλυση κειμένου.
 - android.database - Παρέχει τις χαμηλού επιπέδου κλάσεις που απαιτούνται για το χειρισμό των δρομέων (cursors) κατά τη λειτουργία τους με τις βάσεις δεδομένων.
 - android.content - Το API περιεχομένου (content) χρησιμοποιείται για να διαχειριστεί την πρόσβαση στα δεδομένα και την έκδοσή τους, παρέχοντας υπηρεσίες για την διαχείριση των πόρων, των Παρόχων Περιεχομένου (Content Provider), και των πακέτων.
 - android.view – Οι Όψεις (Views) είναι οι κλάσεις πυρήνα διεπαφών χρήστη. Όλες οι διεπαφές χρήστη κατασκευάζονται χρησιμοποιώντας μια σειρά από Όψεις που παρέχουν τα συστατικά της αλληλεπίδρασης χρηστών.
 - android.widget - Τοποθετημένες στο πακέτο View, οι κλάσεις widget είναι τα στοιχεία της διεπαφής χρήστη (user-interface) που χρησιμοποιούνται στις εφαρμογές. Περιλαμβάνουν τις λίστες, τα κουμπιά, και τις διατάξεις (layouts).
 - com.google.android.maps - Ένα υψηλού επιπέδου API που παρέχει πρόσβαση στις εγγενείς λειτουργίες χαρτών που μπορούν να χρησιμοποιηθούν μέσα σε εφαρμογές. Περιλαμβάνει τη λειτουργία MapView, καθώς επίσης και τις κλάσεις Overlay και MapController που χρησιμοποιούνται για να διαχειριστούν και να ελέγξουν τους ενσωματωμένους χάρτες.
 - android.app - Ένα πακέτο υψηλού επιπέδου που παρέχει πρόσβαση στο μοντέλο εφαρμογών. Το πακέτο εφαρμογών περιλαμβάνει τα Activity και Service APIs που αποτελούν τη βάση για όλες τις Android εφαρμογές.
 - android.provider - Για να διευκολύνει την πρόσβαση των προγραμματιστών σε ορισμένους τυποποιημένους Παρόχους Περιεχομένου (Content Providers) (όπως η βάση δεδομένων επαφών), προσφέρονται κλάσεις που παρέχουν πρόσβαση σε τυποποιημένες βάσεις δεδομένων που περιλαμβάνονται σε όλες τις εκδόσεις Android.
 - android.telephony - Τα APIs τηλεφωνίας επιτρέπουν την άμεση αλληλεπίδραση με το τηλεφωνικό σωρό (phone stack) της συσκευής, δίνοντας τη δυνατότητα, να γίνει λήψη, και έλεγχος των τηλεφωνημάτων, της κατάστασης του τηλεφώνου, και των μηνυμάτων SMS.
 - android.webkit - Το πακέτο WebKit περιλαμβάνει APIs για την εργασία με περιεχόμενο βασισμένο στο διαδίκτυο, συμπεριλαμβανομένης της λειτουργίας wampree για την ενσωμάτωση μηχανών αναζήτησης και ενός διαχειριστή για cookies.
- Εκτός από το Android APIs, ο σωρός του Android περιλαμβάνει ένα σύνολο βιβλιοθηκών C/C++. Αυτές οι βιβλιοθήκες περιλαμβάνουν:
- OpenGL - Βιβλιοθήκη που χρησιμοποιείται για να υποστηρίξει 3D γραφικά βασισμένη στο Open GL ES 1.0 API.
 - FreeType - Υποστήριξη για τη bitmap και τη vector απόδοση γραμματοσειράς.
 - SGL - Βιβλιοθήκη πυρήνων που χρησιμοποιείται για να παρέχει μια μηχανή 2D γραφικών.
 - libc - Τυποποιημένη βιβλιοθήκη C που βελτιστοποιείται για συσκευές βασισμένες σε Linux.
 - SQLite - Ελαφριά μηχανή σχεσιακών βάσεων δεδομένων που χρησιμοποιείται για να αποθηκεύσει δεδομένα εφαρμογών.
 - SSL - Υποστήριξη για χρήση των Secure Sockets Layer του κρυπτογραφικού πρωτοκόλλου για ασφαλείς επικοινωνίες μέσω Διαδικτύου.

1.6.2 Προηγμένες Βιβλιοθήκες Android

Οι βιβλιοθήκες πυρήνα παρέχουν όλες τις λειτουργίες που είναι απαραίτητες για να ξεκινήσει η δημιουργία μιας εφαρμογής για Android, αλλά οι προηγμένες βιβλιοθήκες APIs προσφέρουν πραγματικά συναρπαστικές λειτουργίες.

Επειδή το Android στοχεύει να χρησιμοποιείται από ένα ευρύ φάσμα κινητών, η καταλληλότητα και η εφαρμογή των ακόλουθων APIs ποικίλουν ανάλογα με τη συσκευή επάνω στην οποία εφαρμόζονται.

- `android.location` - Το τοποκεντρικό API (location-based) δίνει πρόσβαση των εφαρμογών σε πληροφορίες για τη τρέχουσα φυσική θέση της συσκευής. Οι υπηρεσίες βασισμένες στη τοποθεσία παρέχουν πρόσβαση σε πληροφορίες τοποθεσίας χρησιμοποιώντας οποιοδήποτε υλικό ή τεχνολογία είναι διαθέσιμη στη συσκευή και βοηθούν στον προσδιορισμό θέσης.
- `android.media` - Τα APIs πολυμέσων παρέχουν υποστήριξη για την αναπαραγωγή και εγγραφή αρχείων ήχου και βίντεο, συμπεριλαμβανομένων των πολυμέσων ροής (streamed).
- `android.opengl` - Το Android προσφέρει μια ισχυρή αποδοτική 3D μηχανή που χρησιμοποιεί το OpenGL ES API με το οποίο δημιουργούνται 3D διεπαφές χρήστη για τις εφαρμογές.
- `android.hardware` - Όπου είναι διαθέσιμο, το API υλικού εκθέτει υλικό αισθητήρων συμπεριλαμβανομένης της φωτογραφικής μηχανής, του επιταχόμετρου (accelerometer) και των αισθητήρων πυξίδα.
- `android.bluetooth`, `android.net.wifi`, και `android.telephony` - Το Android επίσης παρέχει χαμηλού επιπέδου πρόσβαση στην πλατφόρμα υλικού, συμπεριλαμβανομένου του Bluetooth, του WI-FI, και του υλικού τηλεφωνίας.

1.7 Χρόνος Εκτέλεσης (Android Runtime)

Στο ίδιο επίπεδο με τις εγγενείς βιβλιοθήκες, βρίσκουμε και τον χρόνο εκτέλεσης Android. Εδώ υπάρχουν οι βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές από την Google. Η Dalvik τρέχει `.dex` αρχεία, τα οποία είναι `bytecodes` που προέρχονται από αρχεία `.class` και `.jar`. Εν αντιθέσει όμως με τα `.class` αρχεία, τα `.dex` είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java. Κάποια πακέτα και κλάσεις υπάρχουν και στο Android, κάποια άλλα δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα το Android παρέχει και επιπρόσθετα προσαρμοσμένα στις δικές του ανάγκες.

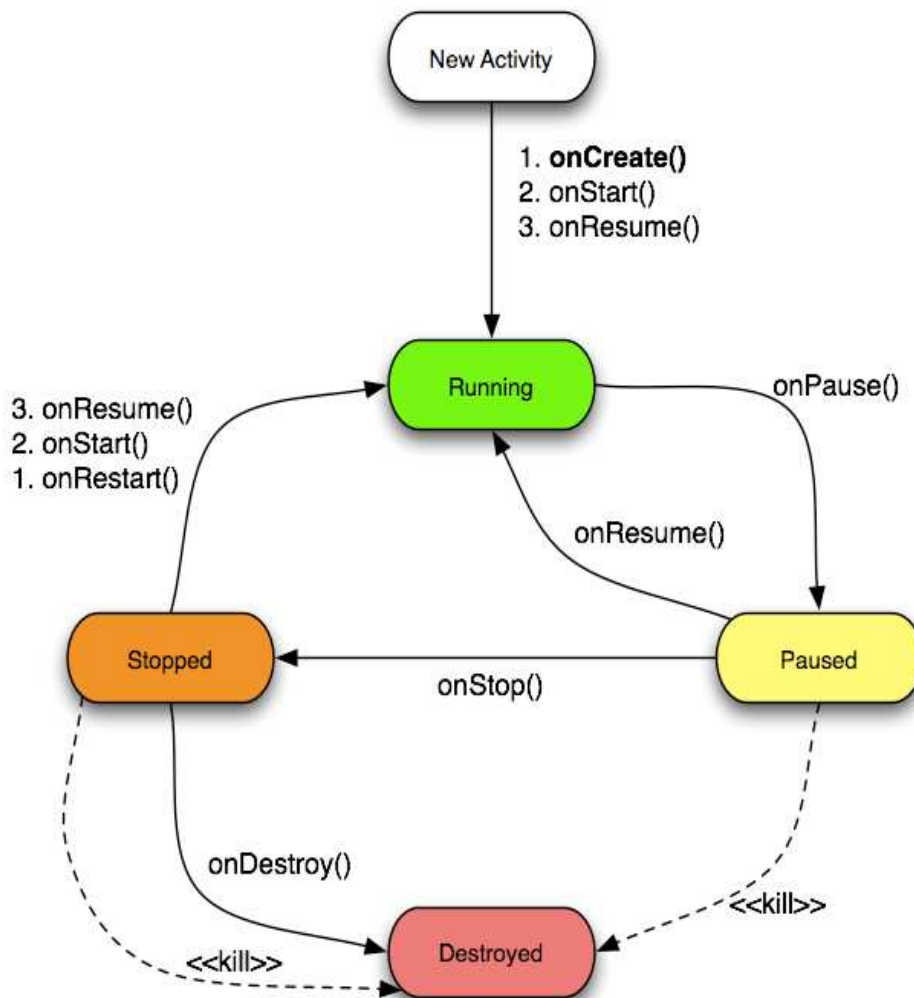
1.8 Πλαίσιο Εφαρμογής (Application Framework)

Πάνω από τις εγγενείς βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής. Αυτό το επίπεδο παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια [5]. Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής Δραστηριοτήτων - Activity Manager: Υπεύθυνος για τον έλεγχο του χρόνου ζωής [Εικόνα 7] των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.

- Διαχειριστής Πόρων - Resource Manager: Οι πόροι είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.
- Διαχειριστής Κοινοποιήσεων - Notification Manager: Ιδανικός τρόπος για την ενημέρωση του χρήστη για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεται η εργασία του.

Activity Lifecycle



Εικόνα 8: Ο κύκλος ζωής μιας Δραστηριότητας Android

1.9 Βασικά Συστατικά εφαρμογών του Android

Οι εφαρμογές Android αποτελούνται από κάποια συστατικά, που συνδέονται χρησιμοποιώντας ένα XML αρχείο, το `AndroidManifest.xml` του κάθε project, το οποίο περιγράφει κάθε συστατικό και πώς αλληλεπιδρά με τα άλλα. Υπάρχουν έξι συστατικά που παρέχουν τις δομικές μονάδες για τις εφαρμογές:

- **Δραστηριότητα (Activity)** – Είναι το επίπεδο παρουσίασης (presentation layer) της εφαρμογής. Μια δραστηριότητα είναι μια απλή οθόνη της εφαρμογής. Κάθε δραστηριότητα υλοποιείται σαν μια κλάση που επεκτείνει (extends) την βασική κλάση Δραστηριότητα (Activity base class). Η συγκεκριμένη κλάση προβάλλει μια διεπαφή χρήστη (user interface) αποτελούμενη από Όψεις (Views) και ανταποκρίνεται σε Συμβάντα (Events). Ένα Activity είναι ισοδύναμο με μια Φόρμα (Form) στην ανάπτυξη desktop εφαρμογών.
- **Υπηρεσία (Service)** - Μια Υπηρεσία είναι κώδικας που τρέχει για μεγάλο χρονικό διάστημα και χωρίς διεπαφή χρήστη (UI), ενημερώνοντας τις πηγές δεδομένων και τις ορατές Δραστηριότητες (Activities), ενεργοποιώντας Ειδοποιήσεις (Notifications). Αυτές τρέχουν και επεξεργάζονται δεδομένα ακόμα και όταν οι Δραστηριότητες της εφαρμογής δεν είναι ενεργές ή ορατές. Ένα καλό παράδειγμα είναι μια εφαρμογή που αναπαράγει μουσική από μια λίστα μουσικών κομματιών (media player). Σε μια τέτοια εφαρμογή, θα υπήρχαν κατά πάσα πιθανότητα μία ή και παραπάνω Δραστηριότητες που επιτρέπουν στον χρήστη να επιλέξει τραγούδια και να τα αναπαράγει. Ωστόσο, η αναπαραγωγή από μόνη της δεν θα έπρεπε να διαχειρίζεται από την Δραστηριότητα γιατί ο χρήστης θα περίμενε την μουσική να συνεχίζει να παίζει ακόμη και μετά την πλοήγησή του σε μια νέα οθόνη. Σε αυτή τη περίπτωση, η Δραστηριότητα της αναπαραγωγής μουσικής θα ξεκινούσε μια Υπηρεσία (Service) για να τρέξει στο παρασκήνιο και να συνεχίσει η μουσική να παίζει. Το σύστημα τότε θα κρατά την Υπηρεσία αναπαραγωγής ενεργή μέχρι να τελειώσει το κομμάτι. Όταν πραγματοποιηθεί σύνδεση σε μια Υπηρεσία, μπορεί να υπάρξει επικοινωνία με αυτή μέσω μιας διεπαφής που προσφέρεται από την Υπηρεσία. Για την Υπηρεσία μουσικής, αυτό θα επέτρεπε την παύση ή την επιστροφή πίσω στο κομμάτι (rewind) κλπ.
- **Παροχέας Περιεχομένου (Content Provider)** - Οι εφαρμογές μπορούν να σώζουν τα δεδομένα τους σε αρχεία, σε μια βάση δεδομένων SQLite ή με οποιοδήποτε άλλο μηχανισμό μπορούν. Ένας Παροχέας Περιεχομένου, ωστόσο, είναι χρήσιμος ώστε τα δεδομένα μιας εφαρμογής να είναι διαθέσιμα και σε άλλες εφαρμογές. Ένας Παροχέας Περιεχομένου είναι μια κλάση που υλοποιεί μια συγκεκριμένη ομάδα μεθόδων που επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να επανακτήσουν δεδομένα του τύπου που διαχειρίζεται ο Παροχέας Περιεχομένου. Οι συσκευές Android περιλαμβάνουν διάφορους εγγενείς Παροχείς Περιεχομένου (Content Providers) που εκθέτουν τις χρήσιμες βάσεις δεδομένων, όπως για παράδειγμα των στοιχείων των επαφών του χρήστη.
- **Προθέσεις (Intents)** - Το Android χρησιμοποιεί μια ειδική κλάση που λέγεται Πρόθεση (Intent) για να κινείται από οθόνη σε οθόνη. Η Πρόθεση περιγράφει τι θέλει η εφαρμογή να γίνει στη συνέχεια. Τα δυο πιο σημαντικά μέρη της δομής δεδομένων της Πρόθεσης είναι η Δράση (Action) και τα δεδομένα βάσει των οποίων αυτή θα εκτελεστεί. Τυπικές τιμές για μια δράση είναι η MAIN (η κεντρική είσοδος της εφαρμογής), VIEW, PICK, EDIT κλπ. Τα δεδομένα εκφράζονται ως URI (Uniform Resource Indicator). Για παράδειγμα, για να εμφανιστεί μια ιστοσελίδα στον φυλλομετρητή (browser), δημιουργείται μια Πρόθεση με Δράση VIEW και τα δεδομένα ως ένα URL.

```
New Intent (android.content.Intent.VIEW_ACTION;  
ContentURI.create("http://addev.org"));
```

Υπάρχει μια σχετική κλάση που λέγεται Φίλτρο Πρόθεσης (IntentFilter). Ενώ μια Πρόθεση είναι στην ουσία ένα αίτημα για να γίνει κάτι, το Φίλτρο Πρόθεσης είναι μια περιγραφή του τι είναι δυνατόν να διαχειριστεί ένας Δέκτης Πρόθεσης (Intent Receiver). Μια Δραστηριότητα που είναι σε θέση να προβάλλει πληροφορίες επικοινωνίας για ένα άτομο, θα ανακοινώνει με ένα Φίλτρο Πρόθεσης (IntentFilter) ότι γνωρίζει πως να διαχειριστεί την VIEW_ACTION όταν τα δεδομένα αντιπροσωπεύουν ένα άτομο. Οι Δραστηριότητες ανακοινώνουν τα Φίλτρα Πρόθεσης (IntentFilters) στο AndroidManifest.xml αρχείο.

Η πλοήγηση από οθόνη σε οθόνη πετυχαίνεται με Προθέσεις. Για να πλοηγηθεί κανείς προς τα μπρος, μια Δραστηριότητα καλεί την startActivity(myIntent). Το σύστημα τότε κοιτά στα Φίλτρα Προθέσεων (Intent Filters) για όλες τις εγκατεστημένες εφαρμογές και διαλέγει την Δραστηριότητα που τα Φίλτρα Πρόθεσης ταιριάζουν καλύτερα με την παράμετρο 'MyIntent' της κλήσης. Η νέα Δραστηριότητα ενημερώνεται για την Πρόθεση και ξεκινά. Η διαδικασία της υλοποίησης των Προθέσεων συμβαίνει κατά τον χρόνο εκτέλεσης της εφαρμογής, όταν δηλαδή καλείται η startActivity, πράγμα που προσφέρει 2 πλεονεκτήματα – κλειδιά:

1. Οι Δραστηριότητες μπορούν να επαναχρησιμοποιούν κάποια λειτουργικότητα από άλλα τμήματα του κώδικα απλά κάνοντας ένα αίτημα υπό την μορφή μιας πρόθεσης.
 2. Οι Δραστηριότητες μπορούν να αντικατασταθούν οποιαδήποτε στιγμή από μια νέα Δραστηριότητα με ένα αντίστοιχο Φίλτρο Πρόθεσης.
- Δέκτες Πρόθεσης (Intent Receiver) - Χρησιμοποιούνται για να εκτελεστεί μια εφαρμογή σε απάντηση ενός εξωτερικού συμβάντος (external event), για παράδειγμα, όταν το τηλέφωνο χτυπά, ή όταν το δίκτυο είναι διαθέσιμο, ή όταν είναι μεσάνυχτα. Οι Δέκτες Πρόθεσης δεν προβάλλουν μια διεπαφή χρήστη (UI), ωστόσο μπορούν να προβάλλουν Ειδοποιήσεις (Notifications) για να ειδοποιήσουν τον χρήστη για κάτι σημαντικό που συνέβη. Οι Δέκτες Πρόθεσης είναι επίσης καταχωρημένοι στο AndroidManifest.xml, αλλά μπορούν επίσης καταχωρηθούν από τον κώδικα χρησιμοποιώντας την Context.registerReceiver(). Η εφαρμογή δεν χρειάζεται να τρέχει για να κληθούν οι Δέκτες Πρόθεσης που έχει. Το σύστημα θα κινήσει την εφαρμογή, αν χρειαστεί, όταν ένας Δέκτης Πρόθεσης ενεργοποιηθεί. Οι εφαρμογές μπορούν επίσης να στέλνουν τις δικές τους Ανακοινώσεις Πρόθεσης (Intent Broadcasts) σε άλλους με την Context.broadcastIntent().
 - Δέκτες Μετάδοσης (Broadcast Receivers) - Βασική Κλάση για τον κώδικα, που θα λαμβάνει τις Προθέσεις (Intents) που στέλνονται από το sendBroadcast (). Μπορεί είτε δυναμικά να καταχωρηθεί μια περίπτωση αυτής της κατηγορίας με Context.registerReceiver(), είτε να καταχωρηθεί ένα στιγμιότυπο αυτής της κλάσης σε μια εφαρμογή μέσω της ετικέτας <receiver> στο AndroidManifest.xml. Οι Δέκτες Μετάδοσης αρχίζουν αυτόματα την εφαρμογή αποκρινόμενοι σε μια εισερχόμενη Πρόθεση, που στην ουσία κάνει την εφαρμογή, μια εφαρμογή οδηγούμενη από τα γεγονότα.

Υπάρχουν δύο σημαντικές κατηγορίες Μεταδόσεων (Broadcasts) που μπορούν να παραληφθούν:

1. Οι Κανονικές Μεταδόσεις - Normal broadcasts (που στέλνονται με Context.sendBroadcast) είναι απολύτως ασύγχρονες. Όλοι οι Δέκτες της Μετάδοσης οργανώνονται σε μια απροσδιόριστη διαταγή, συχνά συγχρόνως. Αυτό είναι αποδοτικότερο, αλλά σημαίνει ότι οι Δέκτες δεν μπορούν να χρησιμοποιήσουν το αποτέλεσμα ή να αποβάλουν APIs που συμπεριλαμβάνονται εδώ.
 2. Οι Μεταδόσεις Διαταγής - Ordered broadcasts (που στέλνονται με Context.sendOrderedBroadcast) παραδίδονται σε έναν Δέκτη τη φορά. Δεδομένου ότι κάθε Δέκτης εκτελείται στη συνέχεια, μπορεί να διαδώσει ένα αποτέλεσμα στον επόμενο Δέκτη, ή μπορεί να αποβάλει τη Μετάδοση έτσι ώστε να μη περάσει σε άλλους δέκτες. Ο χειρισμός της εκτέλεσης των Δεκτών διαταγής μπορεί να γίνει με την ιδιότητα του Φίλτρου Προθέσεων Ταιριάσματος (Matching Intent-Filter). Οι Δέκτες με την ίδια προτεραιότητα τρέχουν με τυχαία σειρά.
- Ειδοποιήσεις (Notifications) - Είναι ένα πλαίσιο ειδοποιήσεων χρήστη. Οι Ειδοποιήσεις δίνουν την δυνατότητα ειδοποίησης του χρήστη χωρίς να υπάρξει παρεμβολή στην τρέχουσα Δραστηριότητα (Activity) και του παρέχουν τις εξής δυνατότητες :
 1. Να δημιουργήσει ένα νέο εικονίδιο status bar.

2. Να επιδείξει τις πρόσθετες πληροφορίες (και να προωθήσουν μια πρόθεση) στο εκτεταμένο παράθυρο status bar.
3. Να ενεργοποιήσει τα φώτα/LEDs.
4. Να δονηθεί το τηλέφωνο.
5. Να ενεργοποιήσει διάφορους ήχους (ringtones, ήχοι του τηλεφώνου κ.τ.λ).

Οι Ειδοποιήσεις είναι ο προτιμότερος τρόπος για τα αόρατα τμήματα της εφαρμογής (Δέκτες μετάδοσης, Υπηρεσίες, και ανενεργές Δραστηριότητες) για να προειδοποιήσουν τους χρήστες ότι κάποια γεγονότα που έχουν εμφανιστεί απαιτούν την προσοχή του χρήστη. Παραδείγματος χάριν, όταν λαμβάνει μια συσκευή ένα μήνυμα κειμένων ή μια εισερχόμενη κλήση, προειδοποιεί με τη λάμψη των φώτων, την παραγωγή ήχων, την επίδειξη εικονιδίων ή με την εμφάνιση μηνυμάτων διαλόγου.

Ο Χειριστής Ειδοποιήσεων (Notification Manager) είναι μια Υπηρεσία συστήματος που χρησιμοποιείται για να χειριστεί τις ειδοποιήσεις. Καλείται μέσω της μεθόδου `getSystemService`, όπως φαίνεται παρακάτω:

```
String svcName = Context.NOTIFICATION_SERVICE;  
NotificationManager notificationManager;  
NotificationManager = (NotificationManager) getSystemService (svcName);
```

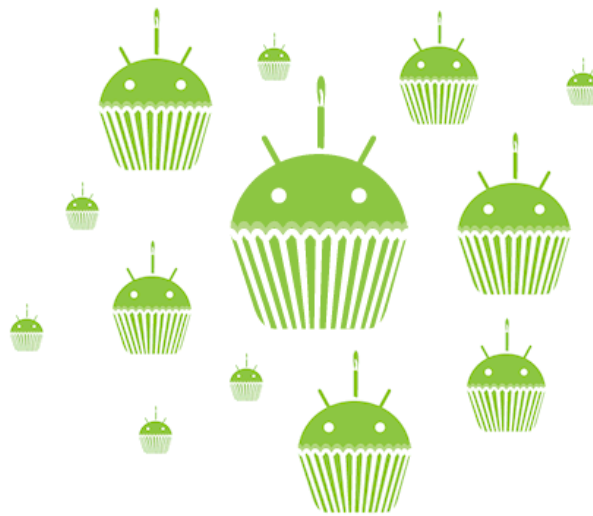
Χρησιμοποιώντας το Χειριστή Ειδοποιήσεων, μπορούν να ενεργοποιηθούν νέες ειδοποιήσεις, να τροποποιηθούν οι υπάρχουσες ή να αφαιρεθούν εκείνες που δεν είναι απαραίτητες ή επιθυμητές.

2 Ιστορικά - Εκδόσεις και χαρακτηριστικά

Η πρώτη έκδοση του Android SDK που εμφανίστηκε τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής, παρά μόνο την χρήση των κλασικών ‘σκληρών’ κουμπιών της συσκευής. [2]

2.1 Android 1.5 CUPCAKE

Τον Μάιο του 2009 εμφανίστηκε η έκδοση Android 1.5, ονόματι ‘Cupcake’ το λογότυπο της οποίας φαίνεται και στην [Εικόνα 8].



Εικόνα 9: Λογότυπο Android 1.5 CUPCAKE

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

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

2.2 Android 1.6 DONUT

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



Εικόνα 10: Λογότυπο Android 1.6 DONUT

Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως:

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

2.3 Android 2.0 ECLAIR

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

ANDROID 2.0



Εικόνα 11: Λογότυπο Android 2.0 ECLAIR

Ανάμεσα στις άλλες αλλαγές είναι και:

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

2.4 Android 2.2 FROYO

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



Εικόνα 12: Λογότυπο Android 2.2 FROYO

Η έκδοση FROYO, ανάμεσα σε άλλες αλλαγές, περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση
- Ενσωμάτωση στην μηχανή αναζήτησης, της μηχανής Javascript του Chrome V8
- Αυξημένη υποστήριξη Microsoft Exchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GAL look-up, remote wipe)
- Βελτιωμένος προωθητής εφαρμογής (application launcher), με συντομεύσεις προς τις εφαρμογές τηλεφώνου και εφαρμογές της Μηχανής Αναζήτησης
- Σύνδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi
- Ανανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση
- Επιλογή για απαγόρευση πρόσβασης δεδομένων μέσω ενός δικτύου κινητής τηλεφωνίας
- Γρήγορη εναλλαγή ανάμεσα σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους
- Φωνητική κλήση και διαμοιρασμός επαφών με Bluetooth
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς
- Η μηχανή αναζήτησης μπορεί να αποτυπώσει κινούμενα GIFs
- Υποστήριξη για πεδία μεταφόρτωσης αρχείων στην μηχανή αναζήτησης
- Υποστήριξη για εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη 14
- Υποστήριξη Adobe Flash 10.1

2.5 Android 2.3 GINGERBREAD

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



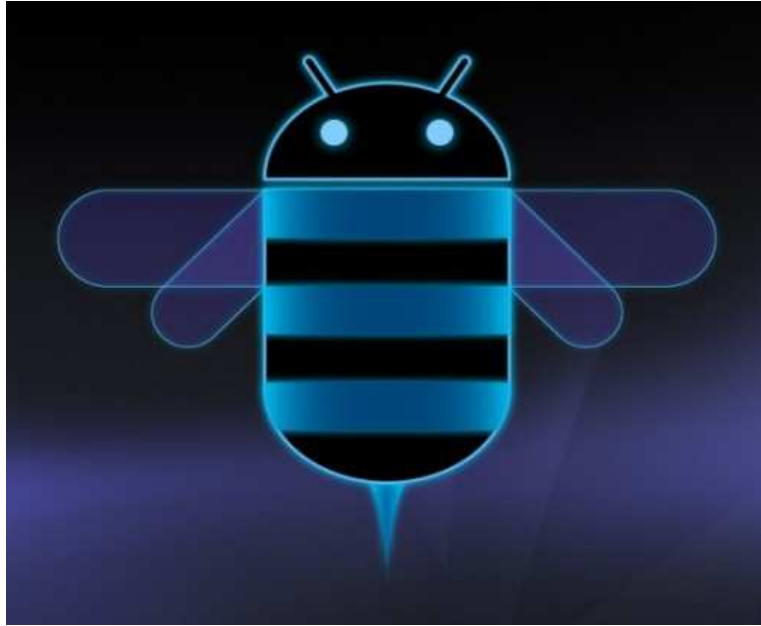
Εικόνα 13: Λογότυπο Android 2.3 GINGERBREAD

Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:

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

2.6 Android 3.0 HONEYCOMB

Η πιο πρόσφατη έκδοση, είναι η έκδοση Android 3.0 με το όνομα “Honeycomb”, η οποία είναι στην διάθεση των χρηστών και προγραμματιστών από τον Φεβρουάριο του 2011, λίγες μέρες μετά την επανέκδοση του Android 2.3.3, και προορίζεται αποκλειστικά για ταμπλέτες, στις οποίες γίνεται αναλυτικότερη αναφορά στην συνέχεια.



Εικόνα 14: Λογότυπο Android 3.0 HONEYCOMB

Μερικά από τα χαρακτηριστικά του είναι:

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

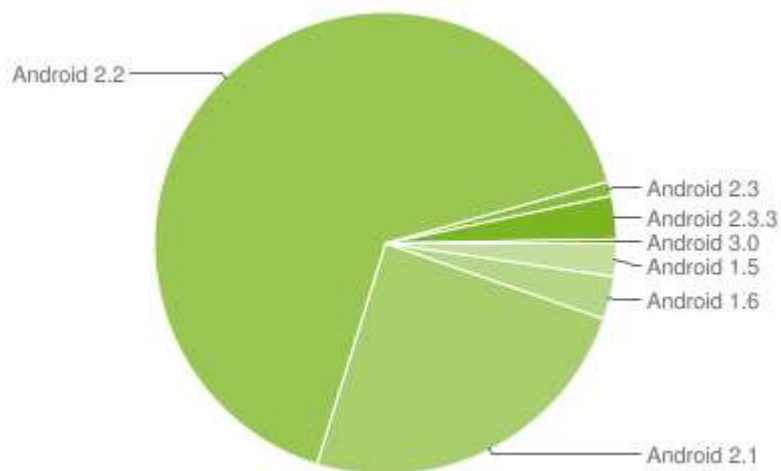
2.7 Android Ice Cream Sandwich

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



Εικόνα 15: Λογότυπο Android Ice Cream Sandwich

Θα είναι διαθέσιμο στο τελευταίο τρίμηνο του 2011 και φιλοδοξία της Google είναι να χρησιμοποιηθεί σε smartphones, tablets, laptops, netbooks κλπ. Το Android Ice Cream Sandwich θα φέρει μαζί του “ολογραφικό” 3D UI, πλουσιότερα widgets, ενισχυμένο multitasking, νέα APIs και φυσικά θα είναι πλήρως open source. Μια καλή πρώτη γεύση του Android Ice Cream Sandwich δόθηκε όταν η Google παρουσίασε τη λειτουργία face tracking, η οποία δεν περιορίζεται μόνο στις κινήσεις του κεφαλιού, αλλά ακολουθεί τα μάτια, τη μύτη και το στόμα. Στην περίπτωση που πραγματοποιείται video κλήση, “ζουμάρει” αυτόματα προς αυτόν που μιλά εκείνη τη στιγμή. Ένα ακόμα ενδιαφέρον στατιστικό που πρέπει να τονιστεί είναι τα ποσοστά των Android εκδόσεων που είναι εγκατεστημένες σε όλες τις Android συσκευές, όπως αυτά ανακοινώθηκαν για τον Απρίλιο του 2011 [Εικόνα 15]. [6][7]



Εικόνα 16: Ποσοστά των εγκατεστημένων εκδόσεων στις Android συσκευές

Όπως φαίνεται στην [Εικόνα 15] πιο πάνω, το Froyo είναι εγκατεστημένο στο μεγαλύτερο μέρος των συσκευών συγκεντρώνοντας το 65.9% αυτών. Το νεότερο Gingerbread συγκεντρώνει μόλις το 3% των συσκευών λόγω του ότι δεν έχει περάσει πολύς καιρός από την επίσημη κυκλοφορία του. Το Honeycomb, ως μια έκδοση αποκλειστικά για ταμπλέτες και ουσιαστικά με μηδαμινό χρόνο κυκλοφορίας, βρίσκεται στο 0.3%.

Τέλος οι παλαιότερες εκδόσεις του Android (Cupcake, Donut, Eclair) με τον καιρό θα συρρικνώνονται όπως είναι λογικό, παρόλο που το Eclair εξακολουθεί να είναι εγκατεστημένο στο 24.5% των Android συσκευών. Το Android έχει καταπληκτικά χαρακτηριστικά και πολλαπλές δυνατότητες, ενώ συνεχίζει να εκσυγχρονίζεται. Επιπρόσθετα, παρέχει καταπληκτικά εργαλεία για την ανάπτυξη εφαρμογών που κάνουν την ζωή του κατασκευαστή πραγματικά πολύ πιο εύκολη. Παρακάτω παρουσιάζεται η βασική αρχιτεκτονική της πλατφόρμας με τέτοιο τρόπο, ώστε να βοηθά στην κατανόηση της πρωτοτυπίας της και γίνεται συζήτηση των λόγων για τους οποίους το Android ευκολύνει την ζωή του προγραμματιστή, ενώ ταυτόχρονα του παρέχει πάμπολλες επιλογές και δυνατότητες για την δημιουργία πρωτοποριακών εφαρμογών. [5]

2.8 Εφαρμογές και Widgets

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

2.9 Γιατί Android

Εν κατακλείδι, παρατίθενται εδώ κάποια σημεία και σκέψεις για το Android, σαν μια τελευταία απόδειξη της πρωτοπορίας του στο χώρο και της ισχυρής δυναμικής του. Γιατί Android λοιπόν;

- Είναι μια πραγματικά ανοιχτή, ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux.
- Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
- Πάμπολλες ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες.
- Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλείδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο smartphone.
- Υψηλής ποιότητας γραφικά και ήχος.
- Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίξουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης.

Αυτοί είναι μερικοί μόνο από τους λόγους για τους οποίους έγινε η επιλογή του Android ως πλατφόρμα ανάπτυξης σε αυτή την πτυχιακή εργασία. Ο οποιοσδήποτε χρήστης θα μπορούσε να επαληθεύσει όλα τα παραπάνω, χρησιμοποιώντας απλά ένα τηλέφωνο ή μια ταμπλέτα που τρέχει το λειτουργικό αυτό σύστημα.

3. Κινητά τηλέφωνα Android

3.1 Η Πρώτη Συσκευή Android

Η πρώτη συσκευή που «έτρεχε» το λειτουργικό σύστημα Android ήταν το HTC Dream, γνωστό και ως T-Mobile G1, που έκανε την εμφάνιση του στα ράφια των καταστημάτων προς το τέλος του 2008. Στη συνέχεια, πολλά νέα Android κινητά έκαναν την εμφάνιση τους με “ναυαρχίδα” αυτών, το Motorola DROID (έκδοση CDMA), το οποίο από το Νοέμβρη του 2009 διατίθεντο αποκλειστικά από την Αμερικανική εταιρία τηλεπικοινωνιών Verizon, ενώ η GSM έκδοση του το Motorola Milestone, από διάφορους παρόχους κινητής τηλεφωνίας ανά τον κόσμο. Στην αυγή του 2010, η Google τάραξε τα νερά στον χώρο των τηλεπικοινωνιών με την ανακοίνωση του Google Nexus One, μιας καινοτόμου συσκευής που έτρεχε την νεότερη έκδοση του Android, 2.1 (που θεωρείται κυρίως βελτίωση της 2.0 ή Eclair). Κατασκευασμένο από την HTC, το Nexus One ξεχωρίζει για τα Live Wallpapers (κινούμενα backgrounds νέου σχεδιασμού που αλληλεπιδρούν με τις κινήσεις δαχτύλου του χρήστη) και τη νέα αναβαθμισμένη Έκθεση εικόνων και βίντεο (gallery). Η συσκευή έχει επίσης, το πλεονέκτημα της άμεσης και απροβλημάτιστης συνεργασίας με όλες τις υπηρεσίες της Google (Gmail, Google Contacts, Google Calendar κ.α.). Στην [Εικόνα 16] μπορούμε να δούμε μερικές τυπικές συσκευές Android. Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματα των Android κινητών τηλεφώνων.



