



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Τίτλος:

Ανάπτυξη Android εφαρμογής με το MIT Inventor

Ζαχαρίας Κατσαράκης (ΑΜ: 1590)

Επιβλέπων καθηγητής : Γεώργιος Τριανταφυλλίδης

Επιτροπή Αξιολόγησης :

Ημερομηνία παρουσίασης:

Ευχαριστίες

Κατ' αρχάς θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας τον Κ. Τριανταφυλλίδη Γεώργιο για την καθοδήγηση του καθ' όλη τη διάρκεια της δημιουργίας της.

Θα ήθελα επίσης να ευχαριστήσω τον Κ. Βιδάκη Νικόλαο για το ενδιαφέρον που έδειξε και την βοήθεια που μου προσέφερε.

Τέλος ένα μεγάλο ευχαριστώ χρωστώ στην οικογένεια μου για τη στήριξη καθ' όλη τη διάρκεια των σπουδών μου και κυρίως κατά την εκπόνηση της πτυχιακής εργασίας δείχνοντας κατανόηση και υπομονή. Επίσης ευχαριστώ θερμά το φιλικό μου περιβάλλον και τα αγαπημένα μου πρόσωπα τα οποία μου συμπαραστέκονται ηθικά.

Περίληψη

Τα τελευταία χρόνια βιώνουμε μια επανάσταση των έξυπνων κινητών τηλεφώνων (smartphones) τα οποία ενσωματώνουν πλέον δυνατότητες που δεν υπάρχουν στα συμβατικά κινητά τηλέφωνα. Τα λειτουργικά συστήματα των συσκευών αυτών ποικίλουν με το Android της Google να περιλαμβάνεται τη στιγμή αυτή στο μεγαλύτερο ποσοστό των smartphone της αγοράς.

Στην αρχή της πτυχιακής γίνεται μια ιστορική αναφορά για το λειτουργικό σύστημα Android δίνονται απαντήσεις σε διάφορα ερωτήματα όπως: Τι είναι το Android; Πότε αρχισαι; Ποιος το ξεκίνησε; και γιατί; Η ιστορική αναδρομή συνεχίζεται με τις πλατφόρμες Android (εκδόσεις και χαρακτηριστικά) φτάνοντας μέχρι και σήμερα.

Επίσης γίνεται μια αναφορά στην αρχιτεκτονική του Android σε διάφορα επίπεδα, στην ανατομία μιας εφαρμογής, το πώς αυτή συμπεριφέρεται χρησιμοποιώντας το λειτουργικό Android και το υλικό της συσκευής.

Στην συνέχεια παρουσιάζονται τα εργαλεία ανάπτυξης λογισμικού για Android με ιδιαίτερη έμφαση στο APP Inventor της Google το οποίο είναι το εργαλείο ανάπτυξης της εφαρμογής που θα δούμε σε αυτήν την πτυχιακή.

Η πτυχιακή τελειώνει με την παρουσίαση βήμα προς βήμα της εφαρμογής Paint Pot με την οποία ο χρήστης μπορεί να επέμβει σε μια εικόνα που έχει τραβήξει με την κάμερα του κινητού του τηλεφώνου και να ζωγραφίσεις πάνω σε αυτήν. Ένας από τους σκοπούς της εφαρμογής αυτής είναι να δείξει πόσο απλά είναι τα πράγματα προγραμματίζοντας κανείς με το APP Inventor της Google.

Abstract

In recent years, we are living through the revolution of the smartphones, which have many more possibilities than typical cell phones. The operating systems for these devices vary, but Google's Android system is the one most widely used.

In the introduction of this thesis, there is a brief historical reference to the Android operating system and some simple questions such as: What is the Android? When did it appear? Who introduced it and why? are answered. Then, there is a reference to several Android platforms (edition and characteristics) from the beginning of their use till today. There is also a reference to the structure of Android in different levels, to the anatomy of the application, to the way the application behaves using the Android operating system, and to the hardware device.

Subsequently, the software development tools of Android are introduced, with emphasis given to Google's App Inventor, which is the software development tool of the application analyzed in this thesis.

At the end, there is a step by step presentation of the PaintPot Application, an application that gives to the user the possibility to draw on a picture taken by the phone's camera. The basic aim of this application is to show how simple things are, when you are programming, using Google's App Inventor.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευχαριστίες.....	2
Περίληψη.....	3
Abstract.....	4

Κεφάλαιο 1

1.1	Εισαγωγή.....	12
1.2	Ιστορία.....	12
1.3	WAP.....	13
1.4	Πλατφόρμες κινητών τηλεφώνων.....	13
1.5	Google.....	15
1.5.1	Η google στον ασύρματο κόσμο.....	16
1.6	Ανοιχτή Συμμαχία Μικροτηλεφώνων (Open Handset Alliance).....	16
1.7	Τι είναι το android.....	17
1.7.1	Η μασκόντ του Android.....	17
1.8	Μία πλατφόρμα νέας γενιάς.....	18
1.8.1	Δωρεάν Ανοιχτή Ολοκληρωμένη Πλατφόρμα.....	18

Κεφάλαιο 2

2.1	Η πλατφόρμα Android εξελίσσεται.....	19
2.2	Εκδόσεις και χαρακτηριστικά.....	19
2.3	Android 1.5 Cupcake.....	20
2.3.1	Χαρακτηριστικά Cupcake.....	20
2.4	Android 1.6 Donut.....	21
2.4.1	Χαρακτηριστικά Donut.....	21
2.5	Android 2.0 Éclair.....	22
2.5.1	Χαρακτηριστικά Éclair.....	22
2.6	Android 2.2 Froyo.....	23
2.6.1	Χαρακτηριστικά Froyo.....	23
2.7	Android 2.3 Gingerbread.....	24
2.7.1	Χαρακτηριστικά Gingerbread.....	24
2.8	Android 3.0 Honeycomb.....	25
2.8.1	Χαρακτηριστικά Honeycomb.....	25
2.9	Android 4 Ice Cream Sandwich.....	26
2.9.1	Χαρακτηριστικά Ice Cream Sandwich.....	26
2.10	Android 4.1 Jelly Bean.....	27
2.10.1	Χαρακτηριστικά Jelly Bean.....	27
2.11	Στατιστικά.....	29

Κεφάλαιο 3

3.1	Αρχιτεκτονική του Android.....	31
3.2	Επίπεδο Εφαρμογών (Applications - widget).....	31
3.3	Επίπεδο Πλαισίου Εφαρμογών (Applications Framework).....	32
3.4	Επίπεδο Βιβλιοθηκών (Libraries).....	32
4.5	Διαχείριση μεταβάσεων δραστηριοτήτων με προθέσεις.....	32
3.6	Dalvik Virtual Machine.....	33
3.7	Πυρήνας του Linux.....	33
3.8	Βιβλιοθήκες Android.....	33

Κεφάλαιο 4

4.1	Ανάπτυξη εφαρμογών Android.....	36
4.2	Η ανατομία εφαρμογών Android.....	36
4.3	Context (περιβάλλον).....	36
4.4	Activity (Δραστηριότητα).....	37
4.5	Διαχείριση μεταβάσεων δραστηριοτήτων με προθέσεις.....	41
4.5.1	Μετάβαση μεταξύ δραστηριοτήτων με προθέσεις.....	41
4.5.2	Εκκίνηση νέας δραστηριότητας με όνομα κλάσης.....	41
4.5.3	Δημιουργία προθέσεων με ενέργειες και δεδομένα.....	42
4.5.4	Εκκίνηση δραστηριότητας που ανήκει σε άλλη εφαρμογή.....	42
4.5.5	Διοχέτευση πρόσθετων πληροφοριών χρησιμοποιώντας προθέσεις.....	42
4.5.6	Οργάνωση δραστηριοτήτων και προθέσεων στην εφαρμογή χρησιμοποιώντας μενού.....	43
4.6	Εργασία με υπηρεσίες.....	43
4.7	Το αρχείο manifest του Android.....	44
4.7.1	Ρύθμιση παραμέτρων του αρχείου manifest του Android.....	44

Κεφάλαιο 5

5.1	Εργαλεία του Android.....	45
5.2	Android SDK.....	46
5.2.1	Μερικά από τα χαρακτηριστικά του SDK.....	46
5.2.2	Μερικά ακόμα εργαλεία.....	47
5.2.3	Τι περιέχεται στο πακέτο (box).....	47
5.3	Android NDK (Native Development Kit).....	48
5.3.1	Μερικά από τα χαρακτηριστικά του NDK.....	48
5.3.2	Τι προβλέπει το NDK.....	49
5.3.3	Πότε γίνεται ανάπτυξη σε ατόμιο κώδικα.....	49
5.3.4	Περιεχόμενα και εργαλεία ανάπτυξης του NDK.....	50
5.3.5	Τεκμηρίωση.....	51
5.4	App Inventor for Android.....	53
5.4.1	Η ανάπτυξη του App Inventor.....	53
5.4.2	Τι είναι το App Inventor.....	53
5.4.3	Τα αρχικά πλεονεκτήματα του App Inventor.....	56
5.5	Hyper Next Android Creator.....	57
5.5.1	HyperNext.....	59
5.5.2	Περιβάλλον προγραμματισμού.....	59

Κεφάλαιο 6

6.1	Ανάπτυξη εφαρμογής με το App Inventor της Google.....	61
6.2	Ξεκινώντας με το App Inventor.....	61
6.2.1	Computer Setup.....	64
6.2.2	Java configuration.....	65
6.2.3	Install the App Inventor Setup Software.....	66
6.3	Σύντομη περιγραφή της κεντρικής οθόνης του App Inventor.....	68
6.3.1	The Blocks Editor.....	70

Κεφάλαιο 7

7.1	Η εφαρμογή Paint Pot.....	73
7.2	Τι θα χρησιμοποιήσουμε για να κάνουμε την εφαρμογή.....	73
7.3	Φτιάχνοντας τα κουμπιά χρώματος.....	74
7.3.1	Χρήση ρυθμίσεων για καλύτερες διατάξεις (Layouts).....	74
7.3.2	Προσθέτοντας τον καμβά.....	75
7.3.3	Οργάνωση κάτω κουμπιών.....	76
7.4	Προσθέτοντας συμπεριφορές στα εργαλεία.....	77
7.4.1	Προσθέτοντας το Touch event για να σχεδιάσετε μια τελεία (κουκίδα).....	77
7.4.2	Προσθέτοντας το γεγονός Drag που σχεδιάζει μια γραμμή.....	80
7.4.3	Προσθήκη στα κουμπιά Event Handlers.....	82
7.4.4	Τραβηγμα φωτογραφίας.....	82
7.5	Αλλαγή του μεγέθους Dot.....	83
7.5.1	Καθορισμό μιας μεταβλητής dotSize (DotSize variable).....	84
7.5.2	Χρησιμοποιώντας μεταβλητές.....	84
7.5.3	Αλλάζοντας τις τιμές των μεταβλητών.....	85
7.6	Ολοκληρωμένη η εφαρμογή PaintPot.....	86
7.7	Test εφαρμογής.....	88

Κεφάλαιο 8

8.1	Συμπεράσματα.....	91
-----	-------------------	----

Πίνακας Εικόνων

Εικόνα 1	Ο Martin Cooper από την Motorola κρατάει το MotorolaDynaTAC το πρώτο κινητό τηλέφωνο.
Εικόνα 2	Κινητά τηλέφωνα που υποστηρίζουν διάφορες πλατφόρμες.
Εικόνα 3	Λογότυπα MobilePlatforms.
Εικόνα 4	Παγκοσμίως οι πωλήσεις smartphones (27-05-2012).
Εικόνα 5	Larry Page και Sergey Brin
Εικόνα 6	Ο πρώτος server της Google
Εικόνα 7	Το λογότυπο της ΟΗΑ.
Εικόνα 8	Το λογότυπο του Android
Εικόνα 9	T-Mobile_G1 το πρώτο κινητό με Android
Εικόνα 10	Το λογότυπο του Android 1.5
Εικόνα 11	Το λογότυπο του Android 1.6
Εικόνα 12	Το λογότυπο του Android 2
Εικόνα 13	Το λογότυπο του Android 2.2
Εικόνα 14	Το λογότυπο του Android 2.3
Εικόνα 15	Το λογότυπο του Android 3
Εικόνα 16	Το λογότυπο του Android 4
Εικόνα 17	Ποσοστά των εγκατεστημένων εκδόσεων στις Android συσκευές.
Εικόνα 18	Το λογότυπο του Android 5
Εικόνα 19	Αρχιτεκτονική του Android
Εικόνα 20	Κύκλος ζωής μιας Activity του Android.
Εικόνα 21	Λογότυπο εργαλείων Android
Εικόνα 22	Λογότυπο App Inventor
Εικόνα 23	Η κεντρική οθόνη του App Inventor
Εικόνα 24	Η κεντρική οθόνη του HyperNext Android Creator (HAC)
Εικόνα 25	Η πρώτη σελίδα του App Inventor.
Εικόνα 26	Σελίδα εισαγωγής στοιχείων λογαριασμού.
Εικόνα 27	Κεντρική οθόνη του App Inventor.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

- Εικόνα 28** Το site απ' όπου μπορούμε να κατεβάσουμε τη Java.
- Εικόνα 29** Εγκατάσταση της Java.
- Εικόνα 30** Έλεγχος Java.
- Εικόνα 31** Κατέβασμα του App Inventor Setup.
- Εικόνα 32** Εκτέλεση του App Inventor Setup.
- Εικόνα 33** Ολοκλήρωση εγκατάστασης του App Inventor Setup.
- Εικόνα 34** Δημιουργία ένα νέο project.
- Εικόνα 35** Κεντρική οθόνη του App Inventor.
- Εικόνα 36** Ανοιγμα του Blocks Editor.
- Εικόνα 37** Ο Blocks Editor.
- Εικόνα 38** Ο Emulator.
- Εικόνα 39** τίτλος οθόνης εφαρμοής.
- Εικόνα 40** Στον Viewer φαίνονται τα τρία κουμπιά που δημιουργήσαμε.
- Εικόνα 41** Στον Viewer φαίνονται τα τρία κουμπιά τοποθετημένα στο Horizontal Arrangement.
- Εικόνα 42** Το εξάρτημα καμβάς τοποθετημένο στο Viewer.
- Εικόνα 43** Το user interface της εφαρμογή μας.
- Εικόνα 44** Το block DrawingCanvas.Touched.
- Εικόνα 45** Τοποθέτηση του block DrawingCanvs.DrawCircle στο block DrawingCanvas.Touched.
- Εικόνα 46** Τα values του block DrawingCanvs.DrawCircle όπως φαίνονται στο My Definitions.
- Εικόνα 47** Τα blocks x,y values τοποθετημένα στο block DrawingCanvs.DrawCircle.
- Εικόνα 48** Ο τελικός χειριστής συμβάντος του DrawingCanvas.Touched.
- Εικόνα 49** Τα arguments που εμφανίζονται στο DrawingCanvas.Dragged block.
- Εικόνα 50** Τοποθέτηση του block DrawingCanvas.DrawLine στο DrawingCanvas.Dragged.
- Εικόνα 51** Τα arguments του DrawingCanvas.DrawLine.
- Εικόνα 52** Προσθήκη στα κουμπιά Event Handlers.
- Εικόνα 53** Τα blocks των TakePictureButton.Click και Camera1.AfterPicture.
- Εικόνα 54** Αρχικοποίηση μεταβλητής.
- Εικόνα 55** Τοποθέτηση του global dotSize block στο DrawingCanvas.DrawLine.
- Εικόνα 56** Τα block αλλαγής μεταβλητών.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

- Εικόνα 57** Ο κώδικας σε block της εφαρμογής Paint Pot 'όπως φαίνεται στον Blocks Editor.
- Εικόνα 58** Η εφαρμογή Paint Pot όπως φαίνεται στην κεντρική οθόνη του App Inventor.
- Εικόνα 59** Η εφαρμογή φορτομένη στον Emulator.
- Εικόνα 60** Test εφαρμογής.
- Εικόνα 61** Test εφαρμογής.

Λίστα Πινάκων

- Πίνακας 1** Ποσοστά των εγκατεστημένων εκδόσεων στις android συσκευές.
- Πίνακας 2** Στοιβα Activity.
- Πίνακας 3** Σύγκριση HyperNext Android Creator (HAC) και Java.

Κεφάλαιο 1

1.1 Εισαγωγή

Τα τελευταία χρόνια βιώνουμε μία επανάσταση των έξυπνων κινητών τηλεφώνων (smart phones) τα οποία ενσωματώνουν πλέον δυνατότητες που δεν υπάρχουν στα συμβατικά κινητά τηλέφωνα. Τα λειτουργικά συστήματα των συσκευών αυτών ποικίλουν με το Android της Google να περιλαμβάνεται τη στιγμή αυτή στο μεγαλύτερο ποσοστό των smart phones της αγοράς.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μίας εφαρμογής που προορίζεται για τα παραπάνω κινητά τηλέφωνα. Φυσικά, η εφαρμογή μπορεί να εκτελεστεί όχι μόνο σε κινητά τηλέφωνα αλλά και σε οποιαδήποτε άλλη ηλεκτρονική συσκευή που περιλαμβάνει λειτουργικό σύστημα Android (όπως tablet pc). Μέσα από την εφαρμογή αυτή, εν ολίγοις, ο χρήστης θα μπορεί να παρεμβαίνει σε φωτογραφίες που έχει τραβήξει με την καμερα του κινητού του τηλεφώνου να ζωγραφίζει ακουμπώντας και σέρνοντας το δάκτυλο του πάνω στην οθόνη του smart phone του.

1.2 Ιστορία

Τα κινητά τηλέφωνα πρώτης γενιάς σχεδιάστηκαν και αναπτύχθηκαν από τους κατασκευαστές σταθερών τηλεφωνικών συσκευών. Ο ανταγωνισμός ήταν άγριος και τα εμπορικά μυστικά φυλάσσονταν με μεγάλη προσοχή. Οι κατασκευαστές δεν ήθελαν να αποκαλύψουν τις εσωτερικές λειτουργίες των συσκευών τους, γι' αυτό και συνήθως ανέπτυσαν οι ίδιοι το λογισμικό των τηλεφώνων τους. Οι προγραμματιστές, αν δεν συμμετείχαν σ' αυτόν τον κλειστό κύκλο, δεν είχαν καμία δυνατότητα να γράψουν εφαρμογές για τα τηλέφωνα αυτά. Τα πρώτα κινητά τηλέφωνα δεν είχαν ιδιαίτερες λειτουργίες εκτός απ' την πραγματοποίηση και λήψη κλήσεων, ακόμα μπορεί να υπήρχε μία απλή εφαρμογή επαφών που μπορούσαμε στην πραγματικότητα να χρησιμοποιήσουμε. Με τον καιρό οι πελάτες άρχισαν να απαιτούν περισσότερα χαρακτηριστικά για τα κινητά τους τηλέφωνα. Αλλά υπήρχε ένα πρόβλημα για τους κατασκευαστές τηλεφωνικών συσκευών: δεν είχαν τους απαραίτητους πόρους για την κατασκευή των εφαρμογών που επιθυμούσαν οι χρήστες. Έπρεπε να έχουν στη διάθεσή τους κάποιον τρόπο, ώστε να παρέχουν μία πύλη για υπηρεσίες ψυχαγωγίας και πληροφοριών, χωρίς να επιτρέπουν άμεση πρόσβαση στη συσκευή, και έτσι στράφηκαν στο Internet.



Εικόνα 1 Ο Martin Cooper από την Motorola κρατάει το MotorolaDynaTAC το πρώτο κινητό τηλέφωνο.

1.3 WAP

Το πρωτόκολλο ασύρματων εφαρμογών WAP (Wireless Application Protocol) αναπτύχθηκε από τον οργανισμό WAP Forum που απαρτίζεται από τους ισχυρότερους τηλεπικοινωνιακούς οργανισμούς. Το WAP ήταν μία εκδοχή του HTTP (το HTTP είναι το βασικό πρωτόκολλο του Internet) με λιγότερες δυνατότητες. Είναι μια τεχνολογία που καθιστά τις ασύρματες-κινητές συσκευές σε χρήσιμα προσωπικά "εργαλεία", μέσω των οποίων μπορεί να εξασφαλιστεί η πρόσβαση των χρηστών σε διάφορες ηλεκτρονικές πηγές πληροφοριών και υπηρεσίες, ενώ αυτοί βρίσκονται εν κινήσει. Αντίθετα από τα παραδοσιακά προγράμματα περιήγησης στο web, τα προγράμματα περιήγησης WAP σχεδιάστηκαν έτσι, ώστε να εκτελούνται μέσα στους περιορισμούς μνήμης, στη μικρή επεξεργαστική ισχύ, το μικρό εύρος ζώνης και τη χαμηλή ανάλυση που είχαν οι οθόνες των κινητών τηλεφώνων τότε. Μερικές από τις πιο δημοφιλείς εφαρμογές WAP που εμφανίστηκαν στην αγορά εκείνη την εποχή ήταν με ταπετσαρίες και ήχους κλήσης, με τους οποίους οι χρήστες μπορούσαν να εξατομικεύσουν τα τηλέφωνα τους για πρώτη φορά.

Η εμπορευματοποίηση των εφαρμογών WAP ήταν δύσκολη και δεν υπήρχε κάποιος ενσωματωμένος μηχανισμός χρέωσης. Η πληρωμή και η επικύρωση ήταν διαδικασίες που αναλάμβαναν διάφοροι μηχανισμοί παράδοσης με υψηλή χρέωση, που βόλευαν τους παρόχους υπηρεσιών όπως το SMS (Short Message Service), το EMS (Enhanced Messaging Service), το MMS (Multimedia Messaging Service) και το WAP Push.

Τα προγράμματα περιήγησης WAP, ιδιαίτερα εκείνες τις πρώτες ημέρες, ήταν αργά και ενοχλητικά. Οι οθόνες των τηλεφώνων ήταν πολύ μικρές και συχνά η περιήγηση σε σελίδες WAP ήταν ιδιαίτερα δύσκολη. Το αποτέλεσμα ήταν μία μέτρια και όχι ιδιαίτερα ενδιαφέρουσα εμπειρία για όλους τους εμπλεκόμενους (κατασκευαστές, προγραμματιστές παρόχους, χρήστες).

1.4 Πλατφόρμες κινητών τηλεφώνων

Όσο οι απαιτήσεις της αγοράς γινόταν όλο και περισσότερες τόσο η δημιουργία εφαρμογών WAP όπως πχ. τα πλούσια γραφικά και τα παιχνίδια γινόταν πιο δύσκολη. Με τη πάροδο του χρόνου η τεχνολογία πήγαινε από το καλό στο καλύτερο. Η χωρητικότητα της μνήμης γινόταν όλο και μεγαλύτερη ενώ η τιμή έπεφτε, οι επεξεργαστές όλο και πιο ισχυροί, το μέγεθος των ηλεκτρονικών εξαρτημάτων γινόταν μικρότερο και οι μπαταρίες καλύτερες με αποτέλεσμα να κερδίζουν ολοένα και περισσότερο έδαφος σε φορητές συσκευές όπως είναι τα κινητά τηλέφωνα.



Εικόνα 2 Κινητά τηλέφωνα που υποστηρίζουν διάφορες πλατφόρμες.

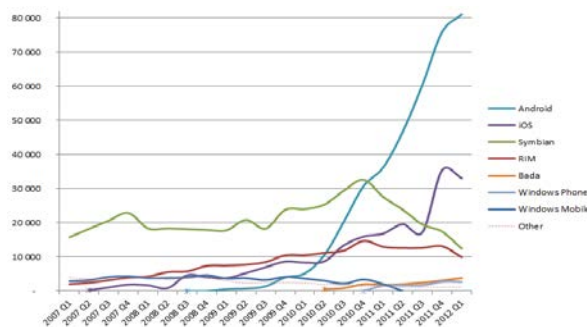
Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Οι κατασκευαστές τηλεφωνικών συσκευών συνειδητοποίησαν ότι έπρεπε να αλλάξουν τις πολιτικές προστατευτισμού που ακολουθούσαν στη σχεδίαση συσκευών και να αποκαλύψουν σε κάποιο βαθμό τις εσωτερικές λειτουργίες των τηλεφώνων τους. Έτσι εμφανίστηκαν πολλές διαφορετικές ιδιωτικές πλατφόρμες όπως το Palm OS (πλέον Garnet OS) και το RIM BlackBerry OS. Η Sun Microsystems προσάρμοσε τη δημοφιλή πλατφόρμα Java και παρουσίασε το J2ME (πλέον γνωστό ως Java Micro Edition Java ME). Η κατασκευάστρια εταιρεία επεξεργαστών Qualcomm ανέπτυξε και κατοχύρωσε την άδεια χρήσης του δικού τους περιβάλλοντος Binary Runtime Environment for Wireless (BREW). Άλλες πλατφόρμες, όπως το Symbian OS, αναπτύχθηκαν από κατασκευαστές τηλεφωνικών συσκευών όπως οι Nokia, Sony Ericsson, Motorola και Samsung. Το λειτουργικό σύστημα του Apple iPhone (OS X iPhone) έκανε την εμφάνισή του το 2008.



Εικόνα 3 Λογότυπα Mobile Platforms.

Πολλές απ' αυτές τις πλατφόρμες συσχετίζονται με προγράμματα ανάπτυξης. Αυτά τα προγράμματα διατηρούν τις κοινότητες προγραμματιστών σε μικρά και ελεγχόμενα επίπεδα, κάτω από ειδικές συμφωνίες ως προς το τι μπορούν και τι δεν μπορούν να κάνουν. Αυτά τα προγράμματα είναι συχνά υποχρεωτικά και οι προγραμματιστές πρέπει να πληρώνουν γι' αυτά. Οι κατασκευαστές και οι πάροχοι κινητής τηλεφωνίας για να αντιμετωπίσουν τον ανταγωνισμό στην αγορά πωλούν τηλέφωνα για όλες τις διαφορετικές πλατφόρμες. Από την άλλη, οι προγραμματιστές έχουν να αντιμετωπίσουν τη μεταφορά εφαρμογών από τη μια πλατφόρμα στην άλλη καθώς πρέπει να δουλέψουν σε διαφορετικά περιβάλλοντα προγραμματισμού με διαφορετικές γλώσσες και διαφορετικά εργαλεία. Συνέπεια όλων αυτών είναι θαυμάσιες εφαρμογές να μη καταφέρνουν να προσεγγίσουν τους χρήστες στους οποίους απευθύνονται και πολλές άλλες ιδέες να μην φτάνουν στο στάδιο υλοποίησης τους.



Εικόνα 4 Παγκοσμίως οι πωλήσεις smartphones (27-05-2012).

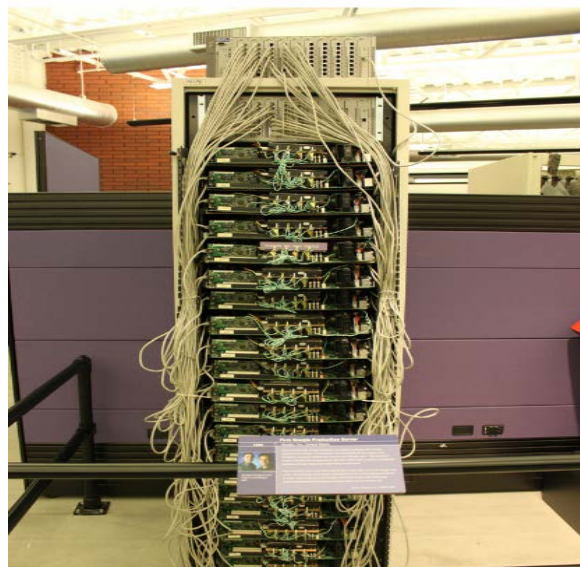
1.5 Google

Η Google είναι μια από τις μεγαλύτερες εταιρείες διαδικτυακών υπηρεσιών. Η λειτουργία της ξεκίνησε στις 27 Σεπτεμβρίου του 1998. Ο στόχος της είναι να οργανώσει όλες τις πληροφορίες του κόσμου και να τις κάνει παγκόσμια διαθέσιμες. Το Google ξεκίνησε σαν μια κολεγιακή εργασία από τον Λάριρ Πέιτζ (Larry Page) και τον Σεργκέι Μπριν (Sergey Brin) το 1996 για μια μηχανή αναζήτησης.