Εικόνα 17: Τυπικές Συσκευές Android

3.2 Εναλλακτικά Πληκτρολόγια

Από πληκτρολόγια που προβλέπουν λέξεις όπως το SwiftKey, μέχρι το καινοτόμο Swype και το Dpen, υπάρχουν πολλές διαφορετικές επιλογές πληκτρολογίου για το Android. Πληκτρολογώντας σε ένα μικροσκοπικό πληκτρολόγιο κινητού δεν αποτελεί διασκέδαση για κανένα, οπότε είναι

σπουδαίο που το Android προσφέρει τόσες πολλές επιλογές για να το κάνει όσο πιο ανώδυνο γίνεται και πολύ εύκολο στην εγκατάσταση.

3.3 Αυτοματοποίηση

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

3.4 Προσαρμοσμένο Γραφικό Περιβάλλον

Οι χρήστες του Android έχουν την δυνατότητα με τα Home Launchers να μπορούν να τροποποιήσουν την αρχική τους οθόνη. Οι Home Launchers προσθέτουν όλα τα είδη επιπλέον χαρακτηριστικών για τις αρχικές οθόνες του κινητού όπως χειρονομίες, διαφόρων ειδών συντομεύσεις, ακόμα και χαμηλού επιπέδου ρυθμίσεις που μπορούν να βοηθήσουν στην αύξηση της ταχύτητας ενός παλιού κινητού. Είτε χρησιμοποιείται το πολύ γρήγορο LauncherPro, είτε το απίστευτα παραμετροποιήσιμο ADWLauncher, είτε το GoLauncher, οι Home Launchers προσφέρουν ένα σωρό πρόσθετες επιλογές στην συσκευή.

3.5 Widgets

Σίγουρα καταλαμβάνουν κάποιο χώρο, αλλά δεν υπάρχει υποκατάστατο για την ευκολία του να υπάρχει ένα μεγάλο widget με τον καιρό στην αρχική οθόνη ή ένα widget μουσικής που να δείχνει το κομμάτι που παίζει εκείνη την στιγμή. Ακόμα πιο χρήσιμο είναι τα to-do-list widgets, που βοηθούν περισσότερο στην παραγωγικότητα του χρήστη και την κάνουν πιο αποτελεσματική. [Εικόνα 17]



Εικόνα 18: Πώς παρουσιάζεται η αρχική οθόνη μιας εφαρμογής Widgets

3.6 Αναιρούμενα Μέσα Αποθήκευσης και Μπαταρία

Δεν είναι μέρος του Android λογισμικού απαραίτητα, αλλά η ανοιχτού λογισμικού φύση του δίνει αρκετά πλεονεκτήματα όπως η αφαίρεση και αναβάθμιση της μπαταρίας και της κάρτας μνήμης. Αν διαπιστωθεί ότι έχει φτάσει στο όριο του χώρου αποθήκευσης στο iPhone, δεν υπάρχει άλλη δυνατότητα εκτός της διαγραφής, ενώ στο Android γίνεται επέκταση του χώρου αποθήκευσης με την εισαγωγή μιας μεγαλύτερης κάρτας μνήμης. Παρόμοια, η αντικατάσταση της μπαταρίας με μια μεγαλύτερης διάρκειας, θα κάνει το κινητό να κρατήσει ακόμα περισσότερο χωρίς να χρειαστεί φόρτιση [Εικόνα 18].



Εικόνα 19: Φωτογραφία μιας καθημερινής κάρτας μνήμης

3.7 Ασύρματη Εγκατάσταση Εφαρμογών

Η αναζήτηση και η ανακάλυψη νέων εφαρμογών πρέπει να είναι κάτι το διασκευαστικό και όχι να αποτελεί πρόκληση λόγω ενός μικροσκοπικού App Store. Η περιήγηση στο App Store και το Cydia App Store δεν είναι και τόσο διασκευαστική μέσω του iPhone και είτε θα πρέπει να γίνει κατέβασμα (download) των εφαρμογών στο κινητό, είτε να γίνει σύνδεση με το iTunes για τον συγχρονισμό. Με το νέο Android Market ή σελίδες τρίτων όπως το AppBrain, είναι δυνατή η αναζήτηση μιας εφαρμογής και με το πάτημα ενός κουμπιού γίνεται αυτόματα η εγκατάσταση στο κινητό, ώστε να είναι έτοιμη προς χρήση την επόμενη φορά. Τι πιο βολικό από αυτό;

3.8 Προσαρμοσμένες ROM

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

3.9 Έλεγχος του Τηλεφώνου από τον Υπολογιστή σου

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

3.10 Flash Player

Για την πλοήγηση στο Internet είναι περισσότερο από απαραίτητη η χρήση του Flash. Ουσιαστικά, αποτελεί τη βάση της πλοήγησης. Είτε πρόκειται για μια σελίδα εξ' ολοκλήρου φτιαγμένη σε Flash, είτε για βίντεο, είτε για παιχνίδια, η εγκατάσταση του Flash στο κινητό ή το Tablet δίνει πρόσβαση σε πολλά πράγματα που διαφορετικά θα ήταν αδύνατη. Το Flash είναι τόσο διαδεδομένο, που η έλλειψη του ισοδυναμεί ουσιαστικά με απουσία Internet [Εικόνα 19].



Εικόνα 20: Πώς εμφανίζεται το uninstall στο Android

3.11 Πραγματική Ενσωμάτωση Εφαρμογών

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

4 Ταμπλέτες Android



Εικόνα 21: Ταμπλέτες Android

4.1 Tablet PC: Με Δυο Λόγια

Μια ταμπλέτα (Tablet PC) είναι ένας ηλεκτρονικός υπολογιστής μεγέθους συνήθως 6 έως 10 ιντσών, η οποία έχει ως βασικά χαρακτηριστικά αυτά ενός κανονικού Η/Υ και συνήθως περιλαμβάνει ένα πλήρες λειτουργικό σύστημα. Ένα φορητό Tablet PC περιλαμβάνει μια οθόνη αφής ως κύρια μέθοδο εισόδου δεδομένων και είναι σχεδιασμένο έτσι ώστε να χρησιμοποιείται για προσωπική χρήση. Ο όρος Tablet PC έγινε γνωστός από μια παρουσίαση της Microsoft το 2001 όπου και χρησιμοποιήθηκε για πρώτη φορά.

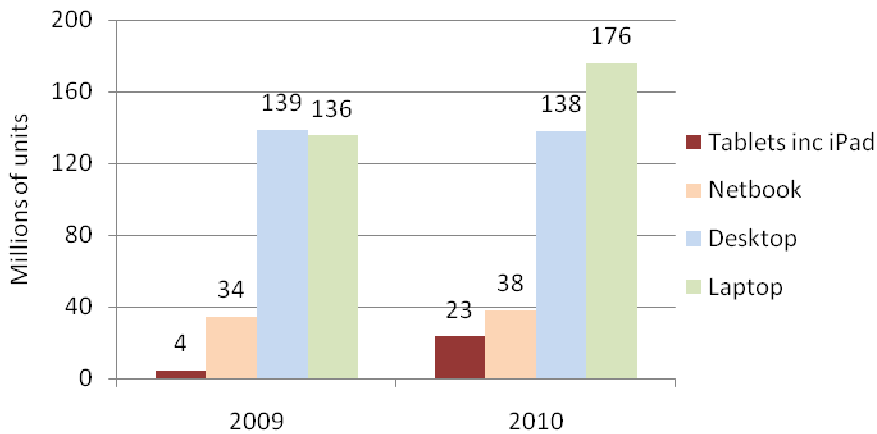
Οι ταμπλέτες χρησιμοποιούν εικονικά πληκτρολόγια και αναγνώριση γραφής, έτσι ώστε να είναι δυνατή η εισαγωγή κειμένου μέσω της οθόνης αφής. Όλες οι ταμπλέτες έχουν δυνατότητα ασύρματης σύνδεσης μέσω Wi-Fi, αλλά και ενσύρματα στο διαδίκτυο. Το λογισμικό τους περιλαμβάνει εφαρμογές γραφείου, προγράμματα περιήγησης στο διαδίκτυο, παιχνίδια, αλλά από την στιγμή που τρέχουν ολοκληρωμένα λειτουργικά συστήματα, μπορούν ουσιαστικά να χρησιμοποιήσουν οποιοδήποτε πρόγραμμα υποστηρίζει στο λειτουργικό τους. Συνήθως βέβαια, η υπολογιστική δύναμη αυτών των συστημάτων είναι περιορισμένη, οπότε δεν είναι πάντα ικανά να ανταπεξέλθουν στους πόρους που χρειάζονται μερικές πιο απαιτητικές εφαρμογές, με αποτέλεσμα να μην προσφέρουν την καλύτερη εμπειρία χρήσης. Σύμφωνα με έρευνες που πραγματοποιήθηκαν στις αρχές του 2011, οι ταμπλέτες βρίσκονται ακόμα στο στάδιο εισαγωγής τους στην αγορά, αφού μόνο το 5% των Αμερικάνων έχει στην κατοχή του μια, εκ των οποίων το 3% είναι οι συσκευές iPad της Apple.

4.2 Ιστορική αναδρομή.

Οι ταμπλέτες υπάρχουν από το 2001, αλλά έγιναν δημοφιλέστερες μετά την έλευση του Apple iPad, κερδίζοντας ένα σημαντικό μερίδιο του ενδιαφέροντος του κοινού. Ειδικά με την έλευση των Android ταμπλετών που πλέον κυριαρχούν στην αγορά, αναμένεται στο άμεσο μέλλον να αντικαταστήσουν πλήρως τα netbooks, μπαίνοντας για τα καλά στην καθημερινότητα των χρηστών. Έρευνα της εταιρίας Strategy Analytics έδειξε πως ως το 2015 η αγορά θα έχει κέρδος από τις πωλήσεις των tablets 49 δισ. Δολάρια [6]. Η αγορά ταμπλετών τελευταίας τεχνολογίας θα είναι η τρίτη σε δύναμη αγορά, μετά από αυτή της τηλεόρασης και των προσωπικών ηλεκτρονικών υπολογιστών. Η εταιρία ερευνών ανέφερε πως, σύμφωνα με τις προβλέψεις, θα έχουν πουληθεί 149 εκατομμύρια μονάδες ως το 2015, δηλαδή οι πωλήσεις θα αυξηθούν κατά 6,5 φορές από 2010.

Παρακάτω φαίνεται καθαρά ο ρυθμός αύξησης πωλήσεων των ταμπλετών, σε σύγκριση με όλα τα άλλα είδη προσωπικών υπολογιστών από το 2009 [7] [Εικόνα 20].

PC and Tablet sales volumes



Εικόνα 22: Αριθμός πωλήσεων PC και Tablet

4.2.1 Πώς εξελίσσεται έως σήμερα.

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

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

Οι ταμπλέτες Android, αποτέλεσαν το 27% της παγκόσμιας αγοράς για το τέταρτο τρίμηνο του '11, παρουσιάζοντας μία άνοδο της τάξεως του 2,3% από το 2010. 4,5 εκατομμύρια ταμπλέτες Android πουλήθηκαν κατά το ίδιο τρίμηνο, ενώ σε σύνολο πουλήθηκαν 16,7 εκατομμύρια ταμπλέτες. Στις πωλήσεις αυτές σημαντική παρουσία είχε η Samsung, με πωλήσεις που άγγιζαν το 1,5 εκατομμύριο.

Ένας ακόμα ανταγωνιστής που παρουσιάστηκε, ήταν και η ταμπλέτα της Amazon Kindle Fire, η οποία τάραξε αρκετά τα νερά. Τέλος και με την πρόσφατη παρουσίαση του «ενωτικού» λογισμικού για ταμπλέτες και smartphones, Android ICS, τα πράγματα έχουν αρχίσει και θερμαίνονται.

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

4.3 Η Adobe φέρνει τις Touch Apps στις ταμπλέτες Android



Εικόνα 23: Το Touch Apps της Android

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

Αυτές οι εφαρμογές είναι διαθέσιμες στο Android Market, για ταμπλέτες που τρέχουν Android 3.1 και πάνω. Το πακέτο περιλαμβάνει μεταξύ άλλων και τα Photoshop Touch, Collage and Proto, καθένα από τα οποία κοστολογείται στα 9,99 δολάρια. Οι εφαρμογές αυτές θα διαδραματίσουν και κεντρικό ρόλο στην επικείμενη πρωτοβουλία της Adobe, Creative Cloud, η οποία θα επιτρέπει στους χρήστες να μοιράζονται, να προβάλλουν και να μεταφέρουν αρχεία σε πολλαπλές συσκευές. Αυτό είναι κάτι το οποίο δεν αναμένεται να ξεκινήσει μέχρι το πρώτο εξάμηνο του επόμενου έτους, ενώ η πλήρης σουίτα των εφαρμογών αφής για iOS χρήστες αναμένεται να κυκλοφορήσει στις αρχές του επόμενου έτους.

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

4.4 Η εξέλιξη των Tablets

Καθώς η αγορά των tablet pc (ή «ταμπλέτες» όπως είναι αλλιώς γνωστές) ωριμάζει, είναι αναμενόμενο να υπάρχουν σημαντικές αλλαγές, τόσο στην εμφάνιση όσο και στο «περιεχόμενο» τους. Παράλληλα, θα υπάρχουν και αναβαθμίσεις στο λειτουργικό τους σύστημα, με το Android 4.0. Οι νέες ταμπλέτες που αναμένονται με το νέο έτος, προβλέπεται να είναι ακόμα πιο λεπτές και ελαφριές. Όσον αφορά τα μοντέλα των 10 ιντσών, το βάρος θα κυμαίνεται στα 540-590gr και το πάχος τους δε θα ξεπερνάει το 1 εκατοστό. Τώρα, μάλιστα, που έχει περάσει η επιτακτική ανάγκη της

γρήγορης εμφάνισης στην αγορά, οι κατασκευαστές των tablet pc αναμένεται να δώσουν μεγαλύτερη έμφαση στις λεπτομέρειες, τις οποίες υπόσχονταν από τα πρώτα κιόλας μοντέλα τους, όπως η ποιοτική απεικόνιση, η καθαρότητα του κειμένου στην οθόνη, τα ηχεία, η θύρα υπέρυθρων (για χρήση της ταμπλέτας ως Τηλε-κοντρόλ). Η ποικιλία στα μεγέθη θα συνεχίζει να υφίσταται, από 7 έως 10.1 ίντσες, καθώς δεν έχει διαμορφωθεί ακόμα σαφής προτίμηση στους χρήστες. Επίσης αναμένονται ακόμα χαμηλότερες τιμές, χάρη στο Amazon Kindle Fire των 199\$, το οποίο πωλείται ελαφρώς χαμηλότερα του κόστους του. Σύμφωνα με το διευθύνοντα της Nvidia οι τιμές των tablet pc που χρησιμοποιούν τον επεξεργαστή Tegra θα έχουν πέσει στα 299\$ ως τα μέσα του 2012.

4.4.1 Χρήση 4-πύρηνων επεξεργαστών

Η Nvidia έβγαλε τον περασμένο Νοέμβριο στην αγορά την πλατφόρμα Tegra 3. Γνωστή ως «project Kal-El», η πλατφόρμα Tegra 3 περιέχει ένα 4πύρηνιο επεξεργαστή ARM Cortex A9, ένα πέμπτο πυρήνα «χαμηλής κατανάλωσης» για να ασχολείται με «δευτερεύοντα» καθήκοντα, όπως η μουσική και ένα 12πύρηνιο επεξεργαστή GeForce για τα γραφικά. Αυτός ο «κινητήρας» είναι παραπάνω από επαρκής για να φέρει τις ταμπλέτες στην ίδια γραμμή εκκίνησης με τα laptop. Απ' ότι φαίνεται το Asus Transformer Prime είναι το πρώτο tablet pc που θα τρέχει με τον Tegra 3 της Nvidia. Αυτή η ταμπλέτα-laptop (εξ ου και η ονομασία Transformer) κυκλοφορεί στην τιμή των 500 δολαρίων με μνήμη 32GB. Τον Ιανουάριο του 2012 στη διεθνή έκθεση CES, αναμένεται να ανακοινωθούν και άλλες ταμπλέτες που θα χρησιμοποιούν τον Tegra 3.

4.4.2 Υψηλότερη ανάλυση

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

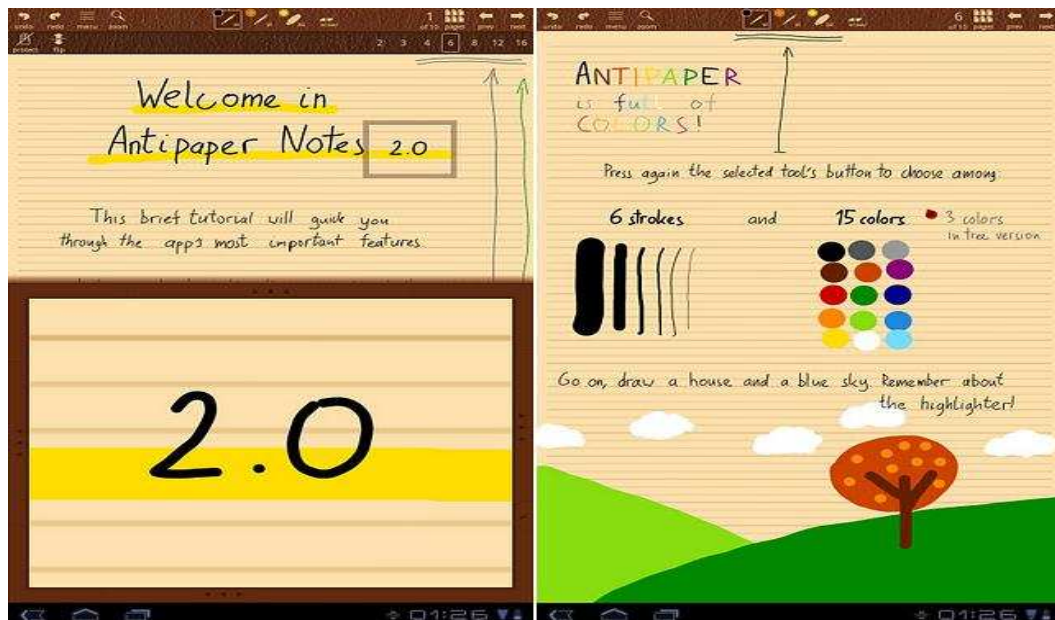
Τα επιπλέον pixels είναι σημαντικά, καθώς απαλύνουν τις ακμές στα γράμματα με αποτέλεσμα καθαρότερο κείμενο στην οθόνη. Μέσα στο 2012 αναμένεται σημαντική εξέλιξη στα λειτουργικά συστήματα των tablet pc. Ήδη έχουν ακουστεί πολλά για το Android 4.0 -το επονομαζόμενο Icescream Sandwich- με το οποίο η Google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

4.5 Μια απλή εφαρμογή στις ταμπλέτες Android



Εικόνα 24: Χειρόγραφες σημειώσεις για ταμπλέτες Android

Οι χρήστες των tablets σίγουρα θα έχουν διαβάσει κάποιο βιβλίο ή θα έχουν κρατήσει χειρόγραφες σημειώσεις. Είναι απλά μαγικό, να “κρατάς ένα βιβλίο” στα χέρια σου, να γυρίζεις τις σελίδες και όλα αυτά με τη βοήθεια μιας οθόνης αφής. Το ίδιο ισχύει και για τις σημειώσεις, που σίγουρα δεν περιλαμβάνουν καθημερινά θέματα, αλλά δημιουργικά θέματα, όπως επινόηση στίχων, συγγραφή μυθιστορημάτων ή ακόμα και προσωρινά πρώιμα σχεδιαγράμματα projects. Όλα αυτά με μια ωραία ρετρό πινελιά, παρέχονται από το Antipaper Notes. Μια εφαρμογή για thinkers που θέλουν να αποτυπώνουν όλες τις ιδέες τους σε χαρτί, για να μην τις ξεχνούν. Περιέχει διάφορα χρώματα “μολυβιών” για ζωγραφική, αλλά και ζουμ στο χαρτί ώστε να είναι δυνατή η συγγραφή με πολύ μικρούς χαρακτήρες.



Εικόνα 25: Μια μικρή ματιά στο γραφικό περιβάλλον του Antipaper

5 Εργαλεία ανάπτυξης λογισμικού

Μέσω του Android software development προκύπτουν ολοένα και νέες εφαρμογές που δημιουργούνται για το λειτουργικό σύστημα Android. Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java που χρησιμοποιεί το Android. Από τον Απρίλιο του 2011, έχουν αναπτυχθεί για το Android πάνω από 200.000 εφαρμογές, με πάνω από 3 δισεκατομμύρια downloads [8][9]. Η χρήση της πλατφόρμας Android έχει επίσης αυξηθεί, γιατί προτιμάται από τους προγραμματιστές των κινητών. Μια έρευνα τον Ιούνιο του 2011 έδειξε ότι πάνω από το 67% των κινητών χρησιμοποιούν εφαρμογές που αναπτύχθηκαν με χρήση της εν λόγω πλατφόρμας.

Παρακάτω παρουσιάζονται τα πιο διαδεδομένα λογισμικά για την ανάπτυξη εφαρμογών σε Android.

- Android SDK
- Native Development Kit
- App Inventor for Android
- Hyper Next Android Creator

Στις επόμενες παραγράφους αναπτύσσεται αναλυτικά το κάθε ένα από τα παραπάνω λογισμικά.

5.1 Android SDK

Το Android SDK παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη προγραμμάτων χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Τα χαρακτηριστικά του παρέχουν τεράστια ευελιξία και δυνατότητα ανάπτυξης έξυπνων εφαρμογών για κινητά τηλέφωνα, παρέχοντας δυνατότητες ανάπτυξης σε επιχειρηματίες όλων των κλάδων, είτε πρόκειται για αυτοματοποίηση πωλήσεων, ψυχαγωγία, παιχνίδια αλλά και κάθε άλλου είδους επιχειρήσεις. Ακόμα, παρέχει APIs για την χρήση web browser, εμφάνιση δισδιάστατων και τρισδιάστατων γραφικών, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού (ήχος, βίντεο, εικόνες), χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο.

Ένα σημαντικό γεγονός που ευνοεί την ανάπτυξη εφαρμογών είναι πως το πακέτο Android SDK συνεργάζεται με το Eclipse και συνεπώς ο προγραμματιστής μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, χωρίς να χρειάζεται να εξάγει κάθε φορά την εφαρμογή και να την εγκαθιστά σε κινητό. Επίσης, ο emulator είναι πολύ αξιόπιστος, καθώς έχει ακριβώς την ίδια συμπεριφορά με αυτή που θα είχε η εφαρμογή εάν είχε εγκατασταθεί σε ένα κινητό τηλέφωνο Android. Ένα άλλο σημαντικό πλεονέκτημα είναι το γεγονός πως σε αναβαθμίσεις του λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη-hardware- του κινητού τηλεφώνου. Παρακάτω, γίνεται ανάπτυξη του εξομοιωτή (emulator).

5.1.1 Μερικά από τα χαρακτηριστικά του SDK

Το Android SDK είναι ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση των «ενδοτέρων» του Android. Παρακάτω, φαίνονται τα βασικότερα χαρακτηριστικά του.

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

5.1.2 Ο εξομοιωτής (emulator).

Προκειμένου να γίνει ευκολότερη η διαδικασία της ανάπτυξης και αποσφαλμάτωσης μιας εφαρμογής, το Android SDK περιλαμβάνει έναν εξομοιωτή μιας εικονικής κινητής συσκευής, η οποία τρέχει το λειτουργικό του Android. Έτσι δεν είναι η αναγκαία η ύπαρξη πραγματικής κινητής συσκευής για την εκτέλεση και δοκιμή των εφαρμογών. Ο εξομοιωτής προσομοιώνει ένα μεγάλο πλήθος λειτουργιών μιας τυπικής συσκευής, η οποία τρέχει το Android:

- Παρέχει μια ποικιλία πλήκτρων πλοήγησης και ελέγχου
- Παρέχει μια οθόνη για την προβολή των εφαρμογών που τρέχουν στον εξομοιωτή
- Επιτρέπει στις εφαρμογές την χρήση των υπηρεσιών που προσφέρει η πλατφόρμα του Android, δηλαδή την κλήση άλλων εφαρμογών, την πρόσβαση στο δίκτυο, την αναπαραγωγή ήχου και βίντεο, την αποθήκευση και επαναφορά δεδομένων, την ειδοποίηση χρήστη, το γραφικό περιβάλλον του Android.

Επίσης παρέχει ένα πλήθος λειτουργιών για την ευκολότερη αποσφαλμάτωση:

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

5.1.3 Άλλα εργαλεία του Android

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

- Το Dalvik Debug Monitor Service (DDMS) το οποίο επιτρέπει την διαχείριση των διεργασιών στον εξομοιωτή ή στην συσκευή. Συγκεκριμένα δίνεται η δυνατότητα port-forwarding υπηρεσιών, λήψη screenshots, εμφάνιση πληροφοριών για τον σωρό και τα νήματα, logcat,

εμφάνιση πληροφοριών ράδιο και πληροφοριών διεργασιών, προσομοίωση εισερχόμενων κλήσεων και μηνυμάτων, προσομοίωση δεδομένων θέσης κ.α.

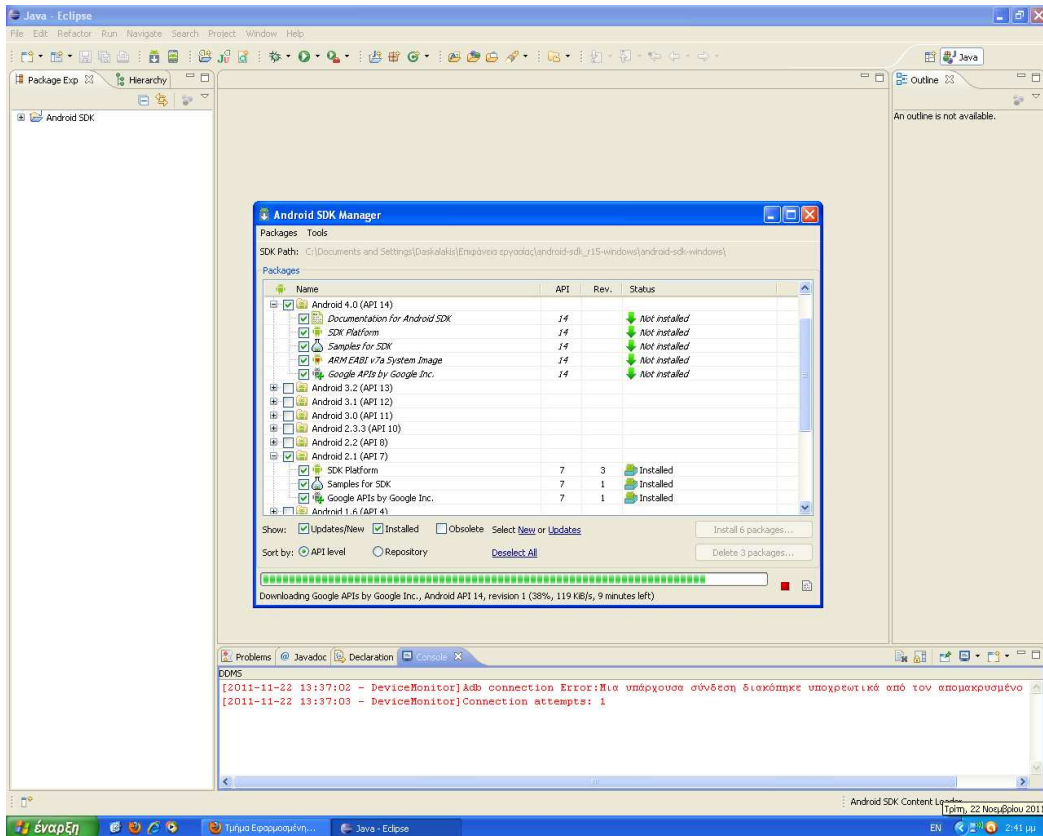
- Την Android Debug Bridge (ADB) η οποία επιτρέπει την διαχείριση της κατάστασης του εξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, η διαχείριση της προώθησης θυρών και η αντιγραφή από και προς την συσκευή ή τον εξομοιωτή.
- Το Android Asset Packaging Tool (AAPT) το οποίο δίνει την δυνατότητα δημιουργίας .apk αρχείων τα οποία περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.
- Την Android Interface Description Language (AIDL) η οποία επιτρέπει την δημιουργία κώδικα που επιτρέπει σε δύο διεργασίες σε μια συσκευή βασισμένη στο Android να συνομιλούν χρησιμοποιώντας διαδιεργασιακή επικοινωνία.
- Το sqlite3 το οποίο επιτρέπει την πρόσβαση στα δεδομένα της SQLite που δημιουργούνται από τις διάφορες εφαρμογές.
- Το Traceview που επιτρέπει την γραφική προβολή της ανάλυσης των trace log data που δημιουργούν οι διάφορες εφαρμογές.
- Το mkshcard το οποίο βοηθά στην δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (όπως η SD card).
- Το dx tool το οποίο μετατρέπει τα αρχεία .class από java bytecode σε Android bytecode.
- Το UI/Application Exerciser Monkey το οποίο είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδό-τυχαίες σειρές από συμβάντα χρήστη όπως clicks, touches, gestures καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.
- Το activitycreator το οποίο είναι ένα script που δημιουργεί Ant build αρχεία τα οποία μπορούν να χρησιμοποιηθούν για την μεταγλώττιση των εφαρμογών.

5.1.4 Μια γρήγορη εφαρμογή για το SDK

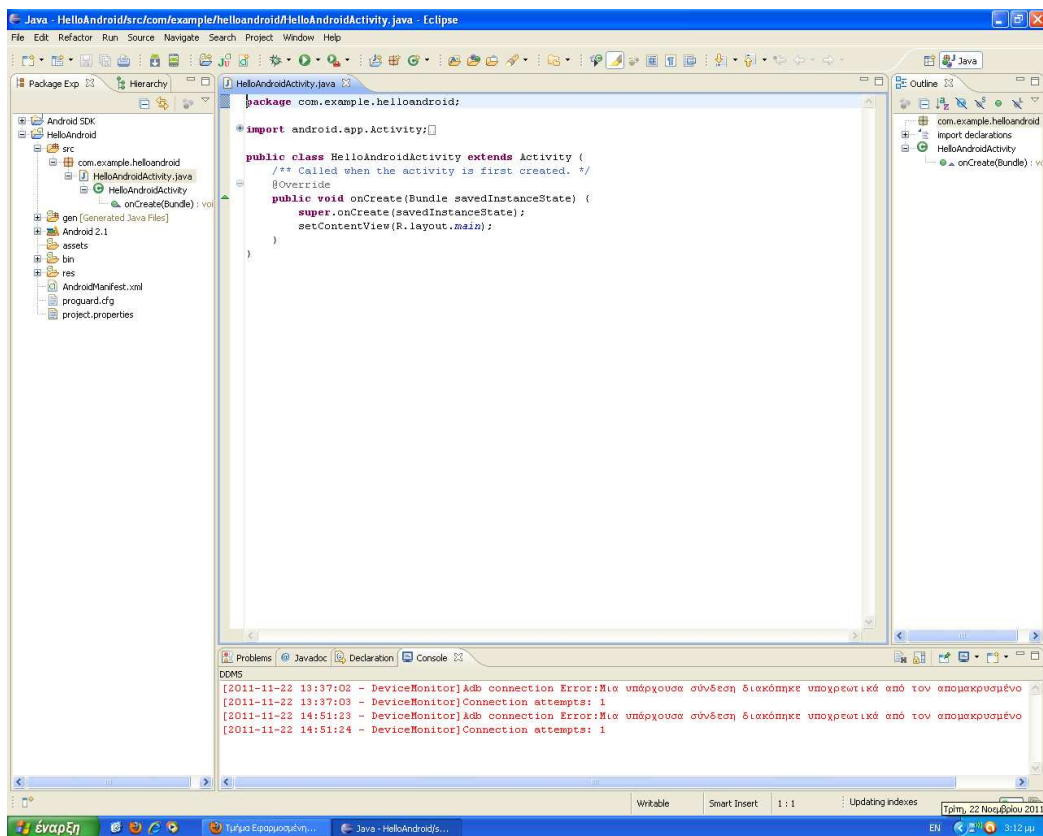
Το μόνο που μένει για να ολοκληρωθεί το κεφάλαιο του Android SDK είναι η ανάπτυξη μιας ιδίας εφαρμογής. Για να επιτευχθεί αυτό, το πρώτο βήμα που πρέπει να γίνει είναι η εγκατάσταση του απαραίτητου λογισμικού, και ακολούθως η ανάπτυξη μιας απλής εφαρμογής. Τα προγράμματα που θα είναι απαραίτητα είναι το Android SDK και στην συνέχεια το περιβάλλον ανάπτυξης λογισμικού Eclipse.

Το αρχικό βήμα είναι η μετάβαση στο developer.android.com/sdk [12], από το οποίο γίνεται το κατέβασμα του αρχείου Android SDK, εφόσον η εργασία πραγματοποιείται σε Windows. Παρακάτω φαίνεται το Quick Start guide, το οποίο παρέχει συγκεκριμένα βήματα για το development με την βοήθεια του Eclipse. Συγκεκριμένα, παραπέμπει το χρήστη να χρησιμοποιήσει ένα δοθέν url , ώστε να γίνει εγκατάσταση του Android Development Tools για το Eclipse κάτι που οδηγεί στο άνοιγμα του προαναφερθέντος προγράμματος.

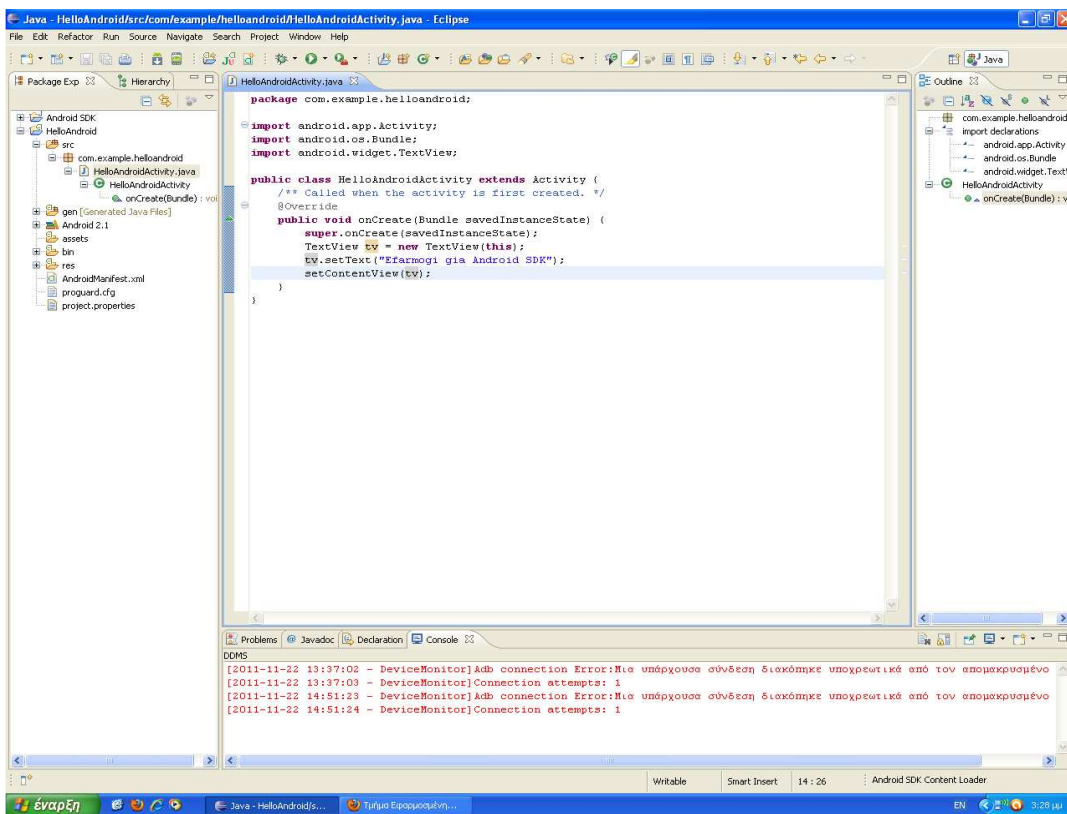
Με την ολοκλήρωση των παραπάνω βημάτων, το σύστημα είναι έτοιμο για την έναρξη της ανάπτυξης μιας νέας παρουσίασης. Ως παράδειγμα, φαίνεται ένα text Κείμενο.