Εικόνα 5 Larry Page και Sergey Brin.

Η λέξη "Google" προήλθε από αναγραμματισμό της λέξης *Googol*, η οποία εκφράζει μαθηματικό όρο (τον οποίο εισήγαγε ο Milton Sirotta) και σημαίνει το «1 ακολουθούμενο από 100 μηδενικά». Με τον όρο αυτόν η Google επιθυμεί να υποδηλώσει την αποστολή της εταιρίας και να οργανώσει το τεράστιο πλήθος πληροφοριών του internet.



Εικόνα 6 Ο πρώτος server της Google

1.5.1 Η Google στον ασύρματο κόσμο

Η Google, πλέον ένα γνωστό σε όλους όνομα, έχει επιδείξει ένα ενδιαφέρον στην εξάπλωση του οράματός της, του εμπορικού ονόματός της, της πλατφόρμας αναζήτησης και διαφήμισης και του πακέτου εργαλείων της στην αγορά της ασύρματης επικοινωνίας. Το επιχειρηματικό μοντέλο της εταιρείας έχει τρομακτική επιτυχία στο Internet. Ένας από τους στόχους της Google ήταν να μεταφέρει τις υπηρεσίες που απολάμβαναν οι χρήστες του internet στους συνδρομητές κινητής τηλεφωνίας. Οι αρχικές απόπειρες της εταιρείας στην αγορά των κινητών τηλεφώνων στιγματίστηκαν από πολλά προβλήματα. Με τη φιλοσοφία της να επικεντρώνεται στο χρήστη και τη σχεδίαση της να ακολουθεί δημοκρατικά πρότυπα, η Google ηγείται ενός κινήματος που επιχειρεί να μετατρέψει την υπάρχουσα κλειστή και φυλασσομένη ασύρματη αγορά σε μία αγορά όπου οι χρήστες τηλεφώνων μπορούν να μετακινούνται από τον ένα φορέα στον άλλο και να έχουν ελεύθερη πρόσβαση σε εφαρμογές και υπηρεσίες. Με τους ατέλειωτους πόρους της, η Google έχει ακολουθήσει μία ευρεία προσέγγιση εξετάζοντας την ασύρματη υποδομή απ' την οπτική του FCC (η Αμερικανική επιτροπή επικοινωνιών) για τις απαιτήσεις από τους κατασκευαστές τηλεφωνικών συσκευών, τις ανάγκες των προγραμματιστών εφαρμογών και τις επιθυμίες των παρόχων κινητής τηλεφωνίας. Στη συνέχεια, η Google στάθηκε στο πλευρό άλλων ομοϊδεατών μελών της ασύρματης κοινότητας και έθεσε το εξής ερώτημα: Τι θα έπρεπε να γίνει, ώστε να κατασκευάσουμε ένα καλύτερο κινητό τηλέφωνο;

1.6 Ανοιχτή Συμμαχία Μικροτηλεφώνων (Open Handset Alliance)

Ο οργανισμός Open Handset Alliance (OHA) σχηματίστηκε το Νοέμβριο του 2007 με σκοπό να δώσει μία απάντηση σ' αυτό το ερώτημα. Στις 5 Νοεμβρίου 2007, η Open Handset Alliance, μια κοινοπραξία διάφορων επιχειρήσεων που περιλαμβάνουν κατασκευαστές επεξεργαστών, κατασκευαστές τηλεφωνικών συσκευών, προγραμματιστές λογισμικού και παρόχους υπηρεσιών όπως τις Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, NVIDIA, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, και Texas Instruments, παρουσιάστηκε με το στόχο να αναπτύξει ανοιχτά πρότυπα (open standards) για τις κινητές συσκευές. Μαζί με το σχηματισμό της η OHA παρουσίασε επίσης το πρώτο προϊόν τους, το Android, μια πλατφόρμα κινητών συσκευών που στηρίχτηκε στο πυρήνα Linux 2.6.



Εικόνα 7 Το λογότυπο της OHA.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Ο Andy Rubin θεωρείται ο πατέρας της πλατφόρμας Android. Η εταιρεία του, Android Inc εξαγοράστηκε από τη Google το 2005. Στις 9 Δεκεμβρίου 2008, αναγγέλθηκε ότι ακόμα 14 νέα μέλη θα συμμετείχαν στο πρόγραμμα του Android συμπεριλαμβανομένων: ARM Holdings Plc, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp και Vodafone. Τα μέλη του OHA, σε συνεργασία με τη Google ξεκίνησαν να αναπτύσσουν μία μη ιδιωτική ανοιχτή πρότυπη πλατφόρμα υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού η οποία βασίζεται σε τεχνολογία της Android Inc. που στοχεύει στην αντιμετώπιση όλων αυτών των προβλημάτων που μαστίζουν την κοινότητα των κινητών τηλεφώνων. Το αποτέλεσμα είναι το Android.

1.7 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα και μία πλατφόρμα λογισμικού πάνω στην οποία αναπτύσσονται εφαρμογές. Ένα βασικό σύνολο από εφαρμογές που συμπεριλαμβάνονται στις συσκευές Android για καθημερινές εργασίες, όπως περιήγηση στο Web και email. Ως προϊόν του οράματος του OHA (Open Handset Alliance) για ένα ανθεκτικό και ανοιχτό περιβάλλον ανάπτυξης για ασύρματες εφαρμογές, το Android είναι μία αναδυόμενη πλατφόρμα ανάπτυξης εφαρμογών κινητών τηλεφώνων. Η πλατφόρμα σχεδιάστηκε για τον αποκλειστικό σκοπό της ενθάρρυνσης μίας ελεύθερης και ανοιχτής αγοράς, που θα μπορούσαν να απολαμβάνουν όλοι οι χρήστες εφαρμογές κινητών τηλεφώνων και που θα μπορούσαν να αναπτύξουν όλοι οι προγραμματιστές λογισμικού.

1.7.1 Η μασκώτ του Android

Η μασκώτ του Android είναι ένα μικρό πράσινο ρομπότ, το οποίο βλέπετε στην Εικόνα 8. Αυτός ο μικρός τύπος (αγόρι ή κορίτσι) συχνά χρησιμοποιείται για την απεικόνιση πραγμάτων που έχουν σχέση με το Android.



Εικόνα 8 Το λογότυπο του Android

1.8 Μία πλατφόρμα νέας γενιάς

Αν και το Android διαθέτει πολλά καινοτόμα χαρακτηριστικά, τα οποία δεν διαθέτουν άλλες πλατφόρμες κινητών τηλεφώνων, οι σχεδιαστές του εκμεταλλεύονται επίσης πολλές δοκιμασμένες προσεγγίσεις, οι οποίες λειτουργούν αποδεδειγμένα στον ασύρματο κόσμο. Πολλά απ' αυτά τα χαρακτηριστικά εμφανίζονται σε υπάρχουσες ιδιωτικές πλατφόρμες, αλλά το Android τα συνδυάζει μ' έναν ελεύθερο και ανοιχτό τρόπο, ενώ ταυτόχρονα λύνει πολλά προβλήματα που έχουν αυτές οι ανταγωνιστικές πλατφόρμες. Το Android είναι η πρώτη πλατφόρμα μίας νέας γενιάς πλατφόρμων κινητών τηλεφώνων, κάτι που χαρίζει στους προγραμματιστές του ένα συγκριτικό πλεονέκτημα. Οι σχεδιαστές του Android εξέτασαν τα πλεονεκτήματα και τα μειονεκτήματα υπάρχοντων πλατφόρμων και κατόπιν ενσωμάτωσαν τα πιο πετυχημένα χαρακτηριστικά τους. Ταυτόχρονα, οι σχεδιαστές του απέφυγαν τα λάθη του παρελθόντος. Από την κυκλοφορία του Android 1.0 SDK και μετά, η ανάπτυξη της πλατφόρμας Android έχει ακολουθήσει μία πολύ γρήγορη πορεία. Για αρκετό καιρό, εμφανιζόταν ένα νέο Android SDK κάθε δύο μήνες. Κάθε SDK συνοδευόταν από ένα όνομα έργου και στην περίπτωση του Android, τα SDK παίρνουν ονόματα γλυκών στα Αγγλικά, τα οποία ακολουθούν αλφαβητική σειρά.

1.8.1 Δωρεάν Ανοιχτή Ολοκληρωμένη Πλατφόρμα

Η ανάπτυξη εφαρμογών Android γίνεται δωρεάν. Δεν απαιτείται η καταβολή χρηματικών ποσών για άδειες χρήσης και πνευματικά δικαιώματα ώστε να προγραμματίσει κάποιος σ' αυτήν την πλατφόρμα. Δεν ζητούνται επίσης πασά για να γίνει κάποιος μέλος της κοινότητας προγραμματιστών. Ούτε καταβάλλεται κανένα ποσό για τις δοκιμές. Κανένα ποσό δεν δίνεται για υπογραφές ή πιστοποιήσεις.

Η πλατφόρμα Android παρέχεται μέσω της διαδικασίας ανοιχτής πηγής. Οι προγραμματιστές απολαμβάνουν ανεμπόδιστη πρόσβαση στα χαρακτηριστικά των συσκευών όταν αναπτύσσουν εφαρμογές.

Οι σχεδιαστές ακολούθησαν μία περιεκτική προσέγγιση στη σχεδίαση της πλατφόρμας Android. Ξεκίνησαν μ' ένα ασφαλές λειτουργικό σύστημα και κατασκεύασαν ένα στιβαρό πλαίσιο λογισμικού το οποίο επιτρέπει την πλούσια ανάπτυξη εφαρμογών.

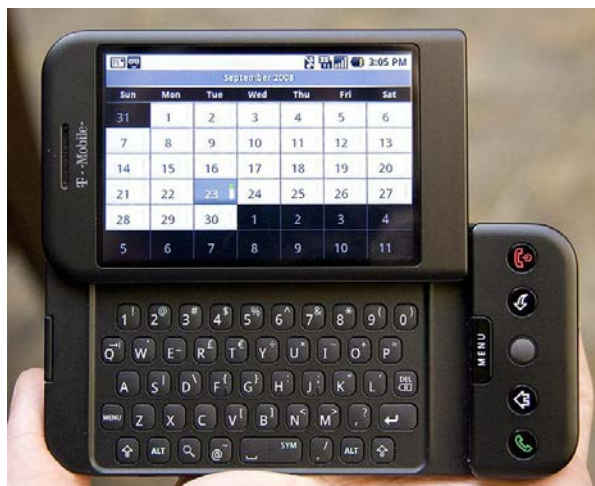
Κεφάλαιο 2

2.1 Η πλατφόρμα Android εξελίσσεται

Κάθε νέα έκδοση του Android SDK παρέχει πολλές ουσιαστικές βελτιώσεις στην πλατφόρμα. Σε πρόσφατες αναθεωρήσεις, η πλατφόρμα Android έχει δεχθεί το απαραίτητο «γυάλισμα» του περιβάλλοντος εργασίας, τόσο όσον αφορά στην οπτική πλευρά του, αλλά και στις επιδόσεις του. Αν και οι περισσότερες αναβαθμίσεις και βελτιώσεις είναι καλοδεχούμενες και απαραίτητες, οι νέες εκδόσεις SDK συχνά προκαλούν κάποια αναστάτωση στην κοινότητα προγραμματιστών του Android. Πολλές δημοσιευμένες εφαρμογές απαιτούν νέες δοκιμές και νέες υποβολές στην αγορά του Android ώστε να συμμορφώνονται με τις προδιαγραφές των νέων SDK, οι οποίες γρήγορα επεκτείνονται στην πράξη σ' όλα τα τηλέφωνα Android με τη μορφή αναβάθμισης του firmware (υλικό λογισμικού) και καθιστούν τις παλαιότερες εφαρμογές απαρχαιωμένες. Ορισμένες παλαιότερες συσκευές Android δεν μπορούν να εκτελέσουν τις τελευταίες εκδόσεις της πλατφόρμας. Αυτό σημαίνει ότι οι προγραμματιστές Android πρέπει συχνά να λαμβάνουν υπόψη τους πολλές διαφορετικές εκδόσεις SDK αν θέλουν να καλύπτουν όλους τους χρήστες. Ευτυχώς, η ανάπτυξη με τα εργαλεία του Android διευκολύνει τη διαδικασία περισσότερο από ποτέ.

2.2 Εκδόσεις και χαρακτηριστικά

Η πρώτη έκδοση του Android SDK έγινε τον Νοέμβριο του 2007. Πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην πραγματικότητα όμως, το Android παρουσίαζε προβλήματα τα οποία παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (SmartPhone), βασισμένο στην πλατφόρμα του Android.



Εικόνα 9 T-Mobile_G1 το πρώτο κινητό με 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.

2.3 Android 1.5 Cupcake

Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5, την ονόμασαν Cupcake, η οποία εισάγει κάποια καινούργια χαρακτηριστικά και ανανεώσεις στη διεπιφάνεια χρήστη (User Interface).



Εικόνα 10 Το λογότυπο του Android 1,5.

2.3.1 Χαρακτηριστικά Cupcake:

- 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.

2.4 Android 1.6 Donut

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



Εικόνα 11 Το λογότυπο του Android 1,6.

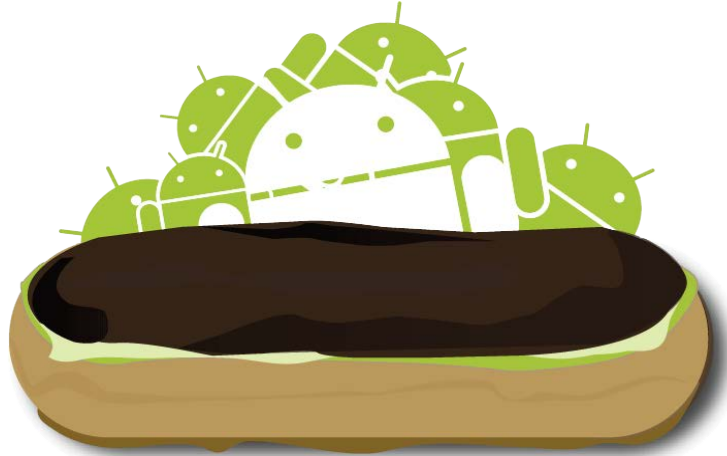
2.4.1 Χαρακτηριστικά Donut:

Η έκδοση Donut ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

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

2.5 Android 2.0 Éclair

Το Android 2.0 το ονόμασαν Éclair. Ήρθε τον Σεπτέμβριο του 2009. Με τις επανεκδόσεις του σε Android 2.0.1 τον Δεκέμβριο 2009 (Éclair 0.1) και τον Ιανουάριο 2010 με το Android 2.1 (Éclair MR1).



Εικόνα 12 Το λογότυπο του Android 2.

2.5.1 Χαρακτηριστικά Éclair:

Η έκδοση Éclair ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

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

2.6 Android 2.2 Froyo

Το Android 2.2 ήρθε τον Μάιο του 2010. Το ονόμασαν Froyo.



Εικόνα 13 Το λογότυπο του Android 2.2

2.6.1 Χαρακτηριστικά Froyo:

Η έκδοση Froyo ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

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

2.7 Android 2.3 Gingerbread

Το Android 2.3 ήρθε τον Δεκέμβριο του 2010. Το ονόμασαν Gingerbread. Έγινε επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.



Εικόνα 14 Το λογότυπο του Android 2,3

2.7.1 Χαρακτηριστικά Gingerbread:

Η έκδοση Gingerbread ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

- Βελτιωμένο UI για απλότητα και ταχύτητα.
- Πιο γρήγορη, πιο διασθητική 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.

2.8 Android 3.0 Honeycomb

Το Android 3.0 ήρθε τον Φεβρουάριο του 2011 λίγες μέρες μετά την επανέκδοση του Android 2.3.3, και προορίζεται αποκλειστικά για ταμπλέτες. Το ονόμασαν Honeycomb.



Εικόνα 15 Το λογότυπο του Android 3

2.8.1 Χαρακτηριστικά Honeycomb:

- 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.

2.9 Android 4 Ice Cream Sandwich

Το Android Ice Cream Sandwich ήρθε τον Νοέμβριο του 2011. Αποτελεί την προσπάθεια ενιαίου λειτουργικού συστήματος για όλες τις συσκευές. Το Android Ice Cream Sandwich φέρνει μαζί του “ολογραφικό” 3D UI, πλουσιότερα widgets, ενισχυμένο multitasking, νέα APIs και φυσικά θα είναι πλήρως open source. Για να πάρουμε μια καλή γεύση του Android Ice Cream Sandwich, η Google παρουσίασε τη λειτουργία face tracking, η οποία δεν περιορίζεται μόνο στις κινήσεις του κεφαλιού, αλλά ακολουθεί τα μάτια, τη μύτη και το στόμα. Στην περίπτωση που πραγματοποιείται video κλήση, “ζουμάρει” αυτόματα προς αυτόν που μιλά εκείνη τη στιγμή.



Εικόνα 16 Το λογότυπο του Android 4

2.9.1 Χαρακτηριστικά Ice Cream Sandwich:

- Χρησιμοποιείται σε Smartphone's, tablets, laptops, netBooks κλπ.
- “Ολογραφικό” 3D UI.
- Πλουσιότερα widgets.
- Ενισχυμένο multitasking.
- Νέα APIs.
- Είναι πλήρως open source
- Υποστηρίζει ένα ενιαίο σύνολο στοιχείων UI, στυλ και δυνατότητες για κινητά τηλέφωνα, tablets, και άλλες συσκευές.
- Πλούσια επικοινωνία με νέο ημερολόγιο API, Android Beam για NFC βάση, Wi-Fi άμεση υποστήριξη, και Bluetooth για υποστήριξη συσκευών Υγείας.
- Βαθιά διαδραστικότητα και παραμετροποίηση, κλείδωμα οθόνης με κάμερα και ηχητικό έλεγχο και βελτίωση της διαχείρισης του προγράμματος εκκίνησης.
- Ακόμα καλύτερη εικόνα εφέ και βίντεο με κάμερα ακριβείας και εφαρμογές για ανίχνευση προσώπου και κινήσεων.
- hardware-accelerated 2D σχεδίασης, νέο grid-based layout, βελτίωση εφαρμογής πληκτρολογίου και ποντικιού, spell-checker API, και γραφίδα υποστήριξης εισόδου.

2.10 Android 4.1 Jelly Bean



Εικόνα 18 Το λογότυπο του Android 5

2.10.1 Χαρακτηριστικά Jelly Bean:

Νέο User Interface

Υπάρχουν νέοι τρόποι να εμφανίζεται το User Interface, με μεγαλύτερες ειδοποιήσεις, υποστήριξη περισσότερων μορφών περιεχομένου στο Notifications bar, με τις apps να μπορούν να συνοδευτούν από έως και τρεις διαφορετικές ειδοποιήσεις. Μερικά από τα βασικά στοιχεία είναι:

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

Οι νέες μορφές εισαγωγής κειμένου και οι δυνατότητες που προσφέρουν

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

Animation και Γραφικά

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

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Νέες μορφές συνδεσιμότητας

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

Νέες δυνατότητες Πολυμέσων

- Πρόσβαση σε Media codecs
- Έξοδος USB Audio
- Έναρξη ηχογράφησης Audio
- Πολυκάναλος ήχος και υποστήριξη κωδικοποίησης/αποκωδικοποίησης ήχου AAC 5.1
- Προεπεξεργασία ήχου
- Συνεχής Αναπαραγωγή χωρίς παύσεις
- Media Router

RenderScript Computation

Με τη δυνατότητα αυτή οι developers έχουν μεγαλύτερη ευελιξία στα scripts, ενώ μπορούν να εφαρμόσουν ταχύτερη vector επεξεργασία και άλλα στοιχεία που δεν ήταν εφικτά με το IEEE 754-2008 standard.

Android Browser και WebView

Τα παρακάτω στοιχεία περιλαμβάνονται στην web λειτουργία του Android 4.1:

- Καλύτερη εμπειρία HTML5 video, με touch-to-play/pause και ομαλή μετάβαση σε full screen mode.
- Μεγαλύτερες ταχύτητες rendering και μειωμένη χρήση μνήμης
- Καλύτερη απόδοση στο HTML5/CSS3/Canvas animation
- Βελτιωμένη εισαγωγή κειμένου
- Ανανεωμένη JavaScript Engine
- Υποστήριξη HTML5 Media Capture

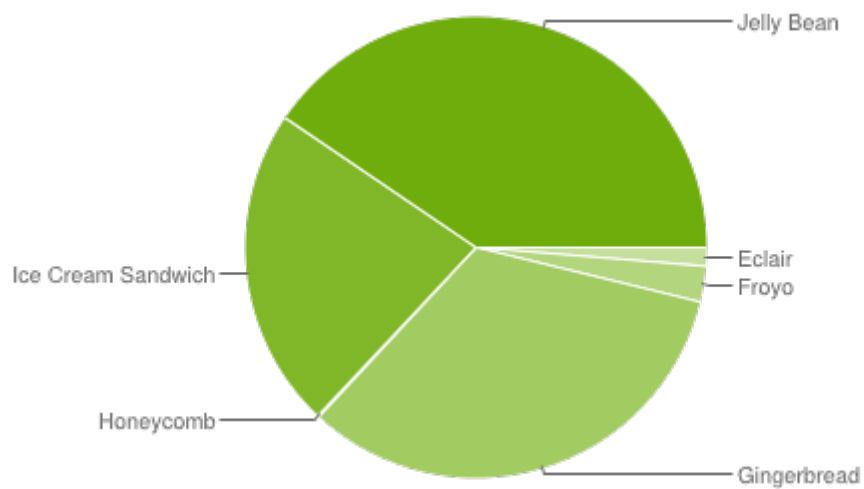
Google APIs και υπηρεσίες

Διαθέσιμες είναι οι ακόλουθες υπηρεσίες:

- Google Cloud Messaging for Android
- Κρυπτογράφηση εφαρμογών
- Έξυπνα App Updates
- Google Play services, όπως η πιστοποίηση και η ενσωμάτωση του Google+ στις εφαρμογές

2.11 Στατιστικά

Ένα ενδιαφέρον στατιστικό που αξίζει να δούμε είναι τα ποσοστά των Android εκδόσεων που είναι εγκατεστημένες σε όλες τις Android συσκευές όπως ανακοινώθηκαν το Μάιο του 2012.



Εικόνα 17 Ποσοστά των εγκατεστημένων εκδόσεων στις Android συσκευές.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

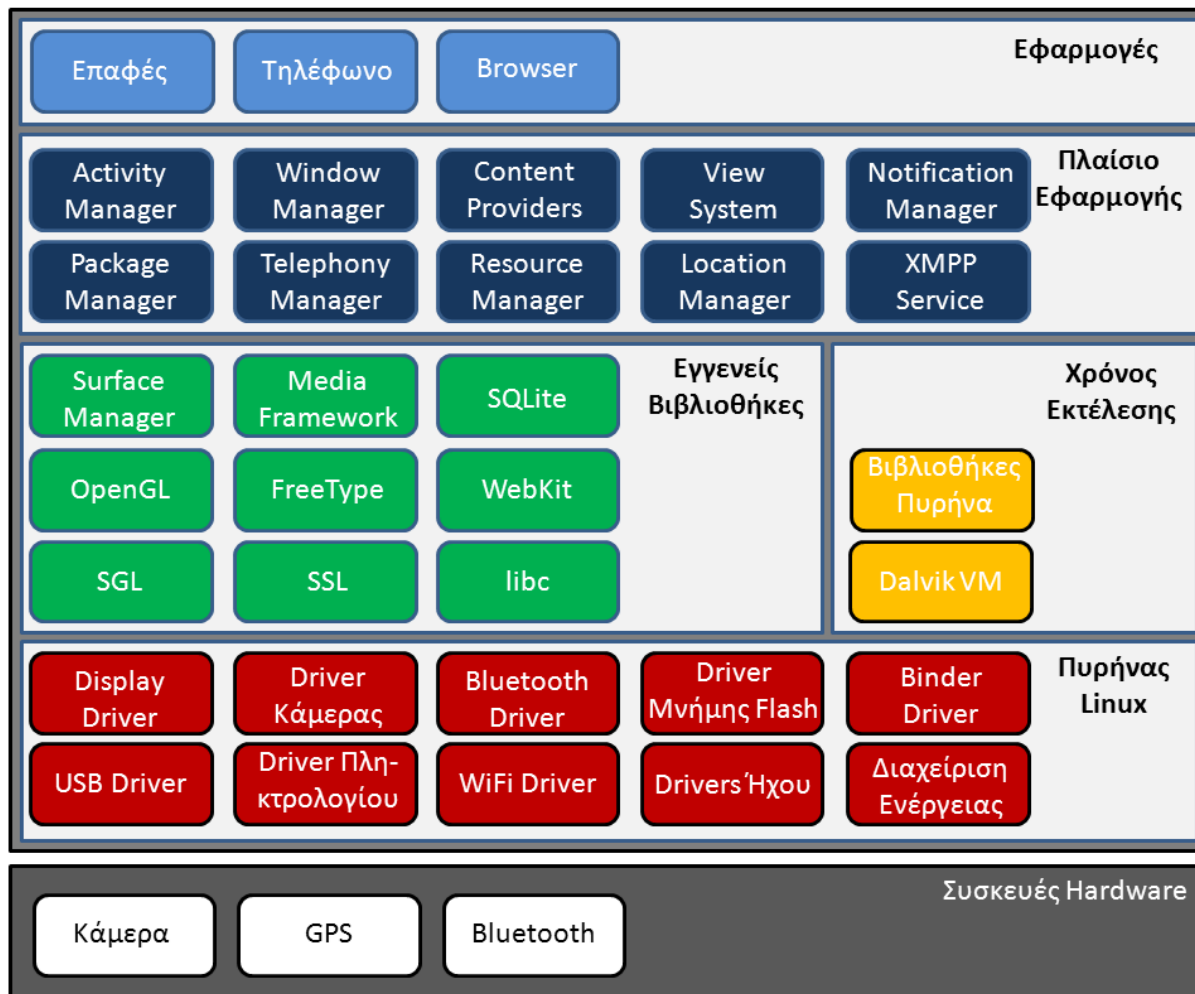
Πλατφόρμα	Κωδική ονομασία	Ποσοστό
Android 1.6	Donut	0.1%
Android 2.1	Éclair	1.2%
Android 2.2	Froyo	2.5%
Android 2.3 Android 2.3.2	Gingerbread	0.1%
Android 2.3.3 Android 2.3.7		33.0%
Android 3.2	Honeycomb	0.1%
Android 4.0.3 Android 4.0.4	Ice Cream Sandwich	22.5%
Android 4.1	Jelly Bean	34.0%
Android 4.2		6.5%

Πίνακας 4 Ποσοστά των εγκατεστημένων εκδόσεων στις android συσκευές. (εκδόσεις με λιγότερο από 0,1% δεν δείχνονται).

Κεφάλαιο 3

3.1 Αρχιτεκτονική του Android

Η αρχιτεκτονική του Android περιλαμβάνει τα εξής επίπεδα, αρχίζοντας από το ψηλότερο και πηγαίνοντας στο χαμηλότερο:



Εικόνα 19 Αρχιτεκτονική του Android

3.2 Επίπεδο Εφαρμογών (Applications - widget)

Το Android είναι εξ αρχής εφοδιασμένο με ένα σύνολο από βασικές εφαρμογές που περιλαμβάνουν ένα email client, ένα πρόγραμμα για SMS μηνύματα, ημερολόγιο, χάρτες (Google Maps), περιηγητή ιστού, πρόγραμμα για δομημένη αποθήκευση των επαφών και άλλα. Όλες οι εφαρμογές είναι γραμμένες στην γλώσσα προγραμματισμού Java. Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Τα widget λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή.

3.3 Επίπεδο Πλαισίου Εφαρμογών (Applications Framework)

Παρέχοντας μια ανοικτή πλατφόρμα ανάπτυξης, το Android προσφέρει στους προγραμματιστές την δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές αφήνονται ελεύθεροι να εκμεταλλευτούν πλήρως το hardware της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να τρέξουν υπηρεσίες στο background, να θέσουν χρονοδιακόπτες για εμφάνιση ειδοποιήσεων και πολλά άλλα. Επίσης, έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο από APIs που έχουν οι βασικές εφαρμογές του Android. Η αρχιτεκτονική είναι διαμορφωμένη με τέτοιο τρόπο που κάθε εφαρμογή μπορεί να χρησιμοποιήσει τις δυνατότητες μιας άλλης και επίσης με τρόπο που δίνει την δυνατότητα στο χρήστη να αλλάξει τα συστατικά κάθε εφαρμογής. Κάτω από το πλαίσιο των εφαρμογών υπάρχει ένα σύστημα από υπηρεσίες και συστήματα τα οποία περιλαμβάνουν:

- Ένα σύνολο από γραφικά στοιχεία (Views) για την δημιουργία γραφικού περιβάλλοντος συμπεριλαμβανομένων λιστών (lists), πλεγμάτων (grids), κουτιών κειμένου (text boxes), κουμπιών (buttons) και άλλων.
- Ένα διαχειριστή περιεχομένου (Content Manager) ο οποίος επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τον διαμοιρασμό των δικών τους δεδομένων με άλλες εφαρμογές.
- Ένα διαχειριστή πόρων (Resource Manager) για την πρόσβαση στους πόρους όπως strings, εικόνες, layout files.
- Έναν διαχειριστή ειδοποιήσεων (Notification Manager) ο οποίος επιτρέπει την προβολή ειδοποιήσεων στην μπάρα κατάστασης (status bar).
- Έναν διαχειριστή δραστηριοτήτων (Activity Manager) ο οποίος διαχειρίζεται τον κύκλο ζωής των εφαρμογών.

3.4 Επίπεδο Βιβλιοθηκών (Libraries)

Το επίπεδο βιβλιοθηκών περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ οι οποίες χρησιμοποιούνται από διάφορα στοιχεία του συστήματος του Android. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να σταθούν 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). Οι δυνατότητες που προσφέρουν αυτές οι βιβλιοθήκες είναι προσβάσιμες στους προγραμματιστές δια μέσου του επιπέδου πλαισίου εφαρμογής.

3.5 Επίπεδο Εκτέλεσης (Android Runtime):

Στο ίδιο επίπεδο με τις βιβλιοθήκες, το Android Runtime παρέχει ένα σύνολο βασικών βιβλιοθηκών που επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας JAVA. Επίσης, περιλαμβάνει την Dalvik Virtual Machine, που επιτρέπει σε κάθε εφαρμογή να τρέξει την δικιά της εργασία, μαζί με την δικιά της ξεχωριστή Dalvik Virtual Machine.

3.6 Dalvik Virtual Machine

Το Android αποτελείται από 4 επίπεδα και 5 ομάδες. Μία απ' αυτές τις ομάδες είναι η Dalvik Virtual Machine. Η “μηχανή” αυτή είναι υπεύθυνη για την δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να “τρέξει” το λειτουργικό σύστημα.

Κάθε εφαρμογή του Android είναι γραμμένη σε γλώσσα Java, την οποία το λειτουργικό σύστημα δεν την αντιλαμβάνεται απευθείας. Για τον λόγο αυτό, η Dalvik Virtual Machine αναλαμβάνει τη δημιουργία των εκτελέσιμων αρχείων .dex (Dalvik Executable) τα οποία εκτελούνται από το λειτουργικό σύστημα. Κάθε εκτελέσιμο πρόγραμμα εκτελείται από την δική του Virtual Machine, ακόμα και όταν εκτελούνται παράλληλα, με αποτέλεσμα τα διαφορετικά προγράμματα να μην επηρεάζουν το ένα το άλλο και σε περίπτωση που προκύψει κάποιο σφάλμα σε ένα απ' αυτά, να μην προκαλέσει προβλήματα στα υπόλοιπα.

3.7 Πυρήνας του Linux:

Το Android βασίζεται στον πυρήνα Linux έκδοση 2.6 για βασικές υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση μνήμης, διαχείριση διεργασιών, στοίβα δικτύου, και οδηγούς συσκευών. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.

3.8 Βιβλιοθήκες Android

Με το Android, οι προγραμματιστές εφαρμογών κινητών τηλεφώνων δεν χρειάζεται πλέον να εφεύρουν ξανά τον τροχό. Αντίθετα, οι προγραμματιστές χρησιμοποιούν οικείες βιβλιοθήκες κλάσεων οι οποίες διατίθενται μέσω των πακέτων java του Android για την εκτέλεση κοινών εργασιών, όπως γραφικά, προσπέλαση βάσεων δεδομένων, προσπέλαση δικτύων, ασφαλείς επικοινωνίες και βοηθητικά προγράμματα (όπως η συντακτική ανάλυση XMI). Μερικές από τις βιβλιοθήκες που είναι διαθέσιμες και υποστηρίζονται από όλες τις συσκευές Android είναι:

- **android.util** - Το πακέτο βοηθήματος (utility) πυρήνων περιέχει χαμηλού επιπέδου κλάσεις όπως τα εξειδικευμένα containers, μορφοποιητές (formatters) συμβολοσειρών, και βοηθήματα για ανάλυση (parsing) XML αρχείων.
- **android.os** - Το πακέτο λειτουργικού συστήματος παρέχει την πρόσβαση στις βασικές υπηρεσίες του λειτουργικού συστήματος όπως τη διαβίβαση μηνυμάτων, την πραγματοποίηση κλήσεων, τις λειτουργίες ρολογιού, και την απασφαλμάτωση (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 δίνεται η 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.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

- **android.content** - Το API περιεχομένου (content) χρησιμοποιείται για να διαχειριστεί την πρόσβαση στα δεδομένα και την έκδοση 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 cookie.

Βιβλιοθήκες C/C++:

- **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.

Προηγμένες βιβλιοθήκες Android:

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Αυτές οι βιβλιοθήκες προσφέρουν συναρπαστικές λειτουργίες και ποικίλουν ανάλογα με τις συσκευές πάνω στις οποίες εφαρμόζονται.

- **android.location** - Το τοποκεντρικό API (location-based) δίνει πρόσβαση των εφαρμογών σε πληροφορίες για την τρέχουσα φυσική θέση της συσκευής. Οι υπηρεσίες βασισμένες στην τοποθεσία παρέχουν πρόσβαση σε πληροφορίες τοποθεσίας χρησιμοποιώντας οποιοδήποτε υλικό ή τεχνολογία είναι διαθέσιμη στη συσκευή και βοηθάει στον προσδιορισμό θέσης.
- **android.media** - Τα APIs πολυμέσων παρέχουν υποστήριξη για την αναπαραγωγή και εγγραφή αρχείων ήχου και βίντεο, συμπεριλαμβανομένων 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 επιταχυμέτρου (accelerometer), και αισθητήρων πυξίδας.
- **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.

Κεφάλαιο 4

4.1 Ανάπτυξη εφαρμογών Android

Το Android SDK παρέχει ένα εκτεταμένο σύνολο περιβαλλόντων προγραμματισμού εφαρμογών (API), τα οποία είναι σύγχρονα και ανθεκτικά. Οι βασικές υπηρεσίες συστήματος για συσκευές Android είναι προσβάσιμες απ' όλες τις εφαρμογές. Όταν τους έχουν χορηγηθεί τα κατάλληλα δικαιώματα, οι εφαρμογές Android μπορούν να μοιράζονται δεδομένα μεταξύ τους και να προσπελαίνουν κοινόχρηστους πόρους του συστήματος με ασφάλεια.

Αντίθετα από άλλες πλατφόρμες ανάπτυξης εφαρμογών κινητών τηλεφώνων, στην πλατφόρμα Android δεν γίνεται καμία διάκριση μεταξύ των εσωτερικών, έμφυτων εφαρμογών και των εφαρμογών που δημιουργούν ανεξάρτητοι προγραμματιστές. Εφόσον η εφαρμογή έχει λάβει τα κατάλληλα δικαιώματα, όλες οι εφαρμογές έχουν την ίδια πρόσβαση σε βασικές βιβλιοθήκες και στις υποκείμενες διεπαφές υλικού. Οι συσκευές Android περιλαμβάνουν ένα πακέτο από εσωτερικές εφαρμογές, όπως το πρόγραμμα περιήγησης στο web και το πρόγραμμα διαχείρισης επαφών. Τρίτες εφαρμογές μπορούν να ενοποιηθούν μ' αυτές τις βασικές εφαρμογές, να τις επεκτείνουν, ώστε να παρέχουν μία πλούσια εμπειρία χρήστη ή να τις αντικαταστήσουν εντελώς με εναλλακτικές εφαρμογές.

4.2 Η ανατομία εφαρμογών Android

Η ανάπτυξη εφαρμογών Android για κινητά τηλέφωνα με περιορισμένους πόρους απαιτεί μία αναλυτική κατανόηση του κύκλου ζωής της εφαρμογής. Το Android επίσης χρησιμοποιεί δική του ορολογία γι' αυτά τα στοιχεία κατασκευής εφαρμογών και παρέχει πιο διεξοδική ανάλυση του τρόπου, που οι εφαρμογές Android λειτουργούν και αλληλεπιδρούν μεταξύ τους. Κάποιοι από τους σημαντικούς πόρους που καλύπτονται σ' αυτό το κεφάλαιο είναι: Context (Περιβάλλον), Activity (Δραστηριότητα) Intent (Πρόθεση) Service (Υπηρεσία).

4.3 Context (περιβάλλον)

Το Context της εφαρμογής είναι η κεντρική θέση για όλες τις λειτουργίες εφαρμογών ανωτέρου επιπέδου. Η κλάση Context μπορεί να χρησιμοποιηθεί για τη διαχείριση λεπτομερειών, ρύθμισης παραμέτρων για συγκεκριμένες εφαρμογές, όπως και για λειτουργίες και δεδομένα για μία ολόκληρη εφαρμογή. Επίσης, μπορούμε να χρησιμοποιήσουμε το Context της εφαρμογής για να προσπελάσουμε ρυθμίσεις και πόρους που μοιράζονται σε πολλαπλά στιγμιότυπα Activity.

Προσπέλαση άλλων λειτουργιών εφαρμογών με τα context:

Το Context της εφαρμογής παρέχει δυνατότητα πρόσβασης σε αρκετά χαρακτηριστικά εφαρμογής κορυφαίου επιπέδου. Μερικά πράγματα που μπορούμε να κάνουμε με το context της εφαρμογής είναι:

- Εκκίνηση στιγμιότυπων Activity.
- Ανάκτηση πόρων που συσκευάζονται μαζί με την εφαρμογή.
- Αίτηση υπηρεσίας συστήματος (π.χ., υπηρεσία Θέσης).
- Διαχείριση ιδιωτικών αρχείων, καταλόγων και βάσεων δεδομένων εφαρμογών.
- Επιθεώρηση και επιβολή δικαιωμάτων εφαρμογών.

4.4 Activity (Δραστηριότητα)

Η κλάση Activity του Android (`android.app.Activity`) αποτελεί θεμέλιο για οποιαδήποτε εφαρμογή Android. Τις περισσότερες φορές ορίζεται και υλοποιείται μία κλάση Activity για κάθε οθόνη στην εφαρμογή.

Ο κύκλος ζωής μίας Activity:

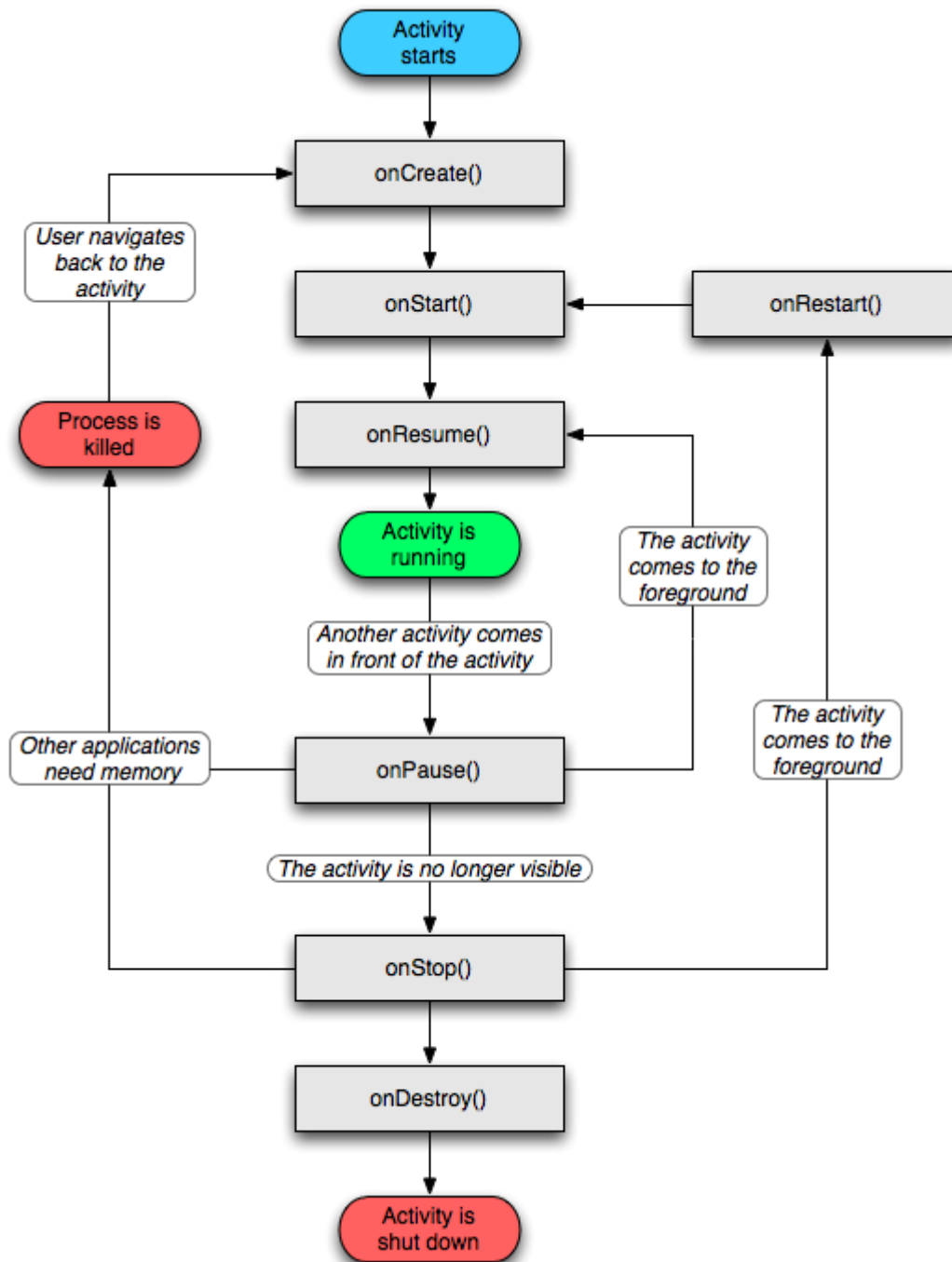
Οι εφαρμογές Android μπορούν να περιλαμβάνουν πολλαπλές διεργασίες και το λειτουργικό σύστημα Android επιτρέπει σε πολλαπλές εφαρμογές να εκτελούνται ταυτόχρονα, αρκεί να υπάρχει διαθέσιμη μνήμη και επεξεργαστική ισχύς. Οι εφαρμογές μπορούν να έχουν διεργασίες στο παρασκήνιο και οι εφαρμογές μπορούν να διακόπτονται και να παύονται όταν υπάρχουν συμβάντα, όπως τηλεφωνικές κλήσεις. Μπορεί να υπάρχει μόνο μία ενεργή εφαρμογή η οποία να είναι ορατή στο χρήστη κάθε φορά. Συγκεκριμένα, σε οποιαδήποτε δεδομένη στιγμή, υπάρχει μόνο μία Activity εφαρμογής στο προσκήνιο. Το λειτουργικό σύστημα Android παρακολουθεί όλα τα αντικείμενα Activity και τα τοποθετεί σε μία στοίβα (στοίβα Activity πίνακας 2). Όταν μία νέα Activity ξεκινά, η Activity στην κορυφή της στοίβας (η Activity που υπάρχει αυτήν τη στιγμή στο προσκήνιο) διακόπτεται προσωρινά και η νέα Activity προωθείται στην κορυφή της στοίβας. Όταν η Activity σταματά, αυτή η Activity αφαιρείται απ' τη στοίβα δραστηριοτήτων και η προηγούμενη Activity της στοίβας ξεκινά πάλι.

Είναι η πιο πάνω δραστηριότητα. Ο χρήστης μπορεί να δει και να αλληλεπιδράσει με αυτήν.
Είναι η δεύτερη δραστηριότητα στη στοίβα. Εάν ο χρήστης πατήσει το Πίσω ή η πιο πάνω δραστηριότητα καταστραφεί, ο χρήστης μπορεί να τη δει και να αλληλεπιδράσει μαζί της ξανά.
Είναι μία δραστηριότητα στη μέση της στοίβας. Οι χρήστες δεν μπορούν να δουν και να αλληλεπιδράσουν μαζί της μέχρι να καταστραφούν όλες οι πιο πάνω από αυτήν.
Είναι μια δραστηριότητα στο κάτω μέρος της στοίβας. Εάν οι δραστηριότητες πάνω από αυτήν χρησιμοποιούν πάρα πολλούς πόρους θα καταστραφεί.

Πίνακας 5 Στοίβα Activity.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Οι εφαρμογές Android είναι υπεύθυνες για τη διαχείριση της κατάστασης, της μνήμης, των πόρων και των δεδομένων τους. Πρέπει να διακόπτονται προσωρινά και να συνεχίζουν χωρίς προβλήματα. Η κατανόηση των διαφορετικών καταστάσεων του κύκλου ζωής Activity είναι το πρώτο βήμα για τη σχεδίαση και ανάπτυξη στιβαρών εφαρμογών Android.



Εικόνα 20 Κύκλος ζωής μιας Activity του Android.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Μέθοδοι που μπορούν να χρησιμοποιηθούν σε ένα Activity:

- **onCreate():**

Όταν μία Activity ξεκινά για πρώτη φορά, καλείται η μέθοδος onCreate(). Η μέθοδος onCreate () έχει μία μόνο παράμετρο, το Bundle, το οποίο έχει τιμή null αν πρόκειται για μία Activity, που μόλις ξεκίνησε. Εάν αυτή η Activity σταμάτησε λόγω απελευθέρωσης μνήμης και τώρα επανεκινείται, το Bundle περιέχει τις πληροφορίες προηγούμενης κατάστασης γι' αυτήν την Activity, ώστε να μπορεί να ξεκινήσει πάλι. Η εκτέλεση οποιασδήποτε ρύθμισης παραμέτρων, όπως για τη διάταξη και τη σύνδεση δεδομένων, μπορεί να γίνει στη μέθοδο onCreate ().

- **onStart():**

Αυτή η μέθοδος υποδεικνύει ότι το Activity πρόκειται να εμφανιστεί στο χρήστη.

- **onResume():**

Όταν η Activity φτάσει στην κορυφή της στοίβας δραστηριοτήτων και γίνεται η διεργασία προσκηνίου, γίνεται κλήση της μεθόδου onResume (). Αν και η Activity ίσως δεν είναι ακόμα ορατή στο χρήστη, ο τρόπος αυτός είναι ο πιο κατάλληλος για την ανάκτηση στιγμιότυπων σε πόρους (με αποκλειστικό τρόπο ή αλλιώς), τους οποίους η Activity πρέπει να εκτελέσει. Συχνά, αυτοί οι πόροι έχουν υψηλές απαιτήσεις και γι' αυτό θα πρέπει να τους χρησιμοποιούμε μόνο για όσο η Activity βρίσκεται στο προσκήνιο. Η μέθοδος onResume() είναι το κατάλληλο σημείο για έναρξη ήχου, βίντεο και κινήσεων.

- **onPause():**

Όταν μία άλλη Activity μετακινείται στην κορυφή της στοίβας δραστηριοτήτων, η τρέχουσα Activity ενημερώνεται ότι θα μεταφερθεί πιο κάτω στη στοίβα δραστηριοτήτων μέσω της μεθόδου onPause(). Εδώ, η Activity θα πρέπει να διακόψει οποιονδήποτε ήχο, βίντεο και κίνηση που έχει ξεκινήσει στη μέθοδο onResume (). Σ' αυτό το σημείο πρέπει επίσης να απενεργοποιήσουμε πόρους όπως αντικείμενα Cursor βάσης δεδομένων, αν έχουμε αποφασίσει να τους διαχειριζόμαστε χειροκίνητα και όχι αυτόματα. Η μέθοδος onPause () μπορεί επίσης να είναι η τελευταία ευκαιρία που έχει η Activity για εκκαθάριση και απελευθέρωση πόρων που δεν χρειάζεται όσο βρίσκεται στο παρασκήνιο. Πρέπει να αποθηκεύσουμε εδώ δεδομένα, τα οποία δεν έχουν υποβληθεί ακόμα, σε περίπτωση που η εφαρμογή μας δεν συνεχιστεί. Η Activity μπορεί επίσης να αποθηκεύσει πληροφορίες κατάστασης σε συγκεκριμένες προτιμήσεις για μία Activity ή για γενικές προτιμήσεις για όλη την εφαρμογή. Η Activity πρέπει να εκτελέσει τα πάντα μέσα στη μέθοδο onPause() γρήγορα. Η νέα Activity προσκηνίου δεν ξεκινά μέχρι να επιστρέψει η μέθοδος onPause(). Γενικά, οποιοδήποτε πόροι και δεδομένα που ανακτώνται στη μέθοδο onResume() θα πρέπει να απελευθερώνονται στη μέθοδο onPause() . Σε αντίθετη περίπτωση, υπάρχει μία πιθανότητα αυτοί οι πόροι να μην μπορούν να απελευθερωθούν καθαρά αν η διεργασία τερματιστεί.

- **onStop():**

Αυτή η μέθοδος καλείται όταν το Activity περάσει σε διακοπή (stopped). Αν οι πόροι του συστήματος είναι χαμηλοί τότε το σύστημα μπορεί να καταστρέψει το Activity χωρίς να την καλέσει.

- **onRestart():**

Αυτή η μέθοδος καλείται όταν το Activity πρόκειται να ξαναεμφανιστεί μετά από μια κατάσταση διακοπής.

- **Αποτροπή της κατάργησης αντικειμένων δραστηριοτήτων**

Σε συνθήκες μικρής ποσότητας ελεύθερης μνήμης, το λειτουργικό σύστημα Android μπορεί να καταργήσει τη διεργασία για οποιαδήποτε Activity, που έχει διακοπεί προσωρινά, διακοπεί οριστικά, ή καταστραφεί. Αυτό ουσιαστικά σημαίνει ότι οποιαδήποτε Activity, που δεν βρίσκεται στο προσκήνιο υπόκειται σε πιθανό κλείσιμο. Εάν η Activity καταργηθεί μετά την onPause(), οι μέθοδοι onStop() και onDestroy() ίσως δεν κληθούν. Όσο περισσότεροι πόροι απελευθερωθούν από μία Activity στη μέθοδο onPause(), τόσο λιγότερο πιθανό είναι η Activity να καταργηθεί όσο βρίσκεται στο παρασκήνιο.

Η ενέργεια της κατάργησης μίας Activity δεν την αφαιρεί από τη στοίβα δραστηριοτήτων. Αντίθετα, η κατάσταση της Activity αποθηκεύεται σ' ένα αντικείμενο Bundle, αν υποθέσουμε ότι η Activity υλοποιεί και χρησιμοποιεί την onSaveInstanceState() για προσαρμοσμένα δεδομένα, αν και κάποια δεδομένα όψης View αποθηκεύονται αυτόματα. Όταν ο χρήστης επιστρέψει στην Activity αργότερα, η μέθοδος onCreate() καλείται ξανά, αλλά αυτή τη φορά ως παράμετρος υπάρχει ένα έγκυρο αντικείμενο Bundle.

Ο σχεδιαστής της εφαρμογής πρέπει να καταφέρει μία ευαίσθητη ισορροπία μεταξύ της συντήρησης των δεδομένων και των πόρων που χρειάζεται για γρήγορη επαναφορά, χωρίς να υποβαθμίζεται η CPU και οι πόροι του συστήματος όταν η εφαρμογή διακόπτεται στο παρασκήνιο.

- **Αποθήκευση κατάστασης δραστηριότητας σ' ένα Bundle με μία onSaveInstanceState().**

Εάν μία Activity είναι ευάλωτη στην κατάργηση από το λειτουργικό σύστημα Android εξαιτίας μικρής ποσότητας μνήμης, η Activity μπορεί να αποθηκεύσει πληροφορίες κατάστασης σ' ένα αντικείμενο Bundle χρησιμοποιώντας τη μέθοδο ανάστροφης κλήσης onSaveInstanceState(). Αυτή η κλήση δεν είναι εγγυημένη σε όλες τις περιστάσεις, γι' αυτό θα πρέπει να χρησιμοποιείται η μέθοδος onPause() για ουσιαστικές υποβολές δεδομένων. Όταν γίνει επιστροφή σ' αυτήν την Activity αργότερα, αυτό το Bundle διοχετεύεται στη μέθοδο onCreate(), δίνοντας τη δυνατότητα στην Activity να επιστρέψει ακριβώς στην ίδια κατάσταση όπου βρισκόταν όταν διακόπηκε προσωρινά. Μπορούμε επίσης να διαβάσουμε τις πληροφορίες του Bundle μετά τη μέθοδο ανάστροφης κλήσης onStart() χρησιμοποιώντας την ανάστροφη κλήση της onRestoreInstanceState().

- **onDestroy():**

Όταν η Activity καταστρέφεται, γίνεται κλήση της μεθόδου onDestroy(). Η μέθοδος onDestroy() καλείται για δύο λόγους. Η Activity έχει ολοκληρώσει τον κύκλο ζωής της εθελοντικά ή η Activity καταργείται από το λειτουργικό σύστημα Android, επειδή χρειάζεται τους πόρους της.

4.5 Διαχείριση μεταβάσεων δραστηριοτήτων με προθέσεις

Στη διάρκεια της ζωής μίας εφαρμογής Android, ο χρήστης μπορεί να περάσει μέσα από πολλά διαφορετικά στιγμιότυπα Activity. Υπάρχουν φορές που μπορεί να υπάρχουν πολλαπλά στιγμιότυπα Activity στη στοίβα δραστηριοτήτων. Οι προγραμματιστές πρέπει να προσέχουν τον κύκλο ζωής κάθε Activity κατά τη διάρκεια αυτών των μεταβάσεων. Ορισμένα στιγμιότυπα Activity όπως η οθόνη εκκίνησης της εφαρμογής εμφανίζονται και κατόπιν καταργούνται μόνιμα όταν αναλάβει δράση η δραστηριότητα της οθόνης του κύριου μενού. Ο χρήστης δεν μπορεί να επιστρέφει στη δραστηριότητα της οθόνης εκκίνησης χωρίς να επανεκκινήσει την εφαρμογή.

Άλλες μεταβάσεις Activity είναι προσωρινές, όπως μία θυγατρική Activity, που εμφανίζει ένα παράθυρο διαλόγου και κατόπιν επιστρέφει στην αρχική Activity (η οποία διακόπηκε προσωρινά στη στοίβα δραστηριοτήτων και τώρα επιστρέφει). Σ' αυτήν την περίπτωση, η γονική Activity προκαλεί επανεκκίνηση της θυγατρικής Activity και αναμένει ένα αποτέλεσμα.