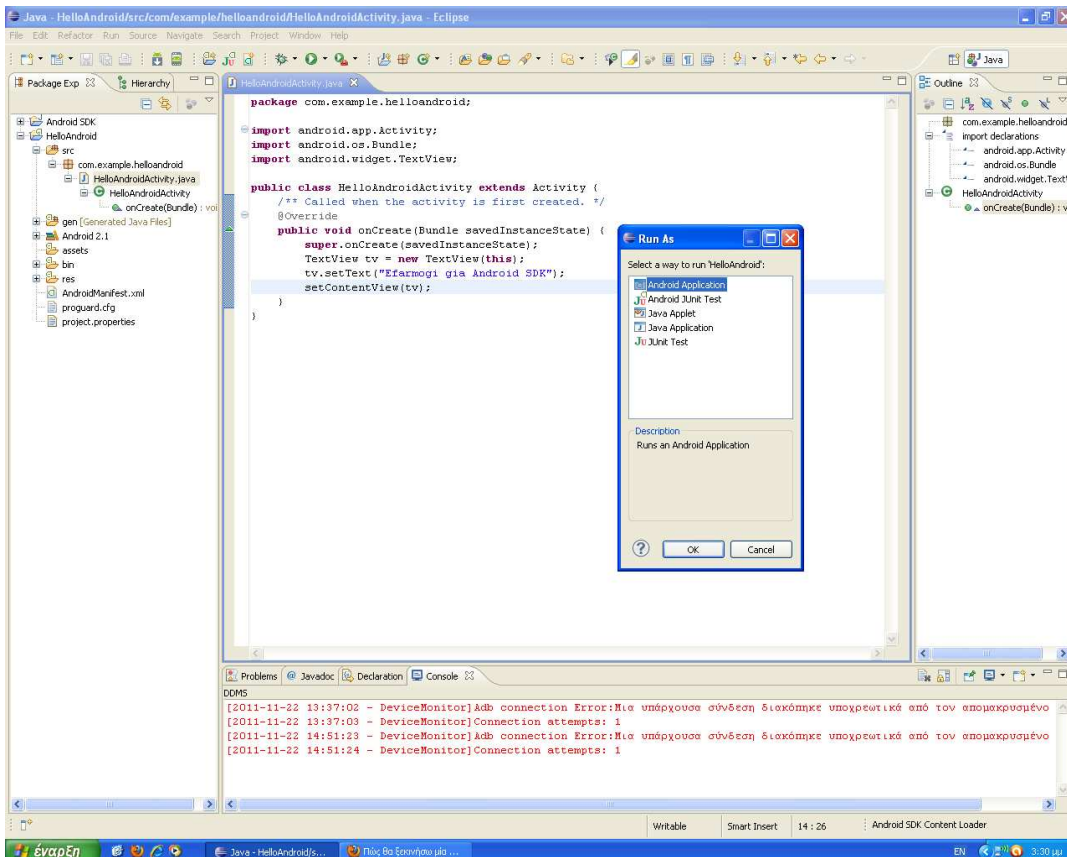
Εικόνα 26: Η εγκατάσταση του Android SDK



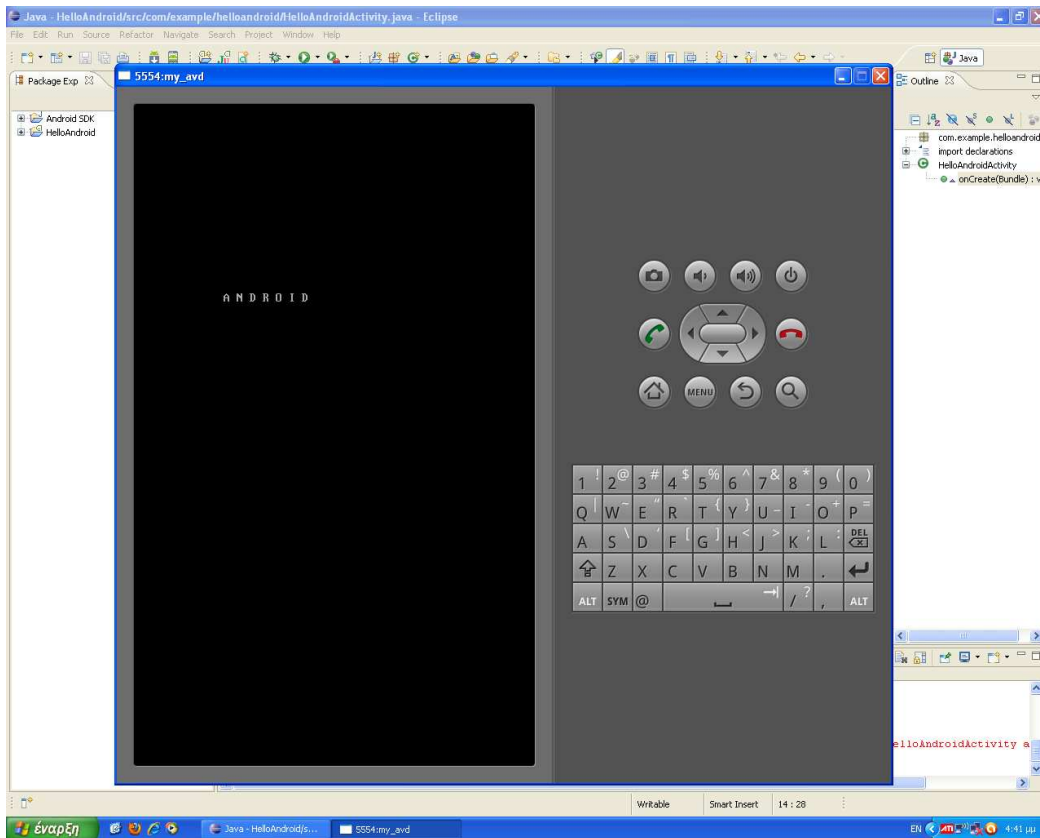
Εικόνα 27: Ο αρχικός βασικός κώδικας που περιέχει το πρόγραμμα



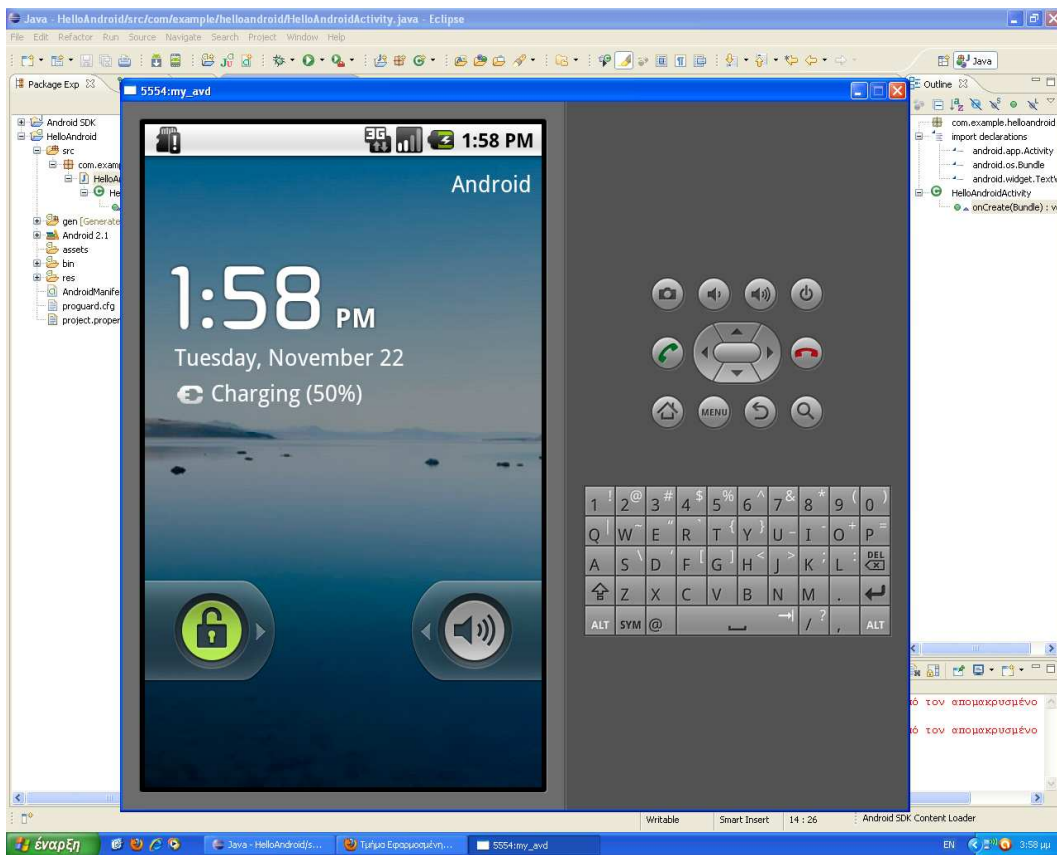
Εικόνα 28: Τροποποίηση του κώδικα για την εφαρμογή



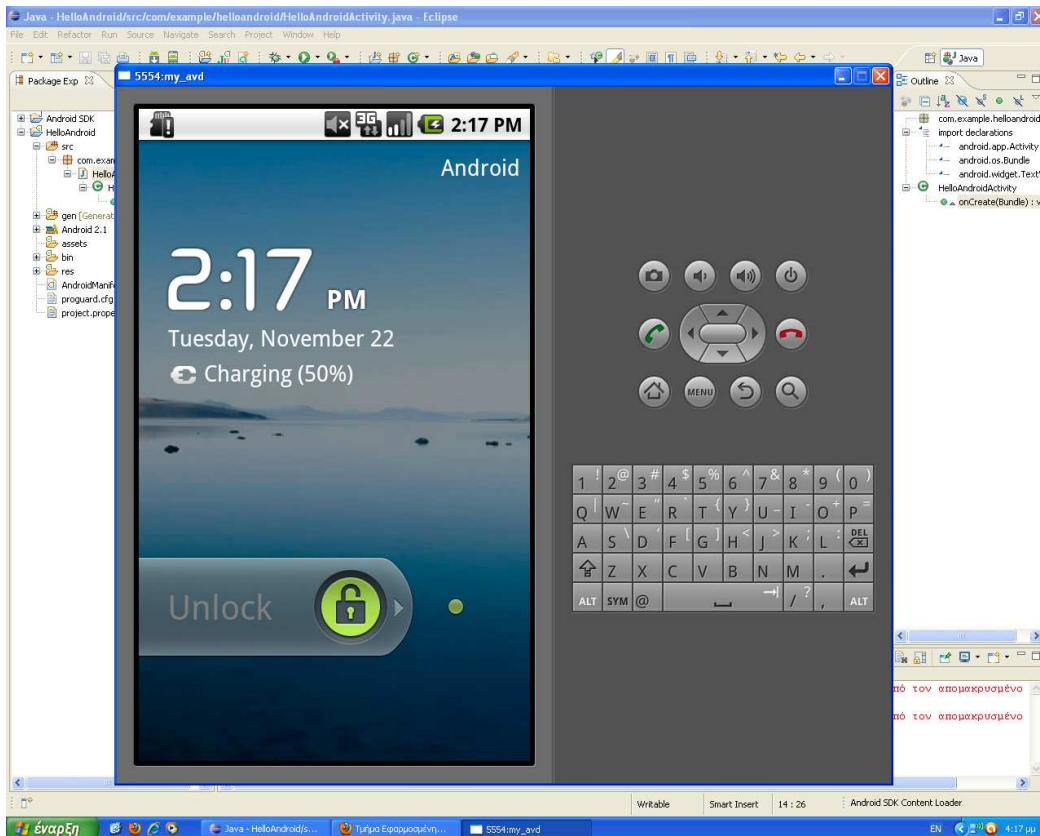
Εικόνα 29: Επιλέγοντας RUN, γίνεται η εκτέλεση του κώδικα



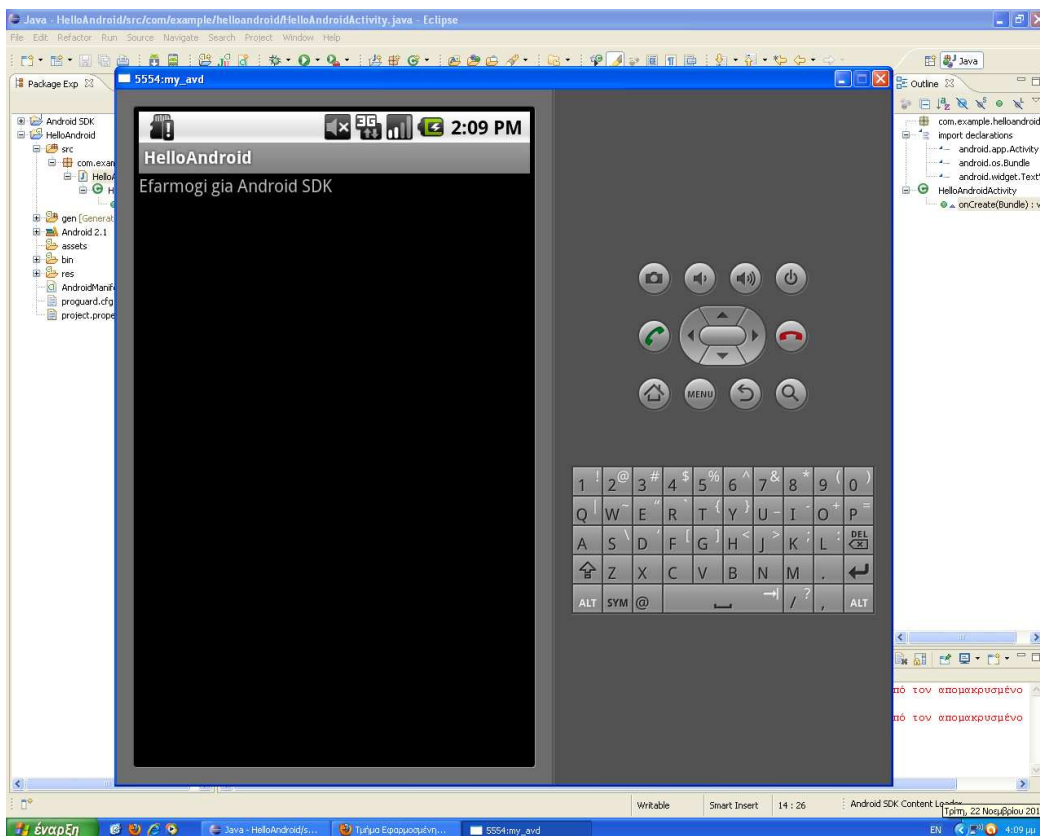
Εικόνα 30: Η εφαρμογή φορτώνει κανονικά



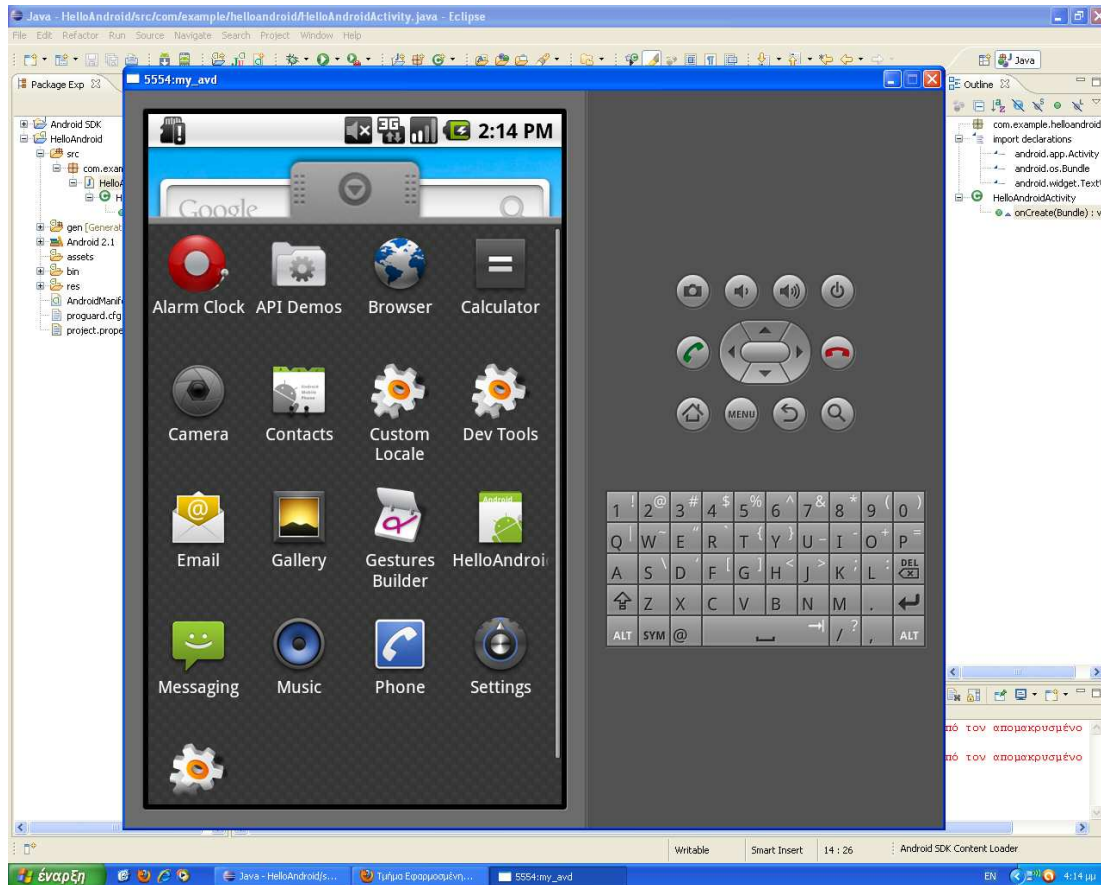
Εικόνα 31: Όπως έχει αναφερθεί, το γραφικό περιβάλλον είναι ίδιο με αυτό μιας Android συσκευής



Εικόνα 32: Ξεκλείδωμα της συσκευής μας για να τρέξει η εφαρμογή. Ο προσομοιωτής Android στη αρχική οθόνη



Εικόνα 33: Το Text κείμενο (που προαναφέρθηκε) “Εφαρμογή για Android SDK”



Εικόνα 34: Ο χειρισμός της εφαρμογής καταδεικνύει την ομοιότητα του προσομοιωτή με ένα πραγματικό κινητό τηλέφωνο Android

```

package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroidActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Efarmogi gia Android SDK");
        setContentView(tv);
    }
}

```

Εικόνα 35: Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή

5.2 Android NDK (Native Development Kit)



Εικόνα 36: Λογότυπο εφαρμογής Android NDK

Το Native Development Kit (NDK) είναι ένα σύνολο εργαλείων που επιτρέπει την ενσωμάτωση μερών (components) που κάνουν χρήση του ατόφιου κώδικα στις Android εφαρμογές. Οι Android εφαρμογές τρέχουν στην Dalvik virtual machine (εικονική μηχανή Dalvik). Είναι ένα συμπλήρωμα του SDK (Software Development Kit) παρέχει εργαλεία που επιτρέπουν την ενσωμάτωση native κώδικα στις εφαρμογές του Android. Το NDK επιτρέπει την εκτέλεση μερών των εφαρμογών, χρησιμοποιώντας γλώσσες ατόφιου κώδικα, όπως η C και η C++. Αυτό μπορεί να παρέχει πλεονεκτήματα σε συγκεκριμένες τάξεις εφαρμογών με τη μορφή της επαναχρησιμοποίησης του υπάρχοντος κώδικα και σε μερικές περιπτώσεις την αυξημένη ταχύτητα. Η ικανότητα άμεσης κλήσης της λειτουργίας στο OpenGL βελτιώνει την απόδοση γραφικών σε μεγάλες οθόνες κινητών. Ένα πολύ καλό website με πολλές πληροφορίες σχετικά με το Android NDK φαίνεται στην σχετική βιβλιογραφία [11].

5.2.1 Τα κύρια χαρακτηριστικά του NDK.

- Είναι ένα εργαλείο που επιτρέπει την ενσωμάτωση components που χρησιμοποιούν C και C++
- Προσφέρει παροχές σε κάποιες κλάσεις εφαρμογών
- Επαναχρησιμοποίηση ήδη υπάρχοντος κώδικα
- Αύξηση της ταχύτητας εκτέλεσης

5.2.2 Το Android NDK παρέχει τα εξής:

- Ένα σύνολο εργαλείων και αρχείων «χτισίματος» (build files) που χρησιμοποιούνται για να παραχθεί απόφιος κώδικα σε βιβλιοθήκες από πηγές C και C++.
- Έναν τρόπο ενσωμάτωσης των αντίστοιχων απόφιων βιβλιοθηκών σε ένα αρχείο πακέτου εφαρμογών (.apk), το οποίο μπορεί να αναπτυχθεί σε Android συσκευές.
- Ένα σύνολο απόφιων επικεφαλίδων και βιβλιοθηκών συστήματος (system headers and libraries), τα οποία θα υποστηρίζονται σε όλες τις μελλοντικές εκδόσεις της Android πλατφόρμας, αρχίζοντας από το Android 1.5. Εφαρμογές που χρησιμοποιούν λειτουργίες απόφιου κώδικα πρέπει να τρέχουν σε Android 2.3 ή μεταγενέστερα.
- Τεκμηρίωση, δείγματα και tutorials (φροντιστήρια).

Η τελευταία έκδοση του NDK υποστηρίζει τα εξής ARM σύνολα οδηγιών:

- ARMv5TE (που συμπεριλαμβάνει τις Thumb-1 εντολές)
- ARMv7-A (που συμπεριλαμβάνει τις Thumb-2 και VFPv3-D16 εντολές, με προαιρετική υποστήριξη για NEON/VFPv3-D32 εντολές)
- x86 εντολές

Ο κώδικας μηχανής ARMv5TE όπως αναμένεται, τρέχει σε συσκευές Android που είναι βασισμένες στο ARM. Ο ARMv7-A τρέχει μόνο σε συσκευές όπως: Verizon Droid ή Google Nexus One που έχουν συμβατή CPU. Η κύρια διαφορά ανάμεσα στα δύο σύνολα εντολών είναι ότι ο ARMv7-A εμπεριέχει hardware FPU, Thumbs-2 και εντολές NEON. Μπορείτε να γίνει χρήση του ενός ή και των δύο συνόλων εντολών – το ARMv5TE είναι το default, αλλά η αλλαγή στο ARMv7-A είναι τόσο εύκολη, όσο η προσθήκη μιας και μόνο γραμμής στο αρχείο εφαρμογών Application.mk, χωρίς να χρειαστεί να αλλάξει οτιδήποτε άλλο στο αρχείο. Επίσης μπορεί να γίνει build και τις δύο αρχιτεκτονικές ταυτόχρονα και αποθήκευση όλων στο τελικό .apk αρχείο. Ολοκληρωμένες πληροφορίες παρέχονται στο CPU-ARCH-ABIS.HTML στο πακέτο NDK.

Το NDK παρέχει σταθερές επικεφαλίδες (headers) για τη libc (βιβλιοθήκη της C), libm (βιβλιοθήκη της Math), OpenGL ES (βιβλιοθήκη 3D γραφικών), τη διεπαφή JNI και άλλες βιβλιοθήκες.

5.2.3 Πότε γίνεται ανάπτυξη σε απόφιο κώδικα

Το NDK δεν θα ωφελήσει στις περισσότερες εφαρμογές. Ένας developer πρέπει να ζυγίσει τα πλεονεκτήματα έναντι των μειονεκτημάτων. Ειδικά, η χρήση απόφιου κώδικα δεν έχει αποτέλεσμα στην αύξηση της αυτόματης απόδοσης, αλλά πάντα αυξάνει την πολυπλοκότητα της εφαρμογής. Γενικά, η χρήση απόφιου κώδικα γίνεται μόνο αν είναι απαραίτητο για την εφαρμογή και όχι για λόγους προτιμήσεων, όπως για παράδειγμα προγραμματισμός σε C/ C++.

Τυπικά καλές υποψήφιες λειτουργίες για το NDK είναι αυτές που είναι αυτοπεριοριζόμενες και που κάνουν εντατική χρήση της CPU και που παρόλα αυτά δεν δεσμεύουν πολλή μνήμη, όπως είναι η επεξεργασία σήματος, η εξομοίωση φυσικής κτλ. Η απλή επανα-κωδικοποίηση μιας μεθόδου σε C, συνήθως δεν φέρνει αποτέλεσμα μεγάλης αύξησης στην απόδοση. Κατά την εξέταση του αν πρέπει ή όχι ο προγραμματισμός πρέπει να γίνει με απόφιο κώδικα, είναι απαραίτητη η επεξεργασία των απαιτήσεων και το αν το API του Android framework παρέχει τη λειτουργικότητα που απαιτείται. Παρόλα αυτά το NDK μπορεί να είναι ένας αποτελεσματικός τρόπος επαναχρησιμοποίησης ενός μεγάλου μέρους του υπάρχοντος κώδικα σε C/C++.

Το Android framework παρέχει δυο τρόπους χρήσης του απόφιου κώδικα:

- Συγγραφή της εφαρμογής χρησιμοποιώντας το Android framework και χρήση του JNI για την προσπέλαση των API που παρέχονται από το Android NDK. Αυτή η τεχνική επιτρέπει την εκμετάλλευση της ευκολίας του Android framework, αλλά επιτρέπει ακόμα και την συγγραφή απόφιου κώδικα όταν χρειάζεται. Είναι δυνατή η εγκατάσταση εφαρμογών που

χρησιμοποιούν ατόφιο κώδικα μέσω του JNI σε συσκευές που τρέχουν Android 1.5 ή μεταγενέστερο.

- Συγγραφή μιας ατόφιας μεθόδου, η οποία επιτρέπει την υλοποίηση των lifecycle callbacks με ατομικό ατόφιο κώδικα. Το Android SDK παρέχει την NativeActivity κλάση, που είναι κλάση που διευκολύνει τον προγραμματιστή, αφού ειδοποιεί τον κώδικά για κάθε δραστηριότητα lifecycle callbacks (onCreate (), onPause (), onResume (), κτλ). Η υλοποίηση των callbacks στον κώδικα μπορεί να γίνει με τέτοιο τρόπο που να χειρίζονται αυτά τα γεγονότα, όταν προκύπτουν. Εφαρμογές που χρησιμοποιούν ατόφιας μεθόδους πρέπει να τρέχουν σε Android 2.5 ή μεταγενέστερο.
- Δεν είναι εφικτές οι προσπελάσεις χαρακτηριστικών όπως Services και Content Providers με ατομικό κώδικα, οπότε αν η χρήση αυτών ή οποιοδήποτε άλλων API frameworks κρίνεται απαραίτητη, θα πρέπει να γίνει συγγραφή κώδικα JNI ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

5.2.4 Περιεχόμενα του NDK

Το NDK περιέχει τα API, τεκμηρίωση και ενδεικτικές εφαρμογές που βοηθούν στην συγγραφή του ατόφιου (ατομικού) κώδικα.

5.2.5 Εργαλεία Ανάπτυξης

Το NDK περιλαμβάνει ένα σύνολο από διασταυρούμενες αλυσίδες εργαλείων (compilers, linkers κτλ) που μπορούν να παράγουν ARM δυαδικά αρχεία σε Linux, OS X, Windows (Με Cygwin) πλατφόρμες. Περιέχει ένα σύνολο από επικεφαλίδες συστήματος για σταθερά ατόφια APIs, που είναι εγγυημένο ότι υποστηρίζονται σε όλες τις μεταγενέστερες κυκλοφορίες/εκδόσεις της πλατφόρμας:

- libc (βιβλιοθήκη C) επικεφαλίδες
- libm (βιβλιοθήκη math) επικεφαλίδες
- επικεφαλίδες JNI διεπαφής
- libz (Zlib συμπίεση) επικεφαλίδες
- liblog (Android καταγραφή) επικεφαλίδα
- OpenGL ES 1.1 και OpenGL ES 2.0 (βιβλιοθήκες 3D γραφικών) επικεφαλίδες
- Libjnigraphics (Pixel buffer access) επικεφαλίδες (for Android 2.2 και μεταγενέστερα)
- Ένα ελάχιστο σύνολο από επικεφαλίδες για υποστήριξη της C++
- OpenSL ES ατόφιας βιβλιοθήκες ήχου
- Android native application APIS

Το NDK επίσης παρέχει ένα σύστημα δόμησης (build system) που επιτρέπει την αποτελεσματική εργασία με τα source files, χωρίς να απαιτείται διαχείριση των λεπτομερειών των toolchain/πλατφόρμας/ CPU/ ABI. Δημιουργούνται πολύ μικρά build files για την περιγραφή ποιων source files θα γίνουν compile και ποια Android εφαρμογή θα τα χρησιμοποιήσει – Το Build system μεταγλωττίζει τα source files και τοποθετεί τις σκόρπιες βιβλιοθήκες κατευθείαν στο project της εφαρμογής.

Προσοχή! Με εξαίρεση τις παραπάνω βιβλιοθήκες, οι ατόφιας βιβλιοθήκες συστήματος στην Android πλατφόρμα δεν είναι σταθερές και μπορεί να αλλάξουν στην μελλοντική έκδοση της

πλατφόρμας. Οι εφαρμογές πρέπει να κάνουν χρήση μόνο των σταθερών ατόφιων βιβλιοθηκών συστήματος που παρέχονται στο NDK.

5.2.6 Τεκμηρίωση

Το πακέτο NDK περιλαμβάνει ένα σύνολο τεκμηριώσεων που περιγράφουν τις ικανότητες του NDK και πώς γίνεται η χρήση του για τη δημιουργία διαμοιρασμένων βιβλιοθηκών για τις Android εφαρμογές. Σε αυτή την έκδοση, η τεκμηρίωση παρέχεται μόνο στο NDK πακέτο που είναι προς κατέβασμα (downloadable). Συμπεριλαμβάνονται τα εξής αρχεία:

- **INSTALL.HTML** — περιγραφή του τρόπου εγκατάστασης του NDK και της ρύθμισης για το σύστημα του χρήστη
- **OVERVIEW.HTML** — παρέχει μια επισκόπηση των ικανοτήτων και της χρήσης του NDK
- **ANDROID-MK.HTML** — περιγράφει τη χρήση του αρχείου `Android.mk`, το οποίο ορίζει τα ατόφια `source files` προς μεταγλώττιση
- **APPLICATION-MK.HTML** — περιγράφει τη χρήση του αρχείου `Application.mk`, το οποίο περιγράφει τα ατόφια `source files` που απαιτούνται από την Android εφαρμογή
- **CPLUSPLUS-SUPPORT.HTML** — περιγράφει την υποστήριξη της C++ που παρέχεται στο Android NDK
- **CPU-ARCH-ABIS.HTML** — μια περιγραφή των υποστηριζόμενων αρχιτεκτονικών CPU
- **CPU-FEATURES.HTML** — μια περιγραφή της στατικής `cpufeatures` βιβλιοθήκης που επιτρέπει στον κώδικα της εφαρμογής να εντοπίσει την στοχευμένη οικογένεια συσκευών στην οποία ανήκει η CPU και τα προαιρετικά χαρακτηριστικά σε runtime.
- **CPU-ARM-NEON.HTML** — περιγραφή του πώς γίνεται `build` με προαιρετικές ARM NEON / VFPv3-D32 εντολές.
- **CHANGES.HTML** — μια ολοκληρωμένη λίστα των αλλαγών του NDK σε όλες τις εκδόσεις.
- **DEVELOPMENT.HTML** — περιγραφή του τρόπου τροποποίησης του NDK και παραγωγή πακέτων εκδόσεων για αυτό
- **HOWTO.HTML** — πληροφορίες σχετικά με κοινές εργασίες που σχετίζονται με την ανάπτυξη του NDK
- **IMPORT-MODULE.HTML** — περιγραφή του τρόπου διαμοιρασμού και επαναχρησιμοποίησης των `modules`
- **LICENSES.HTML** — πληροφορίες σχετικά με τις διάφορες ανοιχτού κώδικα άδειες που διέπουν το Android NDK
- **NATIVE-ACTIVITY.HTML** — περιγραφή του τρόπου υλοποίησης των ατόφιων δραστηριοτήτων
- **NDK-BUILD.HTML** — περιγραφή της χρήσης του `ndk-build` script
- **NDK-GDB.HTML** — περιγραφή του τρόπου χρήσης του debugger ατόφιου κώδικα
- **PREBUILT.HTML** — πληροφορίες σχετικά με το πώς λειτουργούν οι διαμοιρασμένες και στατικές `prebuilt` βιβλιοθήκες
- **STANDALONE-TOOLCHAIN.HTML** — περιγράφει πως γίνεται χρήση του Android NDK `toolchain` σαν μοναδικού `compiler` (ακόμα και σε beta).
- **SYSTEM-ISSUES.HTML** — γνωστά προβλήματα στις Android εικόνες συστήματος που εμφανίζονται κατά τον προγραμματισμό χρησιμοποιώντας το NDK.

- STABLE-APIS.HTML — μια ολοκληρωμένη λίστα των σταθερών APIs εκτεθειμένα με επικεφαλίδες στο NDK

Επιπρόσθετα, το πακέτο περιλαμβάνει λεπτομερείς πληροφορίες σχετικά με τη «βιονική» C βιβλιοθήκη που παρέχεται με την Android πλατφόρμα που είναι επίφοβη, εφόσον ο προγραμματισμός γίνεται με το NDK.

- OVERVIEW.HTML — παρέχει μια επισκόπηση της «βιονικής» C βιβλιοθήκης και των χαρακτηριστικών που αυτή παρέχει.

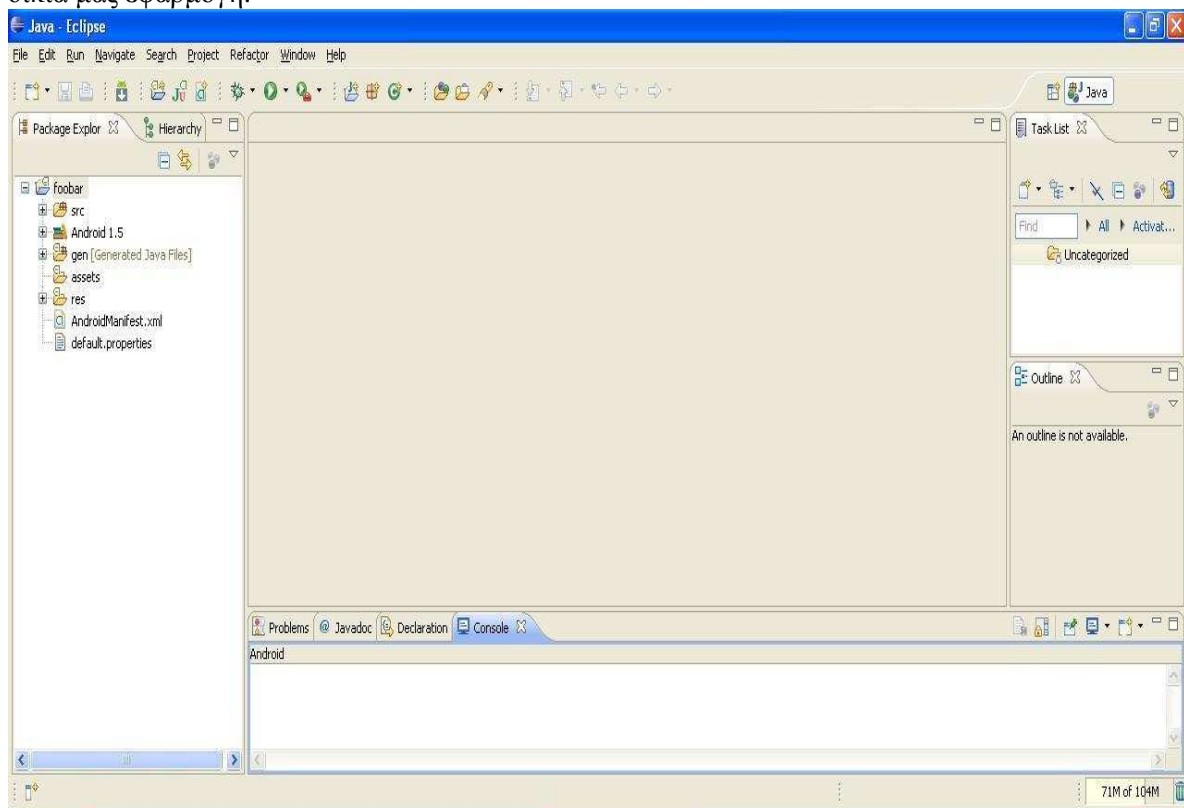
5.2.7 Ανάπτυξη εφαρμογής για το Android NDK

Μετά την ανάπτυξη και επεξήγηση των σχετικών πληροφοριών για το Android NDK , ακολουθεί η ανάπτυξη μιας ίδιας εφαρμογής. Σε αυτό το σημείο, θα ασχοληθούμε με το βασικό περιβάλλον για μια πρώτη εφαρμογή στο Android NDK. Θα χρησιμοποιήσουμε τα θεμέλια που χρησιμοποιεί η εφαρμογή του NDK. Οι λήψεις που είναι απαραίτητες για την αρχική διαμόρφωση του περιβάλλοντος, ώστε να είναι προετοιμασμένο, απαιτούν κάποιο χρόνο (περίπου 30 λεπτά συνολικά).

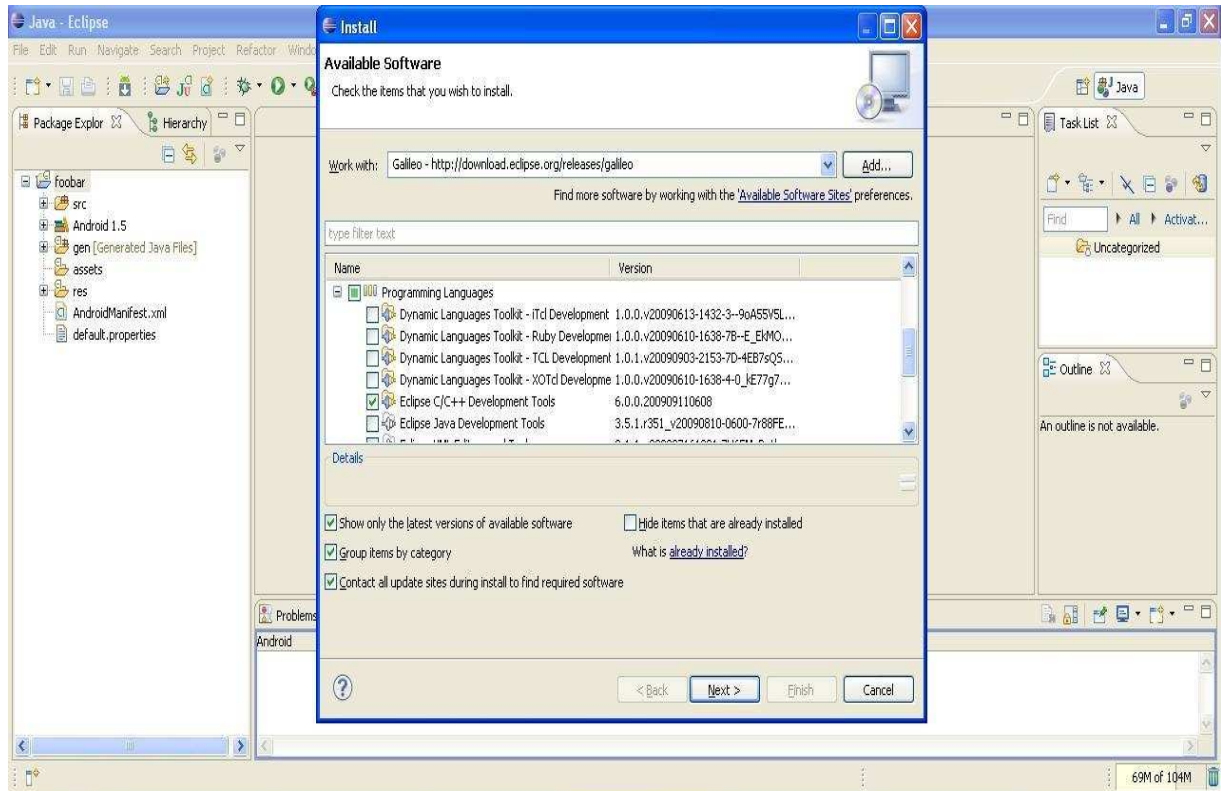
Τα βήματα που πρέπει να ακολουθηθούν για την δημιουργία μιας εφαρμογής σε NDK είναι τέσσερα.

- Βήμα 1 Εγκατάσταση C/C++ support on Eclipse
- Βήμα 2 Εγκατάσταση Cygwin
- Βήμα 3 Εγκατάσταση του Android NDK
- Βήμα 4 Ανάπτυξη μιας NDK εφαρμογής

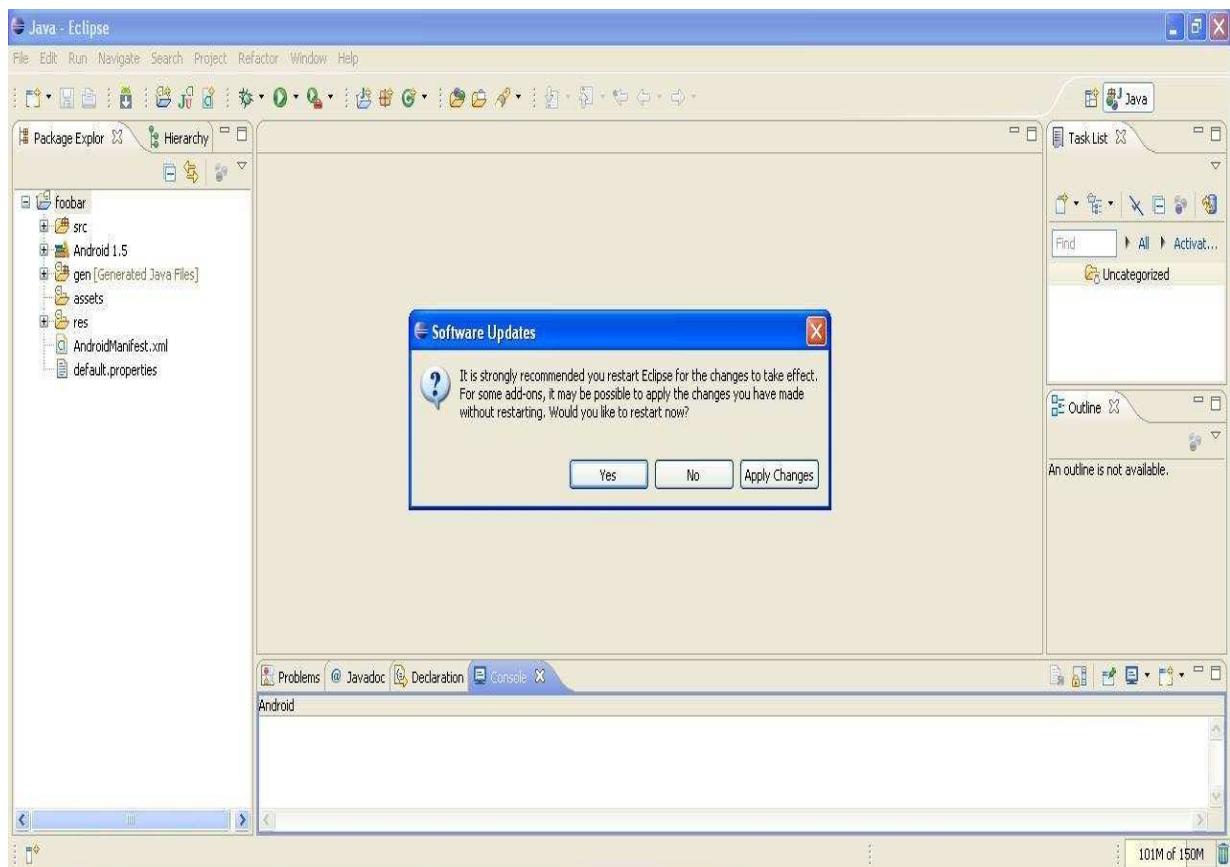
Αφού έχουμε πραγματοποιήσει τα παραπάνω βήματα, είμαστε έτοιμοι να δημιουργήσουμε την δικιά μας εφαρμογή.



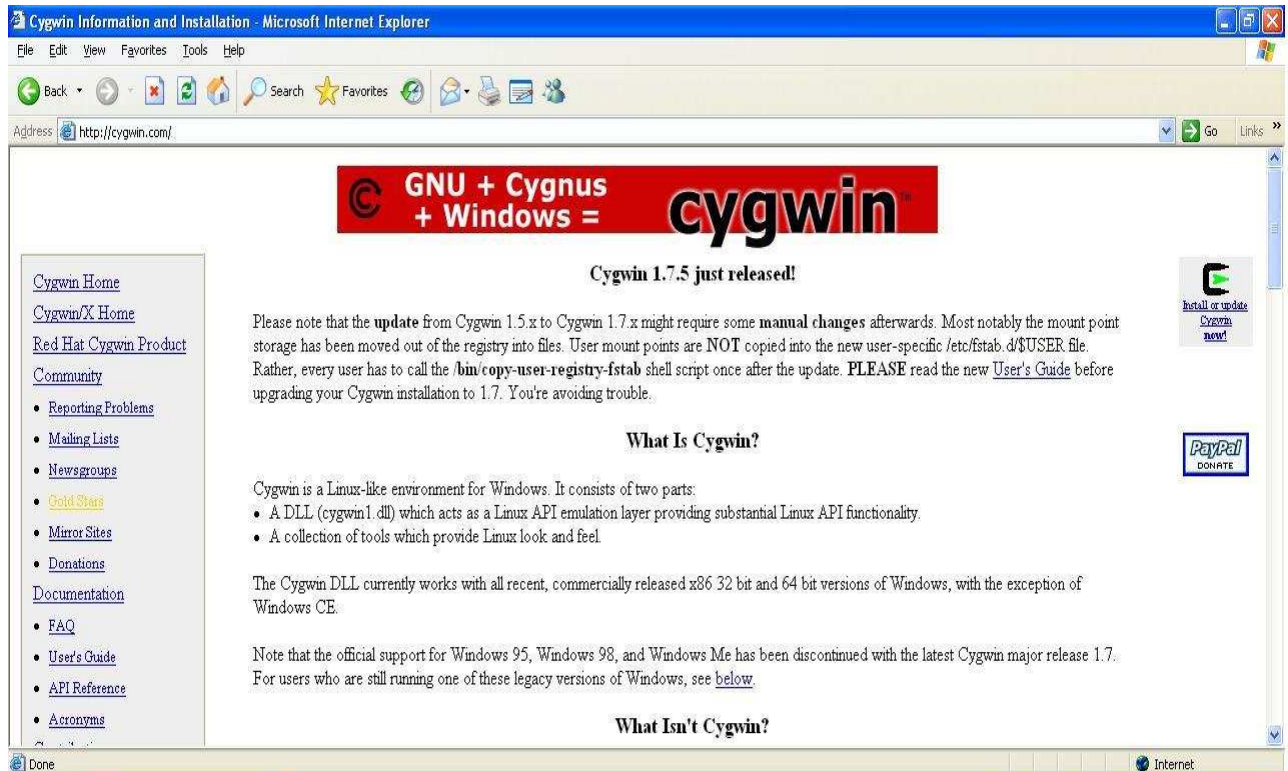
Εικόνα 37: Εγκατάσταση της C/C++ support on Eclipse



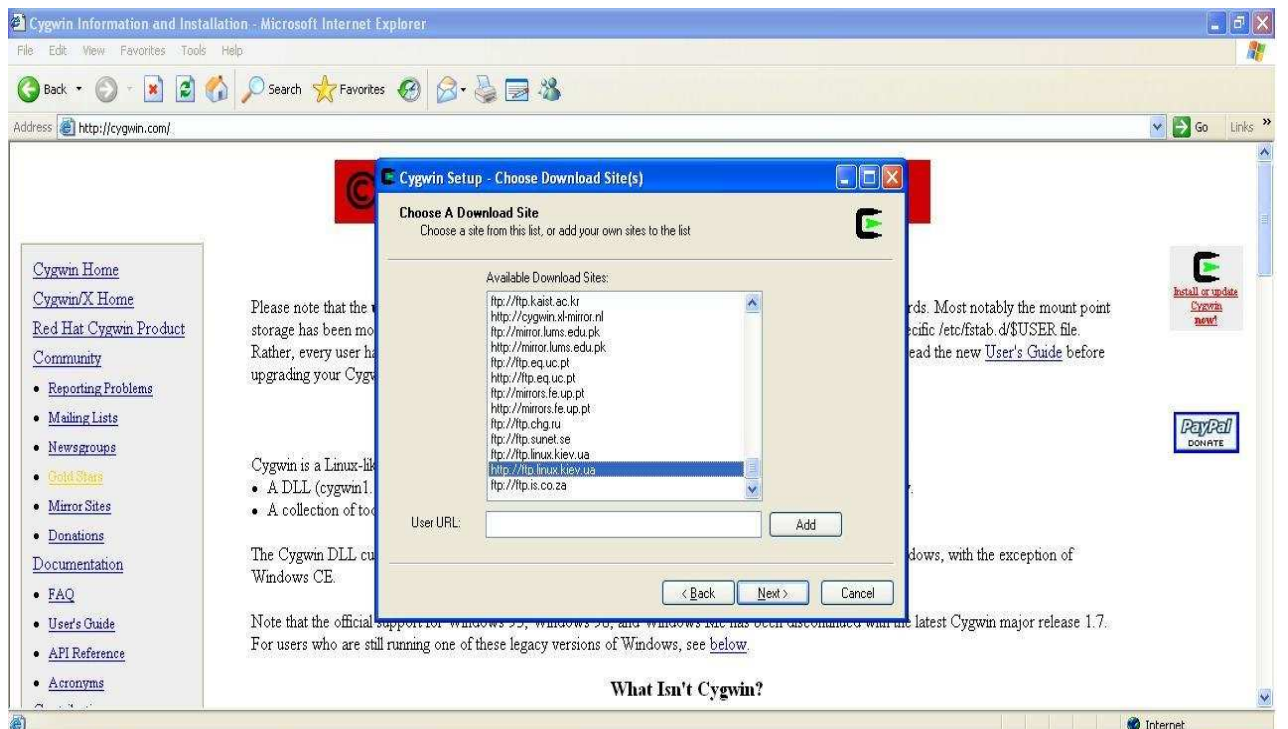
Εικόνα 38: Στο Help, εγκαθιστούμε το New Software στο μενού. Επιλέγουμε Galileo ως το χώρο ενημέρωσης ("Work with"). Αφήνουμε τα στοιχεία να τρέξουν στο Eclipse C / C ++



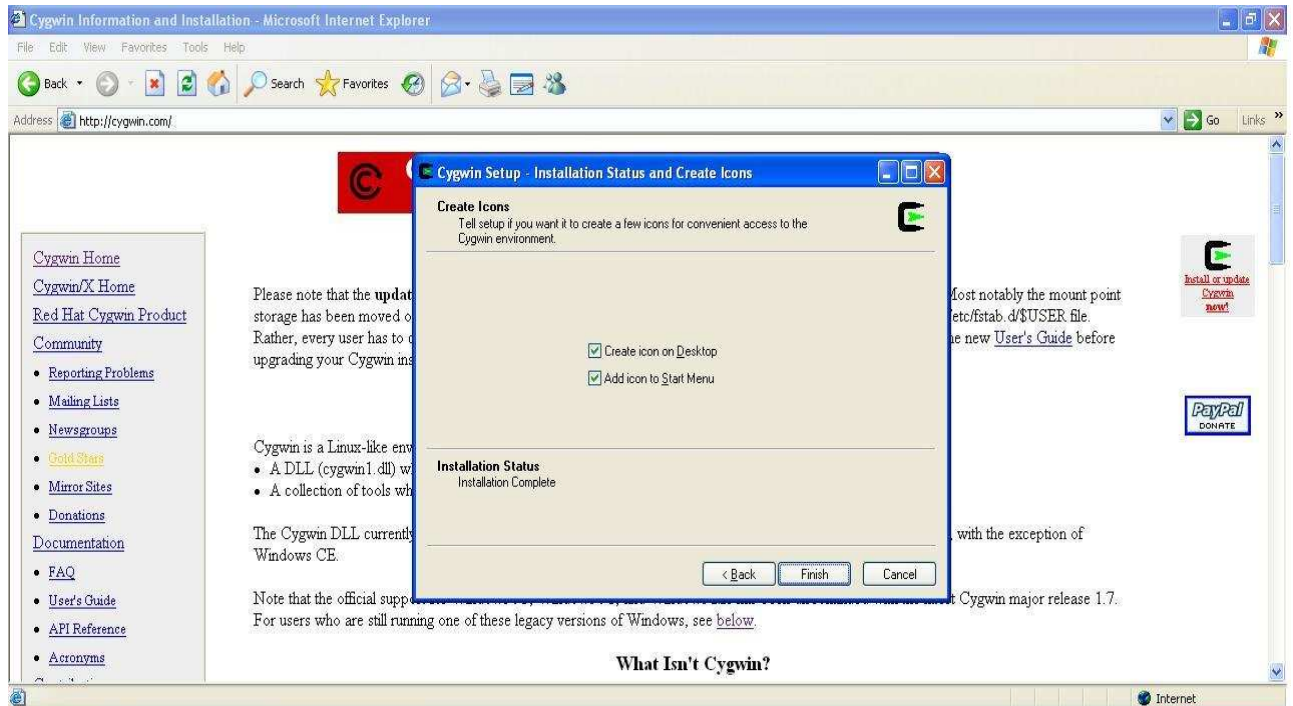
Εικόνα 39: Επιλέγουμε 'ΝΑΙ' σε όλα και περιμένουμε το Eclipse για να κάνει επανεκκίνηση. Έχουμε στο C / C ++ την υποστήριξη στο Eclipse IDE τώρα



Εικόνα 40: Cygwin: ένα σύνολο λογισμικού που προσομοιώνει Unix περιβάλλον των Windows



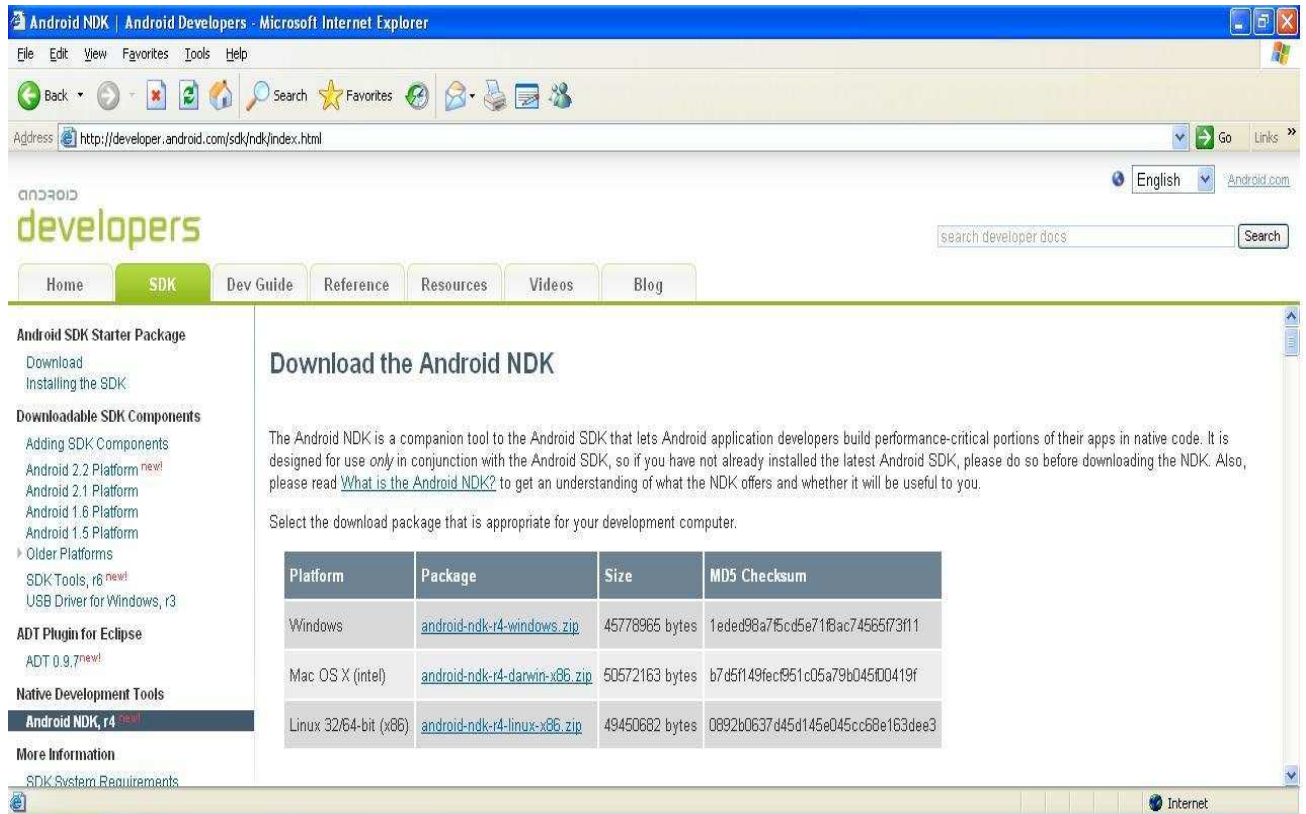
Εικόνα 41: Με την επιλογή του mirror, κάνουμε κλικ στο Next και στην συνέχεια το Cygwin θα κατεβάσει και θα παρουσιάσει τη λίστα των διαθέσιμων πακέτων



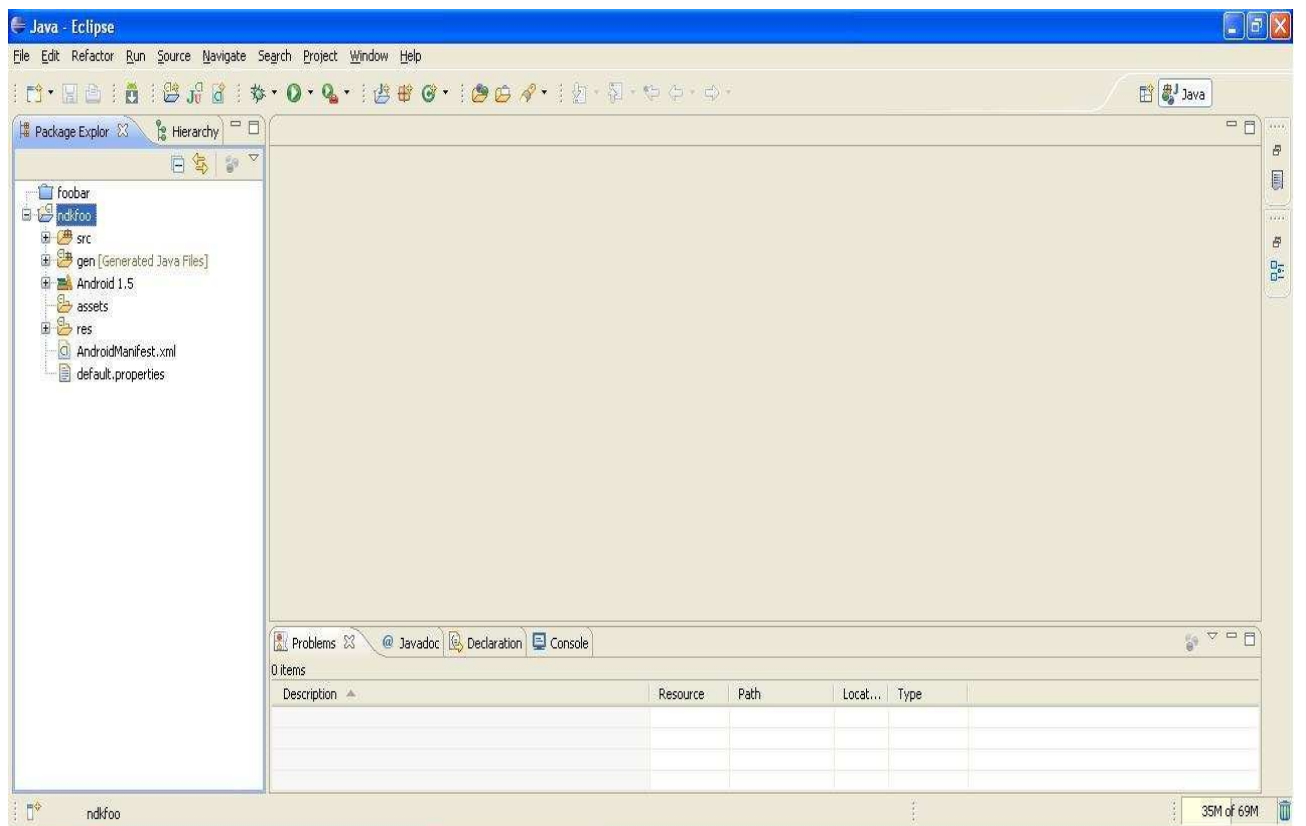
Εικόνα 42: Ολοκλήρωση εγκατάστασης



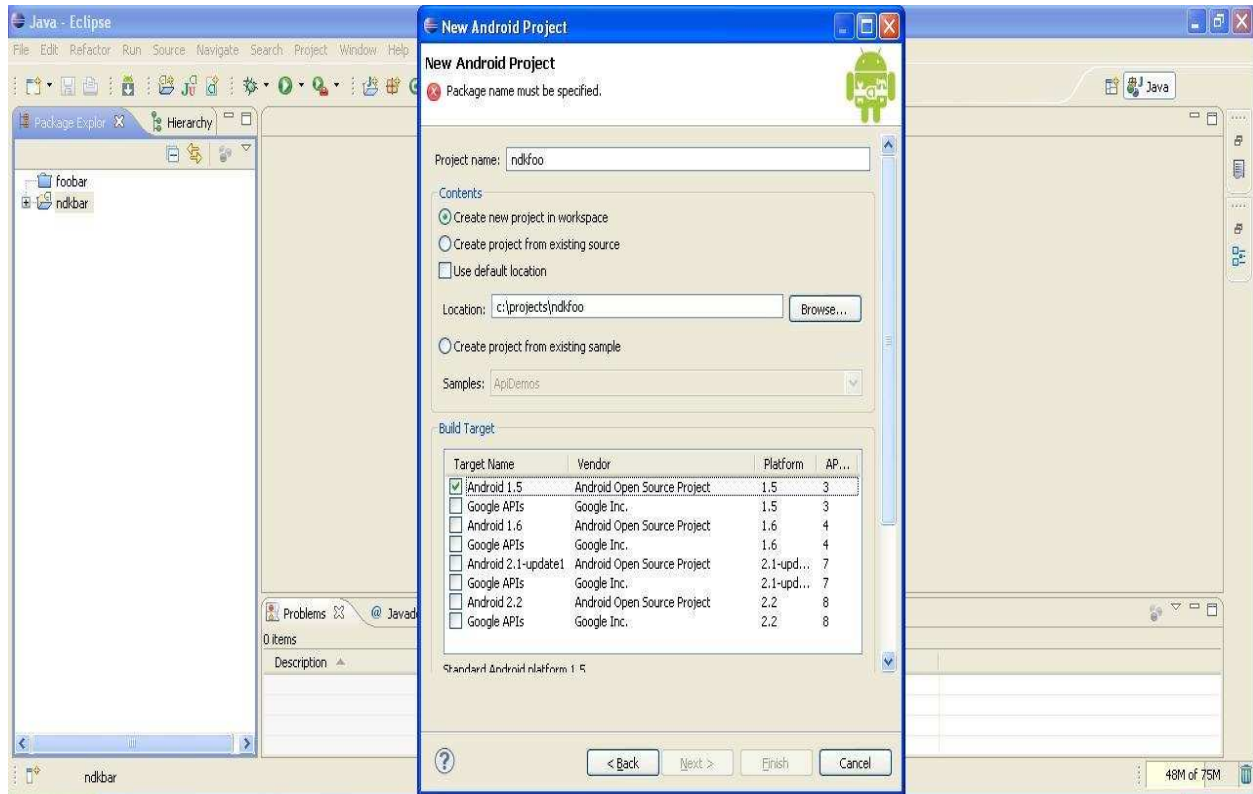
Εικόνα 43: Για τον έλεγχο των εργαλείων που είναι σημαντικά για το Android NDK πληκτρολογούμε make-n στην κονσόλα



Εικόνα 44: Εγκατάσταση του Android NDK



Εικόνα 45: Το γραφικό περιβάλλον του Eclipse



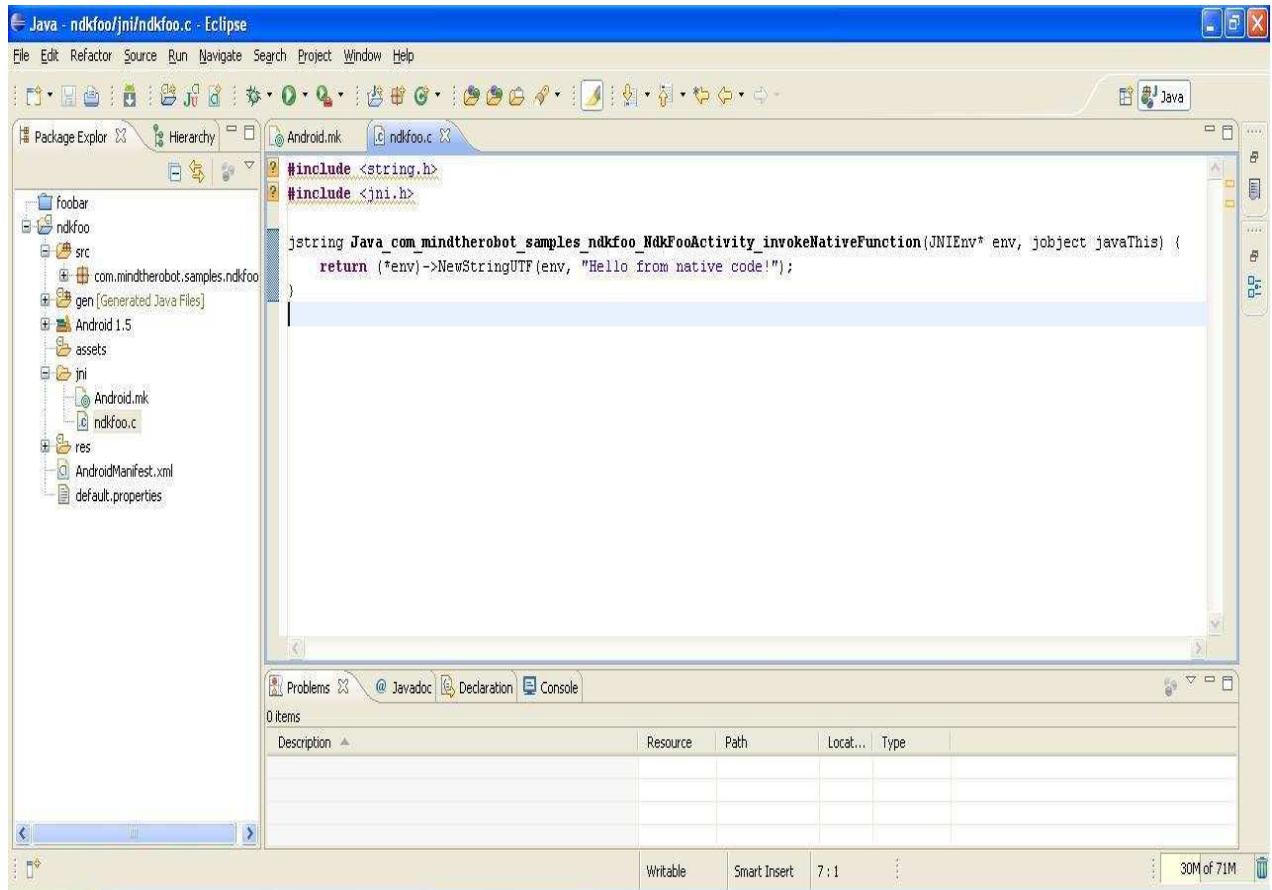
Εικόνα 46: Δημιουργία νέου project

```

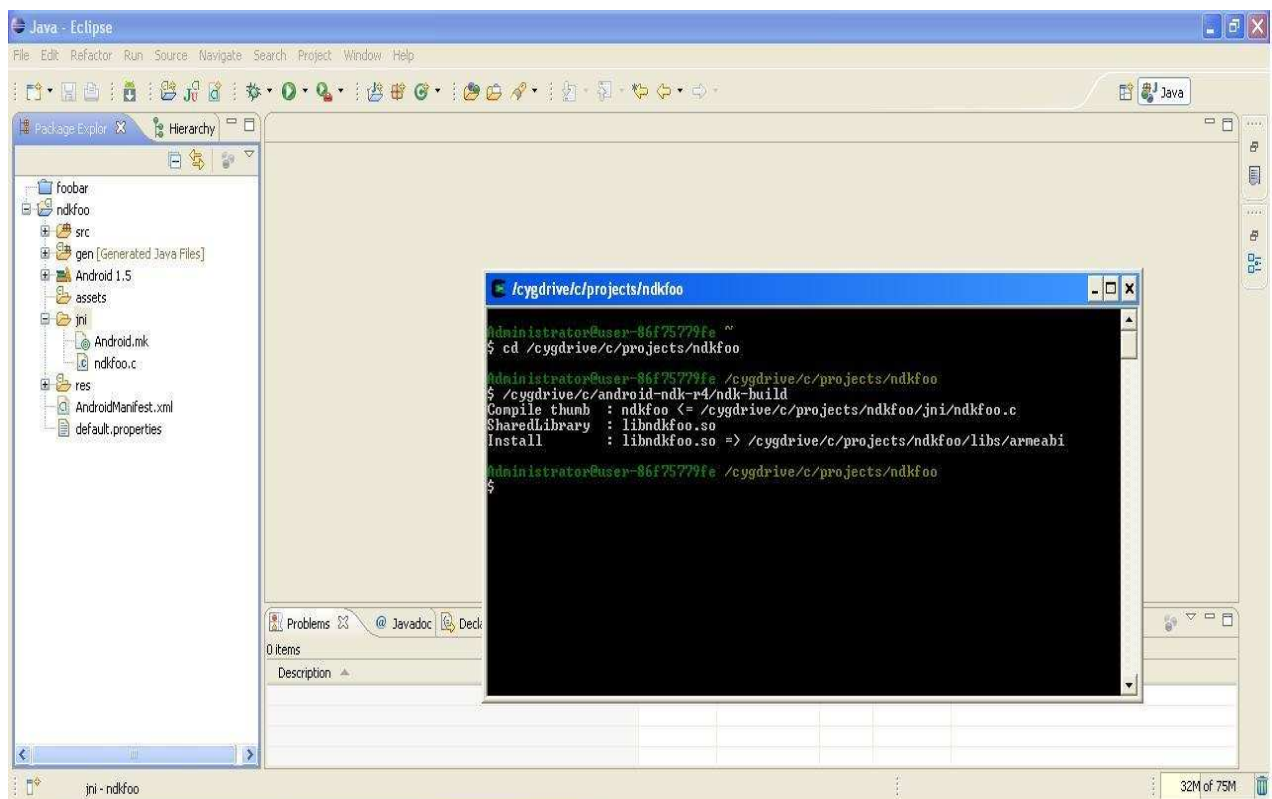
1 | LOCAL_PATH := $(call my-dir)
2 |
3 | include $(CLEAR_VARS)
4 |
5 | # Here we give our module name and source file(s)
6 | LOCAL_MODULE := ndkfoo
7 | LOCAL_SRC_FILES := ndkfoo.c
8 |
9 | include $(BUILD_SHARED_LIBRARY)