4.5.1 Μετάβαση μεταξύ δραστηριοτήτων με προθέσεις

Όπως αναφέρθηκε προηγουμένως, οι εφαρμογές Android μπορούν να έχουν πολλαπλά σημεία εισαγωγής. Δεν υπάρχει καμιά συνάρτηση main(), όπως υπάρχει στο περιβάλλον ανάπτυξης εφαρμογών για το iPhone. Αντίθετα, μία συγκεκριμένη Activity μπορεί να καθοριστεί ως η κύρια Activity που θα εκκινείται εξ' ορισμού μέσα στο αρχείο AndroidManifest.xml.

Άλλες δραστηριότητες μπορούν να καθοριστούν για την εκκίνηση σε άλλες συγκεκριμένες συνθήκες. Για παράδειγμα, μία εφαρμογή μουσικής μπορεί να ορίσει ότι μία γενική Activity θα ξεκινά εξ' ορισμού από το μενού Application, αλλά θα ορίζει επίσης συγκεκριμένες εναλλακτικές δραστηριότητες σημείου εισαγωγής για την προσπέλαση συγκεκριμένων μουσικών λιστών αναπαραγωγής βάσει του ID της λίστας ή του ονόματος του καλλιτέχνη.

4.5.2 Εκκίνηση νέας δραστηριότητας με όνομα κλάσης

Μπορούμε να ξεκινήσουμε τις διάφορες δραστηριότητες με πολλούς τρόπους. Η απλούστερη μέθοδος είναι το αντικείμενο context της εφαρμογής να καλέσει τη μέθοδο the startActivity(), η οποία δέχεται μία μόνο παράμετρο, μία Intent (πρόθεση). Μία Intent (android.content.Intent) είναι ένας ασύγχρονος μηχανισμός μηνυμάτων ο οποίος χρησιμοποιείται απ' το λειτουργικό σύστημα Android κυρίως για την αντιστοίχιση αιτήσεων εργασιών με την κατάλληλη Activity ή Service (την εκκίνησή της, αν είναι απαραίτητη) και για την αποστολή συμβάντων Intent στο σύστημα.

Η παρακάτω γραμμή κώδικα καλεί τη μέθοδο startActivity() με μία ρητή πρόθεση, δηλαδή με μία συγκεκριμένη Intent. Αυτή η Intent αιτείται την εκκίνηση της Activity στόχου που ονομάζεται MyDrawActivity απ' την κλάση της. Αυτή η κλάση υλοποιείται κάπου αλλού μέσα στο πακέτο.

```
startActivity(new Intent(getApplicationContext(), MyDrawActivity.class));
```

Αυτή η γραμμή κώδικα μπορεί να είναι αρκετή για κάποιες εφαρμογές, οι οποίες απλά μεταπηδούν από μία Activity στην επόμενη. Ωστόσο, μπορούμε να χρησιμοποιήσουμε το μηχανισμό Intent μ' έναν πολύ πιο στιβαρό τρόπο. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε τη δομή Intent προκειμένου να περάσουμε δεδομένα από μία δραστηριότητα σε άλλη.

4.5.3 Δημιουργία προθέσεων με ενέργειες και δεδομένα

Είδαμε την απλούστερη περίπτωση χρήσης μίας Intent για την εκκίνηση μίας κλάσης με το όνομά της. Οι Intent, οι προθέσεις, δεν χρειάζεται να προσδιορίζουν με ρητό τρόπο το στοιχείο ή την κλάση που θέλουν να ξεκινήσει. Αντίθετα, μπορούμε να δημιουργήσουμε ένα φίλτρο προθέσεων (Intent Filter) και να το καταχωρίσουμε μέσα στο αρχείο Manifest του Android. Το λειτουργικό σύστημα Android επιχειρεί να αναλύει τις απαιτήσεις των Intent και να ξεκινά την κατάλληλη Activity βάσει των κριτηρίων φιλτραρίσματος.

Το αντικείμενο Intent αποτελείται από δύο βασικά μέρη: την ενέργεια που θα εκτελεστεί, και τα δεδομένα, στα οποία θα ενεργήσει. Μπορούμε επίσης να καθορίσουμε ζεύγη ενεργειών - δεδομένων χρησιμοποιώντας τύπους Intent Action και αντικείμενα Uri. Ένα αντικείμενο Uri αναπαριστά μία συμβολοσειρά, η οποία δίνει τη θέση και το όνομα ενός αντικειμένου. Επομένως, μία Intent ουσιαστικά λέει «κάνε αυτό» (η ενέργεια) σε «εκείνο» (το Uri, το οποίο περιγράφει τον πόρο στον οποίο θα εκτελεστεί η ενέργεια).

Οι πιο κοινοί τύποι ενεργειών ορίζονται στην κλάση Intent, και σ' αυτούς συμπεριλαμβάνονται το ACTION_MAIN (περιγράφει το κύριο σημείο εισαγωγής μίας Activity) και το ACTION_EDIT (χρησιμοποιείται σε συνδυασμό μ' ένα Uri στα δεδομένα, στα οποία γίνεται η επεξεργασία). Θα βρούμε επίσης τύπους Action, οι οποίοι παράγουν σημεία ενοποίησης με δραστηριότητες σε άλλες εφαρμογές, όπως στο πρόγραμμα περιήγησης ή στο πρόγραμμα πραγματοποίησης κλήσεων.

4.5.4 Εκκίνηση δραστηριότητας που ανήκει σε άλλη εφαρμογή

Αρχικά, η εφαρμογή μας μπορεί να ξεκινά μόνο δραστηριότητες, οι οποίες ορίζονται μέσα στο δικό της πακέτο. Ωστόσο, με τα κατάλληλα δικαιώματα, οι εφαρμογές μπορεί επίσης να ξεκινήσουν εξωτερικές δραστηριότητες μέσα σ' άλλες εφαρμογές. Για παράδειγμα, μία εφαρμογή CRM (Customer Relationship Management, Διαχείριση σχέσεων πελατών) θα μπορούσε να ξεκινήσει την εφαρμογή επαφών, ώστε να περιηγηθεί στη βάση δεδομένων επαφών, να επιλέξει μία συγκεκριμένη επαφή και να επιστρέψει το μοναδικό αναγνωριστικό αυτής της επαφής στην εφαρμογή CRM, προκειμένου αυτή να το χρησιμοποιήσει.

4.5.5 Διοχέτευση πρόσθετων πληροφοριών χρησιμοποιώντας προθέσεις

Μπορούμε επίσης να συμπεριλάβουμε πρόσθετα δεδομένα σε μία Intent. Η ιδιότητα Extras μίας Intent αποθηκεύεται σ' ένα αντικείμενο Bundle. Η κλάση Intent επίσης διαθέτει αρκετές χρήσιμες μεθόδους για τη λήψη και τον ορισμό ζευγών ονόματος - τιμής για πολλούς κοινούς τύπους δεδομένων. Οι συμβολοσειρές που χρησιμοποιούμε για να προσδιορίσουμε τις πρόσθετες ιδιότητες (extras) του αντικειμένου μας Intent μπορεί να είναι οτιδήποτε θέλουμε. Ωστόσο, η σύμβαση του Android για το όνομα ενός «extra» δεδομένου επιβάλλει την προσθήκη ενός προθέματος για το πακέτο, για παράδειγμα com.androidbook.Multimedia.SomeStringData.

4.5.6 Οργάνωση δραστηριοτήτων και προθέσεων στην εφαρμογή μας χρησιμοποιώντας μενού

Κάποιες εφαρμογές πιθανώς να έχουν μερικές οθόνες και κάθε μία θα έχει τη δική της Activity. Υπάρχει μία στενή σχέση μεταξύ των μενού, των δραστηριοτήτων και των προθέσεων. Συχνά υπάρχει ένα μενού, το οποίο χρησιμοποιείται με δύο διαφορετικούς τρόπους με τις δραστηριότητες και τις προθέσεις:

- **Κύριο μενού:** Λειτουργεί ως διακόπτης, στον οποίο κάθε στοιχείο του μενού ξεκινά μία διαφορετική Activity στην εφαρμογή μας. Για παράδειγμα, τα στοιχεία μενού για την εκκίνηση της δραστηριότητας παιχνιδιού (Activity Play Game), της δραστηριότητας υψηλών βαθμολογιών (Activity High Scores) και της δραστηριότητας βοήθειας (Activity Help).
- **Αναπτυσσόμενο μενού:** Λειτουργεί ως κατάλογος, στον οποίο κάθε στοιχείο του μενού ξεκινά την ίδια Activity, αλλά κάθε στοιχείο διοχετεύει διαφορετικά δεδομένα για την Intent (π.χ., ένα μενού με όλες τις εγγραφές της βάσης δεδομένων). Η επιλογή ενός συγκεκριμένου στοιχείου θα μπορούσε να εκκινήσει τη δραστηριότητα επεξεργασίας εγγραφής (Activity Edit Record), διοχετεύοντας σ' αυτήν το μοναδικό αναγνωριστικό αυτού του συγκεκριμένου στοιχείου.

4.6 Εργασία με υπηρεσίες

Ένα στοιχείο των εφαρμογών είναι η υπηρεσία. Μία υπηρεσία του Android, μία service, είναι ουσιαστικά μία Activity χωρίς διεπαφή χρήστη. Μπορεί να εκτελείται ως διεργασία παρασκήνιου ή να ενεργεί περίπου όπως μία υπηρεσία web, εκτελώντας την επεξεργασία αιτήσεων τρίτων. Μπορούμε να χρησιμοποιήσουμε τις προθέσεις και τις δραστηριότητες για την εκκίνηση υπηρεσιών χρησιμοποιώντας τις μεθόδους `startService()` και `bindService()`. Οι service που εκτίθενται από μία εφαρμογή Android πρέπει να καταχωρούνται στο αρχείο Manifest του Android. Μπορούμε να χρησιμοποιήσουμε τις υπηρεσίες για διαφορετικές χρήσεις. Γενικά, χρησιμοποιούμε μία υπηρεσία όταν από το χρήστη δεν απαιτείται καμιά είσοδος.

Μερικές περιπτώσεις, στις οποίες θα πρέπει να υλοποιήσουμε ή να χρησιμοποιήσουμε μία υπηρεσία Android:

- Μία εφαρμογή για τον καιρό, το email ή ένα κοινωνικό δίκτυο θα μπορούσε να υλοποιήσει μία υπηρεσία για τακτικό έλεγχο για ενημερώσεις.
- Μία εφαρμογή για φωτογραφίες ή πολυμέσα, η οποία διατηρεί τα δεδομένα συγχρονισμένα στο διαδίκτυο θα μπορούσε να υλοποιήσει μία υπηρεσία για τη συσκευασία και τη μεταφόρτωση νέου υλικού στο παρασκήνιο, όταν η συσκευή βρίσκεται σε κατάσταση αδράνειας.
- Μία εφαρμογή επεξεργασίας βίντεο θα μπορούσε να τοποθετεί μεγάλες ποσότητες επεξεργασίας σε μία ουρά στην υπηρεσία της, ώστε να αποφευχθεί οποιαδήποτε επίδραση στη συνολική απόδοση του συστήματος, τουλάχιστον για όχι αναγκαίες εργασίες.
- Μία εφαρμογή ειδήσεων θα μπορούσε να υλοποιήσει μία υπηρεσία για την εκ των προτέρων φόρτωση περιεχομένου μέσω λήψης ειδήσεων όταν ο χρήστης ξεκινά την εφαρμογή, με απώτερο στόχο τη βελτίωση των επιδόσεων.

Ένας καλός πρακτικός κανόνας είναι ότι αν η εργασία απαιτεί τη χρήση ενός λειτουργικού νήματος και μπορεί να επηρεάσει τη δυνατότητα απόκρισης και τις επιδόσεις μίας εφαρμογής, θα μπορούσαμε

να εξετάσουμε τα οφέλη από την υλοποίηση μίας υπηρεσίας που θα χειρίζεται την εργασία έξω απ' τον κύριο κύκλο ζωής της εφαρμογής.

4.7 Το αρχείο manifest του Android

Τα έργα Android χρησιμοποιούν ένα ειδικό αρχείο ρύθμισης παραμέτρων, το οποίο ονομάζεται αρχείο manifest του Android και το οποίο καθορίζει τις ρυθμίσεις των παραμέτρων της εφαρμογής, ρυθμίσεις όπως το όνομα και η έκδοση της εφαρμογής, όπως και τα δικαιώματα που απαιτούνται για την εκτέλεση της εφαρμογής και από ποια στοιχεία αποτελείται η εφαρμογή.

4.7.1 Ρύθμιση παραμέτρων του αρχείου manifest του Android

Το αρχείο manifest της εφαρμογής Android είναι ένα ειδικά μορφοποιημένο αρχείο XML, το οποίο πρέπει να συνοδεύει κάθε εφαρμογή Android. Αυτό το αρχείο περιέχει σημαντικές πληροφορίες για την ταυτότητα της εφαρμογής. Εδώ ορίζεται το όνομα της εφαρμογής και πληροφορίες για την έκδοσή της, αλλά και τα στοιχεία, στα οποία βασίζεται η εφαρμογή, τα δικαιώματα που απαιτούνται για την εκτέλεσή της και άλλες πληροφορίες για τη ρύθμιση των παραμέτρων της. Το αρχείο manifest του Android ονομάζεται `AndroidManifest.xml` και πρέπει να συμπεριλαμβάνεται στο κορυφαίο επίπεδο οποιουδήποτε έργου Android. Οι πληροφορίες σ' αυτό το αρχείο χρησιμοποιούνται απ' το σύστημα Android, προκειμένου να γίνουν οι εξής εργασίες:

- Εγκατάσταση και αναβάθμιση του πακέτου της εφαρμογής.
- Λεπτομέρειες της εφαρμογής, όπως το όνομά της, μία περιγραφή της και το εικονίδιο που θα χρησιμοποιούν οι χρήστες.
- Προσδιορισμός απαιτήσεων συστήματος για την εφαρμογή, όπως ποια Android SDK υποστηρίζονται, τι ρυθμίσεις υλικού απαιτούνται (π.χ., ειδικές συσκευές πλοήγησης) και σε ποια χαρακτηριστικά πλατφόρμας βασίζεται η εφαρμογή (π.χ., χρησιμοποιεί στοιχεία πολλαπλής αφής).
- Εκκίνηση δραστηριοτήτων εφαρμογής.
- Διαχείριση δικαιωμάτων εφαρμογής.
- Άλλες προχωρημένες παράμετροι της εφαρμογής, όπως η λειτουργία ως υπηρεσία, δέκτης εκπομπής ή πάροχος υλικού.
- Δυνατότητα καθορισμού ρυθμίσεων εφαρμογής, όπως εκσφαλμάτωση και ρύθμιση παραμέτρων υλικού για δοκιμές.

Κεφάλαιο 5

5.1 Εργαλεία του Android

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java ενώ τους τελευταίους μήνες γίνονται προσπάθειες για να συμπεριληφθούν και άλλες γλώσσες όπως η C και C++. Οπότε βασική προϋπόθεση της κατασκευής είναι να διαθέτουμε τα αντίστοιχα εργαλεία της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε. Τα πιο διαδεδομένα λογισμικά για την ανάπτυξη εφαρμογών σε Android είναι:

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



Εικόνα 21 Λογότυπο εργαλείων Android.

5.2 Android SDK

Το βασικότερο εργαλείο αποτελεί το Android SDK (Software Development kit) το οποίο ουσιαστικά μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα συγκεκριμένα για την κατασκευή εφαρμογής Android. Η σύνδεση του Android SDK με το γραφικό μας περιβάλλον (Eclipse) γίνεται μέσω μίας επέκτασης (Android Development Tools ή ADT Plugin) που εγκαθιστάμε στο Eclipse, ώστε να μπορέσουμε να μεταγλωττίσουμε την εφαρμογή μας και έπειτα να την τρέξουμε. Καθώς το λειτουργικό σύστημα Android κυκλοφορεί σε διάφορες εκδόσεις καθίσταται σαφές ότι η κάθε έκδοση θα χρησιμοποιεί και ορισμένα διαφορετικά προγραμματιστικά εργαλεία. Έτσι μέσα από την ADT Plugin μπορούμε να εγκαταστήσουμε τα εργαλεία για την υλοποίηση εφαρμογής σε οποιαδήποτε έκδοση. Για παράδειγμα αν θέλουμε η εφαρμογή μας να είναι συμβατή και σε παλαιότερες εκδόσεις του Android τότε θα πρέπει να εγκαταστήσουμε τα αντίστοιχα εργαλεία και να δοκιμάσουμε την εφαρμογή μας στις εκδόσεις αυτές.

5.2.1 Μερικά από τα χαρακτηριστικά του SDK

Το Android δίνει την ευκαιρία να δημιουργηθούν εφαρμογές που είναι τόσο μέρος του τηλεφώνου όσο και εκτός του πακέτου. Παρακάτω σημειώνονται μερικά από τα πιο αξιοσημείωτα χαρακτηριστικά γνωρίσματα του Android:

- Καμία οικονομική επιβάρυνση όσον αφορά τη χορήγηση αδειών, τη διανομή, ή την αμοιβή ανάπτυξης του λογισμικού.
- WI-FI Πρόσβαση υλικού.
- GSM, EDGE, και 3G δίκτυα για τηλεφωνία ή μεταφορά δεδομένων, που επιτρέπει σε κάποιον να πραγματοποιήσει ή να λάβει κλήσεις ή μηνύματα, ή να στείλει και να ανακτήσει δεδομένα μέσω των δικτύων κινητής τηλεφωνίας.
- Εύκολα και κατανοητά APIs τοποκεντρικών (location-based) υπηρεσιών όπως το GPS.
- Πλήρης έλεγχος του υλικού πολυμέσων συμπεριλαμβανομένης της αναπαραγωγής ήχου και εγγραφής, χρησιμοποιώντας τη φωτογραφική μηχανή και το μικρόφωνο.
- APIs για υλικό επιταχυμέτρων (accelerometer) και πυξίδων (compass).
- Ανάγνωση μηνυμάτων ipc.
- Διαμοιραζόμενες βάσεις δεδομένων.
- Έναν ενσωματωμένο ανοικτού κώδικα βασισμένο σε webKit φυλλομετρητή.
- Πλήρης υποστήριξη εφαρμογών που ενσωματώνουν χειρισμό χαρτών ως μέρος της διεπαφής χρήστη.
- Υποστήριξη Peer-to-peer(P2P) χρησιμοποιώντας το Google Talk.
- Βελτιστοποιημένα γραφικά που περιλαμβάνουν path-based 2D βιβλιοθήκη γραφικών και υποστήριξη για τα 3D γραφικά χρησιμοποιώντας OpenGL ES.
- Πολυμεσικές Βιβλιοθήκες για αναπαραγωγή και για εγγραφή εικόνων και video σε ποικίλες μορφές.
- Ένα πλαίσιο εφαρμογών που ενθαρρύνει την επαναχρησιμοποίηση τμημάτων εφαρμογών και αντικατάσταση των εγγενών εφαρμογών.

5.2.2 Μερικά ακόμα εργαλεία

- Ο εξομοιωτής του Android (The Android Emulator) - Μια εφαρμογή της εικονικής μηχανής του Android σχεδιασμένη να τρέξει στον υπολογιστή. Ο εξομοιωτής μπορεί να χρησιμοποιηθεί για να εξεταστούν και να διορθωθούν οι Android εφαρμογές.
- Η Dalvik Υπηρεσία Παρακολούθησης Απασφαλμάτωσης (Dalvik Debug Monitoring Service (DDMS)) - Η DDMS κατάσταση χρησιμοποιείται για τη παρακολούθηση και τον χειρισμό των Dalvik εικονικών μηχανών στις οποίες γίνεται η απασφαλμάτωση των εφαρμογών.
- Android Asset Packaging Tool (AAPT) - Κατασκευάζει το εκτελέσιμο πακέτο αρχείων με κατάληξη .apk.
- Android Debug Bridge (ADB) - Το ADB είναι μια εφαρμογή πελάτη-εξυπηρετή (client - server) που παρέχει μια σύνδεση με έναν τρέχοντα εξομοιωτή. Δίνει την δυνατότητα να αντιγραφούν αρχεία, να εγκατασταθούν πακέτα εφαρμογών με κατάληξη .apk, και να εκτελεστούν εντολές πυρήνα.
- SQLite3 - Εργαλείο βάσεων δεδομένων που χρησιμοποιείται για να υπάρχει πρόσβαση στα δεδομένα της βάσεως δεδομένων SQLite που δημιουργείται και που χρησιμοποιείται από το Android.
- Traceview - Εργαλείο ανάλυσης γραφικών για την εμφάνιση των trace logs των Android εφαρμογών.
- MksDCard - Δημιουργεί μια εικόνα (image) δίσκων SDCard που μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για να εξομοιώσει μια εξωτερική κάρτα αποθήκευσης.
- dx - Μετατρέπει το .class bytecode της Java σε .dex bytecode του Android.
- activityCreator - Κομμάτι κώδικα που χτίζει αρχεία Ant, τα οποία μπορούν να χρησιμοποιηθούν για να μεταφράσουν εφαρμογές Android χωρίς το ADT plug-in.

5.2.3 Τι περιέχεται στο πακέτο (box)

Το Πακέτο Ανάπτυξης Λογισμικού (software development kit SDK) του Android περιλαμβάνει όλα αυτά που χρειάζεται κανείς για να δημιουργήσει, να ελέγξει και να διορθώσει μια εφαρμογή Android. Στο SDK περιλαμβάνονται:

- **Τα Android APIs** - Ο πυρήνας του SDK είναι οι Android API βιβλιοθήκες που δίνουν στους κατασκευαστές τη δυνατότητα να έχουν πρόσβαση στη συλλογή λογισμικού του Android (Android stack). Αυτές είναι οι ίδιες βιβλιοθήκες που χρησιμοποιούνται από τη Google για να δημιουργήσουν τις εγγενείς εφαρμογές του Android.
- **Τα εργαλεία ανάπτυξης (Development Tools)** - Για τη μετατροπή του πηγαίου κώδικα σε εκτελέσιμες εφαρμογές Android, το SDK περιλαμβάνει διάφορα εργαλεία ανάπτυξης που δίνουν τη δυνατότητα να συνταχτούν και να διορθωθούν οι εφαρμογές .
- **Τον εξομοιωτή του Android (Android Emulator)** - Ο εξομοιωτής του Android είναι ένας πλήρως διαδραστικός εξομοιωτής συσκευών Android που περιέχει διάφορες εναλλακτικές προσόψεις (skins). Μέσω του εξομοιωτή, φαίνεται η όψη και η συμπεριφορά μιας εφαρμογής έτσι όπως θα φαινόταν σε μια πραγματική συσκευή Android. Όλες οι εφαρμογές Android μπορούν να λειτουργήσουν μέσα στη Dalvik VM για το λόγο αυτό ο εξομοιωτής λογισμικού είναι ένα άριστο περιβάλλον – είναι ουδέτερο όσον αφορά την αναπαράσταση κάποιας συσκευής, έτσι παρέχει ένα καλύτερο ανεξάρτητο περιβάλλον δοκιμής από οποιοδήποτε άλλο υλικό δοκιμών εφαρμογών.

- **Πλήρης τεκμηρίωση** - Το SDK περιλαμβάνει εκτενείς και λεπτομερείς πληροφορίες για το τι ακριβώς περιλαμβάνεται σε κάθε πακέτο και κλάση και πώς να χρησιμοποιηθεί. Επίσης εκτός από την τεκμηρίωση του κώδικα, δίνονται οδηγίες για το πώς μπορεί να ξεκινήσει να δημιουργεί κανείς μια εφαρμογή καθώς επίσης και λεπτομερείς εξηγήσεις των βασικών αρχών πίσω από την ανάπτυξη εφαρμογών Android .
- **Δείγματα κώδικα (Sample Code)** - Το SDK του Android περιλαμβάνει έναν αριθμό εφαρμογών που επιδεικνύουν μερικές από τις δυνατότητες που γίνονται διαθέσιμες χρησιμοποιώντας το Android, καθώς επίσης και απλών προγραμμάτων που δίνουν έμφαση στο πώς χρησιμοποιούνται μεμονωμένα χαρακτηριστικά γνωρίσματα του API.
- **Online υποστήριξη** - Αν και σχετικά νέο, το Android έχει δημιουργήσει αναπτυσσόμενη και ενεργή κοινότητα κατασκευαστών λογισμικού. Οι ομάδες της Google στο <http://developer.android.com/index.html>, αποτελούν ενεργό φόρουμ κατασκευαστών Android.

5.3 Android NDK (Native Development Kit)

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) που παρέχει εργαλεία που επιτρέπουν την ενσωμάτωση native κώδικα στις εφαρμογές του Android. Το NDK επιτρέπει την εκτέλεση κομματιών 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.

5.3.1 Μερικά από τα 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.

5.3.2 Τι προβλέπει το NDK

- Ένα σύνολο εργαλείων και αρχείων «χτισίματος» (build files) που χρησιμοποιούνται για να παραχθεί ατόφιος κώδικας σε βιβλιοθήκες από πηγές C και C++.
- Έναν τρόπο ενσωμάτωσης των αντίστοιχων ατόφιων βιβλιοθηκών σε ένα αρχείο πακέτου εφαρμογών (.apk), το οποίο μπορεί να αναπτυχθεί σε Android συσκευές.
- Ένα σύνολο ατόφιων 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 ατόφιου 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 εντολών:

- 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.
- MIPS 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. Μπορεί 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.

5.3.3 Πότε γίνεται ανάπτυξη σε ατόφιο κώδικα

Το NDK δεν θα ωφελήσει στις περισσότερες εφαρμογές. Ένας developer πρέπει να ζυγίσει τα πλεονεκτήματα έναντι των μειονεκτημάτων. Ειδικά, η χρήση ατόφιου κώδικα δεν έχει αποτέλεσμα στην αύξηση της αυτόματης απόδοσης, αλλά πάντα αυξάνει την πολυπλοκότητα της εφαρμογής. Γενικά, η χρήση ατόφιου 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 επανακωδικοποίηση μιας μεθόδου σε C, συνήθως δεν φέρνει αποτέλεσμα μεγάλης αύξησης στην απόδοση. Κατά την εξέταση του αν πρέπει ή όχι ο προγραμματισμός να γίνει με ατόφιο 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++.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Το Android framework παρέχει δυο τρόπους χρήσης του ατόφιου 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 ατόφιου 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 κώδικα 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 ατόφιες 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 οποιωνδήποτε άλλων API frameworks κρίνεται απαραίτητη, θα πρέπει να γίνει συγγραφή κώδικα JNI ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

5.3.4 Περιεχόμενα και εργαλεία ανάπτυξης του NDK

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

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 ατόφιες 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 των source files που θα γίνουν compile και της Android εφαρμογής που θα τα χρησιμοποιήσει – Το Build system μεταγλωττίζει τα source files και τοποθετεί τις σκόρπιες βιβλιοθήκες κατευθείαν στο project της εφαρμογής. Με εξαίρεση τις παραπάνω βιβλιοθήκες, οι ατόφιες βιβλιοθήκες συστήματος στην Android πλατφόρμα δεν είναι σταθερές και μπορεί να αλλάζουν στην μελλοντική έκδοση της πλατφόρμας. Οι εφαρμογές πρέπει να κάνουν χρήση μόνο των σταθερών ατόφιων βιβλιοθηκών συστήματος που παρέχονται στο NDK.

5.3.5 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 crufeatures 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 άδειες ανοιχτού κώδικα 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.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

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.

5.4 App Inventor for Android



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.

5.4.1 Η ανάπτυξη του App Inventor

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 (Το Scratch είναι ένα περιβάλλον προγραμματισμού σχεδιασμένο για εκπαίδευση και ψυχαγωγία) και του Alice (είναι ένα πρωτοποριακό 3D περιβάλλον προγραμματισμού), με τη διαφορά ότι οι εφαρμογές που δημιουργούνται τρέχουν σε έξυπνα τηλέφωνα (smart phones).