```

Εικόνα 47: Δημιουργία του φακέλου JNI στη ρίζα του νέου project



Εικόνα 48: Ο βασικός κώδικας που χρησιμοποιήθηκε για την εφαρμογή



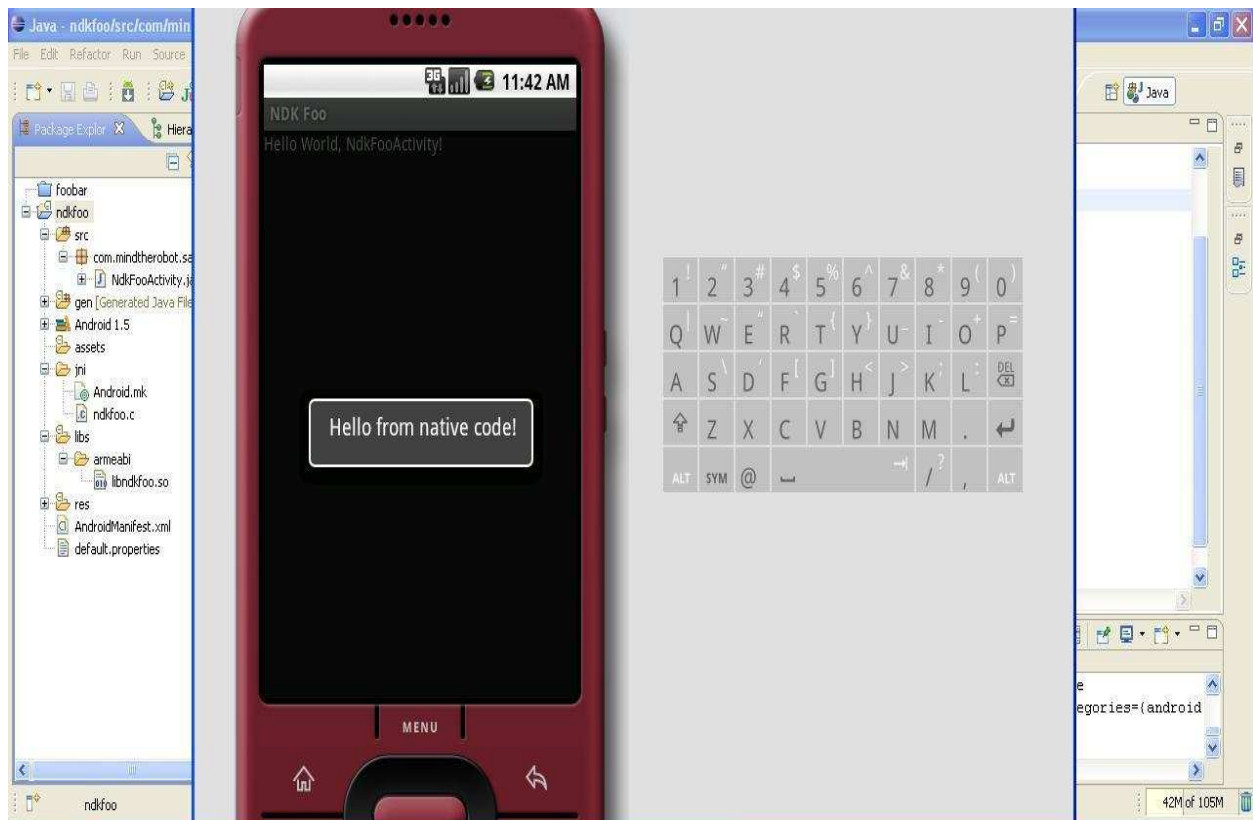
Εικόνα 49: Screenshot τρόπου ελέγχου της εφαρμογής


```

01 package com.mindtherobot.samples.ndkfoo;
02
03 import android.app.Activity;
04 import android.app.AlertDialog;
05 import android.os.Bundle;
06
07 public class NdkFooActivity extends Activity {
08
09     // load the library - name matches jni/Android.mk
10     static {
11         System.loadLibrary("ndkfoo");
12     }
13
14     // declare the native code function - must match ndkfoo.c
15     private native String invokeNativeFunction();
16
17     @Override
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.main);
21
22         // this is where we call the native code
23         String hello = invokeNativeFunction();
24
25         new AlertDialog.Builder(this).setMessage(hello).show();
26     }
27 }

```

Εικόνα 50: Πρέπει να αλλάξετε τον κώδικα της Java της τάξης NdkFooActivity για να χρησιμοποιήσετε τον κώδικα NDK



Εικόνα 51: Σε αυτό το σημείο μπορούμε να διακρίνουμε τον εξομοιωτή (emulator) του eclipse με την εφαρμογή μας. Στην περίπτωση μας το (Hello from native code)

5.3 App Inventor for Android



Εικόνα 52: Λογότυπο εφαρμογής App Inventor

Το App Inventor for Android αποτελεί ένα νέο, δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με Λειτουργικό Σύστημα Android. Είναι πολύ έξυπνη ιδέα το App Inventor, μιας και δίνει την ευκαιρία σε όποιον ενδιαφέρεται αλλά δεν έχει τις απαραίτητες γνώσεις, να φτιάξει μια εφαρμογή για Android. Είναι ουσιαστικά το μοναδικό δωρεάν εργαλείο για εύκολη ανάπτυξη εφαρμογών. Το μόνο αρνητικό χαρακτηριστικό του είναι ότι, κατά τον προγραμματισμό, η διαδικασία χρήσης των blocks είναι μια αργή διαδικασία.

5.3.1 Η ανάπτυξη της εφαρμογής

Οι συγκεκριμένες εφαρμογές τρέχουν και σε emulator. Αναπτύχθηκε στα εργαστήρια της Google από μια ομάδα με επικεφαλής τον καθηγητή του MIT Hal Abelson (Abelson, 2009). Το App Inventor χρησιμοποιείται και δοκιμάζεται ήδη ως πλατφόρμα διδασκαλίας και εισαγωγής στον Προγραμματισμό, τόσο στην Τριτοβάθμια, όσο και στη σχολική εκπαίδευση. Το περιβάλλον του App Inventor έχει πολλές ομοιότητες με το περιβάλλον του Scratch και του Alice, με τη διαφορά ότι οι εφαρμογές που δημιουργούνται τρέχουν σε έξυπνα τηλέφωνα (smart phones).

Σε πρόσφατη έρευνα η οποία πραγματοποιήθηκε από την οργάνωση Ανθρωπιστική δράση με Δωρεάν Λογισμικό Ανοικτού Κώδικα (Humanitarian Free and Open Source Software –HFOSS) και χρηματοδοτήθηκε από το Εθνικό Ίδρυμα Επιστημών των Ηνωμένων Πολιτειών (National Science Foundation) έγινε προσπάθεια να απαντηθεί το ερώτημα αν το app Inventor είναι κατάλληλο για τη διδασκαλία του προγραμματισμού και ειδικότερα της υπολογιστικής σκέψης στη σχολική εκπαίδευση (Morreli et al. 2011).

5.3.2 Τα αρχικά πλεονεκτήματα του App Inventor

Τα πρώτα αποτελέσματα είναι ενθαρρυντικά αφού οι ερευνητές επισημαίνουν τα εξής πλεονεκτήματα:

- Εύκολο στη χρήση περιβάλλον με πολλές δυνατότητες
- Αντικειμενοστραφές μοντέλο οπτικού προγραμματισμού με δομές ελέγχου καθοδηγούμενες από γεγονότα (event-driven)
- Μάθηση μέσω της λύσης προβλημάτων
- Επιπλέον κίνητρα στους μαθητές σε σχέση με το Scratch και Alice, εξαιτίας της φορητότητας και της πρακτικής χρήσης των εφαρμογών που δημιουργούνται
- Ύπαρξη emulator που σημαίνει ότι δεν χρειάζονται πολλές συσκευές για την εισαγωγή στη σχολική τάξη
- Υποστήριξη από τη Google

Η διαφορά σε σχέση με το Android SDK είναι ότι δεν χρειάζονται προγραμματιστικές γνώσεις ή εμπειρία για την δημιουργία του. Σύμφωνα με την Google, ο καθένας μπορεί να αναπτύξει εφαρμογές με το App Inventor, από τους επαγγελματίες προγραμματιστές έως και τους μαθητές, στο επίπεδο μίας σχολικής αίθουσας.

5.3.3 Τι είναι το App Inventor

Το App Inventor είναι μια WYSIWYG (What You See Is What You Get - "Αυτό που βλέπεις, είναι αυτό παίρνεις") λύση της Google, που επιτρέπει τη δημιουργία εφαρμογών για το δημοφιλές λειτουργικό της Android, μέσω απλού drag'n'drop. Θεωρητικά, ακόμη και ένας αρχάριος με μηδενικές γνώσεις προγραμματισμού, μπορεί μέσω του App Inventor μέσα σε λιγότερο από μια ώρα να έχει δημιουργήσει τη δική του εφαρμογή για το Android. Πρακτικά, δεν χρειάζεται καν να έχει κάποιος ένα smartphone ή tablet με αυτό το λειτουργικό σύστημα, αφού μπορεί να βασιστεί σε μια "εξομοιωμένη" εκδοχή του για να δει τους... καρπούς της προσπάθειας του.

Όπως χαρακτηριστικά αναφέρει η Google, «...για πολλούς ανθρώπους, η κινητή τους συσκευή και η πρόσβαση που επιτυγχάνεται στο Διαδίκτυο μέσω αυτής, έχει διάφορους περιορισμούς. Το App Inventor για Android δίνει σε όλους, ανεξαρτήτως προγραμματιστικών ικανοτήτων, την ευκαιρία να ελέγχουν και να μετασχηματίζουν την εμπειρία της επικοινωνίας τους. Παρατηρούμε μάλιστα πως οι άνθρωποι αισθάνονται υπερήφανοι με το να γίνονται δημιουργοί της κινητής τεχνολογίας και όχι απλοί χρήστες της».

Με το App Inventor, οι χρήστες μπορούν να δημιουργήσουν πρακτικά όποια εφαρμογή θέλουν: από ένα απλό παιχνίδι έως εφαρμογές που θα χρησιμοποιούν τους αισθητήρες της συσκευής (εγγύτητας, προσανατολισμού κ.ά.). Επιπλέον, είναι δυνατή και η δημιουργία εφαρμογών εκμάθησης, όπως η δημιουργία ενός κουίζ ερωτήσεων ή η εκμετάλλευση της τεχνολογίας text-to-speech για εκφώνηση διαφόρων εντολών από τη συσκευή.

Όπως ήδη αναφέρθηκε, για την χρήση της εφαρμογής δεν είναι απαραίτητες οι γνώσεις προγραμματισμού, μιας και αντί για τη συγγραφή κώδικα ο δημιουργός αποφασίζει για την εμφάνιση και τη συμπεριφορά της εφαρμογής μέσα από ένα καλαίσθητο παραθυρικό περιβάλλον. Μάλιστα, οι δημιουργοί της πρόσθεσαν διάφορα blocks εντολών, τα οποία «ενεργοποιούνται» πατώντας το αντίστοιχο πλήκτρο και εκτελούν προκαθορισμένες λειτουργίες. Για παράδειγμα, υπάρχουν blocks επανάληψης εντολών τύπου for, do-while κτλ, blocks αποθήκευσης πληροφοριών, καθώς και blocks κώδικα για την εκτέλεση λειτουργιών κάτω από συγκεκριμένες συνθήκες. Επίσης, είναι διαθέσιμα ακόμη και blocks για επικοινωνία με τοποθεσίες κοινωνικής δικτύωσης, όπως π.χ. το Twitter.

Άλλες ενδιαφέρουσες πτυχές του App Inventor είναι η συνεργασία με τους δέκτες GPS, ώστε να «χτιστούν» εφαρμογές που θα χρησιμοποιούν πληροφορίες εντοπισμού θέσης. Ακόμη υπάρχουν αρκετές άλλες εφαρμογές, ίσως πιο πρακτικής φύσεως, οι οποίες εκμεταλλεύονται ή υποβοηθούν τις «καθημερινές» λειτουργίες ενός κινητού τηλεφώνου.

Καταλήγοντας, να σημειώσουμε πως η εφαρμογή App Inventor για Android κάνει χρήση των Java βιβλιοθηκών Open Blocks για την δημιουργία των εικονικών blocks, ενώ ο μεταφραστής της γλώσσας των visual blocks (η οποία ενσωματώθηκε και ρυθμίστηκε ειδικά για το Android) χρησιμοποιεί το Kawa Language Framework.

Εάν κάποιος θέλει να φτιάξει τη δική του Android εφαρμογή χωρίς να αναλωθεί στην συγγραφή εκατοντάδων γραμμών κώδικα, η Google προτείνει το εύχρηστο περιβάλλον ανάπτυξης εφαρμογών App Inventor, για δημιουργία applications από τον καθένα. Πρόκειται για μια πληρέστατη συλλογή εργαλείων που επιτρέπουν σε οποιονδήποτε, δίχως προγραμματιστικό υπόβαθρο, να δημιουργήσει προγράμματα για τα κινητά που τρέχουν Android.

Το «App Inventor» (Εφευρέτης Εφαρμογών) απεικονίζει τις επιμέρους λειτουργίες του τηλεφώνου ως χρωματιστά τουβλάκια, τα οποία οι χρήστες μπορούν να σέρνουν και να συνδέουν πάνω στην οθόνη του υπολογιστή τους. Οι μαθητευόμενοι προγραμματιστές μπορούν έτσι να δημιουργούν παζλ και άλλα παιχνίδια ή προγράμματα που αξιοποιούν ταυτόχρονα πολλές επιμέρους δυνατότητες του Android.

Για παράδειγμα, ένας από τους εθελοντές που δοκίμασαν το App Inventor ανέπτυξε μια εφαρμογή που ενημερώνει τους φίλους του για το πού ακριβώς βρίσκεται μέσω της λειτουργίας GPS. Ακόμα και μαθητές δημοτικού μπόρεσαν να αξιοποιήσουν τα νέα εργαλεία στις δοκιμές που διεξήγαγε η Google εδώ και ενάμιση χρόνο. Το App Inventor βασίστηκε στο Open Blocks, ένα σύστημα προγραμματισμού που αναπτύχθηκε για εκπαιδευτικούς λόγους στο MIT. Το σύστημα περιλαμβάνει μια βιβλιοθήκη κώδικα Java και τη γλώσσα προγραμματισμού Scratch. Η συγκεκριμένη πρόταση αναμένεται να οδηγήσει σε αύξηση των διαθέσιμων εφαρμογών για το Android, το λειτουργικό σύστημα για κινητά που προωθεί η Google.

Να σημειώσουμε πως το App Inventor δεν μπορεί, βέβαια, να δημιουργήσει με λίγα κλικ λογισμικό επαγγελματικού επιπέδου που θα χρειαζόταν χιλιάδες γραμμές κώδικα, αλλά είναι σίγουρα μια καλή αρχή για όποιον θέλει να εξοικειωθεί με τον προγραμματισμό για κινητά, ενώ μπορεί κάλλιστα να χρησιμοποιηθεί και για "prototyping", δηλαδή για "γρήγορο στήσιμο προσχεδίων εφαρμογών" τις οποίες, στη συνέχεια, μπορεί κανείς να συνεχίσει να αναπτύσσει με πρόσθετο κώδικα.



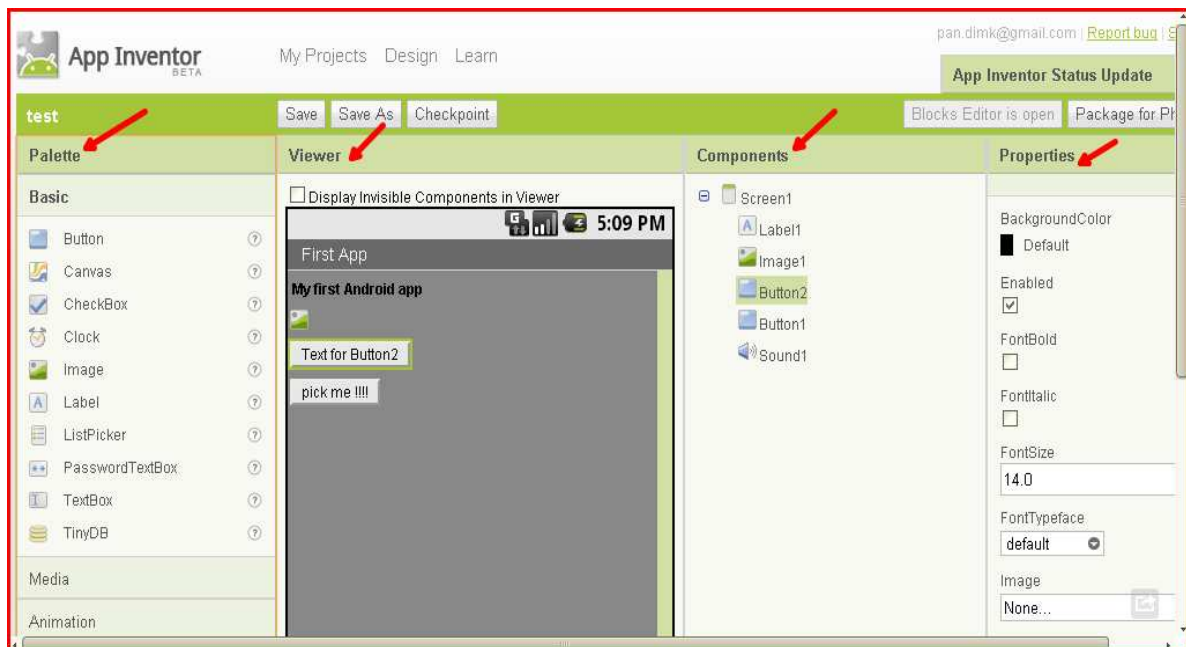
Πλέον είναι αρκετά εύκολη η δημιουργία εφαρμογών για το κινητό. Η Google με το App Inventor, δεν μπορούσε να κάνει τα πράγματα πιο εύκολα για όλους αυτούς που δε θέλουν να ακολουθήσουν τον δρόμο της Java, για να φτιάξουν μια απλή εφαρμογή.

Με το App Inventor, ο χρήστης απλά ενώνει blocks με κώδικα, έτσι ώστε να σχηματιστεί η λογική του προγράμματος που θέλει να φτιάξει. Είναι ο προγραμματισμός στη πιο απλή και

κατανοητή μορφή του. Όλες οι δυνατότητες είναι μπροστά του και ο χρήστης απλά καλείται να επιλέξει και να ενώσει τα κομμάτια που τον ενδιαφέρουν. Κάτι τέτοιο συμβαίνει στο μυαλό του κάθε προγραμματιστή όταν φτιάχνει ένα πρόγραμμα, αλλά χάρις σε αυτό το εργαλείο, αυτή η διαδικασία γίνεται στην οθόνη του καθενός.

Αυτός ο τρόπος προγραμματισμού δεν είναι καινούργιος. Παραπάνω, είχε αναφερθεί το Scratch, μια γλώσσα προγραμματισμού για αρχάριους, που επίσης χρησιμοποιεί blocks κώδικα. Εκνευριστικός τρόπος για να προγραμματίζει ένας προγραμματιστής, αλλά εύκολος για κάποιον αρχάριο.

Όπως στα περισσότερα εργαλεία ανάπτυξης εφαρμογών με γραφικό περιβάλλον, πρώτα σχεδιάζουμε το γραφικό περιβάλλον και μετά τον κώδικα που ανταποκρίνεται στο κάθε συμβάν. Η οθόνη χωρίζεται στην palette, όπου βρίσκονται όλα τα διαθέσιμα controls που μπορεί να έχει η εφαρμογή μας, στον viewer, που αντιπροσωπεύει την εμφάνιση της εφαρμογής μας, τα components που είναι μια λίστα με τα controls που χρησιμοποιούμε και τέλος τα properties, που είναι οι ιδιότητες του κάθε control [Εικόνα 45].



Εικόνα 53: Γραφικό περιβάλλον App Inventor

Μόλις φτιάξουμε το γραφικό περιβάλλον που επιθυμούμε, τότε καλείται ο blocks editor, μια java εφαρμογή που μας επιτρέπει να "κολλήσουμε" μαζί blocks με κώδικα. Το κάθε control έχει τα δικά του blocks κώδικα, ανάλογα με την λειτουργία του. Έτσι με ευκολία έχουμε πρόσβαση στις κυριότερες λειτουργίες της συσκευής μας.

Η δοκιμή της εφαρμογής μπορεί να γίνει είτε στον emulator που υπάρχει, είτε απευθείας στην συσκευή. Συγκριτικά, η καλύτερη λύση είναι οι δοκιμές να γίνονται σε μια συσκευή, καθώς ο emulator είναι κάπως αργός.

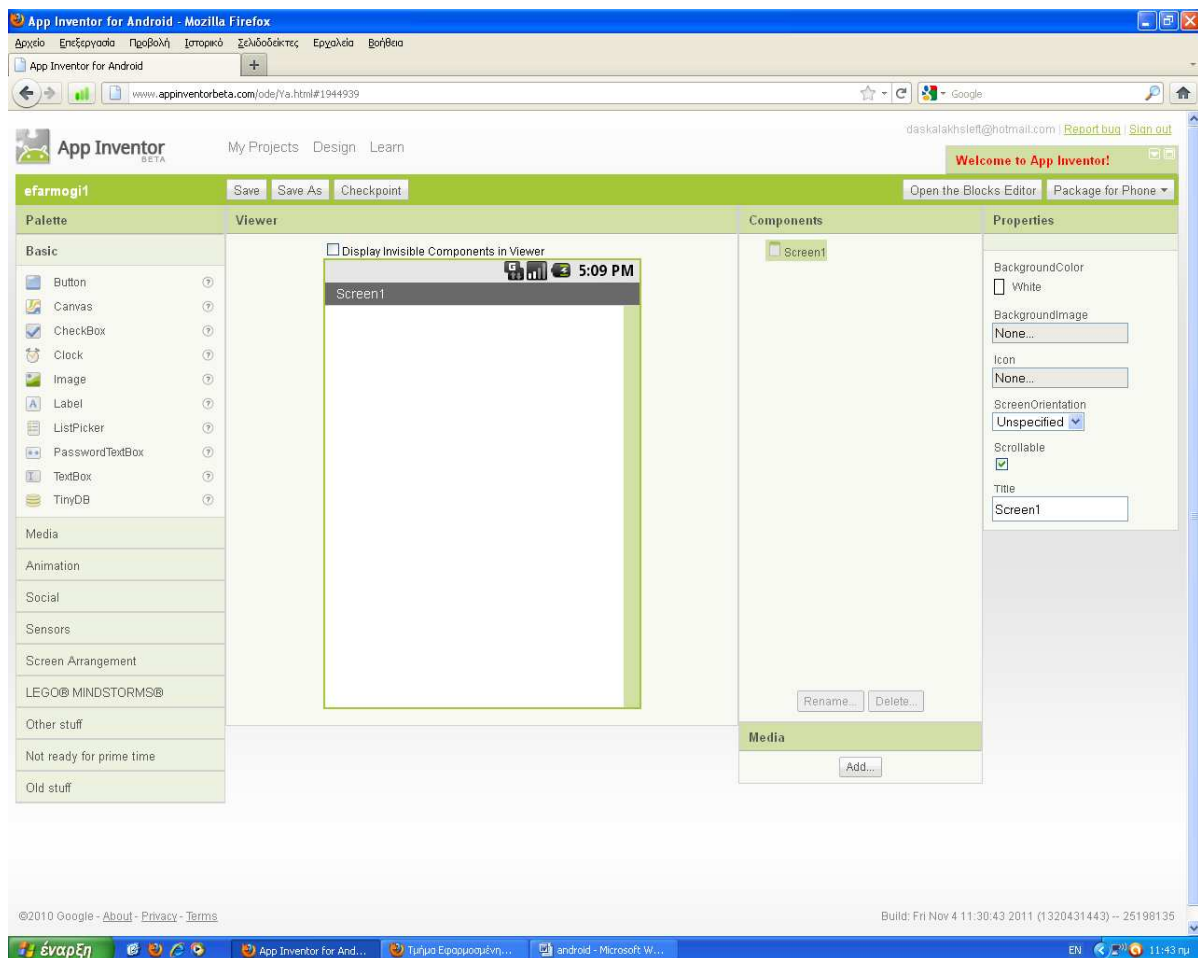
Είναι πολύ έξυπνη ιδέα το App Inventor, μιας και δίνει την ευκαιρία σε όποιον ενδιαφέρεται αλλά δεν έχει τις απαραίτητες γνώσεις, να φτιάξει μια εφαρμογή για Android. Είναι ουσιαστικά το μοναδικό δωρεάν εργαλείο για εύκολη ανάπτυξη εφαρμογών.

5.3.4 Μια γρήγορη εφαρμογή για το App Inventor

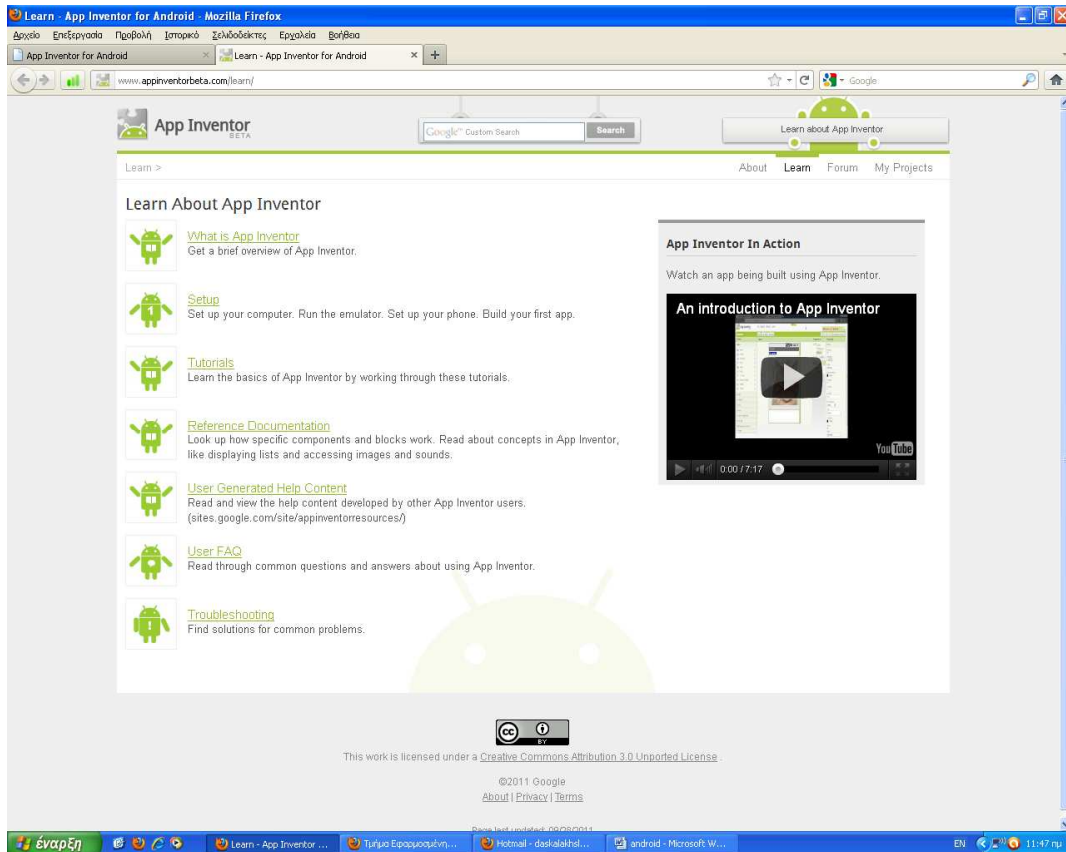
Σε αυτό το σημείο και αφού έχουμε αναπτύξει και εξηγήσει την λειτουργία του App Inventor το μόνο που απομένει είναι να δημιουργήσουμε μια δική μας εφαρμογή.

Για να το πετύχουμε αυτό, τα βήματα που πρέπει να ακολουθήσουμε είναι πολύ απλά. Το πρώτο βήμα είναι να δημιουργήσουμε ένα δικό μας λογαριασμό στην Google, κάτι το οποίο το πετύχουμε πηγαίνοντας στην σελίδα www.appinventorbeta.com [12], η οποία μας ανακατευθύνει στην κεντρική οθόνη του App Inventor. Το δεύτερο βήμα είναι να επιλέξουμε την επιλογή learn, από την γραμμή εργαλείων, εκεί όπου βρίσκονται γενικές πληροφορίες. Εμείς επιλέγουμε το setup και μόλις ανοίξει αυτή η πλατφόρμα, υπάρχουν οδηγίες ακριβώς ως προς το τι απαιτείται να είναι εγκατεστημένο έτσι ώστε να μπορούμε να τρέξουμε το πρόγραμμα μας. Το τρίτο βήμα είναι να εγκαταστήσουμε το λειτουργικό στον υπολογιστή μας. Στην περίπτωση μας, εφόσον δουλεύουμε σε Windows, δεν έχουμε παρά να επιλέξουμε το instructions for windows.

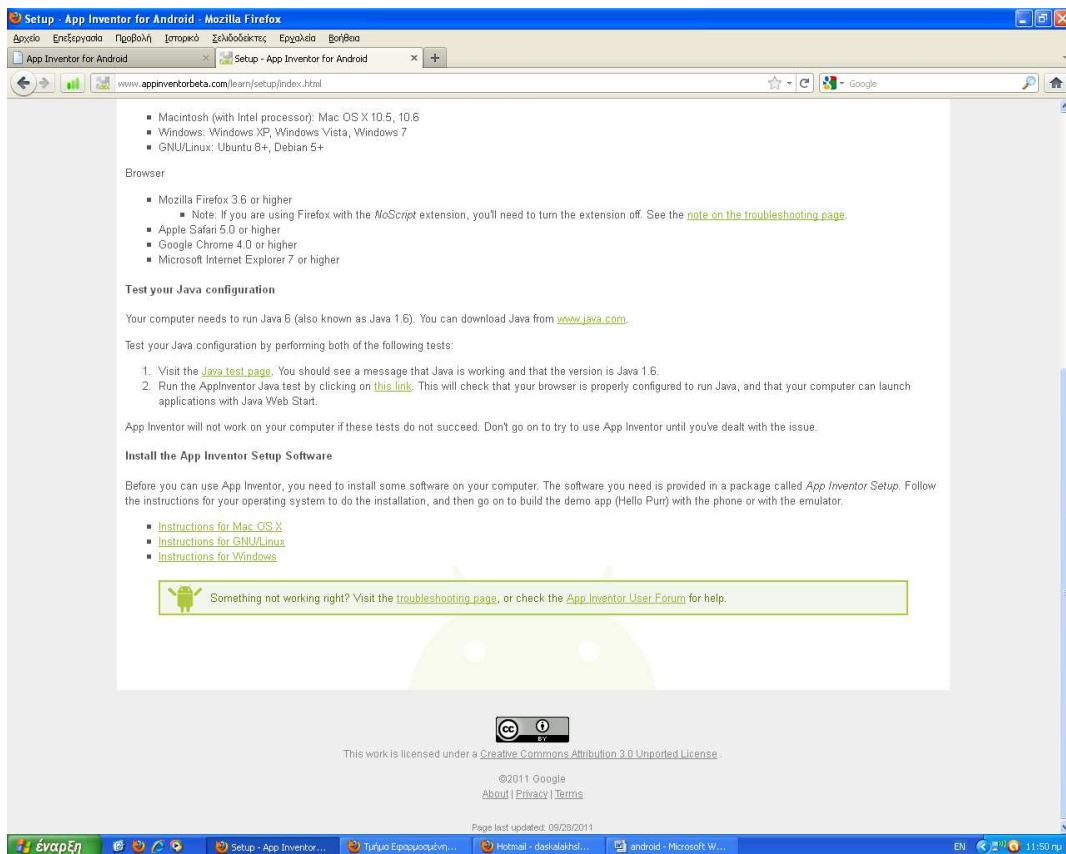
Μόλις πραγματοποιήσουμε όλα τα παραπάνω δεν έχουμε παρά να πάμε στο App Inventor και να δημιουργήσουμε το δικό μας project κάτι το οποίο φαίνεται στην παρακάτω εικόνα.



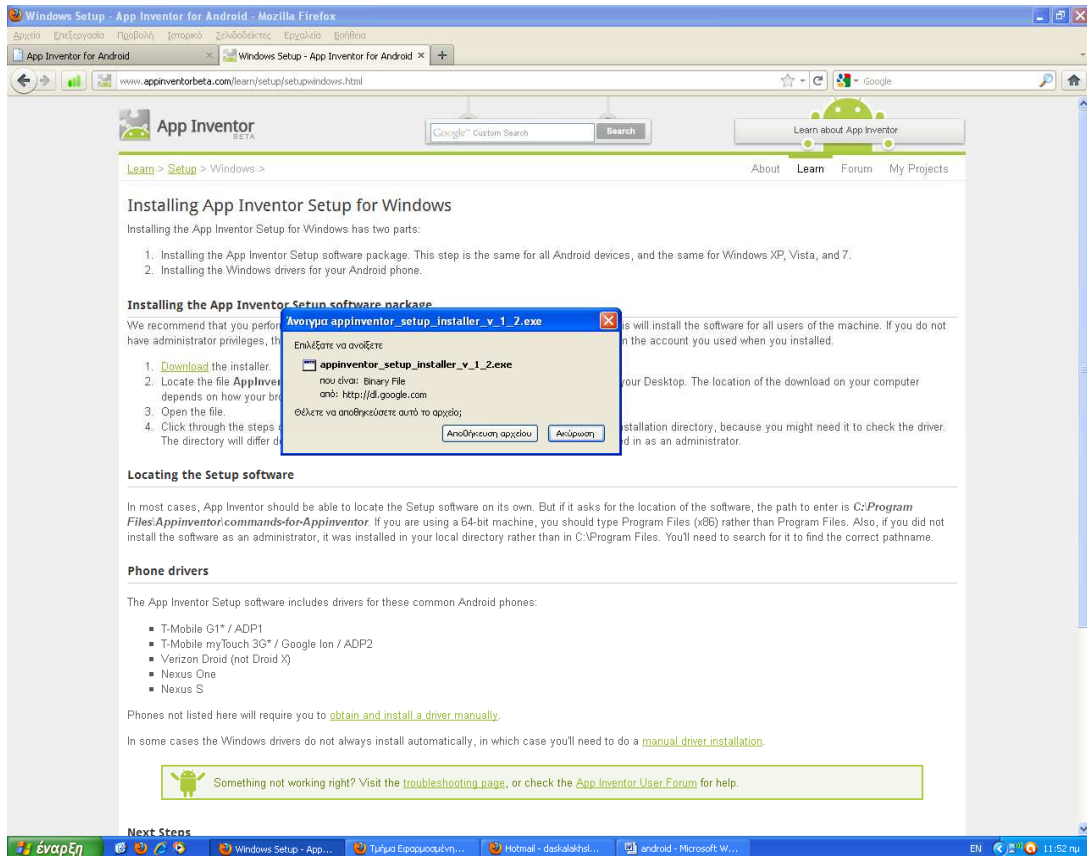
Εικόνα 54: Κόρυα οθόνη του App Inventor



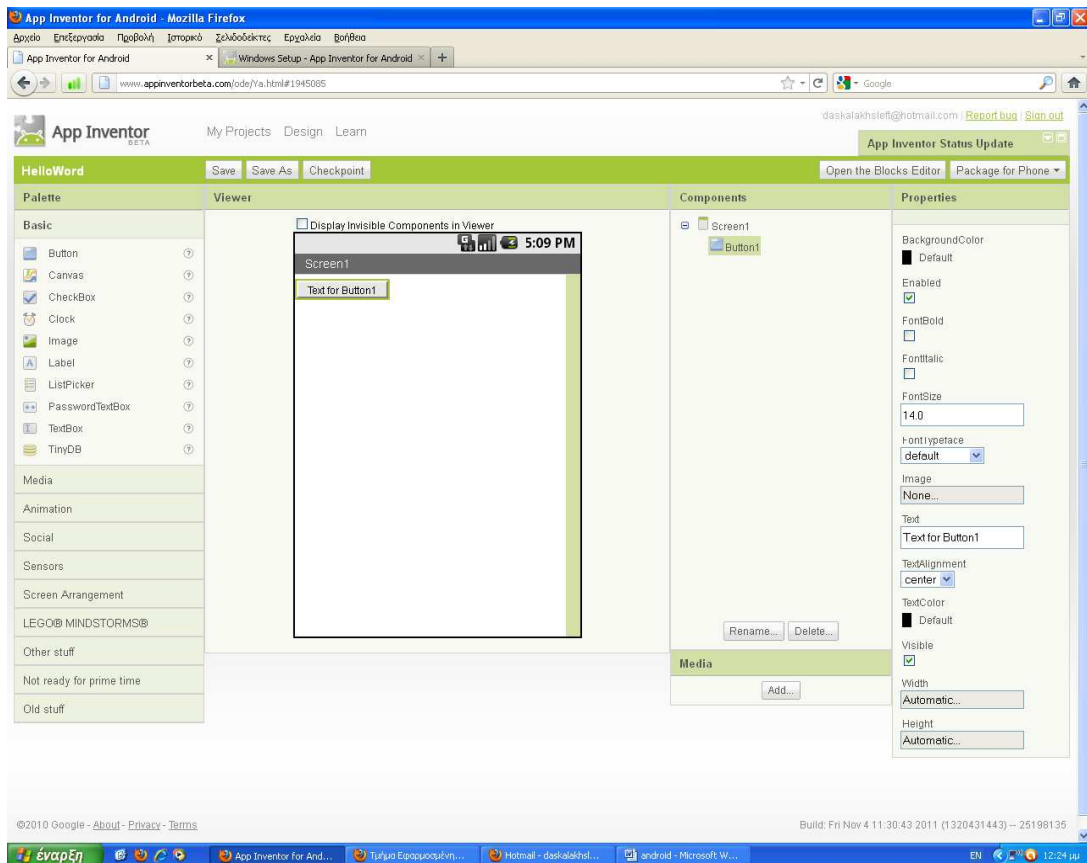
Εικόνα 55: Έχουμε επιλέξει το learn σε δεύτερο παράθυρο



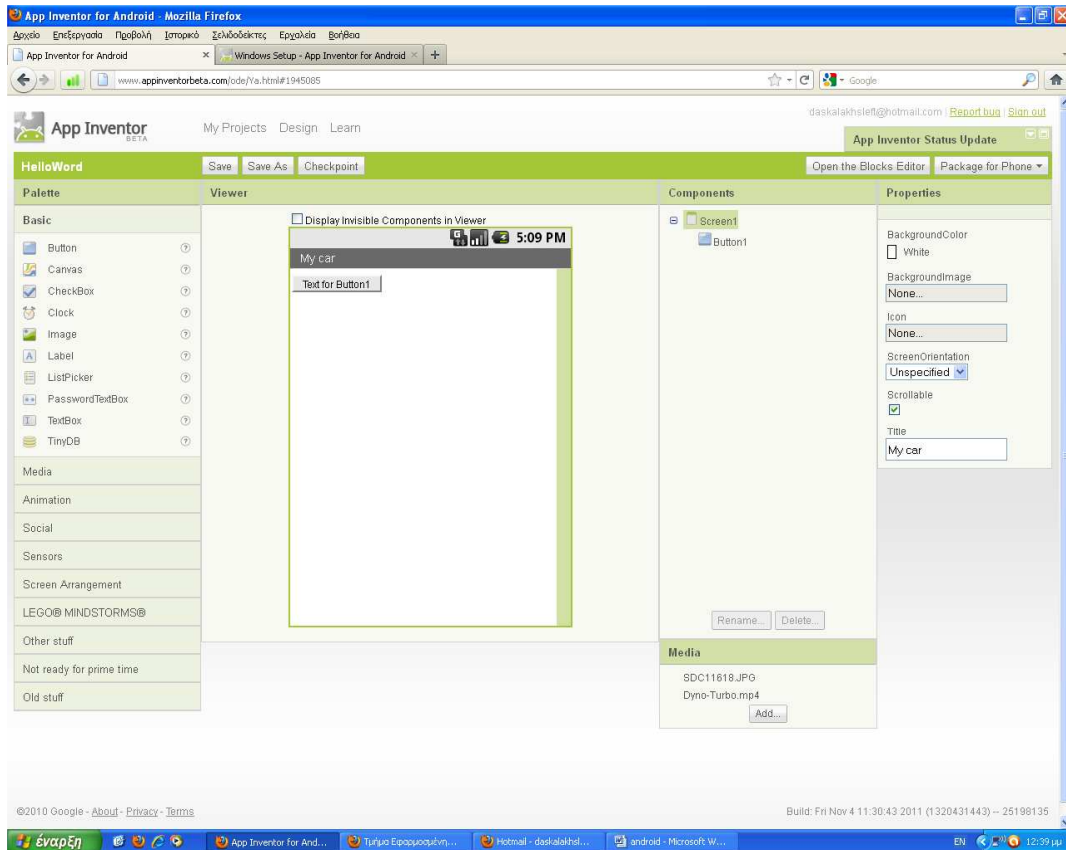
Εικόνα 56: Εγκατάσταση λογισμικού που χρειαζόμαστε. Στην περίπτωση μας instructions for Windows



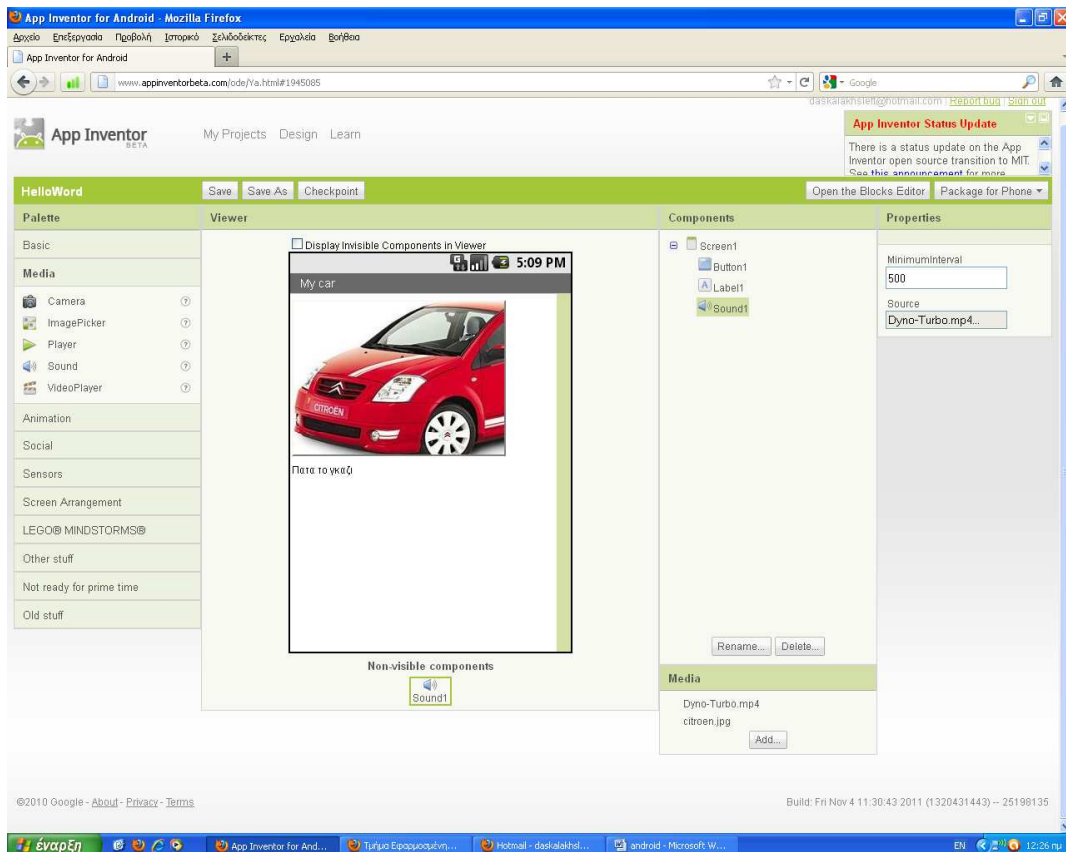
Εικόνα 57: Εγκατάσταση και εκτέλεση εφαρμογής



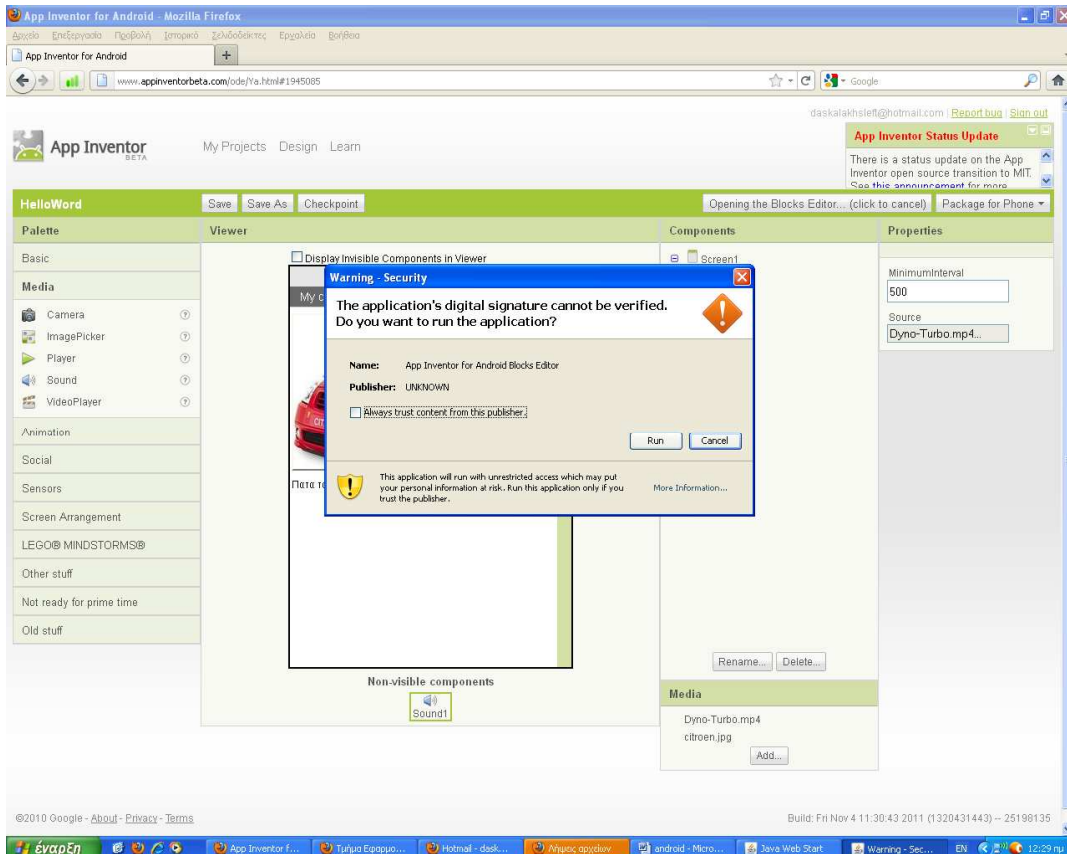
Εικόνα 58: Προσθέτουμε το πρώτο Button



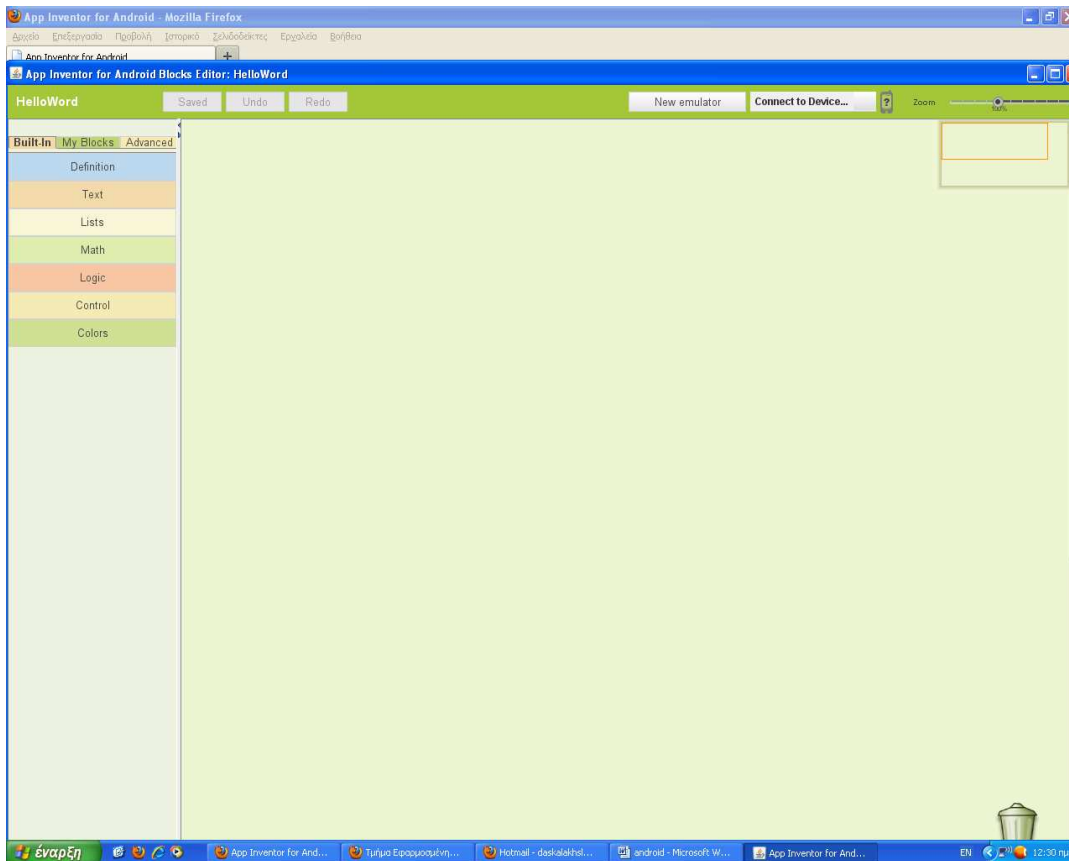
Εικόνα 59: Σε αυτό το σημείο έχουμε προσθέσει στο components την φωτογραφία και τον ήχο που θα χρησιμοποιήσουμε



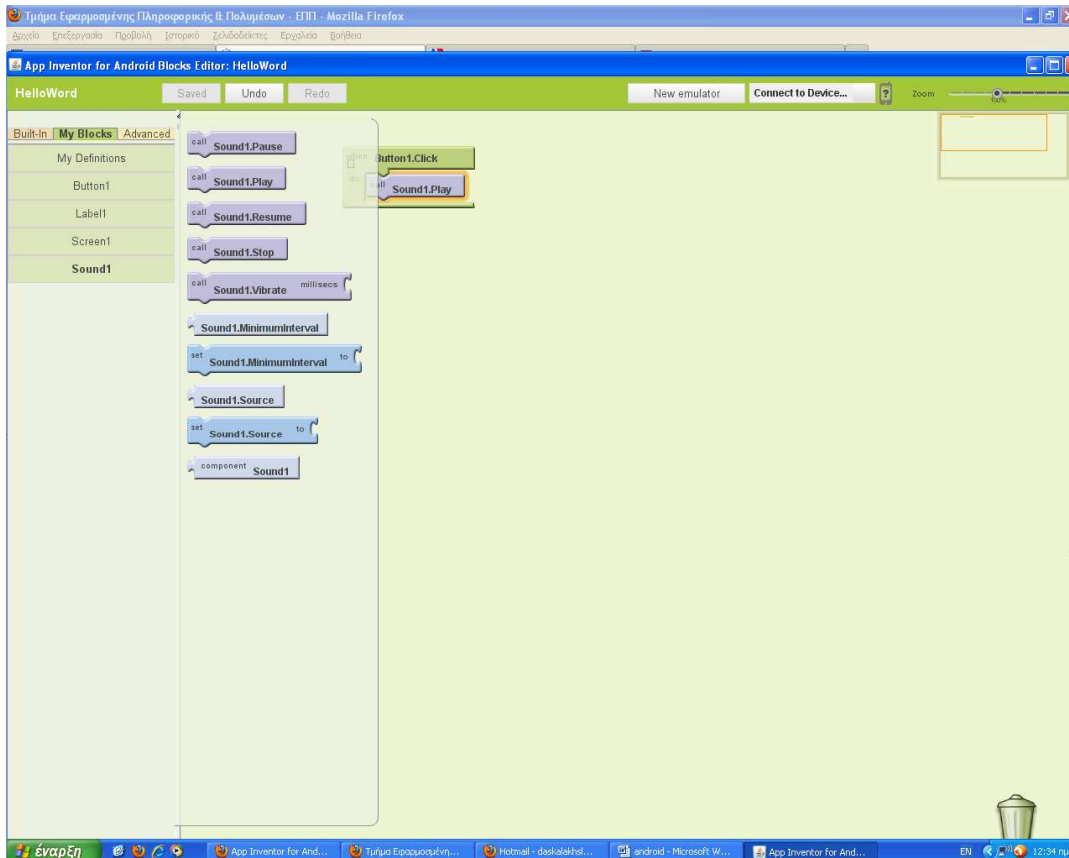
Εικόνα 60: Εδώ διακρίνουμε καθαρά τον ήχο που έχουμε προσθέσει



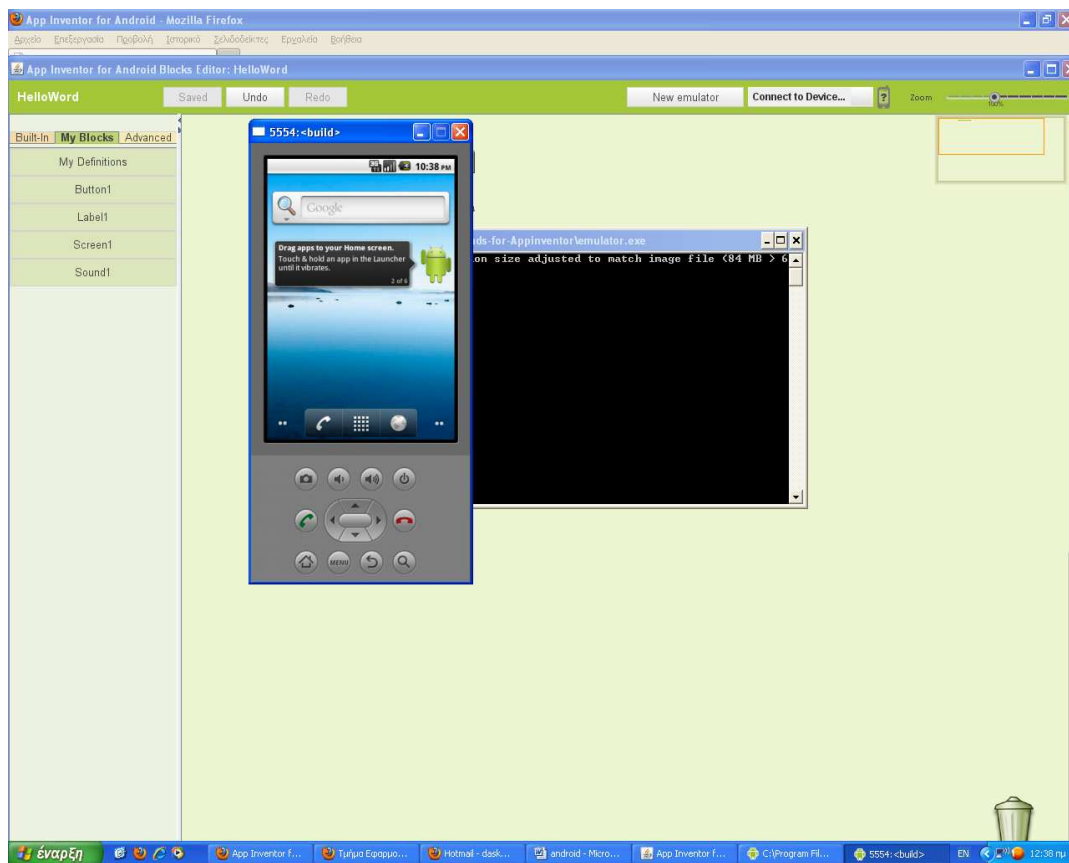
Εικόνα 61: Τρέχουμε την εφαρμογή μας και ταυτόχρονα εγκαθιστούμε τον κώδικα που χρειαζόμαστε



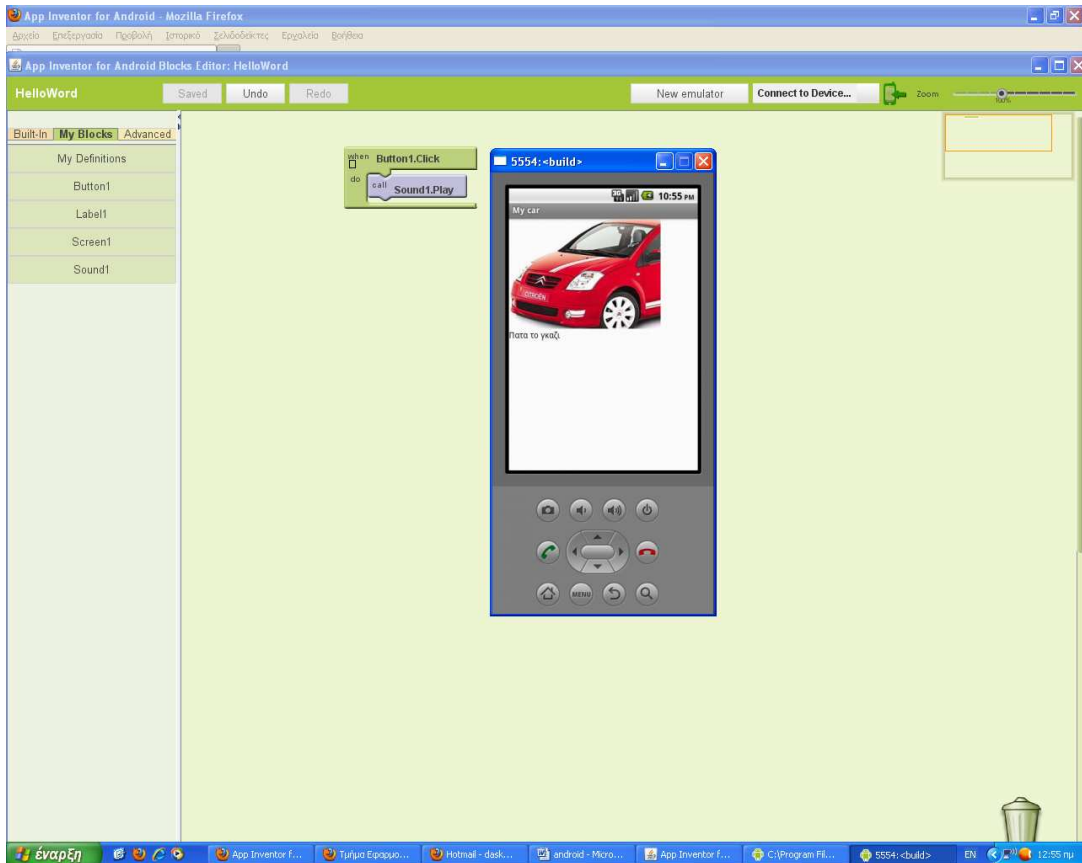
Εικόνα 62: Εδώ μπορούμε να διακρίνουμε το block editor του App Inventor



Εικόνα 63: Έχουμε ανοίξει το My block, προσθέτουμε το button και τον ήχο μέσα σε αυτό

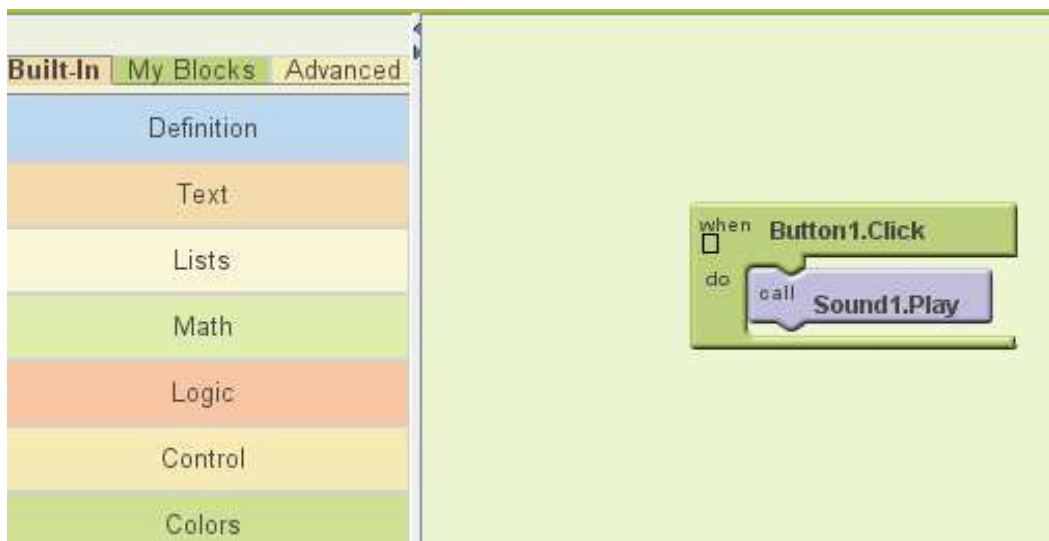


Εικόνα 64: Επιλέγουμε το Connect Device και στην συνέχεια τρέχουμε το new emulator



Εικόνα 65: Με τον εξομοίωτη (emulator), όπως μπορούμε να διακρίνουμε, εκτελούμε την εφαρμογή μας

Σε αυτό το σημείο να τονιστεί πως με το πάτημα πάνω στην εικόνα του αυτοκινήτου ακούγεται ένας ήχος, στο παράδειγμα μας το γκάζι από ένα αυτοκίνητο.



Εικόνα 66: Ο κώδικας μας στην προκειμένη περίπτωση

5.4 Hyper Next Android Creator



Εικόνα 67: Λογότυπο εφαρμογής HyperNext Android

Το HyperNext Android Creator (HAC) είναι ένα ισχυρό αλλά και εύκολο στη χρήση λογισμικό συστήματος, που επιτρέπει σε οποιονδήποτε να αρχίσει γρήγορα να δημιουργεί της δικές του Android εφαρμογές. Με το HyperNext δεν υπάρχει καμία ανάγκη για την εκμάθηση java ή Android SDK και είναι πολύ πιο εύκολο στην χρήση από το Eclipse. Το περιβάλλον του HyperNext έχει μόνο ένα παράθυρο σχεδίασης και μια απλή γραμμή εργαλείων. Έλεγχοι, όπως για παράδειγμα κουμπιά, μπορούν να τοποθετηθούν γρήγορα και η επεξεργασία τους γίνεται με την χρήση του Script Editor. Η όλη λειτουργία του γίνεται σε τρία απλά βήματα :

- Σχεδίαση
- Προεπισκόπηση
- Run

Το HyperNext Android Creator (HAC) παρέχει τα εξής: Υπάρχει η δυνατότητα test εφαρμογών demo με απευθείας τηλεφόρτωση στην Android συσκευή. Δοκιμάζοντας demos παιχνίδια όπως Lunar Lander, Pre-School app ή Stock Quotes app φαίνεται πόσο διασκεδαστικό είναι. Το HAC έρχεται με example projects που είναι πλήρως εκτελέσιμα και το μόνο που μένει είναι να αρχίσουμε τις δικές μας Android εφαρμογές αμέσως. Εάν θέλουμε να δημιουργήσουμε το δικό μας app για το Android, στη συνέχεια θα προσπαθήσουμε με το HAC [13]:

- Εύκολο στη χρήση και ιδανικό για αρχάριους
- Δεν χρειάζεται η εκμάθηση Java ή το Android SDK
- Πολλά παραδείγματα για το ξεκίνημα
- Κατέβασμα demo εφαρμογών για Android συσκευή
- Απλό περιβάλλον εργασίας: ένα παράθυρο, σχεδιασμός και ένα Toolbar
- Το περιβάλλον εργασίας έχει τρεις λειτουργίες: Σχεδιασμός, Προεπισκόπηση και Run
- Μπορεί να τρέξει τα καθήκοντα φόντου
- Δημιουργία sprite με βάση τα παιχνίδια, όπως το Lunar Lander.
- Λειτουργία του Email ανεξάρτητη από το Android OS.
- Ενσωματώνεται με Android SDK και Java JDK.
- Στην δοκιμαστική έκδοση μπορούμε να δημιουργήσουμε, να αποθηκεύσουμε και να εκτελέσουμε εφαρμογές για 5 λεπτά
- Δημιουργεί και για τα δύο Emulator, και Android smartphones και tablets.
- Τα Apps λειτουργούν με Android OS 1.6 και με μεταγενέστερα.
- Η έκδοση για PC τρέχει σε Windows XP, Vista και 7.
- Η OS X έκδοση έρχεται σύντομα.
- Τακτικές ενημερώσεις για μεγαλύτερη λειτουργικότητα και νέα χαρακτηριστικά.

Επισκόπηση για Αρχάριους και Προγραμματιστές: Ο χρήστης είτε είναι αρχάριος, είτε έμπειρος προγραμματιστής, θα βρει κάτι στο HAC να τον ενδιαφέρει:

Αρχάριοι: Η HAC έχει ένα απλό interface και μια English-like γλώσσα προγραμματισμού, έτσι ώστε ακόμη και μη-προγραμματιστές να μπορούν γρήγορα να ξεκινήσουν να δημιουργούν το δικό τους λογισμικό. Το περιβάλλον ανάπτυξης HAC οπτικά είναι σχεδιασμένο έτσι ώστε να επιτρέπει την ταχεία τοποθέτηση των πεδίων, κουμπιών και άλλων στοιχείων ελέγχου, και στη συνέχεια γρήγορη πρόσβαση στις ιδιότητες και τα σενάρια τους.

Προγραμματιστές: Το σύστημα HyperNext είναι τόσο ισχυρό και επεκτάσιμο με πολλές εκατοντάδες εντολές και λειτουργίες. Σύντομα η HyperNext Android Developer θα κυκλοφορήσει επιτρέποντας plugins που πρέπει να γίνουν για τη HAC και προγράμματα από κοινού με την HAC.

Σύγκριση HAC και Java: Πώς λειτουργεί μια εφαρμογή στο HAC σε σύγκριση με μια εφαρμογή σε Java; Σε έναν αρχάριο προγραμματιστή, θα φανεί χρήσιμη η παρακάτω σύγκριση:

	HAC	Java & Android SDK
Easy to learn	yes	no
Time to understand	hours	months
Android functionality	growing subset	full
Language	English-like	Object Orientated
Variable types	1	Multiple
Commands per function	1	>10
Error checking	automatic	script
Screen and control creation	Designer	XML file and script
Screen scaling	automatic	script
Menu creation	Designer	script
Images & video	Library	script
Image Banks	yes	no
Sprite Animation	built-in	script
Device missing functions	automatic	script
Overview Help	built-in	search web
Built demos .apk files	our website	search web
Support	our forums	search web
Price	\$59.95	free
Time	low	very high

Απαιτήσεις υπολογιστή για την HAC:

- Windows XP, Vista ή 7
- Ταχύτητα του επεξεργαστή των 2GHz ή και περισσότερο
- Μέγεθος οθόνης 1024x768 pixels (συνιστάται)
- Ενεργοποίηση Άδειας για σύνδεση στο internet

5.4.1 HyperNext

Το HyperNext είναι ένα freeware visual software development system που στοχεύει στους αρχάριους προγραμματιστές και τρέχει επάνω σε Macintosh και προσωπικούς υπολογιστές (pc). Εμπνεύστηκε από την HyperCard και περιλαμβάνει τον έλεγχο GUI όπως κουμπιά και listboxes, και αποτελεί μια English-like γλώσσα προγραμματισμού. Το HyperNext αποτελεί επίσης ένα υψηλού επιπέδου αντικειμενοστραφές σύστημα συνταγμένο σε BASIC. Το πακέτο HyperNext Studio περιλαμβάνει τρεις συμπληρωματικές εφαρμογές που μπορούν να βοηθήσουν τους χρήστες να δημιουργήσουν και να τρέξουν το λογισμικό στις πλατφόρμες των Windows, MAC OS X και MAC OS 9.

- Ο HyperNext Creator είναι παρόμοιος με το Hypercard και επιτρέπει στους χρήστες να δημιουργήσουν το δικό τους λογισμικό. Η σχεδιαστική διεπαφή του Creator περιλαμβάνει μόνο ένα παράθυρο σχεδιασμού, μια toolbar και έναν Mode switcher. Τα τρία Modes: Design, Preview και Run, επιτρέπουν τη γρήγορη εναλλαγή μεταξύ της δημιουργίας/ επεξεργασίας και το τρέξιμο των προγραμμάτων.
- Ο HyperNext Developer κάνει built plugins/ βιβλιοθήκες για τον HyperNext Creator και επιτρέπει στους χρήστες να επεκτείνουν τη λειτουργία του Creator με τα δικά τους plugins. Ο HyperNext Developer έχει μια αρκετά παρόμοια διεπαφή με τον Hypernext Creator και τοποθετεί τα ήδη built plugins άμεσα στο φάκελο των plugins του Creator. Για τη γρηγορότερη ανάπτυξη των plugins είναι δυνατό να είναι ταυτόχρονα ανοιχτοί και ο Creator αλλά και ο Developer.
- Ο HyperNext Player είναι μια αυτόνομη εφαρμογή, παρόμοια με τον Hypercard Player, δεδομένου ότι τρέχει τους σωρούς που δημιουργούνται από την «γονική» του εφαρμογή, που είναι ο HyperNext Creator. Ο HyperNext Player είναι δωρεάν λογισμικό και υπάρχουν εκδόσεις για τις πλατφόρμες των Windows, MAC OS X και MAC OS 9.

5.4.2 Περιβάλλον προγραμματισμού

Το HyperNext έχει μια σχετικά απλή διεπαφή σχεδίου, που επιτρέπει στα προγράμματα τη γρήγορη εκτέλεση και τον τάχιστο έλεγχο. Έλεγχοι όπως τα κουμπιά, μπορούν να επιλεγθούν από τη γραμμή εργαλείων και να προστεθούν έπειτα στο παράθυρο σχεδιασμού. Μόλις τοποθετηθούν, οι ιδιότητες των ελέγχων μπορούν να αλλαχθούν και τα scripts τους να επεξεργαστούν, χρησιμοποιώντας τον ενσωματωμένο scripts editor.