5.4.2 Τι είναι το App Inventor

Το App Inventor είναι μια WYSIWYG (συντομογραφία της φράσης "What You See Is What You Get" (Αυτό βλέπεις είναι αυτό που θα εμφανιστεί)) 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" έκδοσή του για να δει τους... καρπούς της προσπάθειάς 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.



Εικόνα 22 Λογότυπο App Inventor

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 τη 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

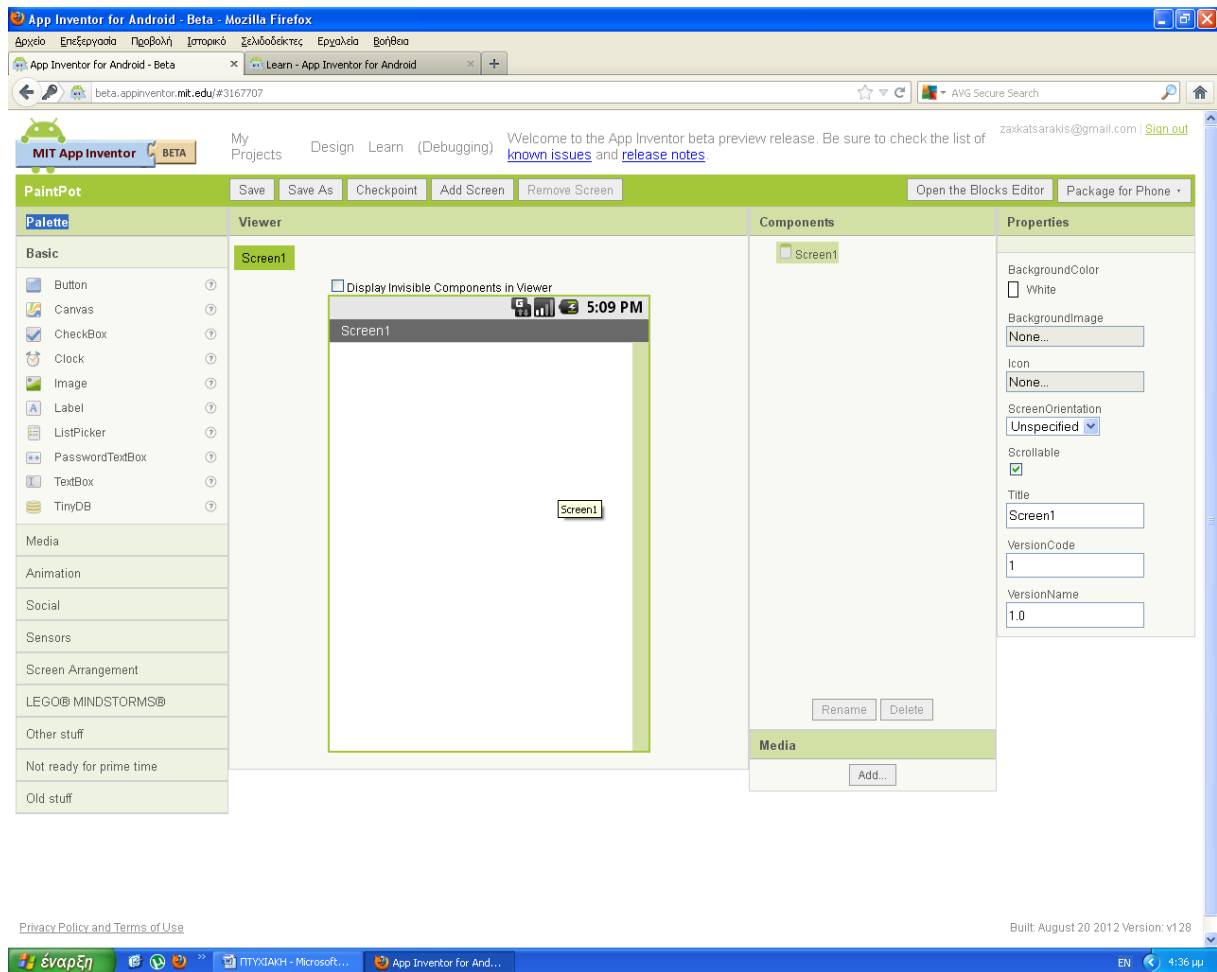
κώδικα για την εκτέλεση λειτουργιών κάτω από συγκεκριμένες συνθήκες. Επίσης, είναι διαθέσιμα ακόμη και blocks για επικοινωνία με τοποθεσίες κοινωνικής δικτύωσης, όπως π.χ. το Twitter.

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.

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.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 23 Η κεντρική οθόνη του App Inventor

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

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

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 componets που είναι μια λίστα με τα controls που χρησιμοποιούμε και τέλος τα properties, που είναι οι ιδιότητες του κάθε control.

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.

5.4.3 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.

5.5 Hyper Next Android Creator



Το HyperNext Android Creator (HAC) είναι ένα ισχυρό αλλά και εύκολο στη χρήση λογισμικό συστήματος, που επιτρέπει σε οποιονδήποτε να αρχίσει γρήγορα να δημιουργεί τις δικές του Android εφαρμογές. Με το Hyper Next δεν υπάρχει καμία ανάγκη για την εκμάθηση java ή Android SDK και είναι πολύ πιο εύκολο στην χρήση από το Eclipse. Το περιβάλλον του Hyper Next 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 Hyper Next 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.

Μερικά από τα πλεονεκτήματα του HAC είναι:

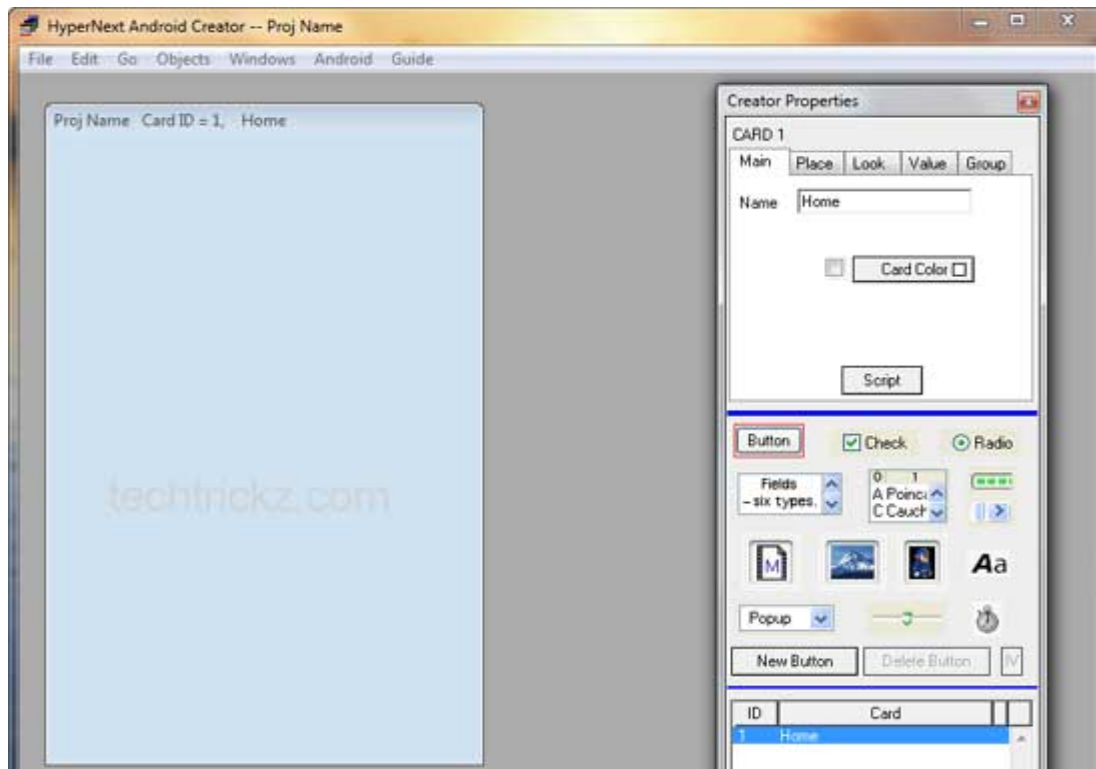
- Εύκολο στη χρήση και ιδανικό για αρχάριους
- Δεν χρειάζεται η εκμάθηση Java ή το Android SDK
- Πολλά παραδείγματα για το ξεκίνημα
- Κατέβασμα demo εφαρμογών για Android συσκευή
- Απλό περιβάλλον εργασίας: ένα παράθυρο, σχεδιασμός και ένα Toolbar
- Το περιβάλλον εργασίας έχει τρεις λειτουργίες: Σχεδιασμός, 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.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 24 Η κεντρική οθόνη του HyperNext Android Creator (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 forms	Search web
Price	\$59.95	Free
Time	low	Very high

Πίνακας 6 Σύγκριση HyperNext Android Creator (HAC) και Java

5.5.1 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.

5.5.2 Περιβάλλον προγραμματισμού

Το HyperNext έχει μια σχετικά απλή διεπαφή σχεδίου, που επιτρέπει στα προγράμματα τη γρήγορη εκτέλεση και τον τάχιστο έλεγχο. Έλεγχος 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 αλλαχτούν 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 επεκταθεί 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 προγραμματίσιμες 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

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

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.

Κεφάλαιο 6

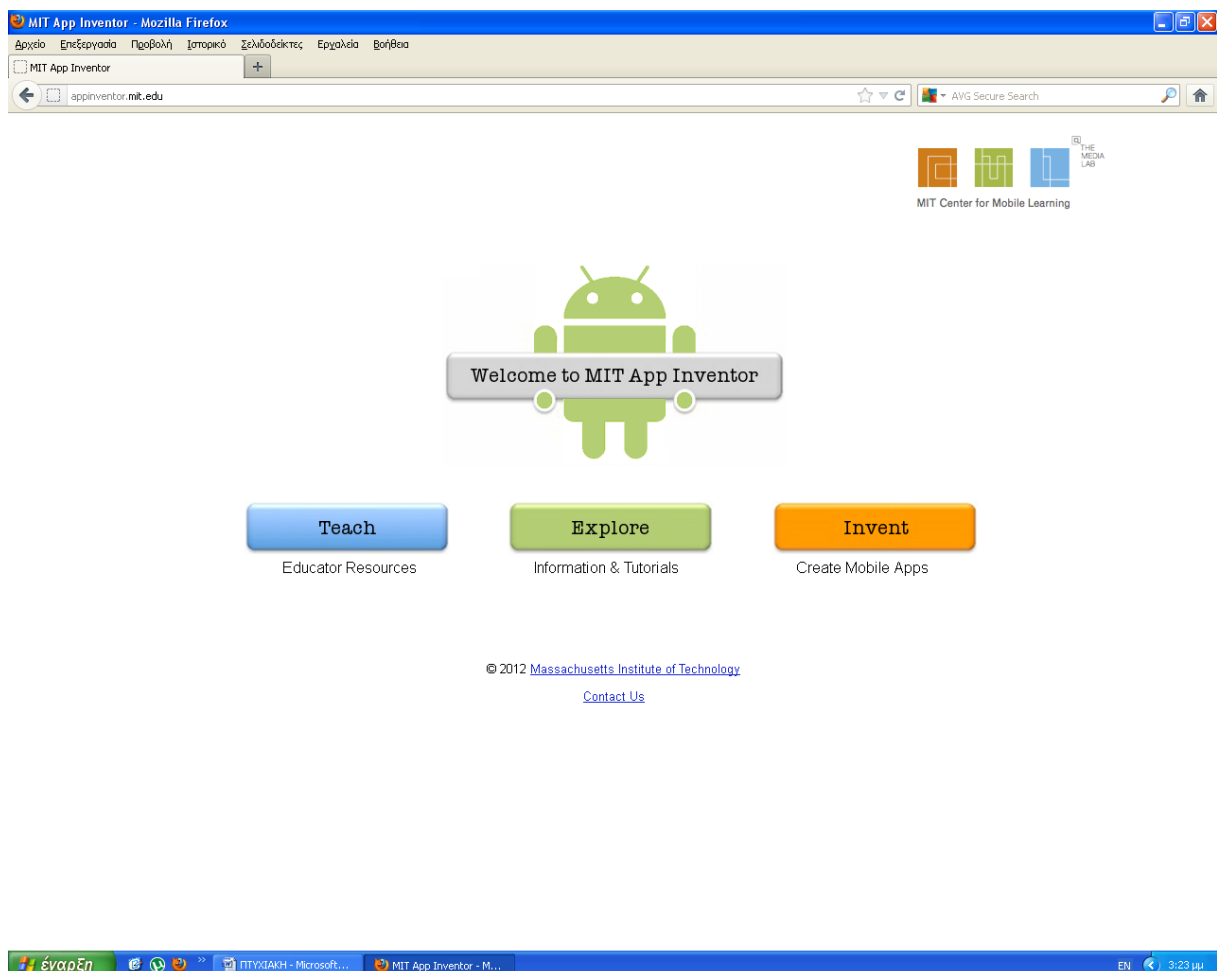
6.1 Ανάπτυξη εφαρμογής με το App Inventor της Google

Ο σκοπός αυτής της εφαρμογής είναι να παρουσιάσει το App Inventor της Google ως ένα εργαλείο ανάπτυξης εφαρμογών Android και να βοηθήσει όσους ενδιαφέρονται να γράψουν apps για android.

6.2 Ξεκινώντας με το App Inventor

Το App Inventor τρέχει μέσω ενός προγράμματος περιήγησης στο Web. Το link για το site του App Inventor είναι: <http://www.appinventor.mit.edu>

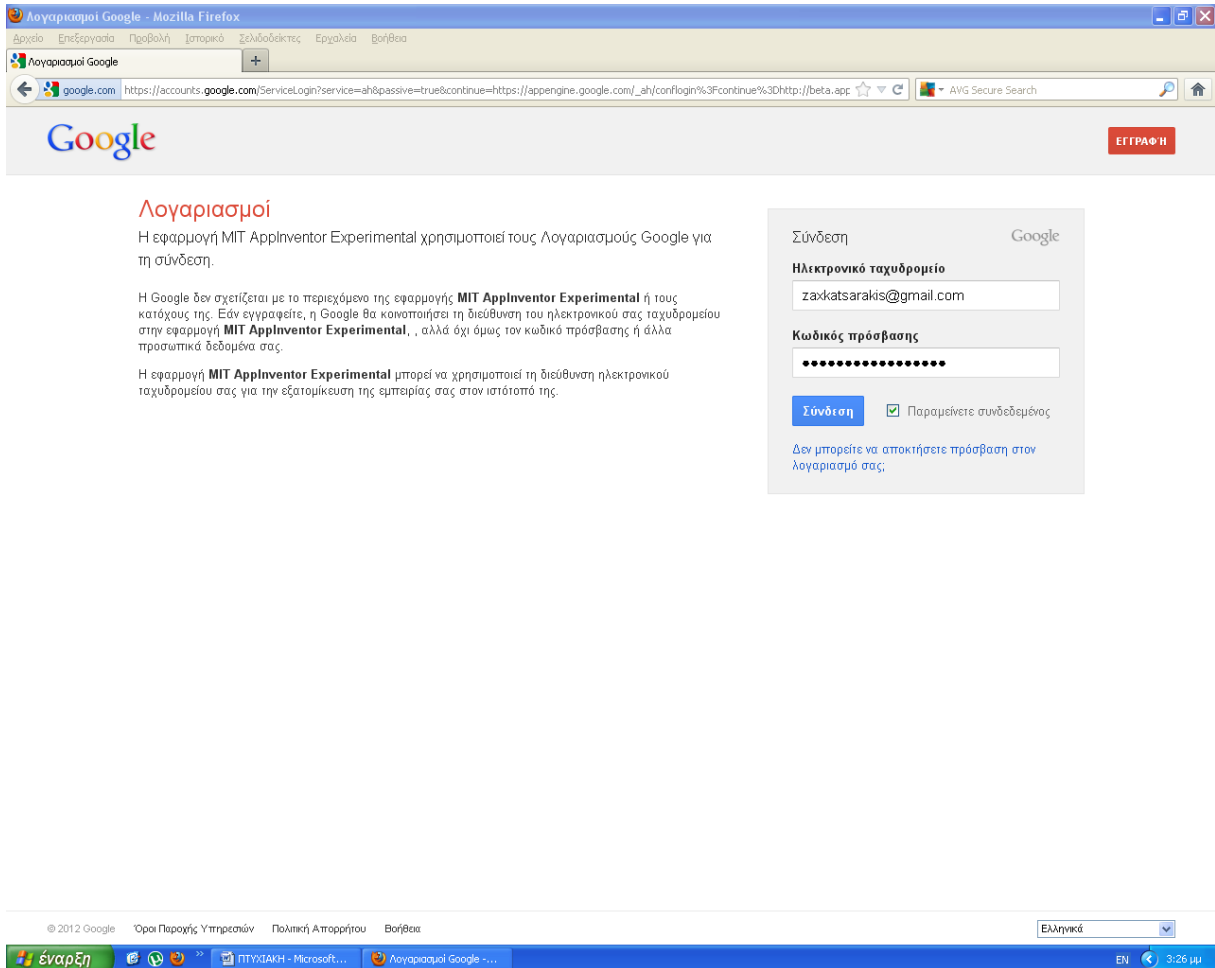
Η πρώτη σελίδα που μας βγάζει είναι:



Εικόνα 25 Η πρώτη σελίδα του App Inventor.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

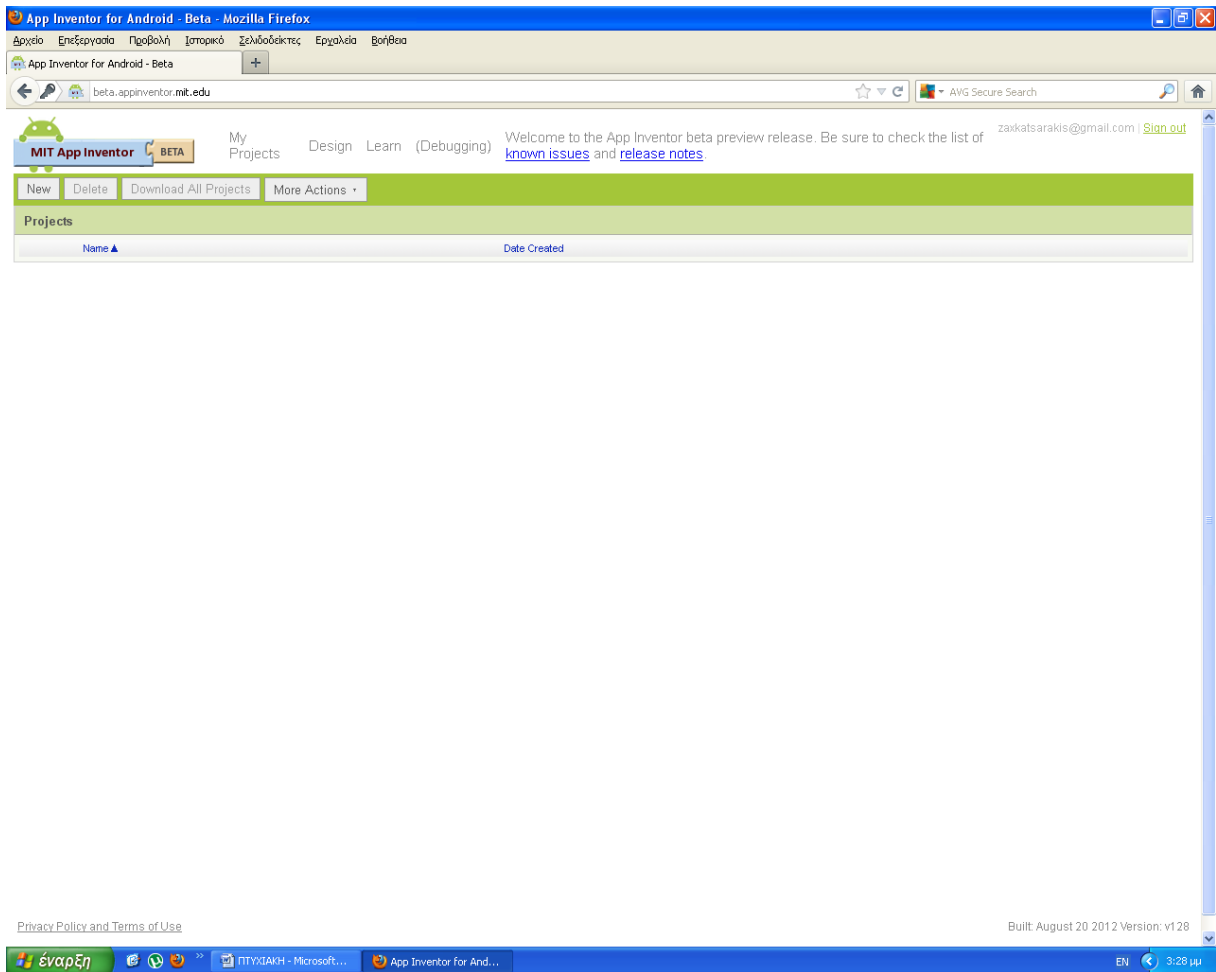
Πατάμε το Invent (πρέπει να έχουμε δημιουργήσει ένα λογαριασμό στη Google) και μας πηγαίνει στη σελίδα για να περάσουμε τα στοιχεία του λογαριασμού μας.



Εικόνα 26 Σελίδα εισαγωγής στοιχείων λογαριασμού.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Αφού συμπληρώσουμε τα στοιχεία του λογαριασμού μας πατάμε σύνδεση και μας πηγαίνει στην κεντρική σελίδα του App Inventor:



Εικόνα 27 Κεντρική οθόνη του App Inventor.

Στη συνέχεια πατάμε το Learn από τη γραμμή εργαλείων όπου μπορούμε να βρούμε γενικές πληροφορίες όπως: τι είναι το App Inventor, πως κάνουμε το setup, τα Tutorials και διάφορα άλλα.

6.2.1 Computer Set up

Απαιτήσεις συστήματος:

Computer and operating system

- Macintosh (with Intel processor): Mac OS X 10.5, 10.6
- Windows: Windows XP, Windows Vista, Windows 7
- GNU/Linux: Ubuntu 8+, Debian 5+

Browser

- Mozilla Firefox 3.6 or higher
- Apple Safari 5.0 or higher
- Google Chrome 4.0 or higher
- Microsoft Internet Explorer 7 or higher

6.2.2 Java configuration

Ο υπολογιστής μας θα πρέπει να τρέχει Java 1,6 και πάνω.

Μπορούμε να κατεβάσουμε τη Java από: www.java.com.



Εικόνα 28 Το site απ' όπου μπορούμε να κατεβάσουμε τη Java.



Εικόνα 29 Εγκατάσταση της Java.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Μπορούμε να τεστάρουμε αν η java έχει εγκατασταθεί σωστά στον υπολογιστή μας από το link:
<http://beta.appinventor.mit.edu/learn/setup/misc/JWSTest/AppInvJWSTest.html>



Εικόνα 30 Έλεγχος Java.

6.2.3 Install the App Inventor Setup Software

Για να μπορέσουμε να χρησιμοποιήσουμε το App Inventor, θα πρέπει να εγκαταστήσουμε το App Inventor Setup στον υπολογιστή μας.

Ανάλογα με το λειτουργικό που χρησιμοποιούμε μπορούμε να το κατεβάσουμε από:

- Instructions for Mac OS X :

<http://beta.appinventor.mit.edu/learn/setup/setupmac.html>

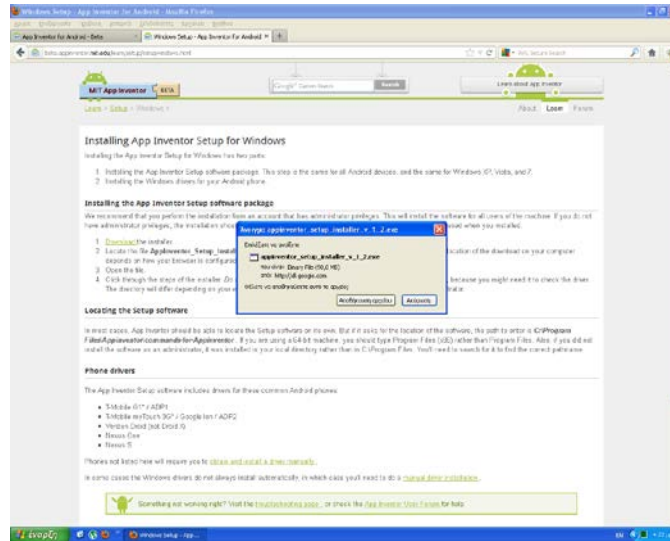
- Instructions for GNU/Linux:

<http://beta.appinventor.mit.edu/learn/setup/setuplinux.html>

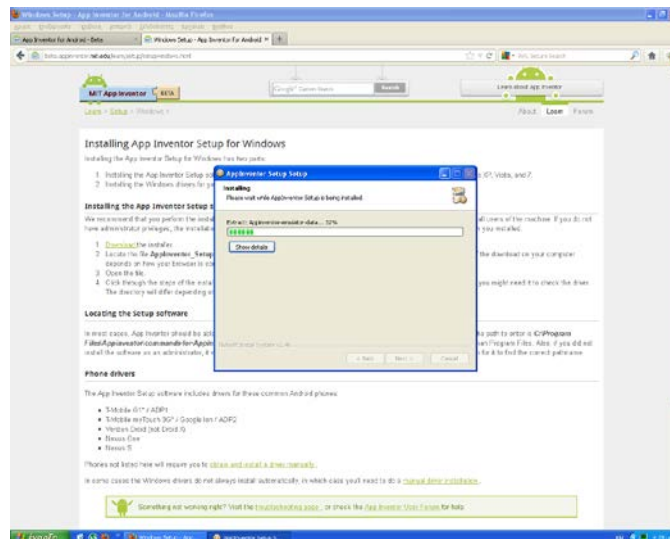
- Instructions for Windows :

<http://beta.appinventor.mit.edu/learn/setup/setupwindows.html>

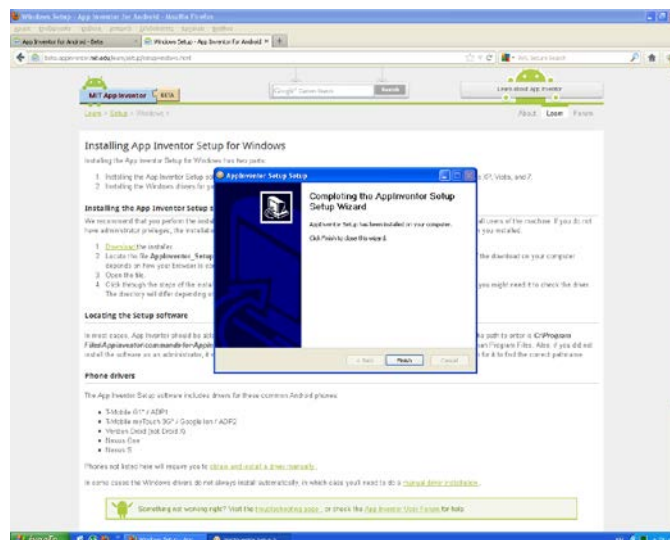
Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 31 Κατέβασμα του App Inventor Setup.



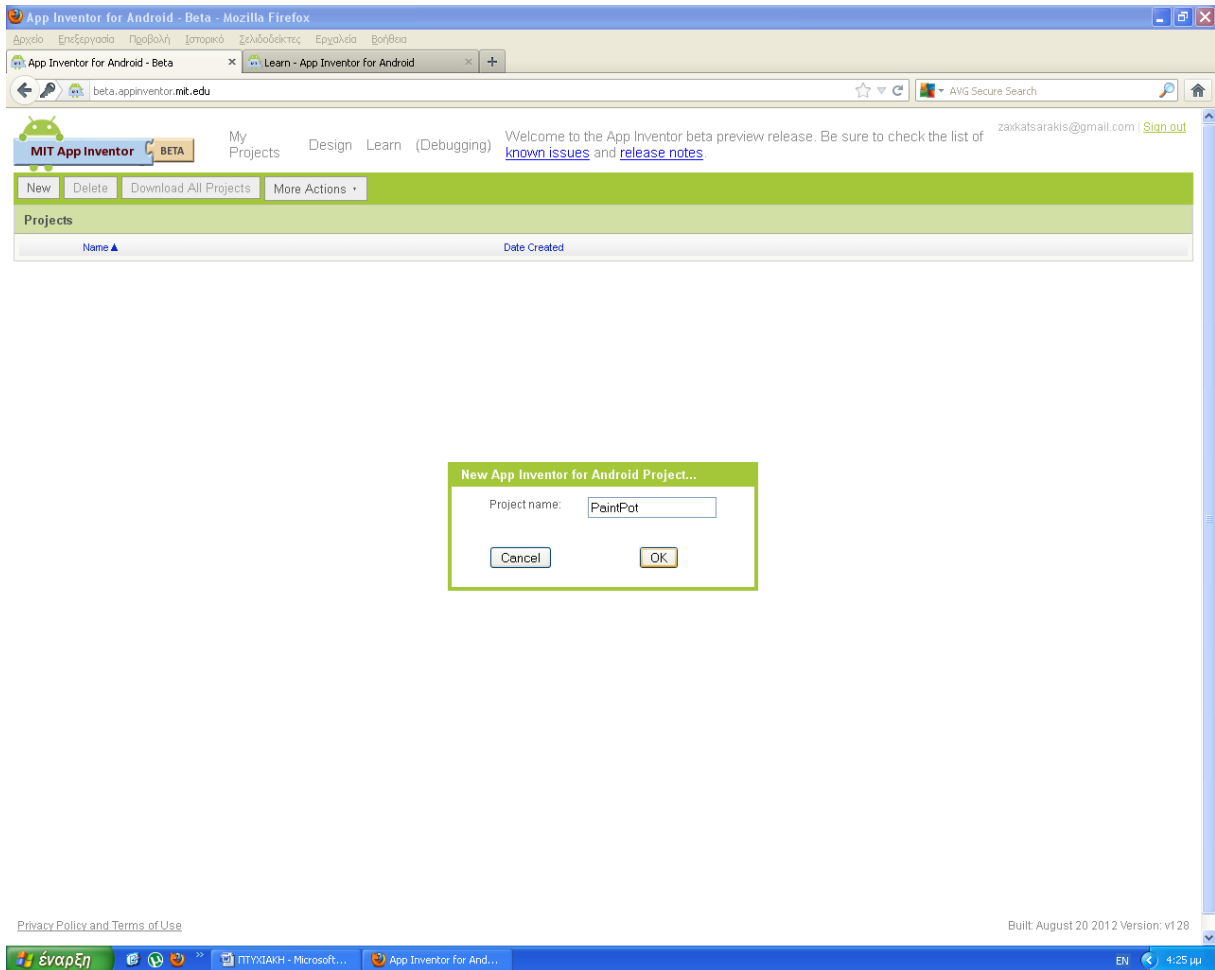
Εικόνα 32 Εκτέλεση του App Inventor Setup.



Εικόνα 33 Ολοκλήρωση εγκατάστασης του App Inventor Setup.

6.3 Σύντομη περιγραφή της κεντρικής οθόνης του App Inventor

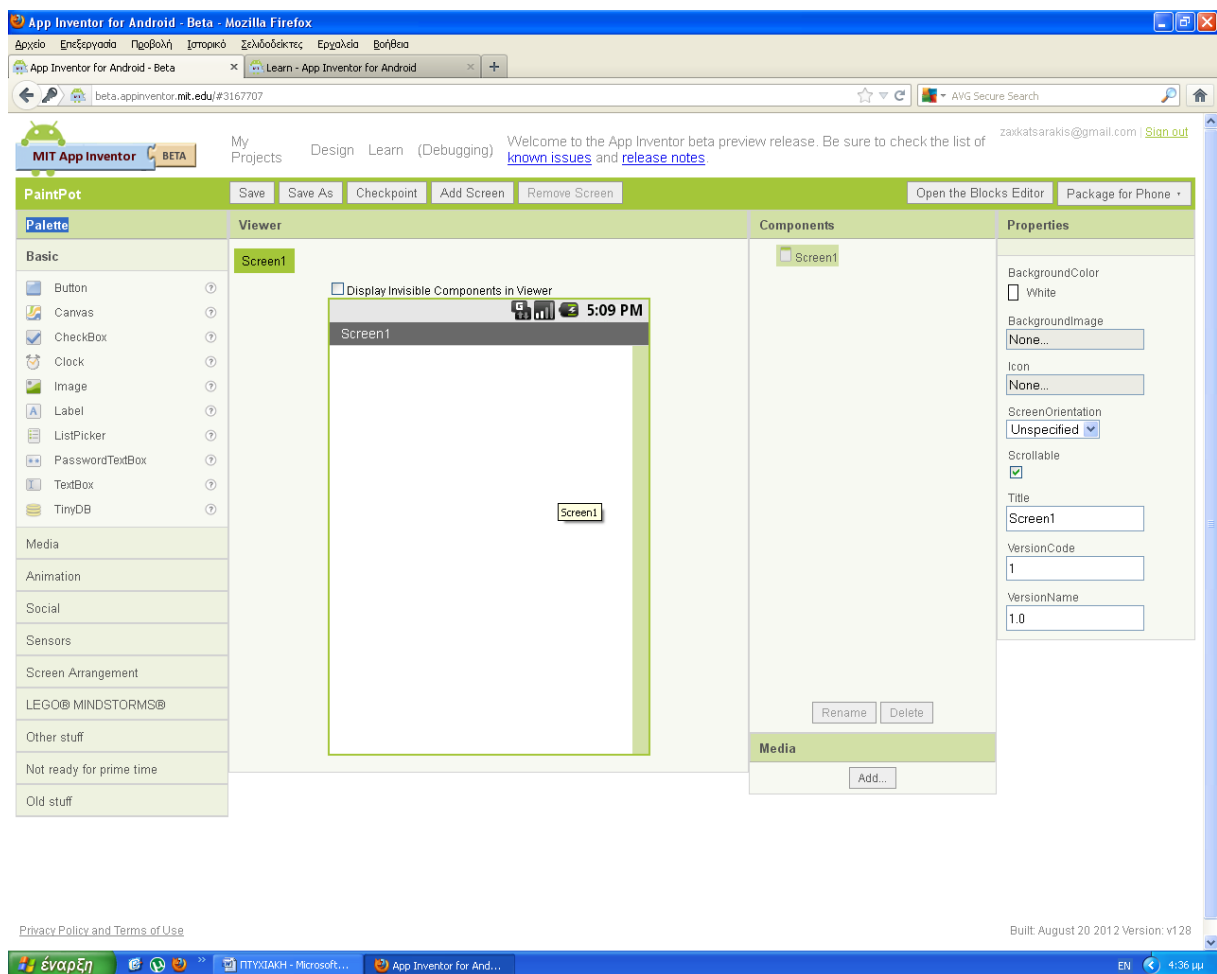
Αφού πραγματοποιήσουμε όλα τα παραπάνω είμαστε έτοιμοι δημιουργήσουμε την εφαρμογή μας. Στη κεντρική οθόνη του App Inventor πατάμε New για να δημιουργήσουμε ένα νέο project με το όνομα: PaintPot (για τη συγκεκριμένη εφαρμογή).



Εικόνα 34 Δημιουργία ένα νέο project.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πατάμε ok και μας πηγαίνει στην οθόνη όπου μπορούμε να δημιουργήσουμε την εφαρμογή μας.

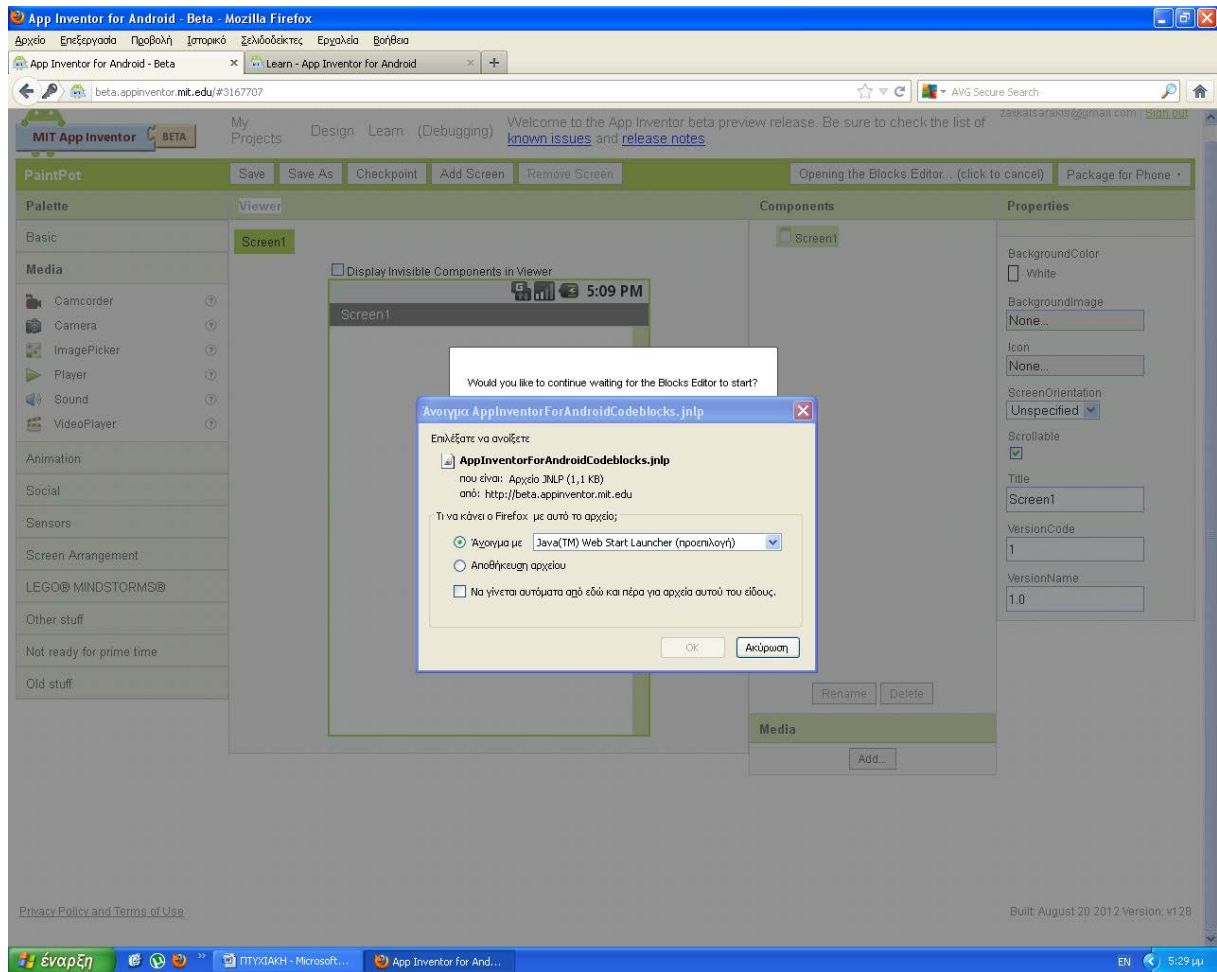


Εικόνα 35 Κεντρική οθόνη του App Inventor.

Αυτή η οθόνη είναι χωρισμένη σε τέσσερα μέρη: Το πρώτο από αριστερά είναι η παλέτα (Palette) που περιέχει όλα αυτά που μπορούμε να βάλουμε στην εφαρμογή μας, όπως Button, Canvas, checkbox, μπορούμε να ελέγξουμε την κάμερα να βάλουμε ένα player που να παίζει ήχο ένα video player κ.α. Το δεύτερο είναι ο Viewer που είναι στην ουσία το μέρος όπου θα φτιάχνουμε την εφαρμογή μας. Το τρίτο είναι το Components, εκεί εμφανίζεται οτιδήποτε βάζουμε στον Viewer. Και το τέταρτο είναι τα Properties (ιδιότητες) του κάθε Components που έχουμε μέσα στον Viewer.

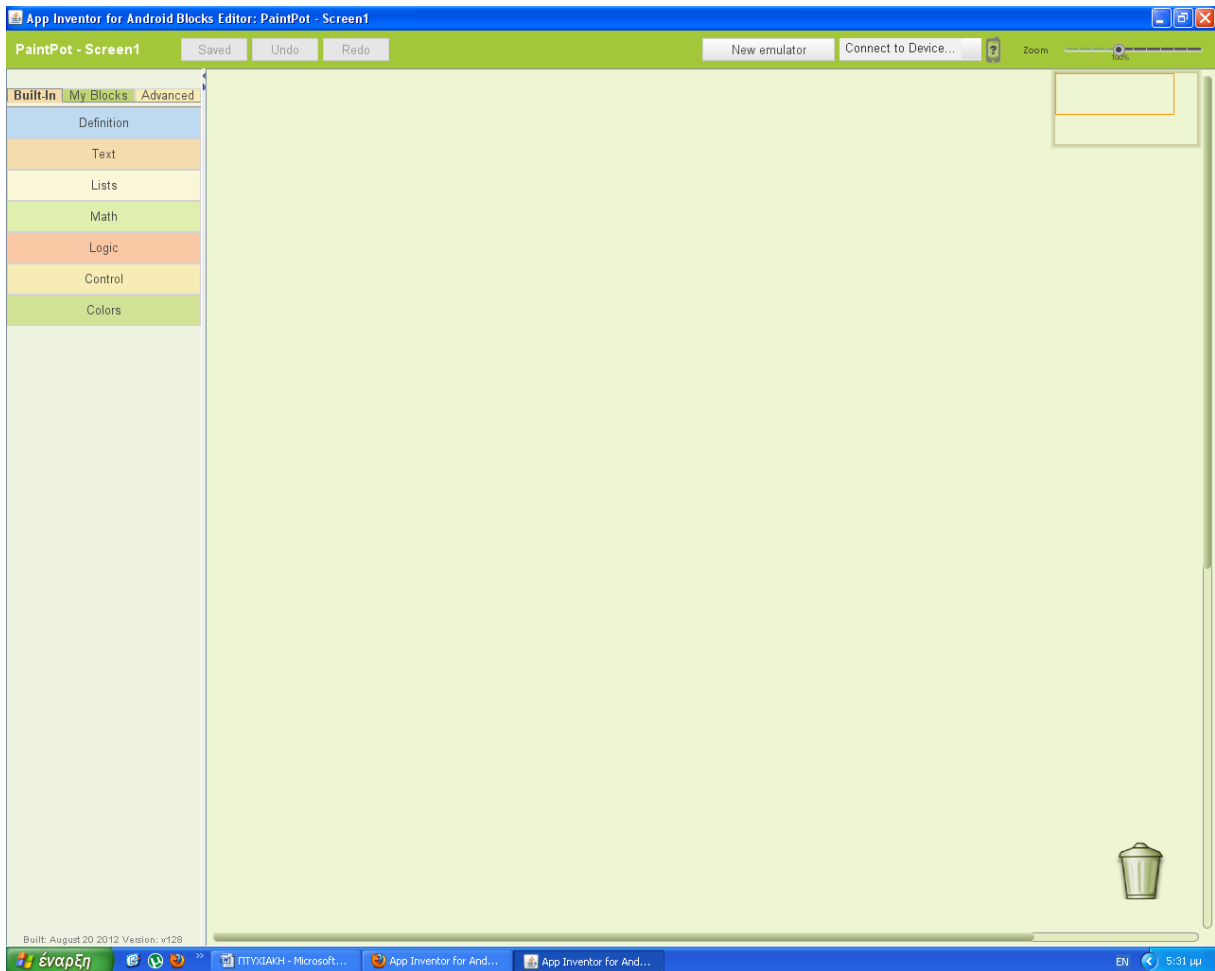
6.3.1 The Blocks Editor

Ο κώδικας της εφαρμογής που έχουμε δημιουργήσει στον Viewer φαίνεται στον Blocks Editor, εκεί μπορούμε να <<γράψουμε>> τον κατάλληλο κώδικα ώστε να κάνει η εφαρμογή μας αυτό που θέλουμε. Πατώντας το κουμπί Open the Blocks Editor μας φέρνει ένα αρχείο από το internet το οποίο για να το ανοίξουμε χρειαζόμαστε την java που πρέπει να είναι εγκατεστημένη στον υπολογιστή μας όπως είπαμε παραπάνω. Όποτε αλλάζουμε την εφαρμογή στον Viewer στην κεντρική οθόνη του App Inventor, αυτόματα θα αλλάξει και ο κώδικας στον Blocks Editor.



Εικόνα 36 Ανοίγμα του Blocks Editor.

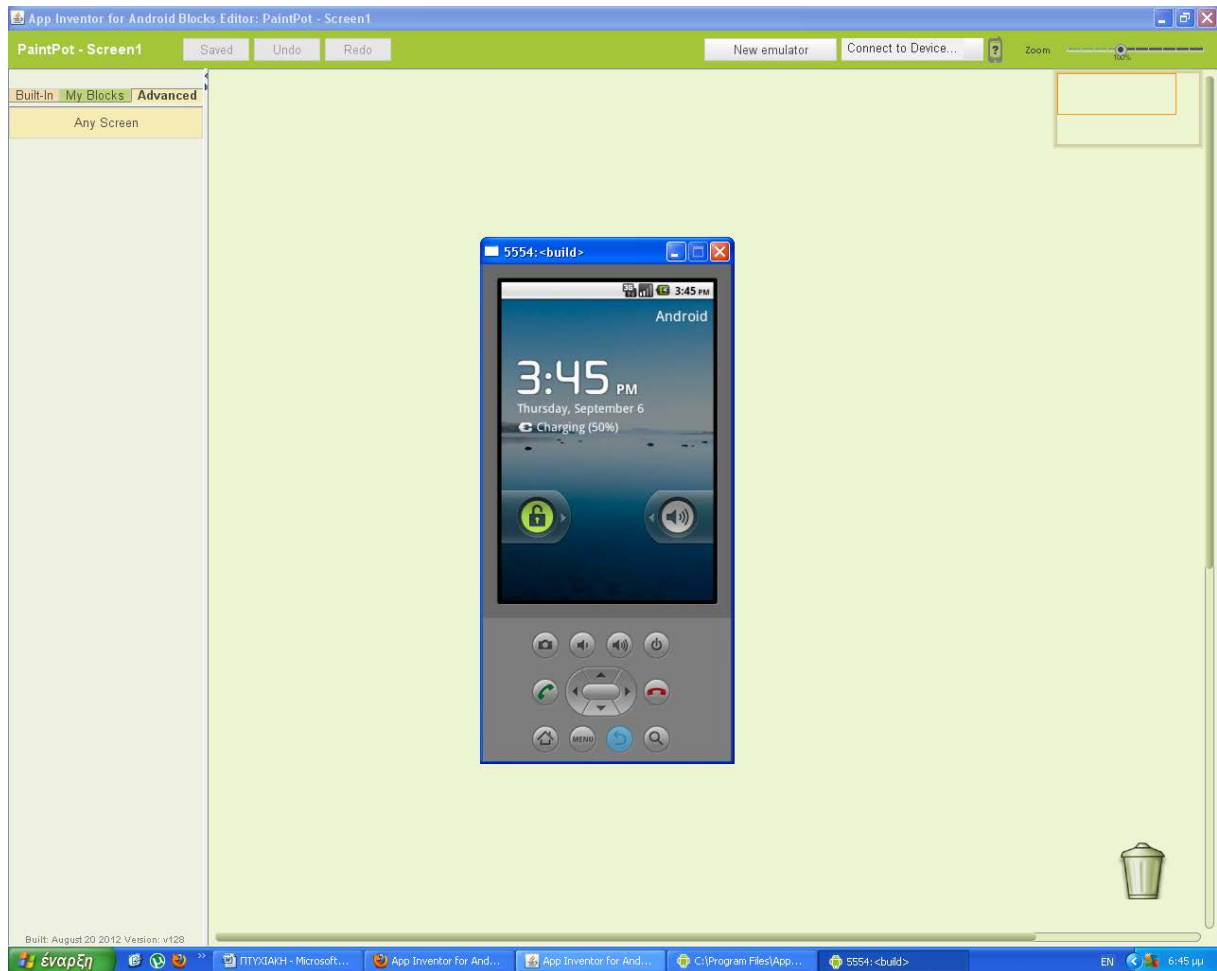
Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 37 Ο Blocks Editor.

Η αριστερή μεριά του Blocks Editor χωρίζεται σε τρία μέρη (Built-In, My Blocks, Advanced) εκεί είναι όλα όσα χρειαζόμαστε για να δημιουργήσουμε τον κατάλληλο κώδικα για την εφαρμογή μας. Όλα είναι σε στιλ πάζλ και με χρώματα για να μπορούμε να τα ξεχωρίζουμε εύκολα. Στο Built-In υπάρχουν βασικά κομμάτια κώδικα όπως ερωτήσεις λογικής, ελέγχου κ.α., και στο My Blocks υπάρχουν αυτά που έχουμε προσθέσει εμείς στον Viewer στην κεντρική οθόνη του App Inventor. Στο κέντρο του Blocks Editor είναι το μέρος όπου δημιουργούμε τον κώδικά μας και κάτω δεξιά είναι ο κάδος που πηγαίνουν τα κομμάτια κώδικα που διαγράφουμε. Μπορούμε μέσω του Blocks Editor να τρέξουμε τον Emulator πατώντας το κουμπί New Emulator που βρίσκεται στην γραμμή εργαλείων του Block Editor και να δοκιμάσουμε την εφαρμογή μας.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



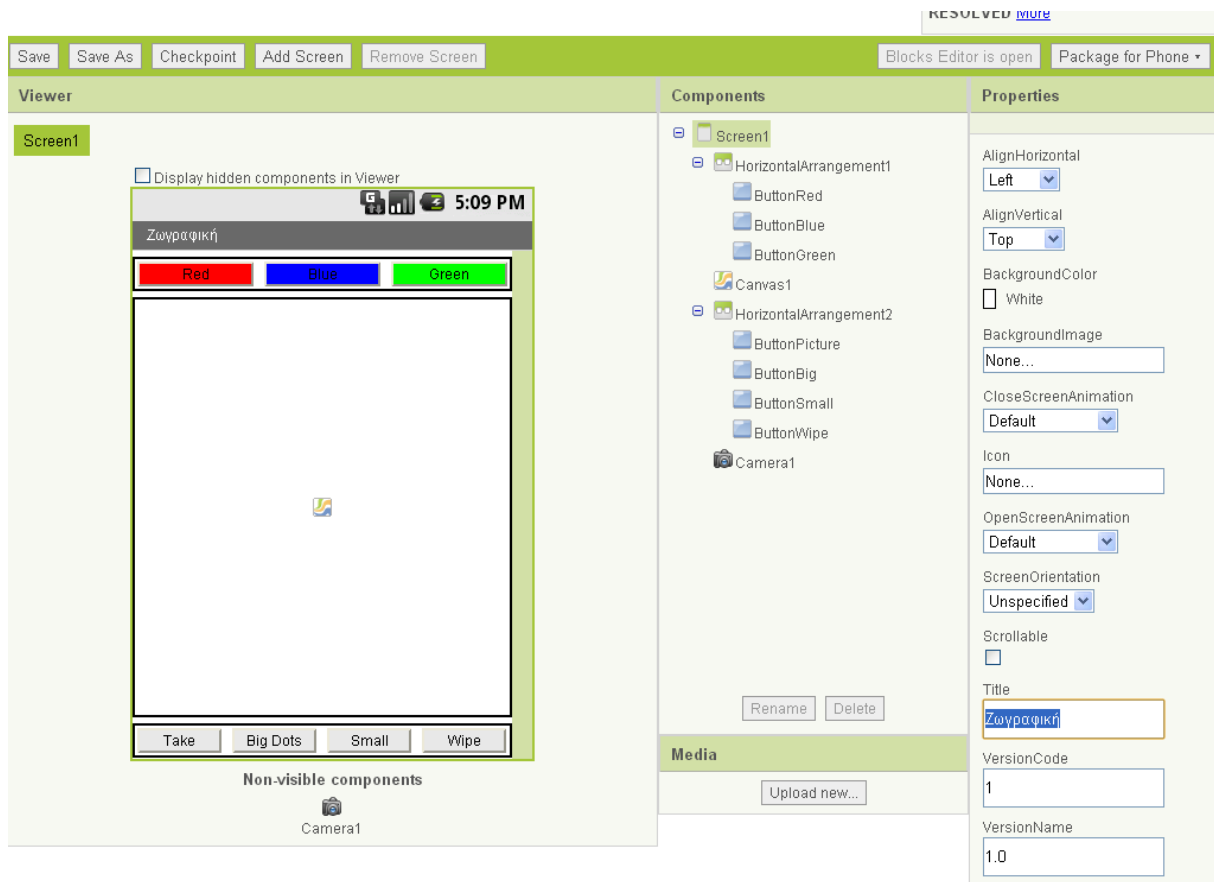
Εικόνα 38 Ο Emulator.

Μπορούμε ακόμα να τεστάρουμε την εφαρμογή μας χωρίς να την έχουμε κατεβάσει (απλά να φαίνεται) σε μια Android συσκευή συνδέοντάς την με καλώδιο usb. Αυτό γίνεται πατώντας το κουμπί Connect to Device από τη γραμμή εργαλείων του Blocks Editor.

Κεφάλαιο 7

7.1 Η εφαρμογή Paint Pot

Είναι μια εφαρμογή με την οποία μπορούμε να τραβάμε φωτογραφίες και να ζωγραφίζουμε πάνω σε αυτές σέρνοντας το δάκτυλο μας στην οθόνη του smart phone. Ξεκινώντας, πηγαίνουμε στον πίνακα ιδιοτήτων (properties) στα δεξιά του Σχεδιαστή (Designer) και αλλάζουμε τον τίτλο οθόνης σε «Ζωγραφική» όπως φαίνεται στην εικόνα.



Built: September 20 2013 Version: v134

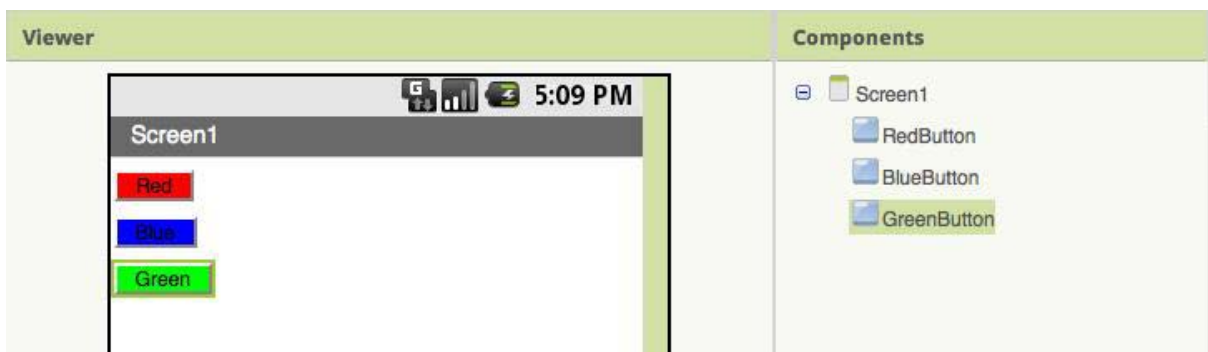
Εικόνα 39 Τίτλος οθόνης εφαρμογής.

7.2 Τι θα χρησιμοποιήσουμε για να κάνουμε την εφαρμογή:

- Τρία κουμπιά (three button component) για να επιλέγουμε κόκκινο, μπλε ή πράσινο χρώμα κι ένα εξάρτημα οριζόντιας διάταξης (horizontal arrangement component) για να τα οργανώσουμε.
- Ένα κουμπί για τον καθαρισμό του σχεδίου, και δύο για να αλλάζουμε το μέγεθος των κουκίδων.
- Έναν «καμβά» (Canvas component) που είναι η επιφάνεια σχεδίασης.
- Και ένα κουμπί για τη λείψη φωτογραφιών.