Η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται από το HyperNext, ονομάζεται απλά HyperNext και είναι ελαφρώς βασισμένη στη γλώσσα του Hypercard, τη HyperTalk. Η HyperNext είναι μια interpreted English-like γλώσσα και έχει πολλά χαρακτηριστικά που επιτρέπουν τη δημιουργία ολοκληρωμένων εφαρμογών και σωρών. Τα χαρακτηριστικά αυτά περιλαμβάνουν μια μεγάλη ποικιλία λέξεων-κλειδιών, με τις οποίες επηρεάζουν και λαμβάνουν το feedback των πολλαπλών τύπων ελέγχου GUI και των ικανοτήτων των πολυμέσων, χρησιμοποιώντας το Apple Quicktime. Οι Midi νότες και οι ήχοι μπορούν να αναπαραχθούν χρησιμοποιώντας το Note Player και η επιφάνεια στοιχείων του HyperNext έχει διάφορες χρήσεις, συμπεριλαμβανομένης της ανάπτυξης παιχνιδιών. Οι εκδόσεις για Macintosh περιλαμβάνουν τις απαραίτητες λειτουργίες για να τρέξουν την AppleScript, που είναι μια γλώσσα scripting.

Η γλώσσα HyperNext μπορεί να εγκαταθεί με καθορισμένα από το χρήστη plugins ή plugins τρίτων, τα οποία αναπτύσσονται με τη χρήση του HyperNext Developer. Η HyperNext έχει επίσης τη δυνατότητα να τρέξει RBScripts σε πραγματικό χρόνο εκτέλεσης, γεγονός που επιτρέπει στους χρήστες να φτιάξουν τις δικές τους προγραμματισμένες εφαρμογές. Υπάρχει ένα μεγάλο εύρος RBScript κώδικα και παρόμοιου κώδικα BASIC που είναι ήδη διαθέσιμα σε διάφορα sites στο

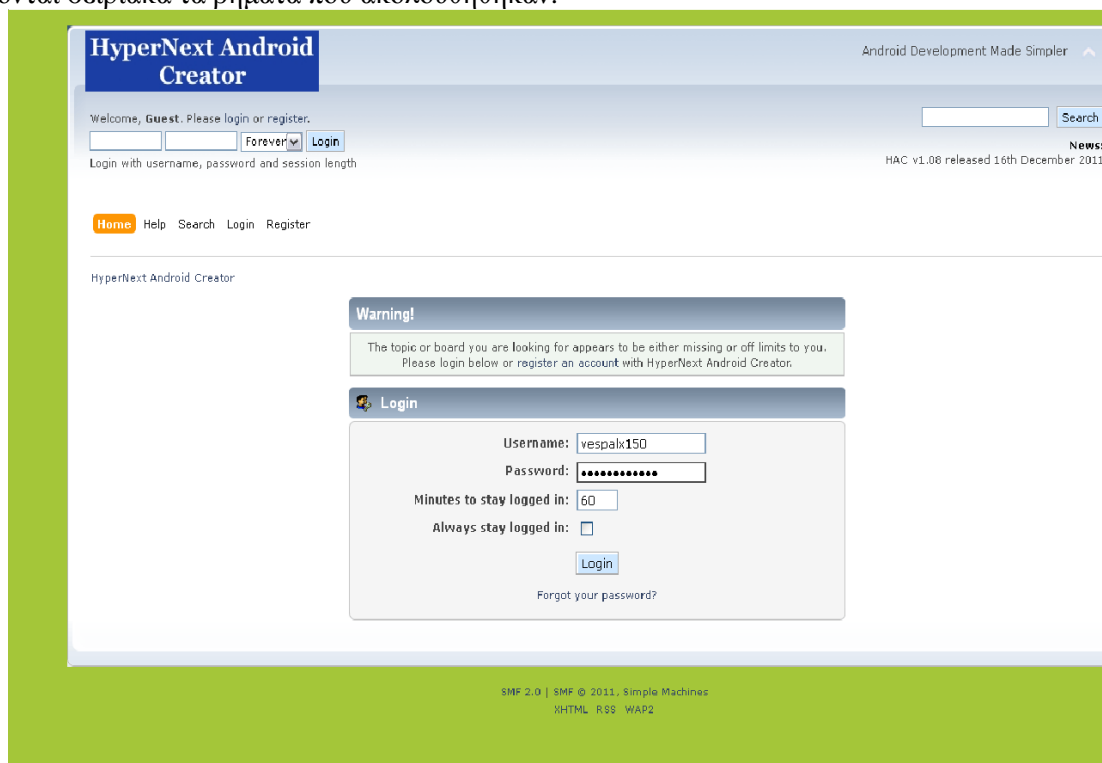
διαδίκτυο. Το RBScript είναι μια έκδοση της REALbasic, ένα μοντέρνο αντικειμενοστραφές σε BASIC, που έχει αναπτυχθεί από τη Real Software.

Η HyperNext παρέχει το πλαίσιο για την ανάπτυξη νευρωνικών δικτύων και έρχεται με παραδείγματα, τα οποία επιδεικνύουν πώς γίνεται η ανάπτυξη της εξάσκησης και της δοκιμής ενός νευρωνικού δικτύου. Τα projects και οι σωροί που δίνονται ως παραδείγματα περιλαμβάνουν την ανάλυση των στοιχείων ασθενών, ακολουθιών DNA έως και των τιμών μεταχειρισμένων αυτοκινήτων.

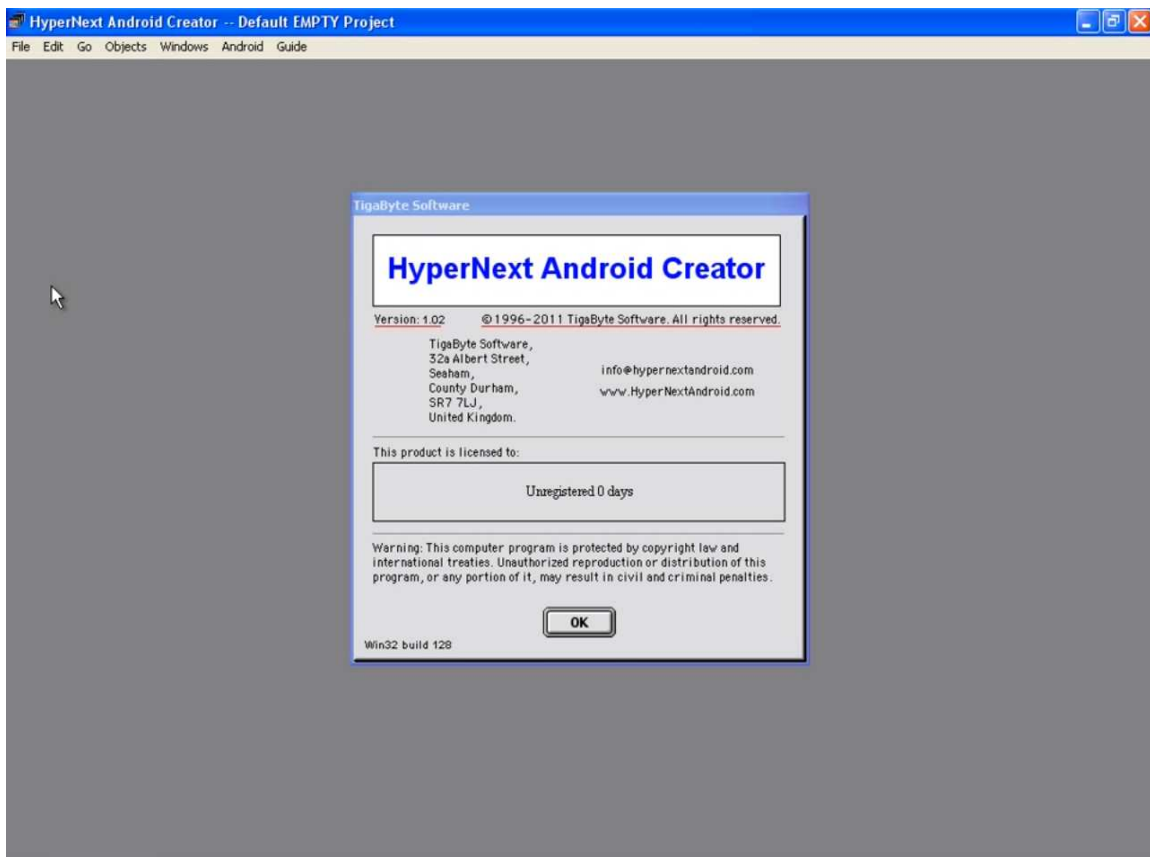
Μια πρόσφατη εισαγωγή στη HyperNext είναι η δυνατότητα ελέγχου των συσκευών βιο-ανάδρασης, όπως το LightStone που χρησιμοποιεί USB HID σύνδεση. Η συσκευή LightStone στέλνει τα δεδομένα που μπορούν να υποβληθούν σε επεξεργασία, έτσι ώστε να δώσουν τιμές που αφορούν τον ρυθμό της καρδιάς και της δερματικής αγωγιμότητας, τα οποία μπορούν έπειτα να προσπελαστούν από ένα πρόγραμμα για τη χορήγηση φαρμακευτικής αγωγής ή για να γίνουν προτάσεις στο χρήστη για βιο-αναδραστικά παιχνίδια που μπορεί αυτός να παίξει. Η HyperNext παρέχει ένα τέτοιο project-παράδειγμα που απεικονίζει γραφικά τα δεδομένα μιας συσκευής LightStone.

5.4.3 Εφαρμογή για το HyperNext Android

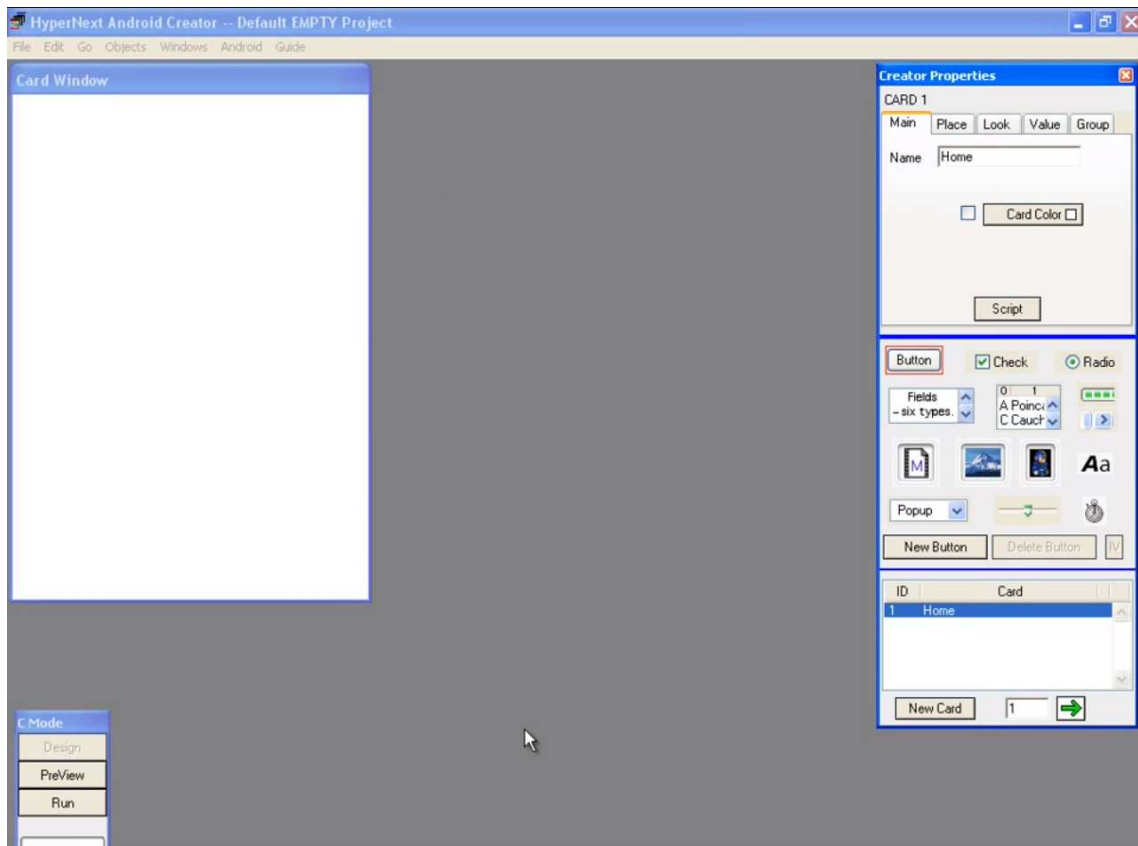
Σε αυτήν την εφαρμογή τα βήματα που πρέπει να ακολουθήσουμε είναι πολύ απλά. Αφού έχουμε αναπτύξει και εξηγήσει την λειτουργία του HyperNext Android δεν έχουμε παρά να υλοποιήσουμε την δικιά μας εφαρμογή. Το αρχικό μας βήμα είναι να πάμε στην ηλεκτρονική διεύθυνση www.HyperNextAndroid.com και να κάνουμε εγγραφή, έτσι ώστε να μπορούμε να κατεβάσουμε το πρόγραμμα της εφαρμογής μας. Στο e-mail που έχει αποσταλεί στην ηλεκτρονική διεύθυνση μας, αναγράφονται ακριβώς τα βήματα που πρέπει να ακολουθήσουμε για την εγκατάσταση. Δεξιά στην σελίδα, μπορούμε να διακρίνουμε το Installation. Ανοίγοντας αυτό το παράθυρο, βλέπουμε τα πέντε βήματα εγκατάστασης για το HAC. Μόλις εγκαταστήσουμε το πρόγραμμα, είμαστε έτοιμοι να ξεκινήσουμε την ανάπτυξη της δικιάς μας εφαρμογής. Παρακάτω φαίνονται σειριακά τα βήματα που ακολουθήθηκαν:



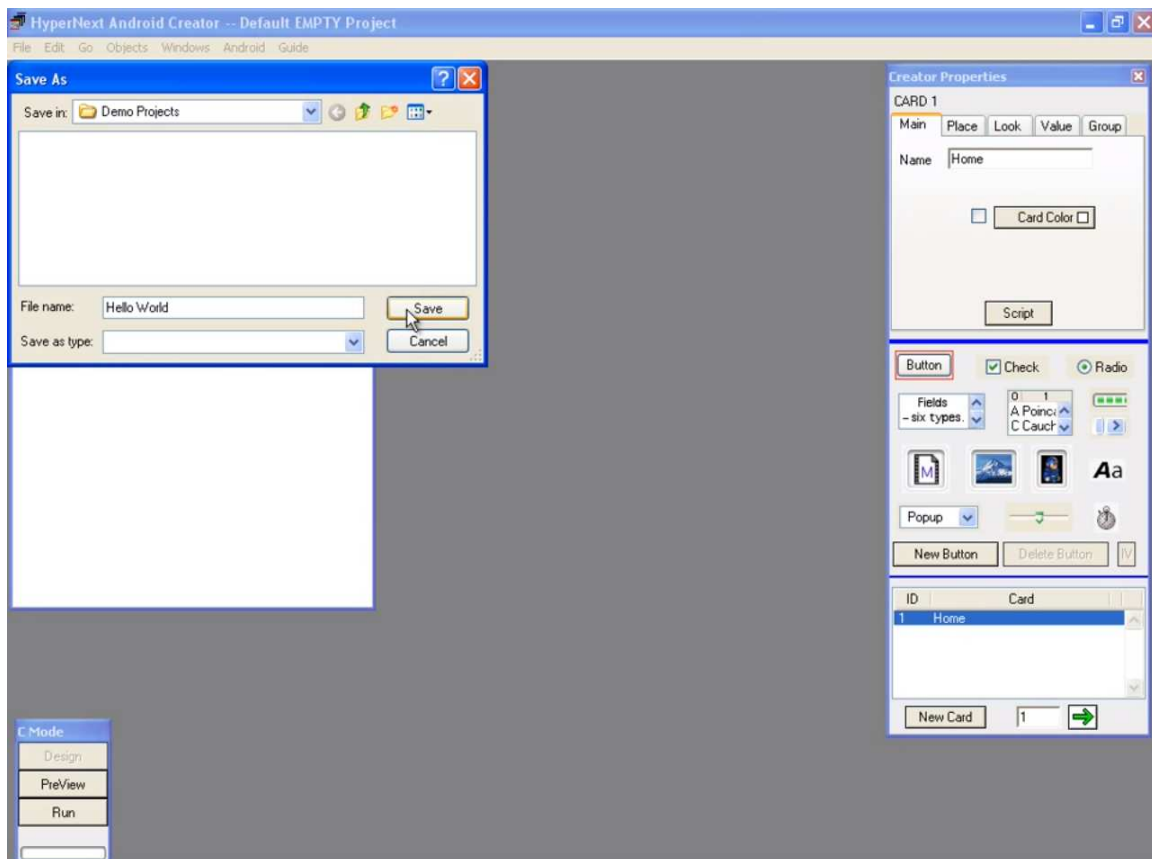
Εικόνα 68: Μόλις έχουμε ανοίξει την ηλεκτρονική σελίδα Hypernext Android και κάνουμε την εγγραφή μας



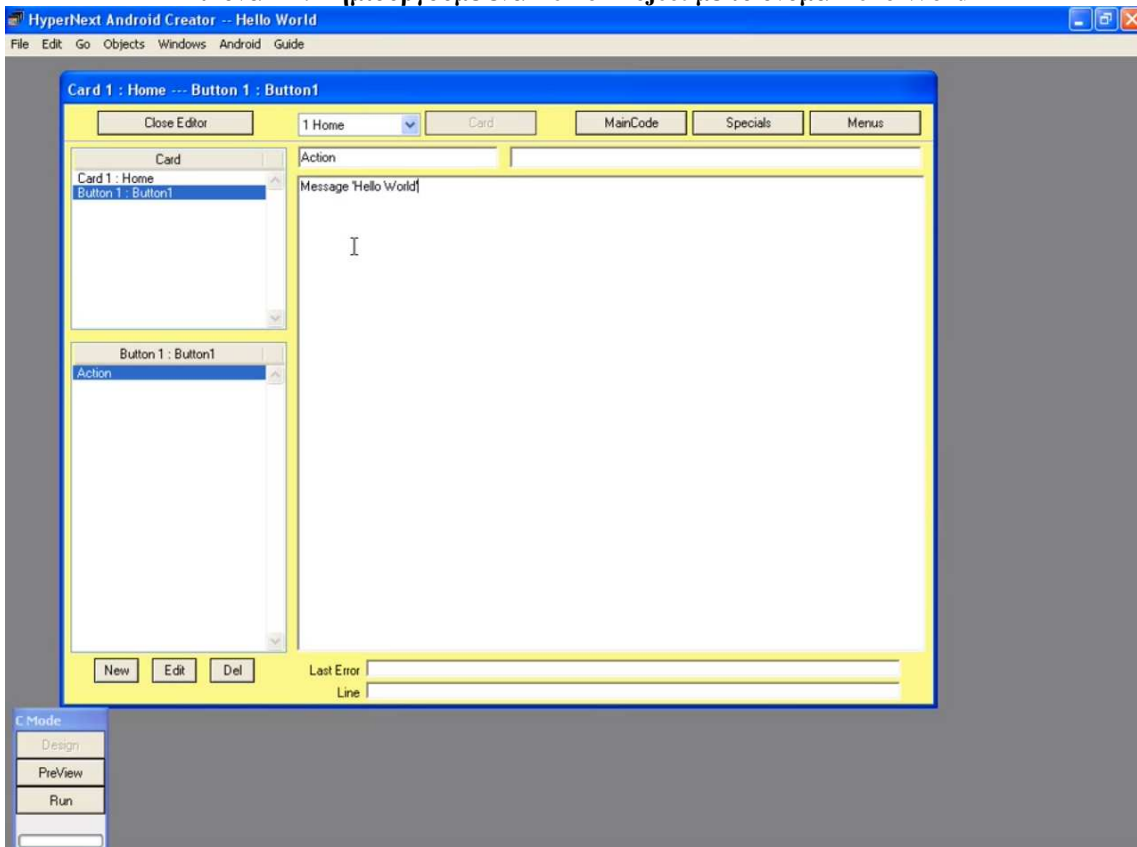
Εικόνα 69: Εικόνα κατά την έναρξη του HyperNext Android Creator



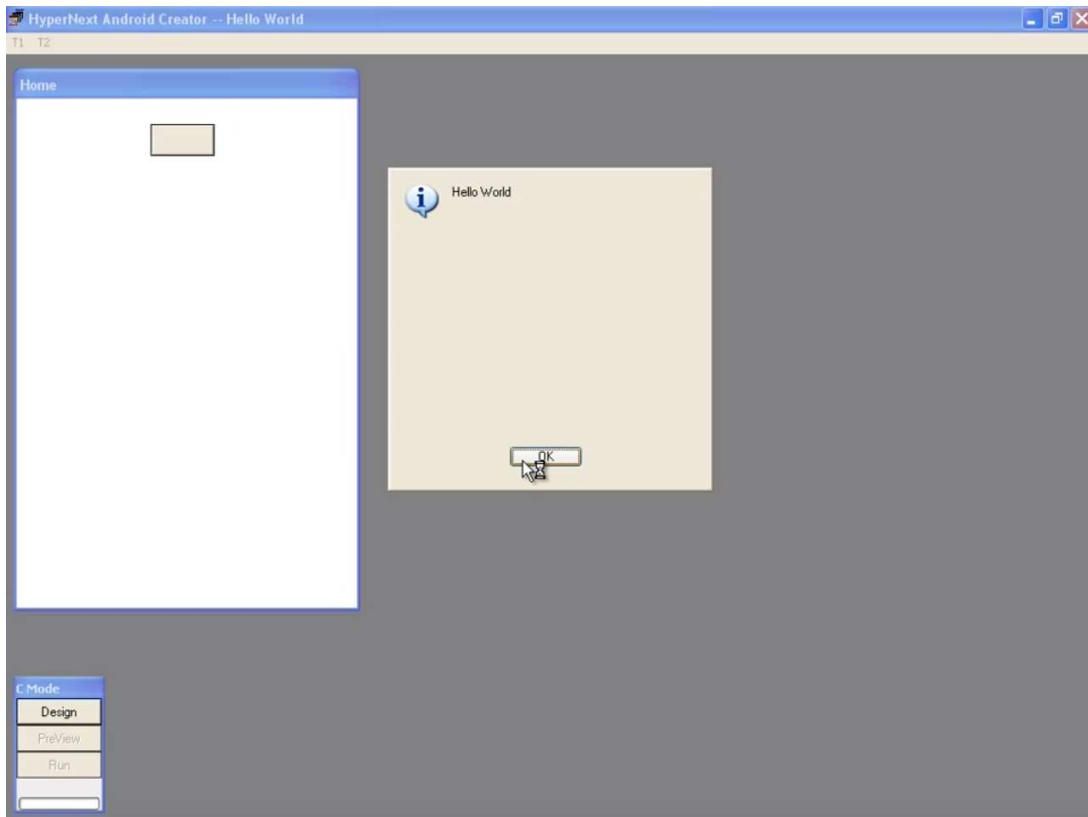
Εικόνα 70 : Σε αυτό το σημείο μπορούμε να διακρίνουμε το γραφικό περιβάλλον του HAC



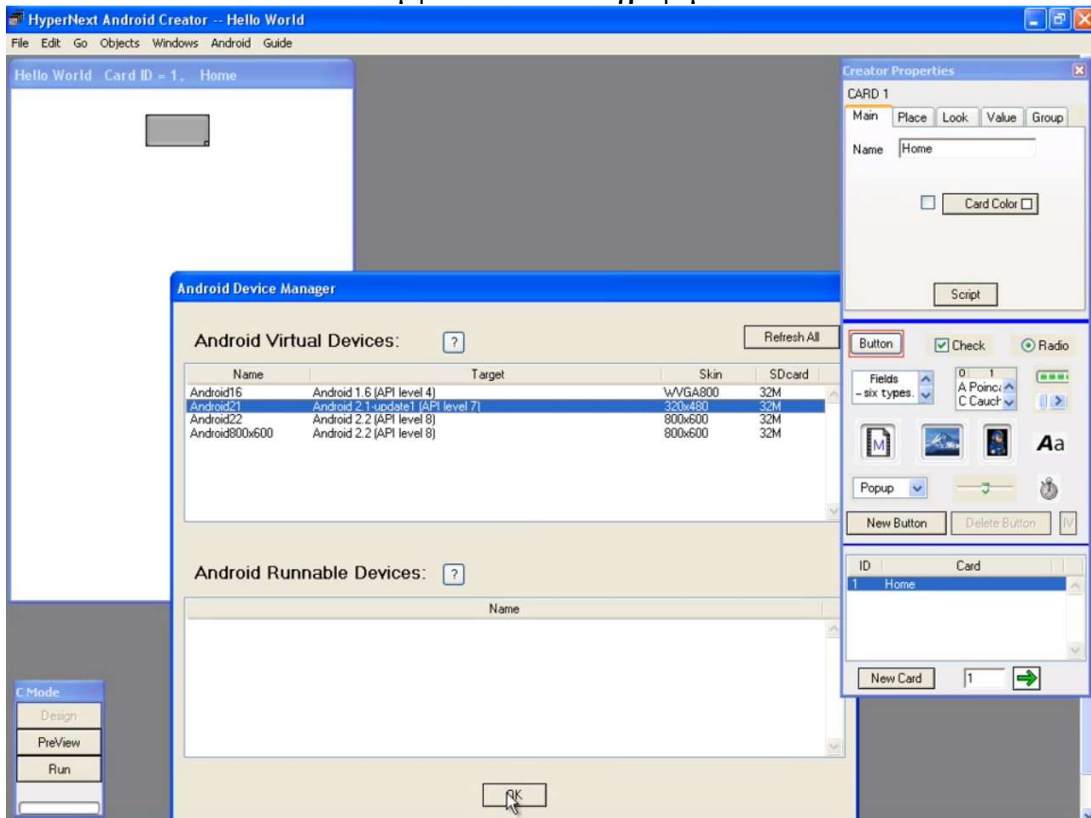
Εικόνα 71 : Δημιουργούμε ένα Demo Project με το όνομα Hello Word



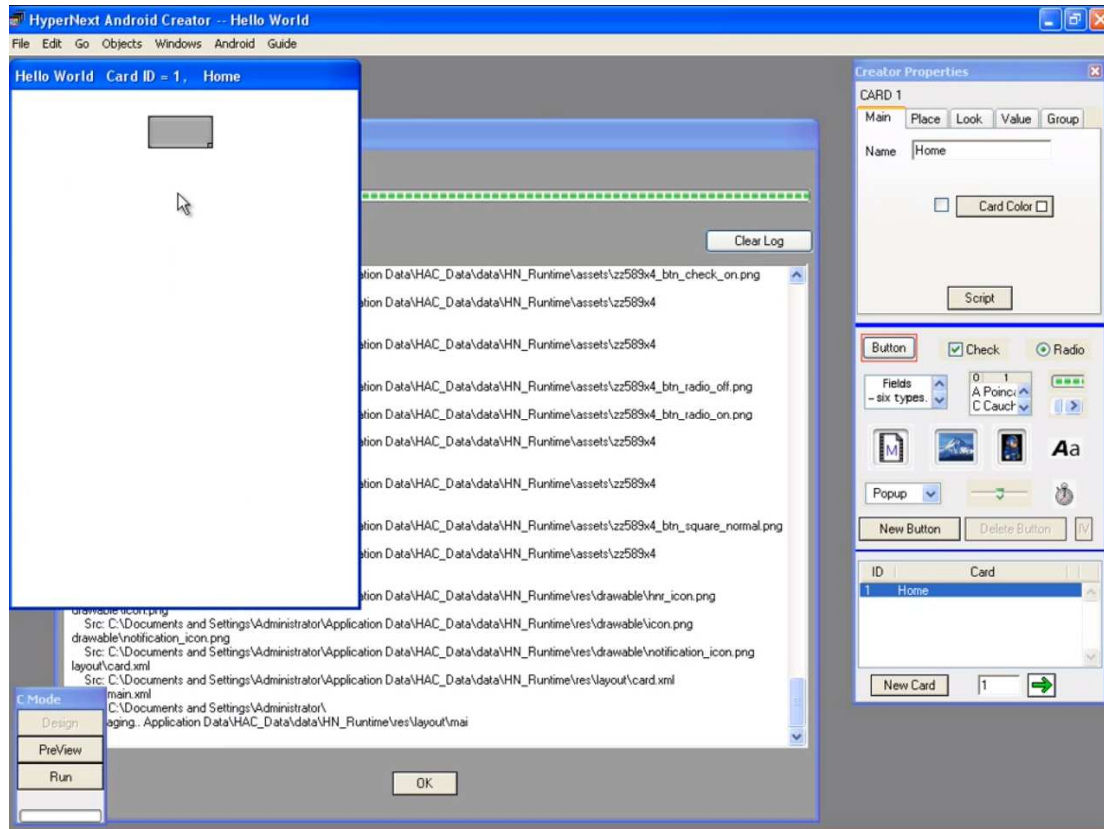
Εικόνα 72 : Έχουμε προσθέσει ένα Button με την επιλογή New Button και επιλέγουμε την ιδιότητα Script έτσι ώστε να γράψουμε αυτό που θα εμφανίζεται πατώντας το Button.



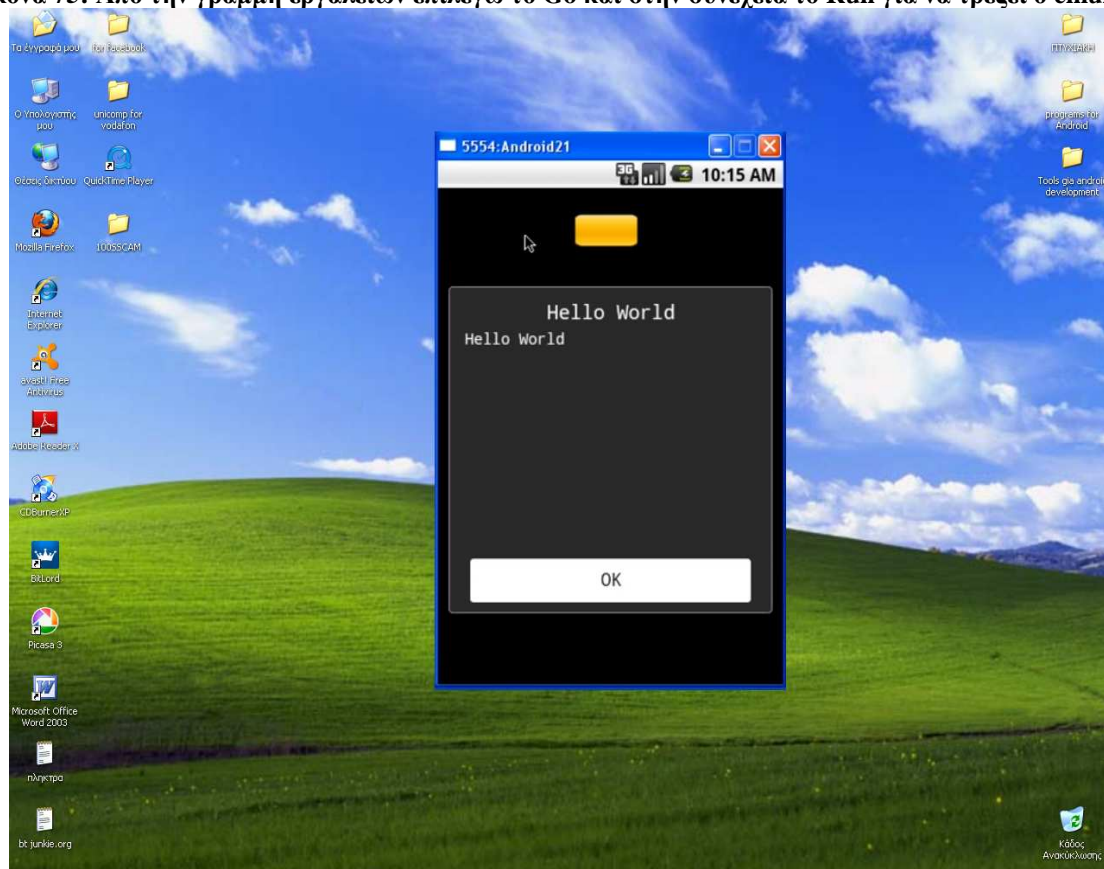
Εικόνα 73 : Έχουμε πατήσει το Run και στην συνέχεια πάνω στο Button της εφαρμογής μας για να εμφανίσει αυτό που γράψαμε



Εικόνα 74 : Στο C Mode επιλέγουμε Design. Στην συνέχεια, επιλέγουμε το Manager Devices από την γραμμή εργαλείων Android



Εικόνα 75: Από την γραμμή εργαλείων επιλέγω το Go και στην συνέχεια το Run για να τρέξει ο emulator



Εικόνα 76: Έχει ανοίξει ο emulator του Android. Πατώντας πάνω στο Button που έχουμε δημιουργήσει και εμφανίζεται η εφαρμογή μας. Στην περίπτωση μας, ένας κειμενογράφος που γραφεί “Hello world”.

6 Android application

6.1 Μια Android application με την χρήση των SDK και Eclipse

Ο σκοπός αυτής της εφαρμογής είναι να βοηθήσει όσους ενδιαφέρονται να γράψουν apps για Android. Σε αυτήν την εφαρμογή θα εξετάσουμε ολόκληρο τον κύκλο της ανάπτυξης εφαρμογής, ξεκινώντας με ένα καθαρό υπολογιστή μέχρι το τελικό .apk αρχείο.

6.2 Προετοιμασία

Για την προετοιμασία της εφαρμογής απαιτείται η εγκατάσταση μερικών λειτουργικών προγραμμάτων που αναφέρονται λεπτομερώς πιο κάτω. Δεν θα υπάρξουν δραματικές αλλαγές αν χρησιμοποιείτε άλλη έκδοση των Windows ή Linux.

- Εγκατάσταση του Eclipse IDE
- Εγκατάσταση ADT plugin για το Eclipse
- Εγκατάσταση JDK και JRE
- Εγκατάσταση του Android SDK

6.2.1 Εγκατάσταση του Eclipse IDE

Επιλέγουμε την ενότητα Downloads στο επίσημο site και εγκαθιστούμε την έκδοση Classic. Στη συνέχεια αποσυμπιέζουμε το αρχείο οπουδήποτε θέλουμε (εν προκειμένω, στο φάκελο C:\Eclipse Εγκατάσταση του περιβάλλοντος Java JDK και JRE). Αν δεν έχουμε ήδη εγκαταστήσει το Java Development Kit (JDK) και Java Runtime Environment (JRE), τότε θα πρέπει να το κάνουμε.