7.3 Φτιάχνοντας τα κουμπιά χρώματος

1. Σέρνουμε ένα στοιχείο “Button” στον viewer και αλλάζουμε το χαρακτηριστικό κειμένου σε “Κόκκινο” και το χρώμα του, background κόκκινο.
2. Επιλέγουμε το button 1 στην λίστα με τα Components και κάνουμε κλικ στο «Μετονομασία» (Rename) για να αλλάξετε το όνομα από Button1 σε RedButton.
3. Ομοίως, κάνουμε δύο ακόμα κουμπιά για το μπλε και το πράσινο με ονόματα BlueButton και GreenButton.

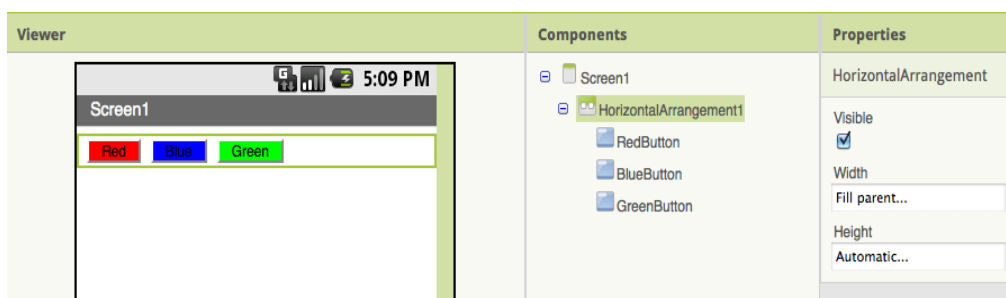


Εικόνα 40 Στον Viewer φαίνονται τα τρία κουμπιά που δημιουργήσαμε.

7.3.1 Χρήση ρυθμίσεων για καλύτερες διατάξεις (Layouts)

Ως τώρα έχουμε τρία κουμπιά κολλημένα το ένα πάνω στο άλλο. Αλλά για αυτήν την εφαρμογή θα θέλουμε όλα στη σειρά το ένα δίπλα στο άλλο, στο πάνω μέρος της οθόνης. Αυτό το κάνουμε χρησιμοποιώντας το εργαλείο HorizontalArrangement:

1. Από την κατηγορία Palette's Screen Arrangement (παλέτα διάταξης οθόνης) σέρνουμε έξω ένα εξάρτημα HorizontalArrangement και το τοποθετούμε κάτω από τα κουμπιά.
2. Στον πίνακα ιδιοτήτων αλλάζουμε το Width of the HorizontalArrangement σε “Fill parent” έτσι ώστε να καλύπτει όλο το πλάτος της οθόνης
3. Μετακινούμε τα κουμπιά ένα ένα στο εργαλείο HorizontalArrangement. Μια μπλε κάθετη γραμμή μας δείχνει πού θα πάει το κομμάτι που σέρνουμε.

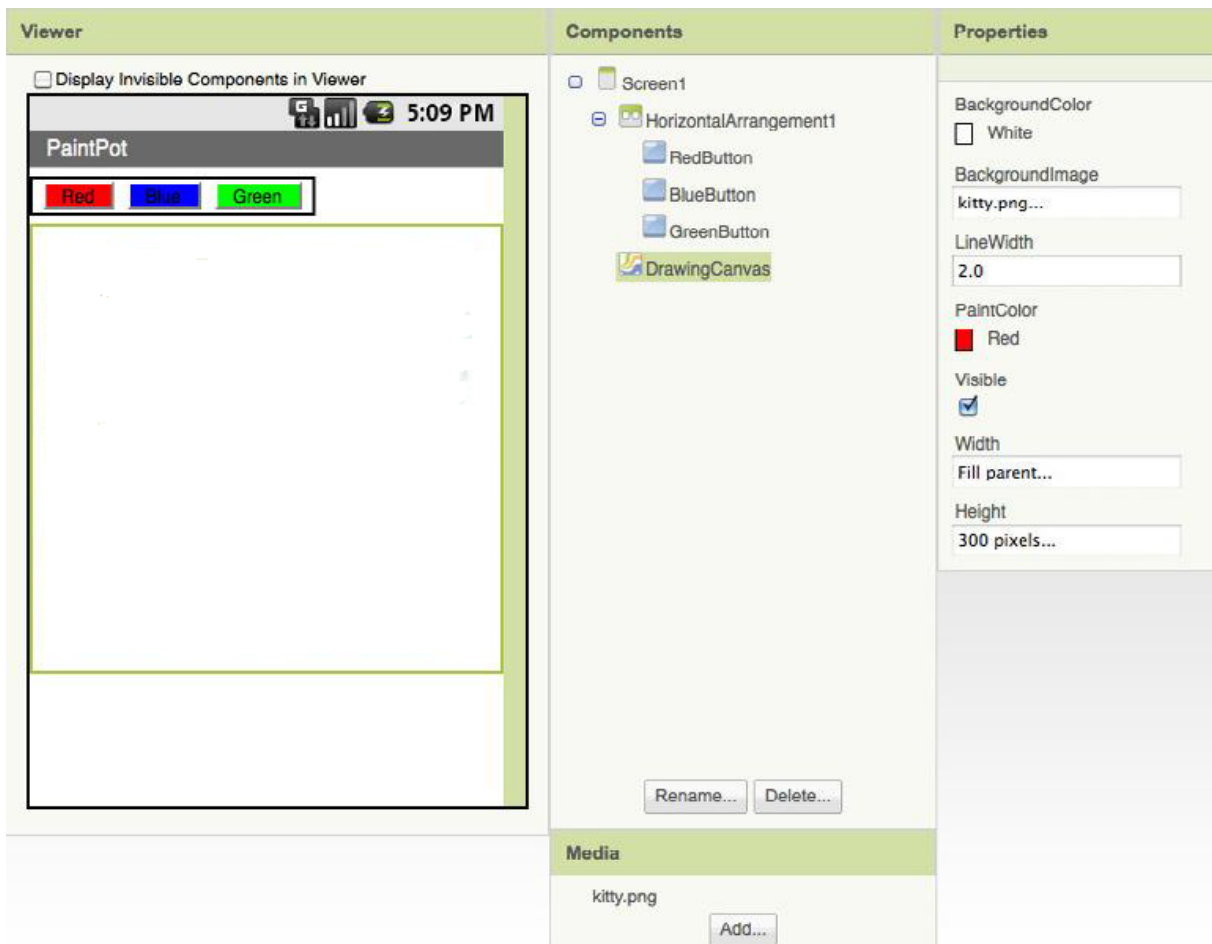


Εικόνα 41 Στον Viewer φαίνονται τα τρία κουμπιά τοποθετημένα στο Horizontal Arrangement.

7.3.2 Προσθέτοντας τον καμβά

Ο καμβάς είναι εκεί όπου ο χρήστης θα σχεδιάζει κύκλους και γραμμές.

1. Από την κατηγορία Palette's Basic, σέρνουμε ένα εξάρτημα καμβά στο Viewer. Βάζουμε το πλάτος τους σε "Fill parent" και το ύψος σε 300 pixels.
2. Ορίσαμε το PaintColor του καμβά μαύρο έτσι ώστε όταν ο χρήστης ξεκινήσει την εφαρμογή αλλά δεν έχει επιλέξει ακόμα κάποιο χρώμα, να σχεδιάζει με μαύρο.

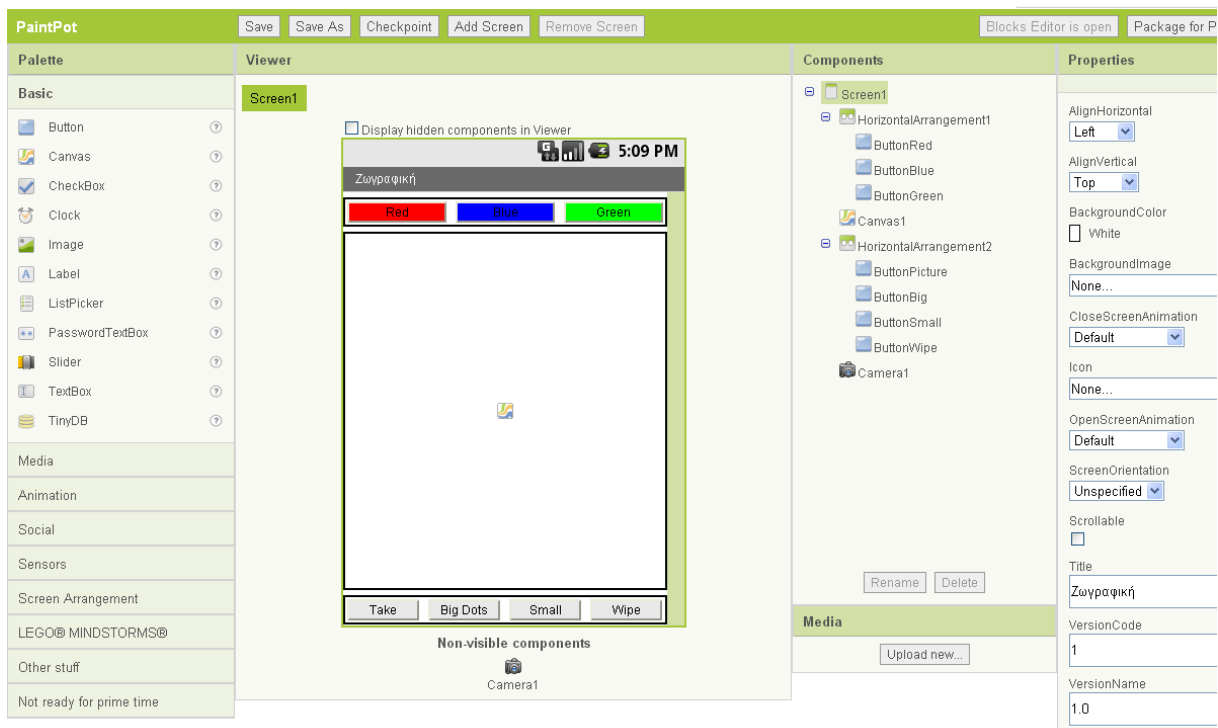


Εικόνα 42 Το εξάρτημα καμβάς τοποθετημένο στο Viewer.

7.3.3 Οργάνωση κάτω κουμπιών.

1. Από την παλέττα (Palette) σέρνουμε ένα δεύτερο HorizontalArrangement και το τοποθετούμε κάτω από τον καμβά. Έπειτα σέρνουμε δύο ακόμα κουμπιά (Button) και τα τοποθετούμε στο κάτω HorizontalArrangement. Αλλάζουμε το όνομα του πρώτου κουμπιού σε TakePicture. Αλλάζουμε το όνομα του δεύτερου κουμπιού σε WipeButton.
2. Σέρνουμε δύο ακόμα κουμπιά από το Palette στο HorizontalArrangement και τοποθετείστε τα δίπλα στο WipeButton.
3. Ονομάζουμε τα κουμπιά BigButton και SmallButton.
4. Από το Media Palette σερνουμε το camera component στο Viewer.

Έχουμε τώρα συμπληρώσει τα βήματα για να φτιάξουμε την εμφάνιση της εφαρμογής όπως δείχνει το σχήμα.



Εικόνα 43 Το user interface της εφαρμογής μας.

7.4 Προσθέτοντας συμπεριφορές στα εργαλεία

Το επόμενο βήμα είναι να ορίσουμε πώς θα συμπεριφέρονται τα εργαλεία. Η δημιουργία ενός προγράμματος σχεδίασης μπορεί να φαίνεται δύσκολη, αλλά όπως έχουμε ήδη προαναφέρει το AppInventor κάνει μεγάλο μέρος της βαριάς δουλειάς για εμάς. Υπάρχουν εύκολα στη χρήση κομμάτια (blocks) για να διαχειρίζονται το άγγιγμα και το σύρσιμο του χρήστη και για να σχεδιάζουν και να παίρνουν φωτογραφίες.

Στο σχεδιαστή (Designer) προσθέσαμε ένα εργαλείο καμβά με το όνομα DrawingCanvas. Όπως όλα τα εργαλεία καμβά, το DrawingCanvas έχει εκδήλωση αφής (Touching event) και σύρσης (Dragging event). Θα προγραμματίσουμε το DrawingCanvas.Touched event έτσι ώστε να ζητά DrawingCanvas.DrawCircle. Στη συνέχεια θα προγραμματίσουμε το DrawingCanvas.Dragged event να ζητά DrawingCanvas.Drawline. Έπειτα, θα προγραμματίσουμε τα κουμπιά για να ρυθμίσουμε την ιδιότητα DrawingCanvas.PaintColor, και να αλλάξουμε το BackgroundImage με φωτογραφία που θα λαμβάνουμε από την κάμερα.

7.4.1 Προσθέτοντας το Touch event για να σχεδιάσετε μια τελεία (κουκίδα)

Αρχικά θα οργανώσουμε τα πράγματα έτσι ώστε όταν αγγίζουμε το DrawingCanvas, να σχεδιάζετε μια κουκίδα στο σημείο που αγγίζουμε:

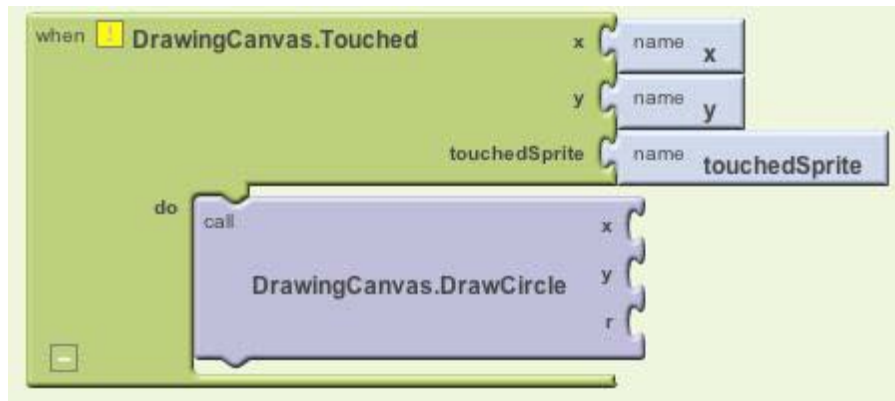
1. Στο Block Editor κάναμε κλικ στο My Blocks, επιλέξαμε το σχεδιαστή για το DrawingCanvas, και σύραμε το block DrawingCanvas.Touched στο χώρο εργασίας. Μόλις σύραμε το block, οι τρεις τάπες (plugs) στα δεξιά συμπληρώνοντε αυτόματα με ονόματα block για x, y και touchedSprite, όπως φαίνεται στο σχήμα.



Εικόνα 44 Το block DrawingCanvas.Touched.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

- Σέρνουμε την εντολή `DrawingCanvas.DrawCircle` και τη τοποθετούμε μέσα στο (χειριστή συμβάντος) `DrawingCanvas.Touched` event όπως φαίνεται στο σχήμα.



Εικόνα 45 Τοποθέτηση του block `DrawingCanvas.DrawCircle` στο block `DrawingCanvas.Touched`.

Στη δεξιά πλευρά του block `DrawingCanvas.DrawCircle`, θα δούμε τρεις υποδοχές για τα arguments που χρειάζεται να συμπληρώσουμε: `x`, `y` και `r`. Τα arguments `x`, `y` ορίζουν που θα πρέπει να σχεδιαστεί ο κύκλος, και το `r` ορίζει την ακτίνα του κύκλου. Το κίτρινο κουτί ειδοποιήσεων με το θαυμαστικό στο πάνω μέρος του `DrawingCanvas.Touched` event, υποδεικνύει ότι αυτές οι υποδοχές δεν έχουν συμπληρωθεί ακόμα.

Θα πρέπει να θυμόμαστε ότι τα `x`, `y` για το `DrawingCanvas.Touched` event μας λένε πού έχει αγγίξει ο χρήστης, ενώ τα `x`, `y` για το `DrawingCanvas.DrawCircle` event είναι ανοιχτές υποδοχές για να ορίσουμε εμείς πού πρέπει να σχεδιαστεί ο κύκλος.

Επειδή θέλουμε να σχεδιάσουμε τον κύκλο εκεί που θα αγγίξει ο χρήστης, συνδέουμε τις τιμές του `x` και `y` από `DrawingCanvas.Touched` όπως οι τιμές των παραμέτρων `X` και `Y` στο `DrawingCanvas.DrawCircle`.

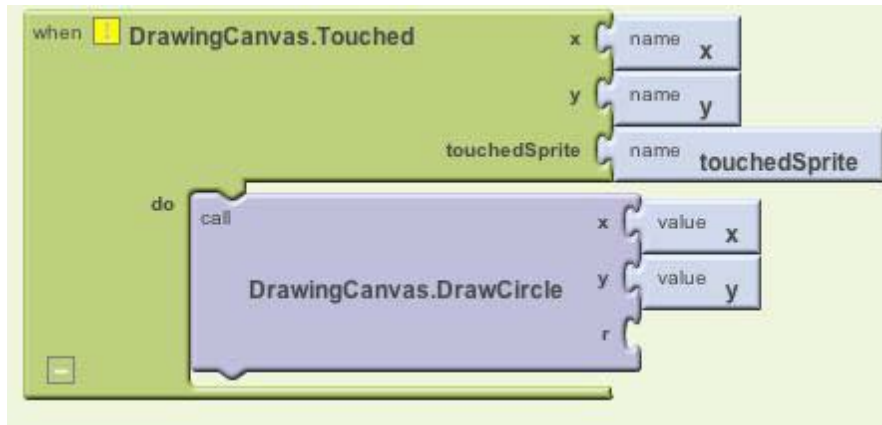
*Σημείωση: δεν θα πρέπει να πάρουμε κατευθείαν τα arguments του `Touched`, παρόλο που αυτό μπορεί να φαίνεται λογικό. Το γεγονός ότι τα arguments μπορούν ακόμη να πιαστούν είναι μια ατυχής όψη του App Inventor. Αντιθέτως, θέλουμε να πάρουμε αυτά τα values από το σχεδιαστή `My Definitions` όπως φαίνεται στο σχήμα.



Εικόνα 46 Τα values του block `DrawingCanvas.DrawCircle` όπως φαίνονται στο `My Definitions`.

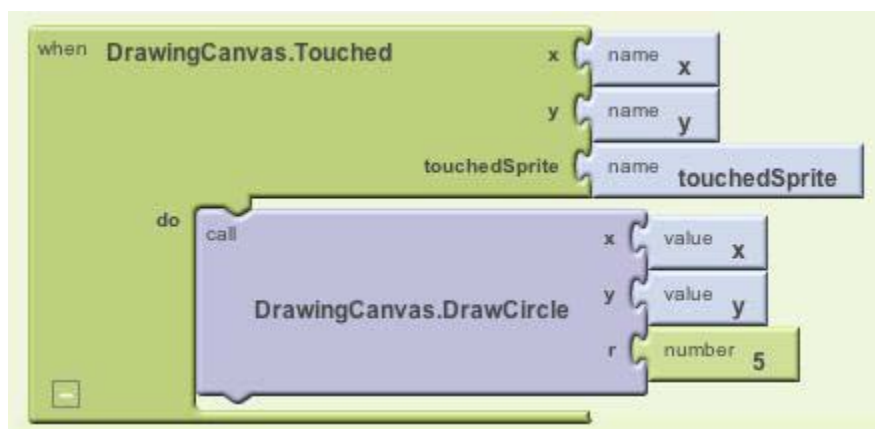
Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

3. Ανοίγουμε το σχεδιαστή My Definitions μέσα στο My Blocks και βρούμε τα blocks για value x και y. Τα blocks είχαν αυτόματα δημιουργηθεί για εμάς από το AppInventor όταν σύραμε το block DrawingCanvas.Touche event: είναι αναφορές των x, y arguments αυτού του συμβάντος. Σέρνουμε τα blocks των x, y values και τα βάζουμε στις αντίστοιχες υποδοχές στο block DrawingCanvas.DrawCircle, έτσι ώστε να μοιάζουν όπως δείχνει το σχήμα.



Εικόνα 47 Τα blocks x,y values τοποθετημένα στο block DrawingCanvs.DrawCircle.

4. Θα χρειαστεί επίσης να συγκεκριμενοποιήσουμε την ακτίνα, r, του κύκλου που θα σχεδιάσουμε. Η ακτίνα μετριέται σε pixels, που είναι η μικρότερη κουκίδα που μπορεί να σχεδιαστεί στην οθόνη. Προς το παρόν ορίζουμε σε 5: κάνουμε κλικ στην κενή περιοχή στην οθόνη για να εμφανιστεί το μενού συντόμευσης, και μετά επιλέγουμε το φάκελο Math. Επιλέγουμε 123 από τη λίστα για να δημιουργήσουμε ένα block αριθμών. Αλλάζουμε το 123 σε 5 και το βάζουμε στην υποδοχή του r. Όταν το κάνουμε, το κίτρινο κουτί στην πάνω αριστερή γωνία θα εξαφανιστεί καθώς όλες οι υποδοχές έχουν συμπληρωθεί. Το σχήμα απεικονίζει πώς θα έπρεπε να φαίνεται ο τελικός χειριστής του συμβάντος DrawingCanvas.Touched.



Εικόνα 48 Ο τελικός χειριστής συμβάντος του DrawingCanvas.Touched.

7.4.2 Προσθέτοντας το γεγονός Drag που σχεδιάζει μια γραμμή.

Στη συνέχεια, προσθέτουμε το πρόγραμμα χειρισμού συμβάντων drag. Εδώ είναι η διαφορά ανάμεσα σε μια επαφή και ένα σύρσιμο:

- Ένα άγγιγμα είναι όταν τοποθετούμε το δάχτυλό μας πάνω στον καμβά και το σηκώνουμε χωρίς κίνηση.
- Ένα drag είναι όταν τοποθετούμε το δάχτυλό μας πάνω στον καμβά και το μετακινούμε διατηρώντας το σε επαφή με την οθόνη.

Σε ένα πρόγραμμα ζωγραφικής, αυτό που πραγματικά γίνεται σέρονοντας το δάχτυλό μας πάνω στην οθόνη για να δημιουργήσουμε μια καμπύλη γραμμή είναι η σχεδίαση εκατοντάδων μικροσκοπικών ευθείων γραμμών. Κάθε φορά που κινείτε το δάχτυλό μας, έστω και λίγο, θα παρατείνει την γραμμή από την τελευταία θέση του δακτύλου μας στη νέα του θέση.

1. Από το DrawingCanvas, σέρνουμε το DrawingCanvas.Dragged block.

Στο DrawingCanvas.Dragged event εμφανίζονται επτά arguments:

startx, starty:

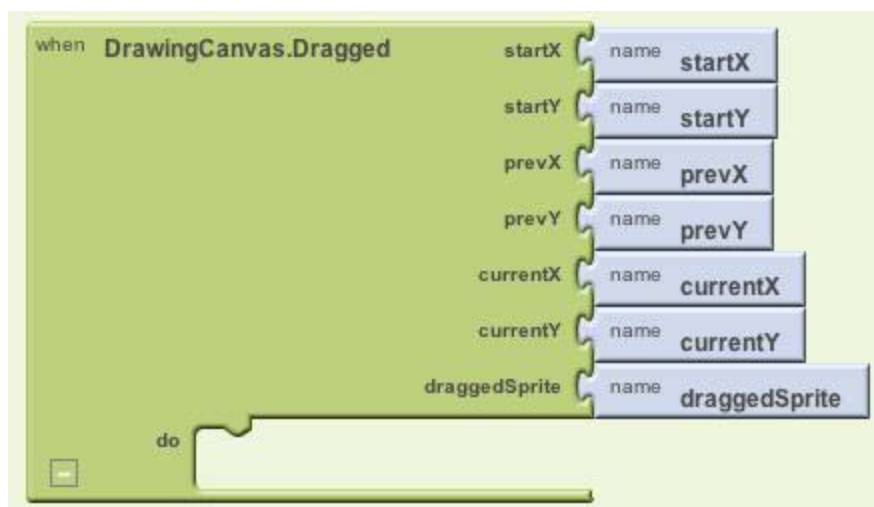
Η θέση του δάχτυλου μας, στην περίπτωση που έρχεται σε επαφή με την οθόνη.

currentx, currenty:

Η θέση του δάχτυλου μας αυτή τη στιγμή.

Prevx, prevy:

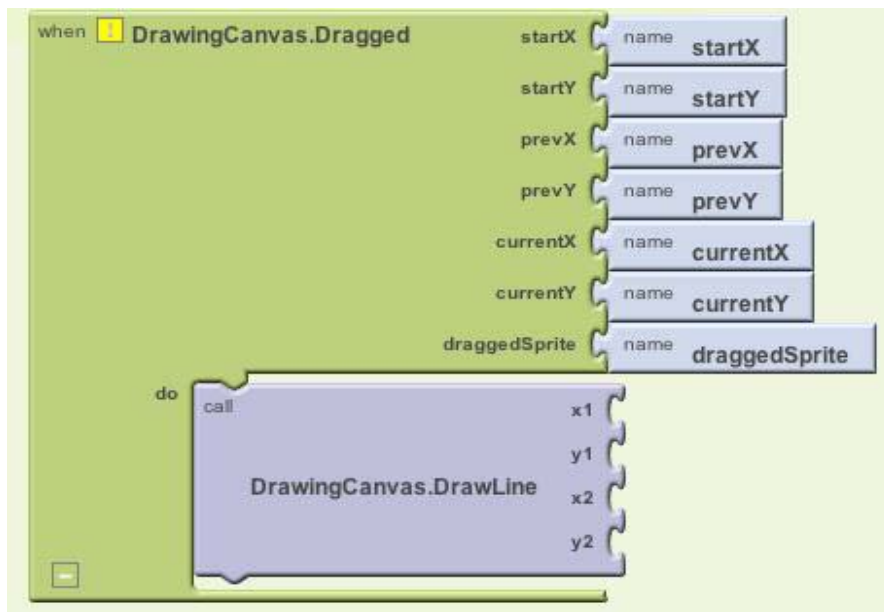
Η αμέσως προηγούμενη θέση του δακτύλου μας.



Εικόνα 49 Τα arguments που εμφανίζονται στο DrawingCanvas.Dragged block.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

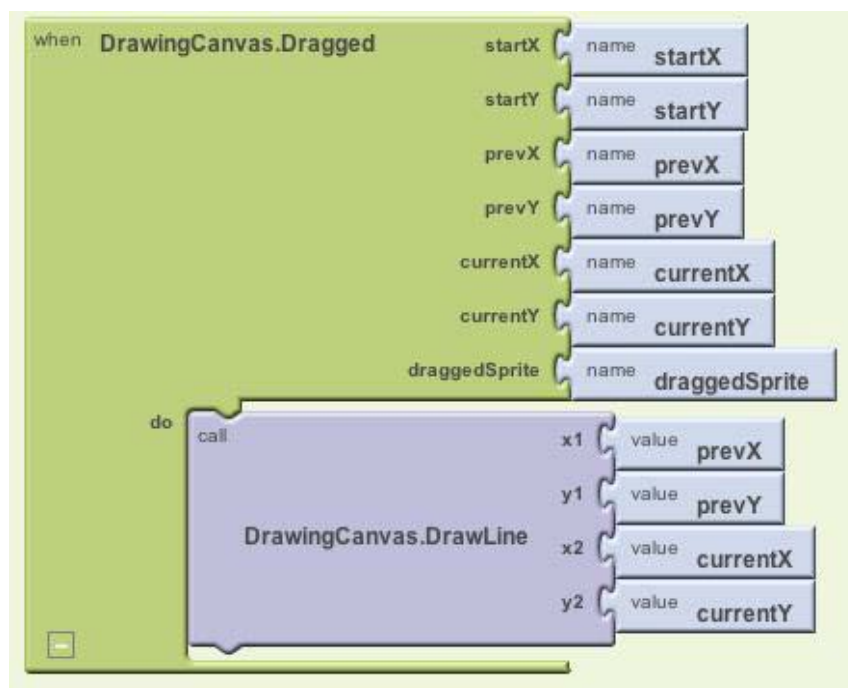
2. Στη συνέχεια απο το DrawingCanvas, σέρνουμε το block DrawingCanvas.DrawLine όπως φαίνεται στο Σχήμα.