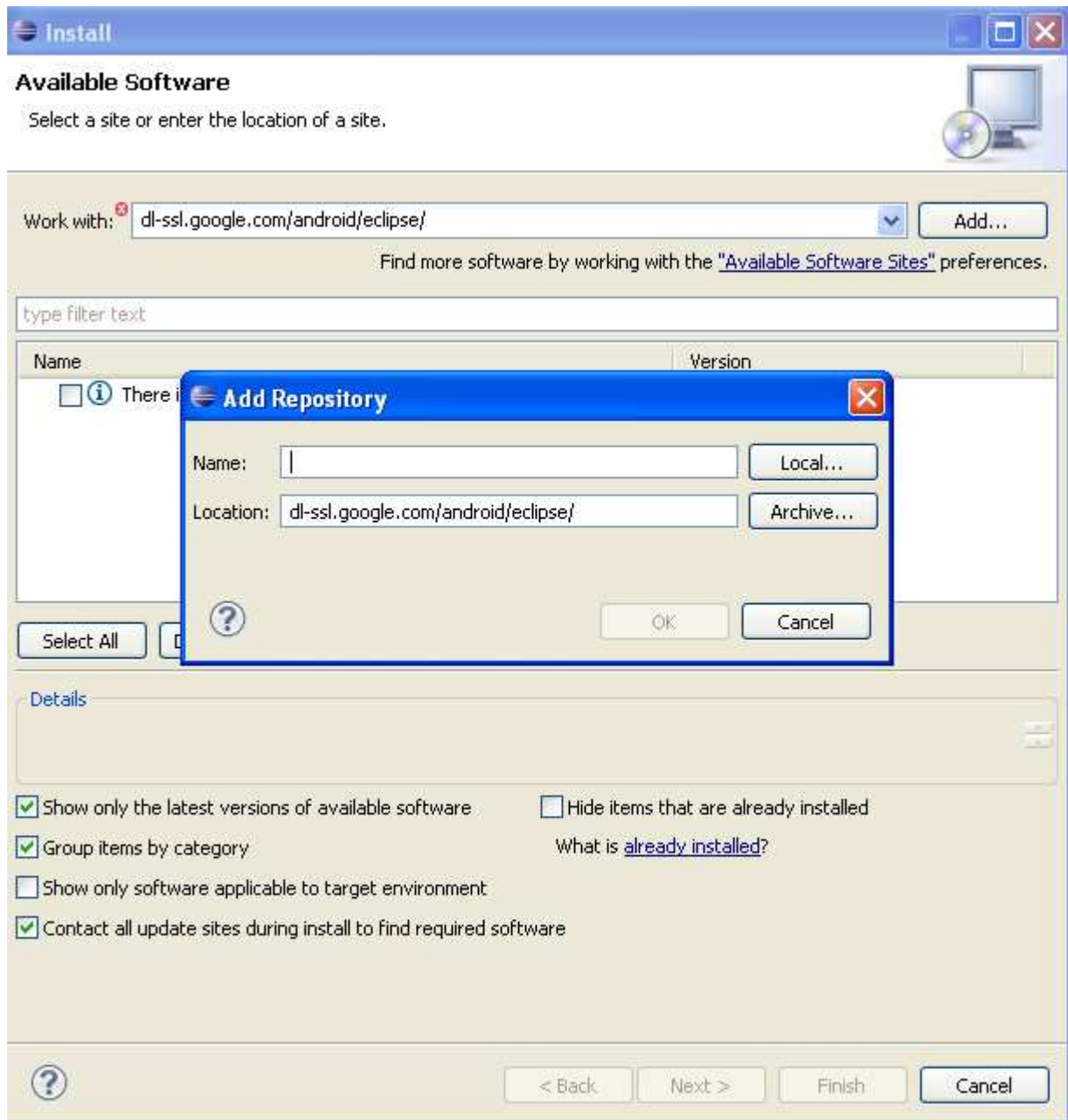
Αυτό είναι κάτι που γίνεται στο site της Oracle (www.oracle.com/technetwork/java/javase/downloads/index.html), επιλέγοντας «Λήψη και εγκατάσταση του JDK και JRE».

Java Platform, Standard Edition		
Java SE 6 Update 24 This release includes security enhancements and bug fixes. Learn more ▶	<input type="button" value="Download JDK"/>	<input type="button" value="Download JRE"/>
What Java Do I Need? You must have a copy of the JRE (Java Runtime Environment) on your system to run Java applications and applets. To develop Java applications and applets, you need the JDK (Java Development Kit), which includes the JRE.	JDK 6 Docs <ul style="list-style-type: none"> Installation Instructions ReadMe ReleaseNotes Oracle License Third Party Licenses Supported System Configurations 	JRE 6 Docs <ul style="list-style-type: none"> Installation Instructions ReadMe ReleaseNotes Oracle License Third Party Licenses Supported System Configurations

Εικόνα 77 : Java Platform

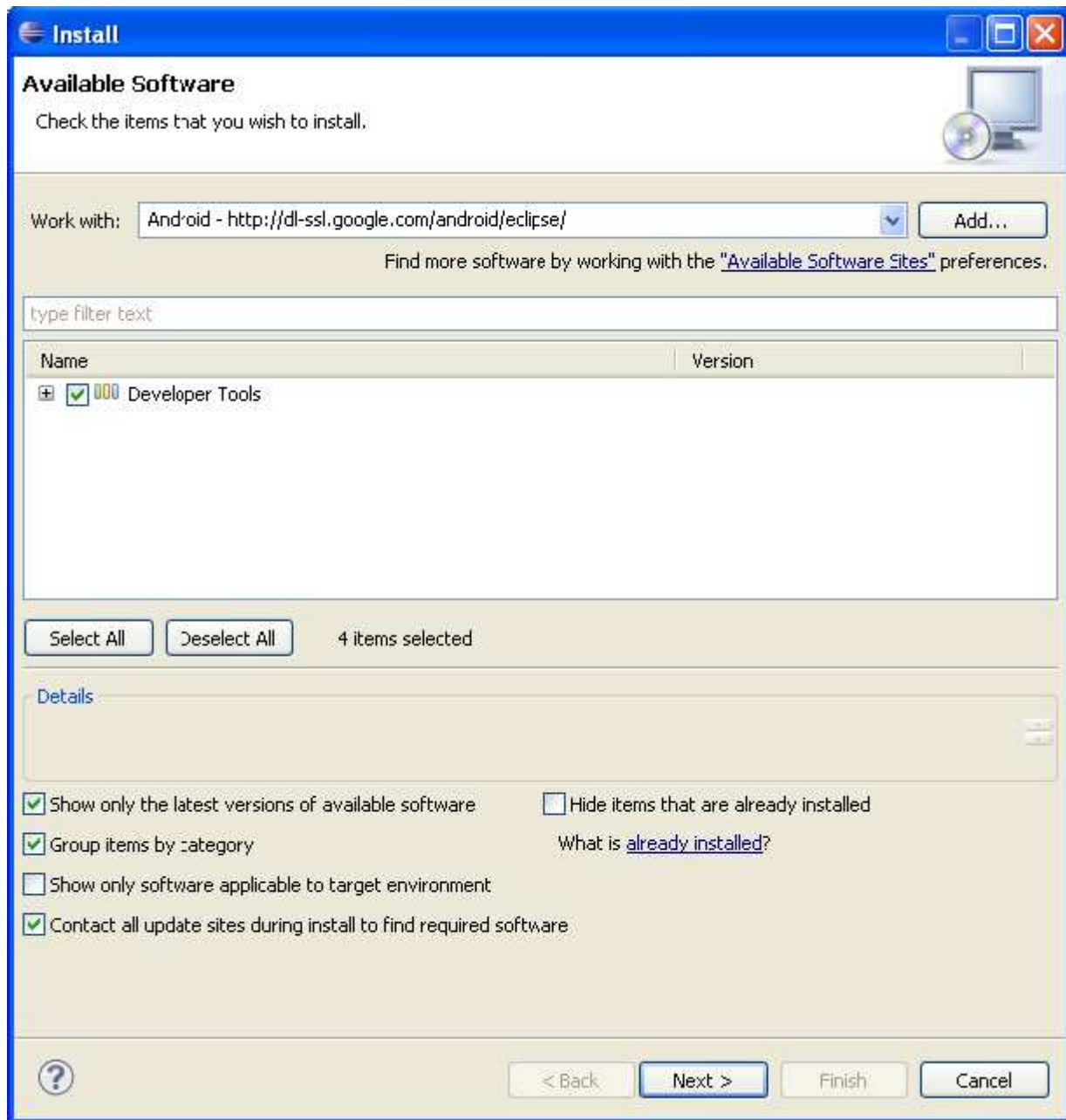
6.2.2 Εγκατάσταση του ADT plugin για το Eclipse

Τρέχουμε το Eclipse. Αρχικά θα μας ζητηθεί να επιλέξουμε τον φάκελο που θα αποθηκεύονται τα projects μας. Για την παρούσα εργασία επιλέχθηκε το C:\Eclipse\Projects. Τώρα πρέπει να εγκαταστήσουμε το plugin, επιλέγοντας στο Help->Install New Software. Στο πεδίο “Work With” γράφουμε <https://dl-ssl.google.com/android/eclipse/> και πατάμε add. Στο παράθυρο που θα ανοίξει γράφουμε ένα όνομα ώστε αργότερα να μπορούμε να τα διαχωρίζουμε.



Εικόνα 78: Εγκατάσταση του Eclipse

Επιλέγουμε το “Developer tools” και πατάμε next.

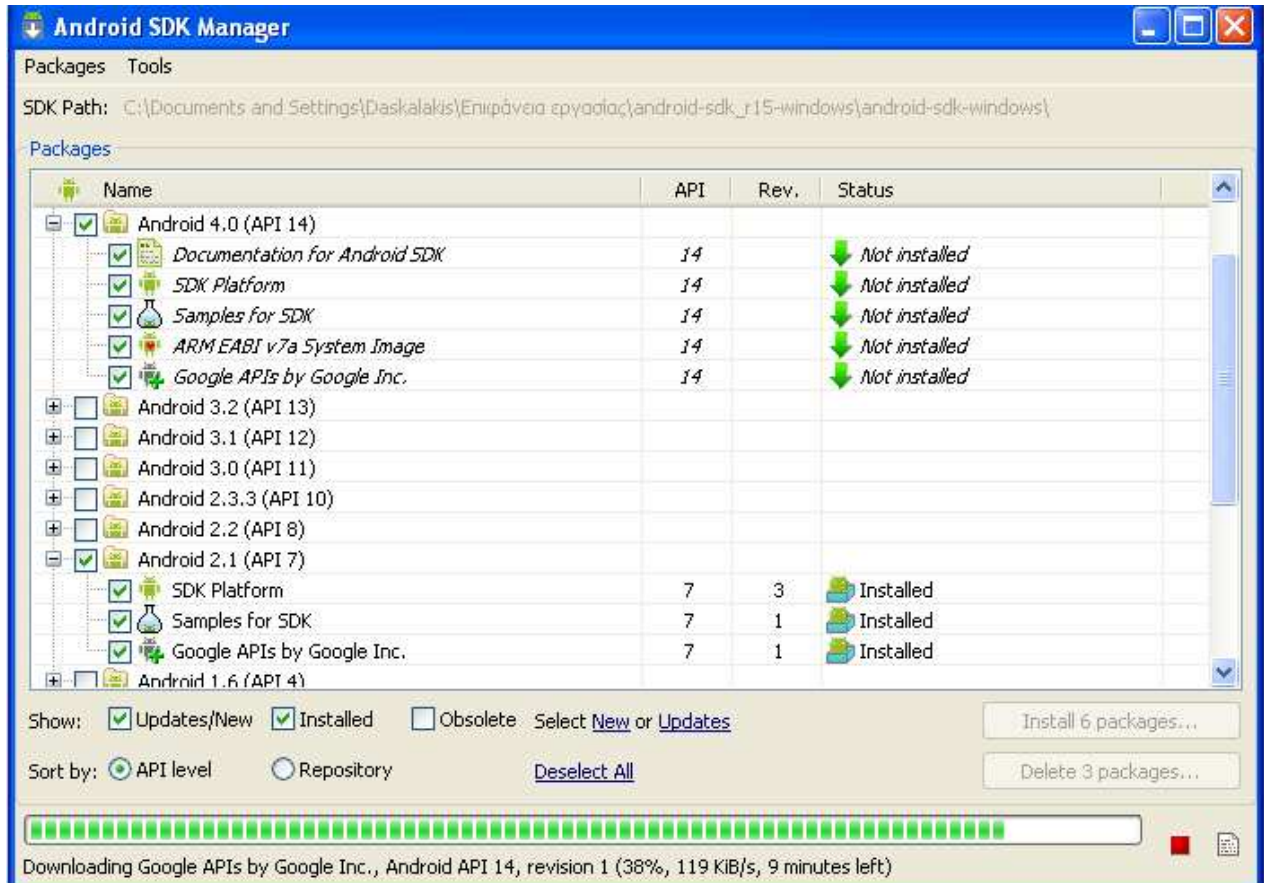


Εικόνα 79 : Εγκατάσταση του Eclipse

Αφού ολοκληρωθεί η εγκατάσταση κάνουμε επανεκκίνηση του Eclipse.

6.2.3 Εγκατάσταση του Android SDK

Κατεβάζουμε το SDK από το επίσημο site του Android . Για τις ανάγκες της εργασίας, επιλέχθηκε το .zip αρχείο και έγινε unzip στο C:\Android. Τρέχουμε το αρχείο SDK Manager.exe που βρίσκεται στον φάκελο Android και στο παράθυρο που θα ανοίξει, επιλέγουμε τις εκδόσεις του Android και διάφορα άλλα στοιχεία που θέλουμε να χρησιμοποιήσουμε και πατάμε το install.



Εικόνα 80 : Εγκατάσταση του SDK

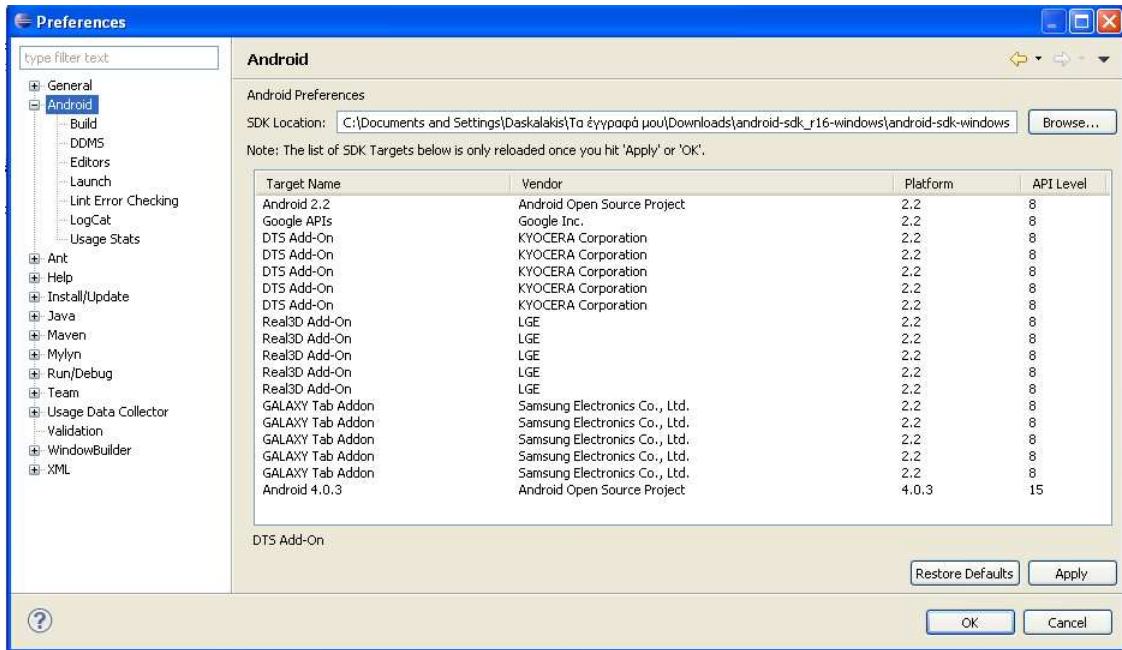
Όταν ολοκληρωθεί η εγκατάσταση, πατάμε Restart και κλείνουμε τα παράθυρα.

6.3 Δημιουργία Android-εφαρμογής

Μια καλή συμβουλή πριν από τη δημιουργία της πρώτης εφαρμογής, είναι η δημιουργία μιας εικονικής Android συσκευής για τον ταχύτερο έλεγχο του λογισμικού. Το Android Virtual Device (AVD) είναι ένα τέτοιο εικονικό smartphone που τρέχει Android, στο οποίο μπορούμε εύκολα να εκτελέσουμε οποιοδήποτε πρόγραμμα έχουμε δημιουργήσει.

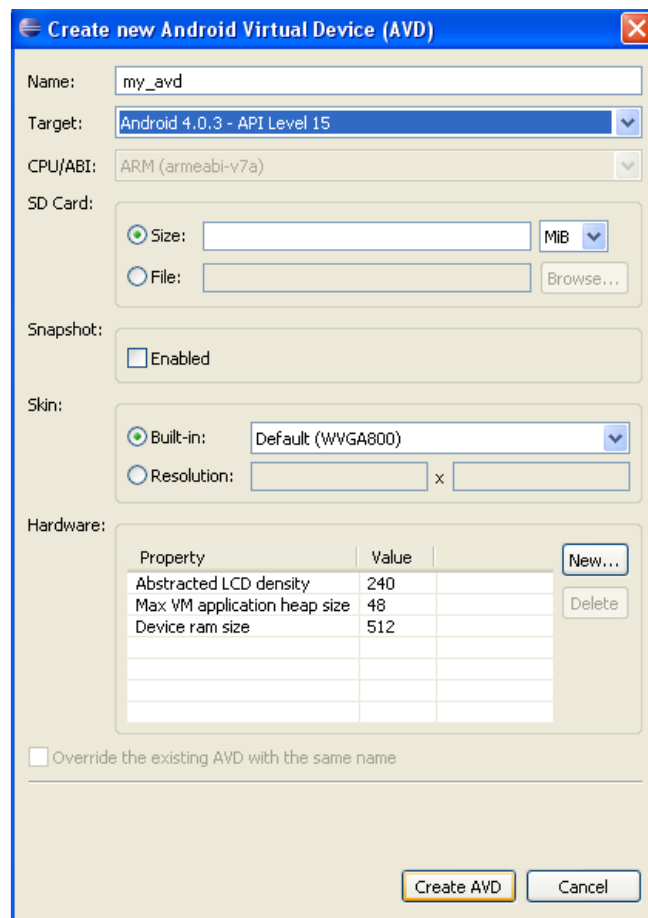
Το προφανές πλεονέκτημα του AVD είναι ότι μπορούμε να δούμε πώς το λογισμικό μας θα προβάλλεται σε διαφορετικά smartphones τα οποία υποστηρίζουν Android, αντί να τα αγοράσουμε (κάτι που είναι πρακτικά και κυρίως, οικονομικά αδύνατον) και να κάνουμε τις απαραίτητες δοκιμές στο καθένα ξεχωριστά.

Πριν φτιάξουμε το AVD, πρώτα θα συνδέσουμε το Android SDK, επιλέγοντας στο Window->Preferences. Στο παράθυρο που ανοίγει, πάμε στο Android και εκεί επιλέγουμε το φάκελο που έχουμε κάνει Unzip το SDK, πατάμε Apply και τέλος OK.



Εικόνα 81: Android Virtual Device (AVD)

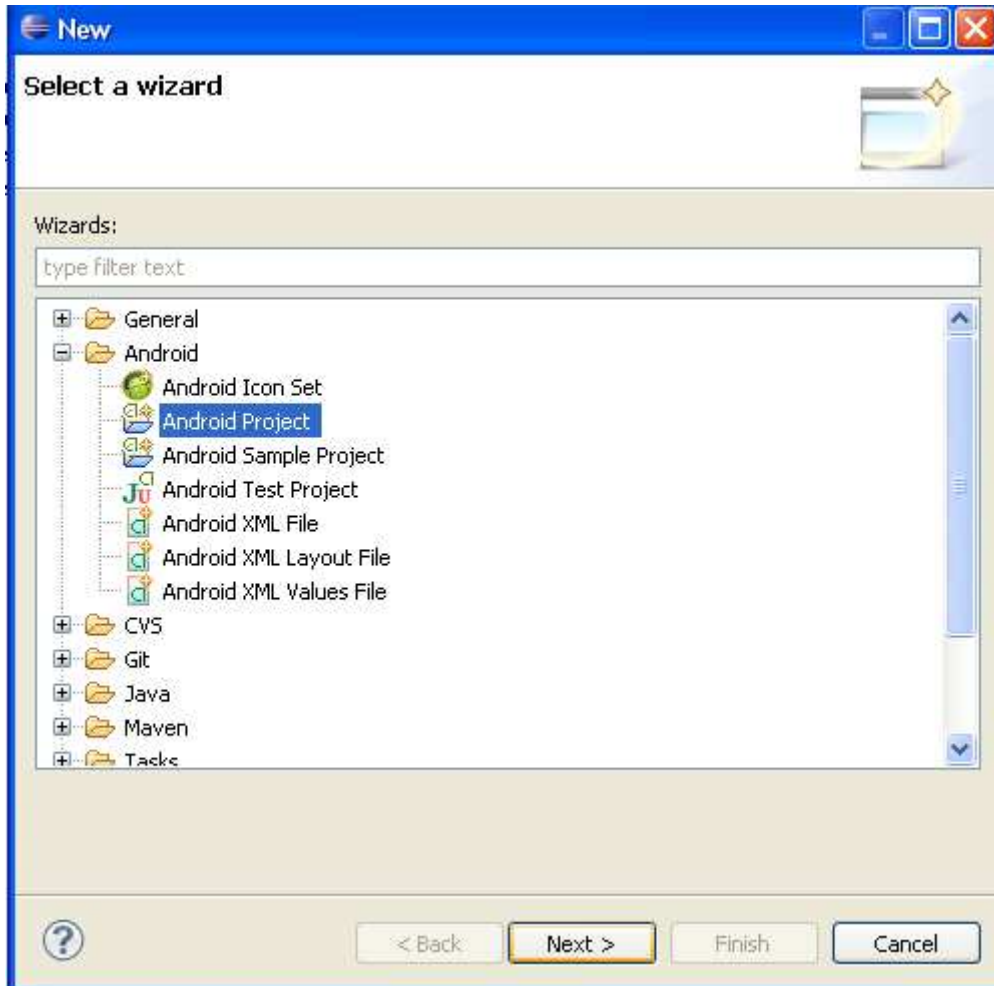
Τώρα πάμε στο Window -> Android SDK and AVD Manager, πατάμε New και διαλέγουμε τα χαρακτηριστικά που θέλουμε να έχει το εικονικό smartphone μας.



Εικόνα 82: Android Virtual Device (AVD)

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

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



Εικόνα 83: Android Virtual Device (AVD)

Πατάμε Finish

Για τις ανάγκες του tutorial, αναπτύξαμε ένα απλό πρόγραμμα που δείχνει την θερμοκρασία στο Λουτράκι, οπότε οι οδηγίες που αναφέρονται στο εξής αφορούν την δημιουργία αυτού του app.

Μετά την δημιουργία του project, στα αριστερά θα εμφανιστεί ένα δέντρο φακέλων. Αρχικά μπορούμε να επιλέξουμε την εικόνα του app μας. Για να το κάνουμε αυτό, πάμε στο res και κάνουμε drag&drop μια png εικόνα 72×72 στον φάκελο drawable-hdpi, 48×48 στον drawable-mdpi και 36×36 στον drawable-ldpi.

Το πρόγραμμα χρειάστηκε μόλις 3 στοιχεία: ImageView (μια εικόνα, το logo), Button (το κουμπάκι που θα κάνει refresh όταν το πατήσουμε), TextView (εδώ θα εμφανίζεται η θερμοκρασία). Όλα αυτά τα στοιχεία πρέπει να περιγράφουν στο αρχείο main.xml (res->layout).

Εδώ φαίνεται το αρχείο main.xml της συγκεκριμένης εφαρμογής:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="fill_parent"
    android:background="#313031"
    android:layout_width="match_parent"
    android:orientation="vertical">

    <TableLayout android:id="@+id/tableLayout1"
        android:layout_height="86px"
        android:layout_width="match_parent">

        <ImageView android:layout_height="86px"
            android:src="@drawable/wheather"
            android:layout_width="263px"
            android:id="@+id/imageView1"
            android:layout_weight="8"></ImageView>
    </TableLayout>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="..."
        android:textSize="100dip"
        android:gravity="center_horizontal"
        android:id="@+id/temper"
        android:textColor="#EEE3C4"
    />

    <Button android:gravity="center_horizontal"
        android:padding="20px"
        android:layout_gravity="center_horizontal|center"
        android:layout_width="wrap_content"
        android:text="Refresh"
        android:textSize="30px"
        android:layout_height="wrap_content"
        android:id="@+id/refrbutton"
        android:layout_marginTop="10dip"></Button>

</LinearLayout>

```

Εικόνα 84: Κώδικας στο main.xml για να δηλώσουμε τα 3 στοιχεία μας

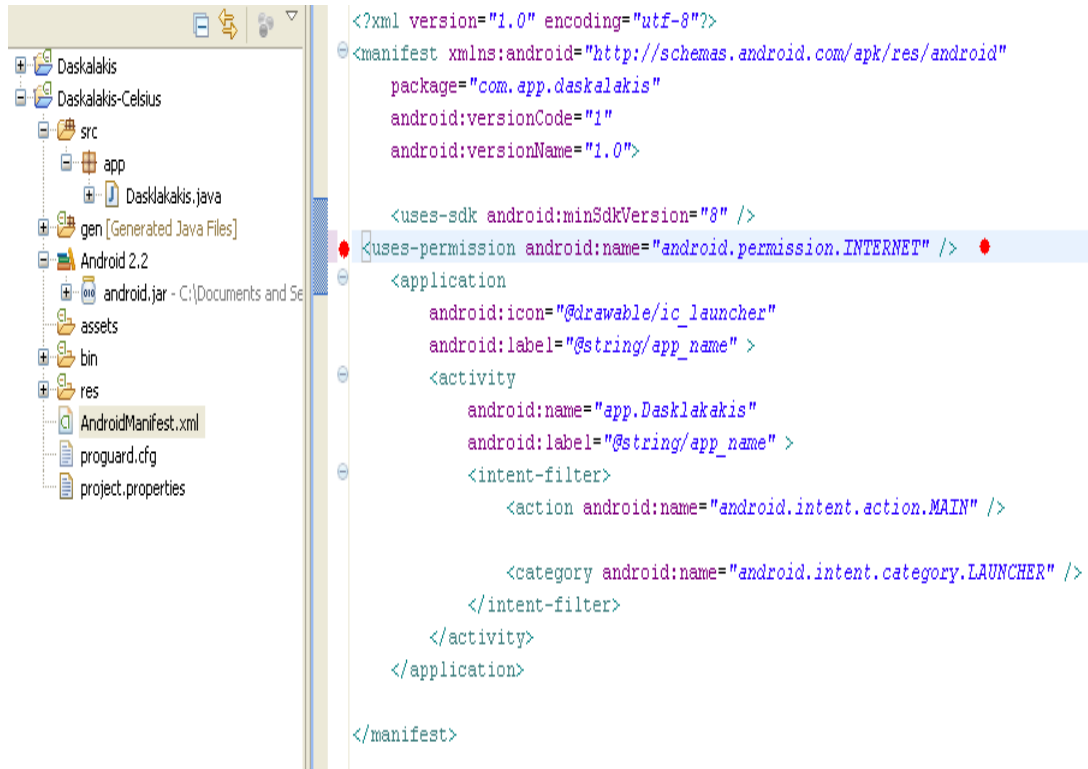
Ο παραπάνω κώδικας χρειάζεται κάποια περαιτέρω ανάλυση, είναι πολύ απλός. Αυτό που έχει σημασία εδώ είναι ο τρόπος με τον οποίο φυτρώθηκε το Logo στο ImageView. Στο src γράφουμε την διαδρομή προς την εικόνα, ξεκινώντας από το σύμβολο @. Όπως φαίνεται, στο res φτιάξαμε νέο φάκελο με το όνομα drawable και εκεί τοποθετήσαμε το Logo.

Ο κώδικας του app βρίσκεται στην διαδρομή src->"package name"->"app name".java. Το Eclipse φτιάχνει αυτόματα ένα Header, που από κάτω μπορούμε να γράψουμε τον κώδικά μας. Η λογική του παρόντος app είναι η εξής:

- Κατεβάζουμε την ιστοσελίδα που περιέχει την θερμοκρασία

- Χρησιμοποιώντας regular expressions, βγάζουμε την ίδια την θερμοκρασία
- Δείχνουμε την θερμοκρασία στο TextView

Να σημειώσουμε ότι όταν μια εφαρμογή απαιτεί κάποιου είδους δικαιώματα, αυτά πρέπει να καθοριστούν στο AndroidManifest.xml. Επειδή στην προκειμένη περίπτωση έγινε χρήση σύνδεσης στο Internet, έπρεπε να προστεθεί η παρακάτω γραμμή, η οποία απεικονίζεται ανάμεσα στις δυο κόκκινες κουκίδες.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.app.daskalakis"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name="app.Dasklakakis"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

Εικόνα 85: Κώδικας στο AndroidManifest.xml

Ο κώδικας που επεξηγήθηκε προηγουμένως, φαίνεται στην [Εικόνα 85], που ακολουθεί:

```

package app;
|
import com.app.daskalakis.R;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Button;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import java.util.regex.*;
import com.app.daskalakis.R.id;
import com.app.daskalakis.R.layout;

@SuppressWarnings("unused")

public class Dasklakakis extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(layout.main);
        final Button button = (Button) findViewById(id.refrbutton);
        button.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) // κλικ πάνω στο κουμπί
            {
                RefreshTemper();
            }
        });

        RefreshTemper(); // φορτώνουμε την θερμοκρασία μόλις τρέξει το πρόγραμμα
    };
    //-----
    public String GetTemper(String urlsite) // η function που φορτώνει την θερμοκρασία
    {
        String matchtemper = "";
        try
        {
            // φορτώμα της ιστοσελίδας

            URL url = new URL(urlsite);
            URLConnection conn = url.openConnection();
            InputStreamReader rd = new InputStreamReader(conn.getInputStream());
            StringBuilder allpage = new StringBuilder();
            int n = 0;
            char[] buffer = new char[40000];
            while (n >= 0)

```

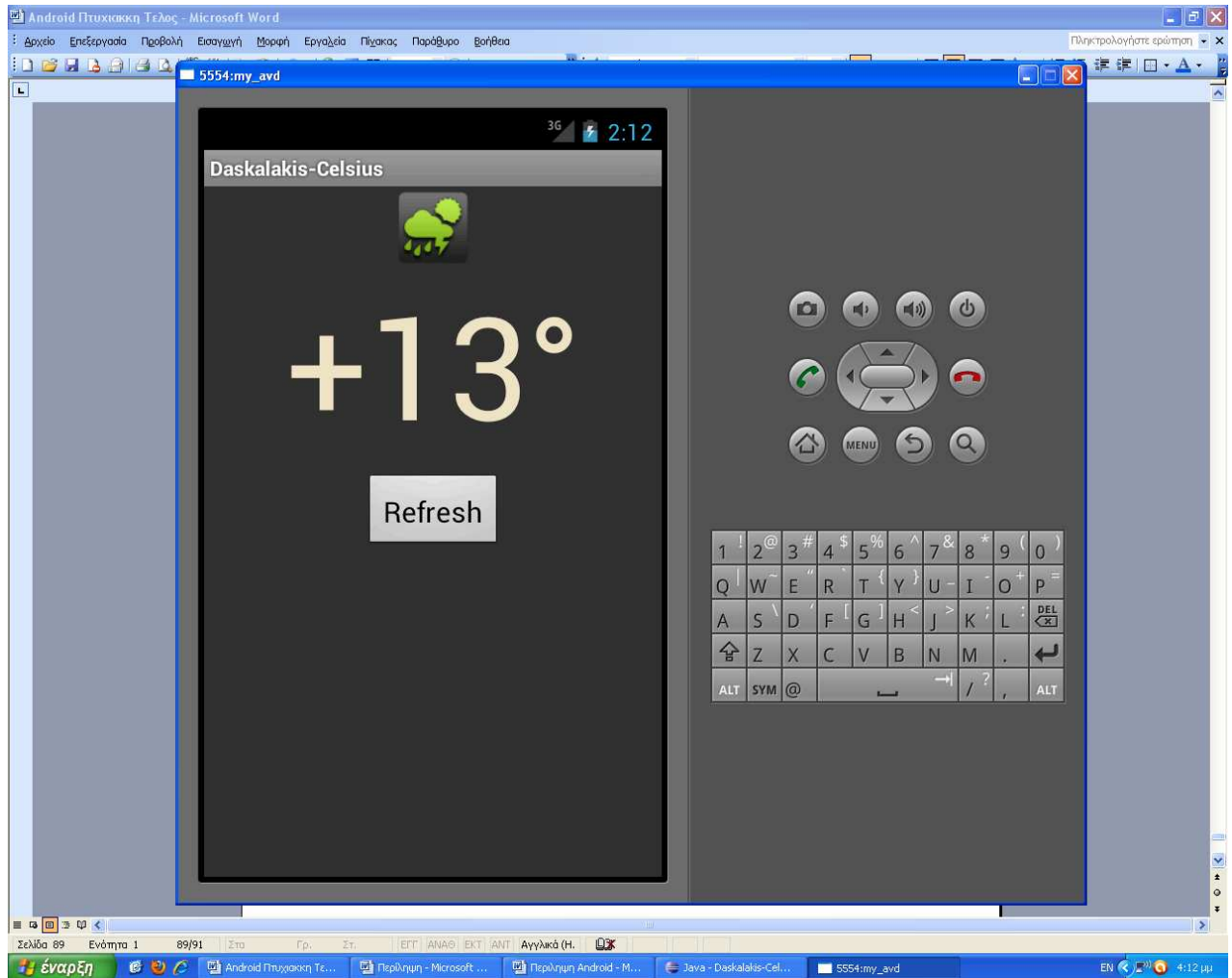
```

        {
            n = rd.read(buffer, 0, buffer.length);
            if (n > 0)
            {
                allpage.append(buffer, 0, n);
            }
        }
        // κολουουμε parsing και βγαζουμε την θερμοκρασια
        final Pattern pattern = Pattern.compile
        ("

```

Εικόνα 86: Συνολικός κώδικας

Για να εκτελεστεί η εφαρμογή, πατάμε Run->Run ή Ctr+F11. Αυτό θα τρέξει το AVD που φτιάξαμε πιο πριν και θα εκτελέσει τον κώδικα που μόλις γράψαμε.



Εικόνα 87: Ολοκλήρωση εφαρμογής

6.4 Συμπέρασμα

Εν κατακλείδι, θα μπορούσαμε να πούμε ότι η ανάπτυξη εφαρμογών για το Android είναι ευχάριστη, πολύ απλή και ενδιαφέρουσα. Φυσικά, έχοντας υπόψη μας, πως αυτό αποτελεί μόνο την κορυφή του παγόβουνου.

Βιβλιογραφία

- [1] Millennial Media's May 2011 Mobile Mix Report
<http://www.mobilemarketingwatch.com/millennial-medias-may-2011-mobile-mix-report-shows-major-gains-across-the-mobile-landscape-16274/>
- [2] Android Developers: What is Android?
<http://developer.android.com/guide/basics/what-is-android.html>
- [3] Android versions comparasion
<http://socialcompare.com/en/comparison/android-versions-comparison>
- [4] Τι είναι το Android
<http://www.adds.gr/company/technology/what-is-android/>
- [5] Activity Lifecycle
<http://developer.android.com/reference/android/app/Activity.html>
- [6] Android Chart
<http://techblog.gr/tablet/android-honeycomb-google-statistics-3786/>
- [7] Android Developers: Platform Versions
<http://developer.android.com/resources/dashboard/platform-versions.html>
- [6] Strategy Analytics: Global Tablet Revenues
<http://www.businesswire.com/news/home/20110419005114/en/Strategy-Analytics-Global-Tablet-Revenues-Reach-US49>
- [7] Richardblundell: Statistics
http://richardblundell.net/2010/12/tablet_sales_2010/
- [8] AndroLib
"Android App Stats".
- [9] Leena Rao
"Google: 3 Billion Android Apps Installed; Downloads Up 50 Percent From Last Quarter".
- [10] Download the Android SDK
<http://developer.android.com/sdk/index.html>
- [11] Ότι χρειάζομαστε για το Android NDK
<http://www.ibm.com/developerworks/opensource/tutorials/os-androidndk/os-androidndk-pdf.pdf>
- [12] App inventor
<http://www.appInventorbeta.com>
- [13] App inventor
<http://www.hypernextandroid.com/hnfiles/about.html>