Εικόνα 50 Τοποθέτηση του block DrawingCanvas.DrawLine στο DrawingCanvas.Dragged.

3. Το μπλοκ DrawingCanvas.DrawLine έχει τέσσερα arguments, δύο για κάθε σημείο που καθορίζει τη γραμμή: (x1, y1) (x2, y2).

Κάνουμε κλικ στο My Definitions. Θα πρέπει να δούμε τα blocks για τα arguments που χρειαζόμαστε. Σέρνουμε τα αντίστοιχα μπλοκ στις κατάλληλες υποδοχές όπως φαίνεται στο Σχήμα.



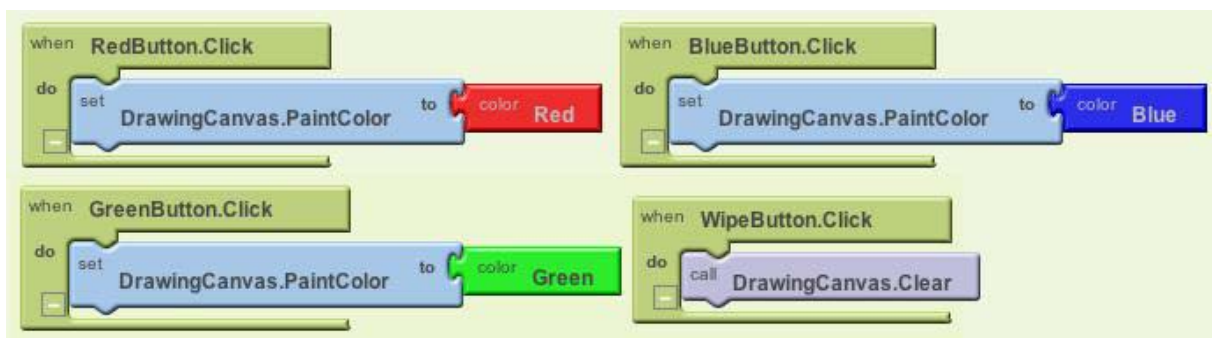
Εικόνα 51 Τα arguments του DrawingCanvas.DrawLine.

7.4.3 Προσθήκη στα κουμπιά Event Handlers.

Στην εφαρμογή που έχουμε φτιάξει μέχρι τώρα μπορούμε να ζωγραφίσουμε μόνο με μαύρο. Στη συνέχεια, θα προσθέσουμε τα event για τα κουμπιά χρώματος και για το WipeButton έτσι ώστε ο χρήστης να μπορεί να αλλάξει το χρώμα βαφής και να καθαρίσει την οθόνη για να ξεκινήσει από την αρχή.

Στον Blocks Editor:

1. Στο My blocks επιλέγουμε το ButtonRed και σέρνουμε το block RedButton.Click.
2. Στο My block επιλεγουμε το DrawingCanvas και σέρνουμε το block DrawingCanvas.PaintColor και το τοποθετούμε στο "do" του RedButton.Click.
3. Από το Built-In επιλέγουμε το block για το κόκκινο και το τοποθετούμε στο DrawingCanvas.PaintColor.
4. Επεναλαμβάνουμε τα βήματα και για το blue και green button.
5. Για να δημιουργήσουμε το WipeButton πηγαίνουμε πάλι στο My block και σέρνουμε το block WipeButton.Click. Στη συνέχεια από το DrawingCanvas σέρνουμε το DrawingCanvas.Clear και το τοποθετούμε στο WipeButton.Click.



Εικόνα 52 Προσθήκη στα κουμπιά Event Handlers.

7.4.4 Τραβηγμα φωτογραφίας

Οι εφαρμογές με το App inventor μπορούν να αλληλεπιδράσουν με τα ισχυρά χαρακτηριστικά μιας συσκευής Android, όπως είναι η φωτογραφική μηχανή.

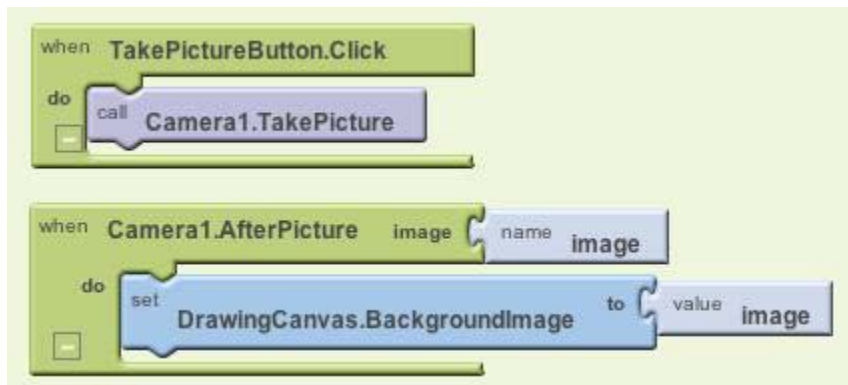
Η συγκεκριμένη εφαρμογή επιτρέπει στο χρήστη να τοποθετήσει στο παρασκήνιο του σχεδίου του μια εικόνα που έχει τραβήξει με την κάμερα.

1. Το Camera component “συστατικό κάμερα” έχει δύο βασικά μπλοκ. Το μπλοκ Camera.TakePicture ξεκινά την κάμερα της συσκευής. Το Camera.AfterPicture event ενεργοποιείται όταν ο χρήστης έχει τελειώσει τη λήψη της φωτογραφίας. Θα προσθέσουμε το κατάλληλο μπλοκ στο Camera.AfterPicture event για να ρυθμίσουμε το DrawingCanvas.Background.Image.
2. Από το My blocks επιλέγουμε ButtonPicture και σέρνουμε το TakePictureButton.Click event στο χώρο εργασίας.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

3. Από το Camera1, σέρνουμε το Camera1.TakePicture και τοποθετούμε στο TakePictureButton.click event.
4. Από το Camera1, σέρνουμε το Camera1.AfterPicture event μέσα στο χώρο εργασίας.
5. Από το DrawingCanvas, σέρνουμε το DrawingCanvas.BackgroundImage και το τοποθετούμε στο πρόγραμμα χειρισμού συμβάντων Camera1.AfterPicture.
6. Το Camera1.AfterPicture έχει ένα argument που ονομάζεται Image, το οποίο είναι η εικόνα που λαμβάνεται. Από το My Definitions palette σέρνουμε το Value Image και το συνδέουμε στο DrawingCanvas.BackgroundImage.

Τα blocks θα πρέπει να φαίνονται όπως στην Εικόνα.



Εικόνα 53 Τα blocks των TakePictureButton.Click και Camera1.AfterPicture.

7.5 Αλλαγή του μεγέθους Dot

Έως τώρα ο χρήστης δεν μπορεί να αλλάξει το μέγεθος της κουκίδας την οποία την έχουμε ορίσει 5 pixel. Σε αυτό το σημείο θα τροποποιήσουμε το πρόγραμμα με χρήση μεταβλητών ώστε ο χρήστης και όχι μόνο ο προγραμματιστής να μπορεί να αλλάξει το μέγεθος της κουκίδας, κάνοντας click στα κουμπιά BigDots όπου το μέγεθος θα είναι 8 pixel και SmallDots όπου το μέγεθος θα είναι 2 pixel.

7.5.1 Καθορισμό μιας μεταβλητής dotSize (DotSize variable):

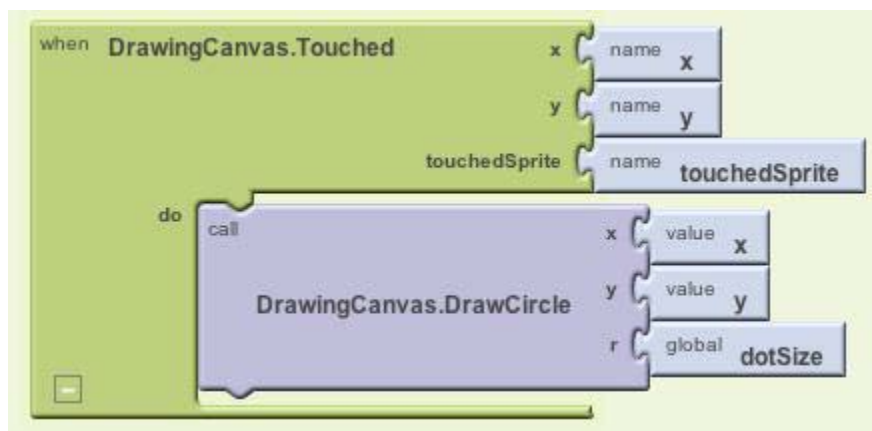
1. Στον Blocks Editor επιλέγουμε Built-in και μετα Definition και σέρνουμε στο χώρο εργασίας το def variable Block και το ονομάζουμε dotSize.
2. Το def dotSize μπλοκ έχει μια ανοιχτή υποδοχή. Εκεί μπορούμε να καθορίσουμε την αρχική τιμή για τη μεταβλητή, ή την προεπιλεγμένη τιμή όταν η εφαρμογή ξεκινά. (αρχικοποίηση μεταβλητής). Στη συνέχεια δημιουργούμε ένα αριθμό κάνοντας κλικ στο Built-in και μετά από το math επιλέγουμε το number βάζοντας του τιμή 2 και το τοποθετούμε στο dotSize όπως φαίνεται στο σχήμα.



Εικόνα 54 Αρχικοποίηση μεταβλητής.

7.5.2 χρησιμοποιώντας μεταβλητές:

1. Εάν παμε στο My Blocks → My Definitions και ρίξουμε μια ματιά στα Blocks θα δούμε ότι υπάρχουν δύο νέα Blocks το global dotSize block και το set global dotSize to block. Αυτά τα blocks δημιουργήθηκαν αυτόματα όταν φτάσαμε την dotSize μεταβλητή.
2. Στη συνέχεια πηγαίνουμε στο DrawingCanvas.Touched και σέρνουμε το block με τον αριθμό 5 από την υποδοχή και τον τοποθετούμε στον κώδο ανακύκλωσης. Βάζουμε στη θέση του το global dotSize block όπως φαίνεται στο σχήμα έτσι ώστε όταν ο χρήστης αγγίζει τον καμβά, η εφαρμογή θα καθορίσει την ακτίνα από τη μεταβλητή dotSize.



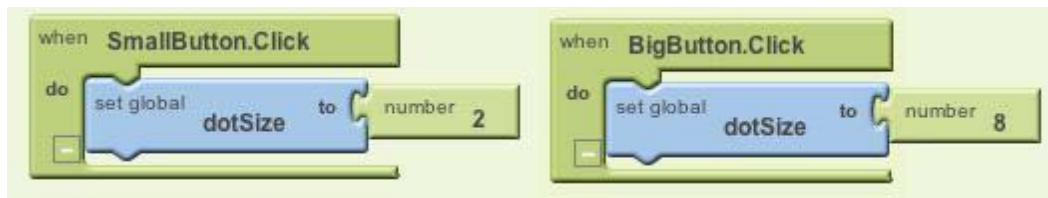
Εικόνα 55 Τοποθέτηση του global dotSize block στο DrawingCanvas.DrawLine.

7.5.3 Αλλάζοντας τις τιμές των μεταβλητών:

Η μεταβλητή dotSize επιτρέπει στο χρήστη να επιλέξει το μέγεθος του κύκλου.

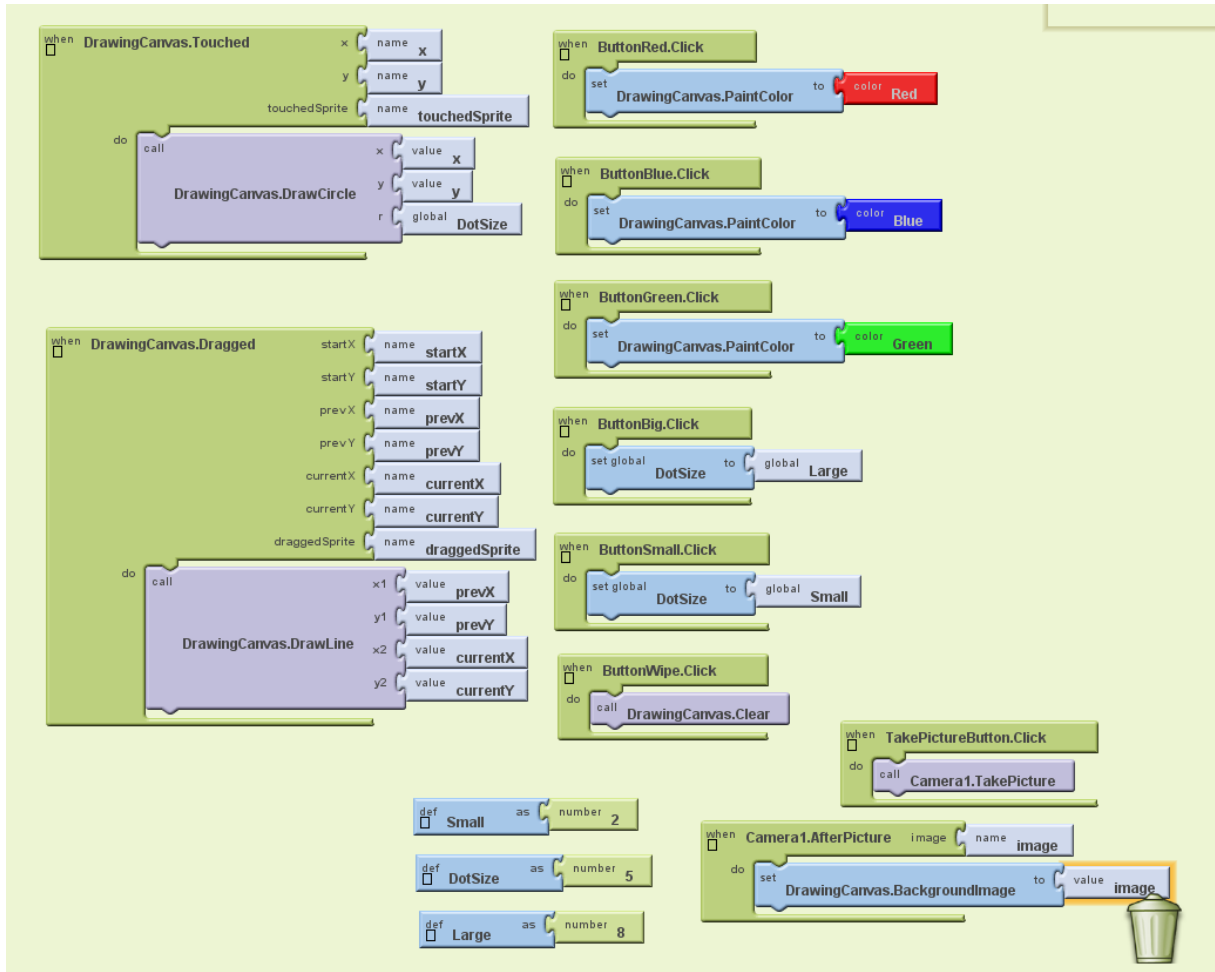
Θα εφαρμόσουμε αυτή τη συμπεριφορά με τον προγραμματισμό του SmallButton.Click και BigButton.Click:

1. Από το My Blocks →buttonSmall σέρνουμε το block SmallButton.Click. στη συνέχεια σέρνουμε από το My Definition το set global DotSize block και το συνδέουμε στο SmallButton.Click. τέλος δημιουργούμε ένα block number 2 και το τοποθετούμε στο block set global dotSize to.
2. Την είδια διαδικασία κάνουμε και για το BigButton.Click μόνο που το number block είναι 8.



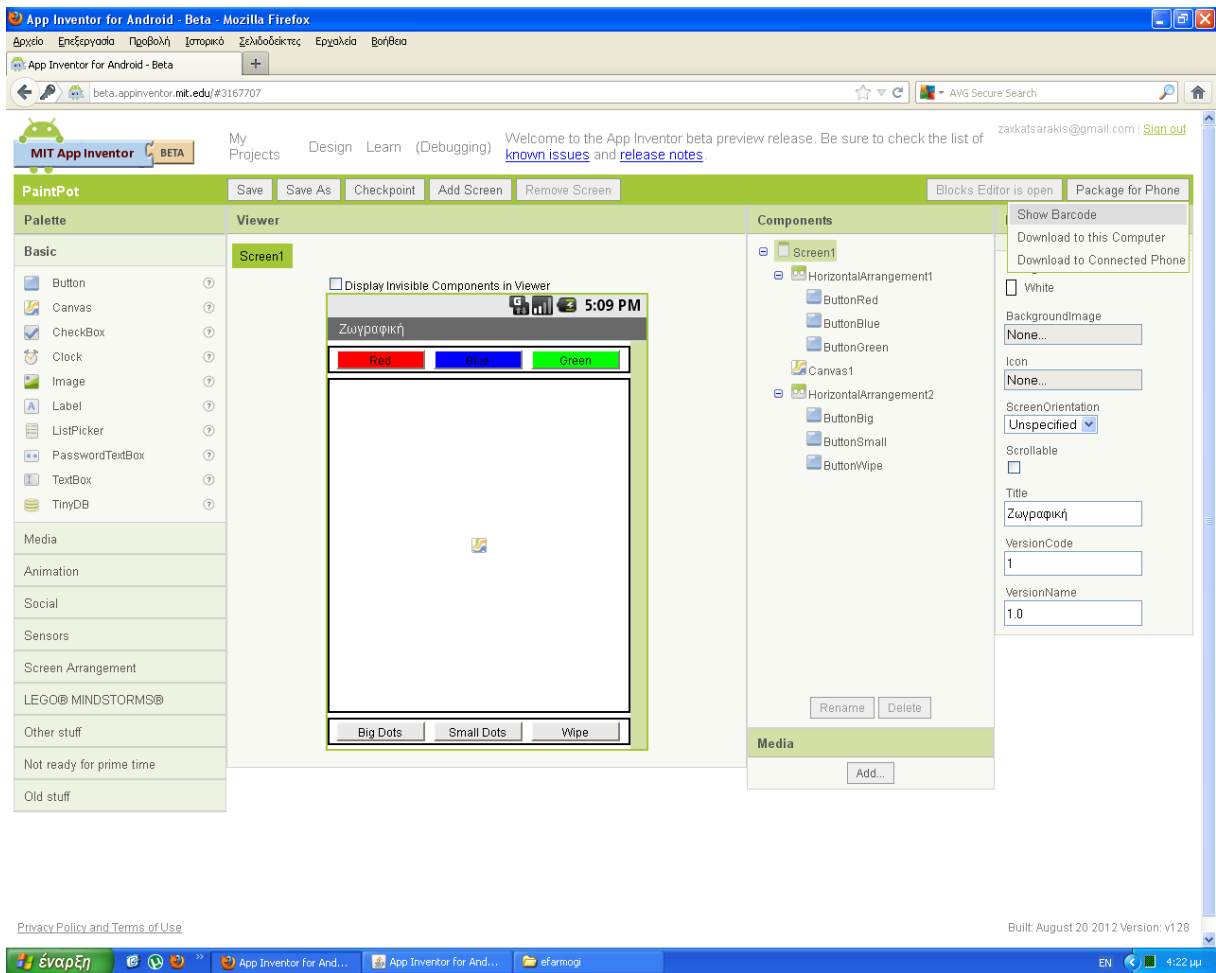
Εικόνα 56 Τα block αλλαγής μεταβλητών.

7.6 Ολοκληρωμένη η εφαρμογή PaintPot:



Εικόνα 57 Ο κώδικας σε block της εφαρμογής Paint Pot 'όπως φαίνεται στον Blocks Editor.

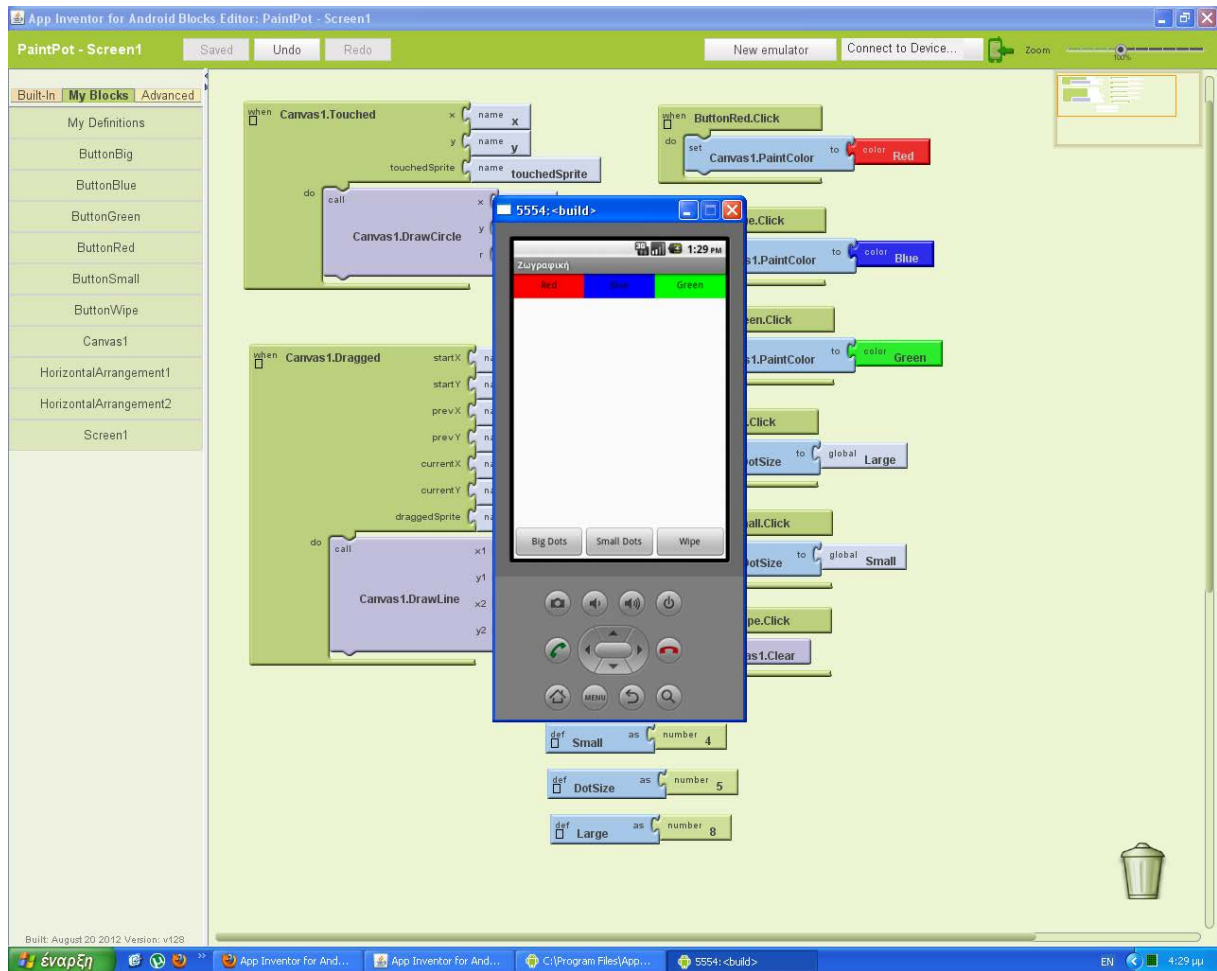
Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 58 Η εφαρμογή Paint Pot όπως φαίνεται στην κεντρική οθόνη του App Inventor.

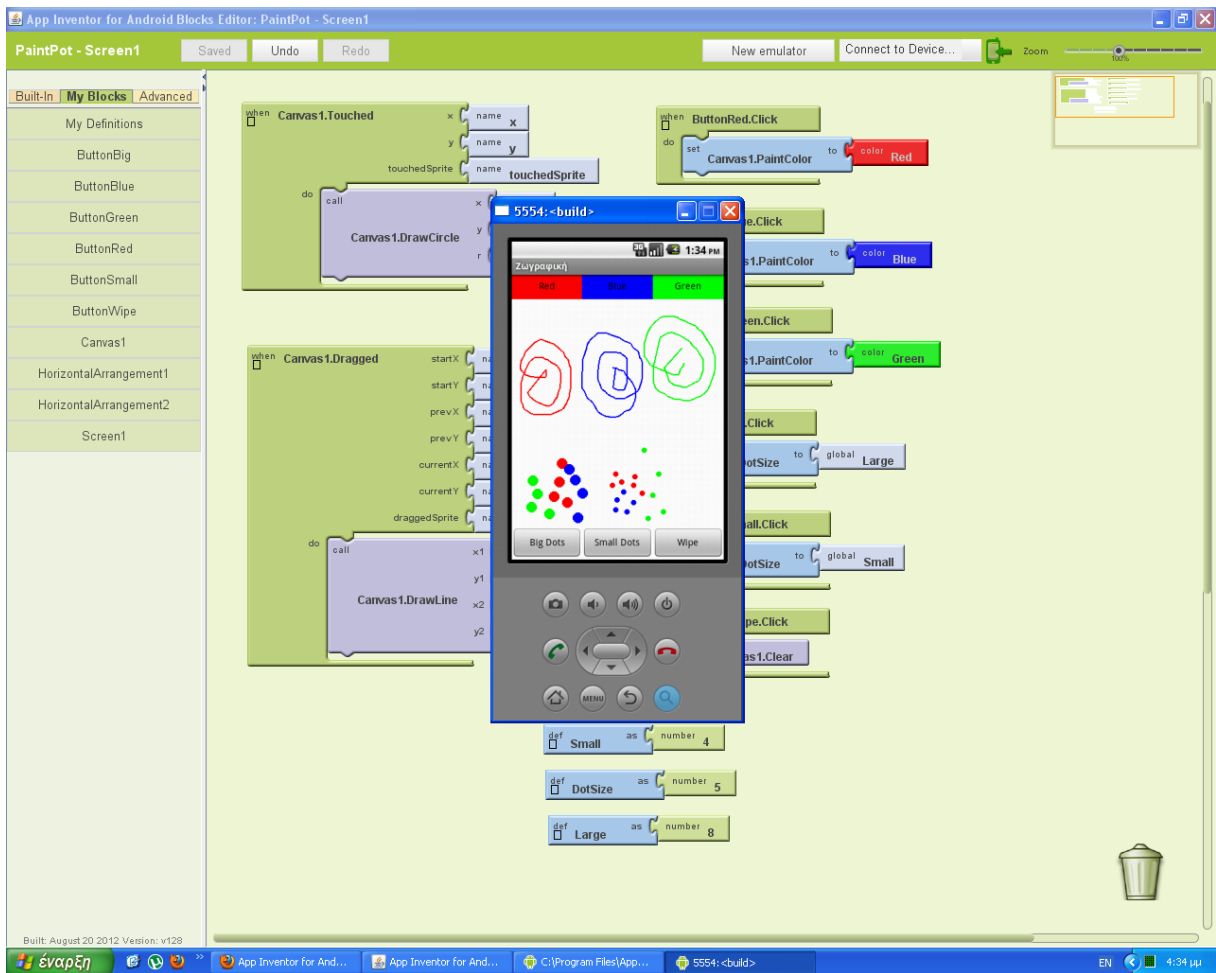
7.7 Test εφαρμογής

Στην συνέχεια για να τσεκάρουμε τον κώδικα και γενικά την εφαρμογή έχουμε εκκινήσει τον Emulator από τον Blocks Editor και έχουμε φορτώσει την εφαρμογή όπως φαίνεται στις παρακάτω εικόνες.



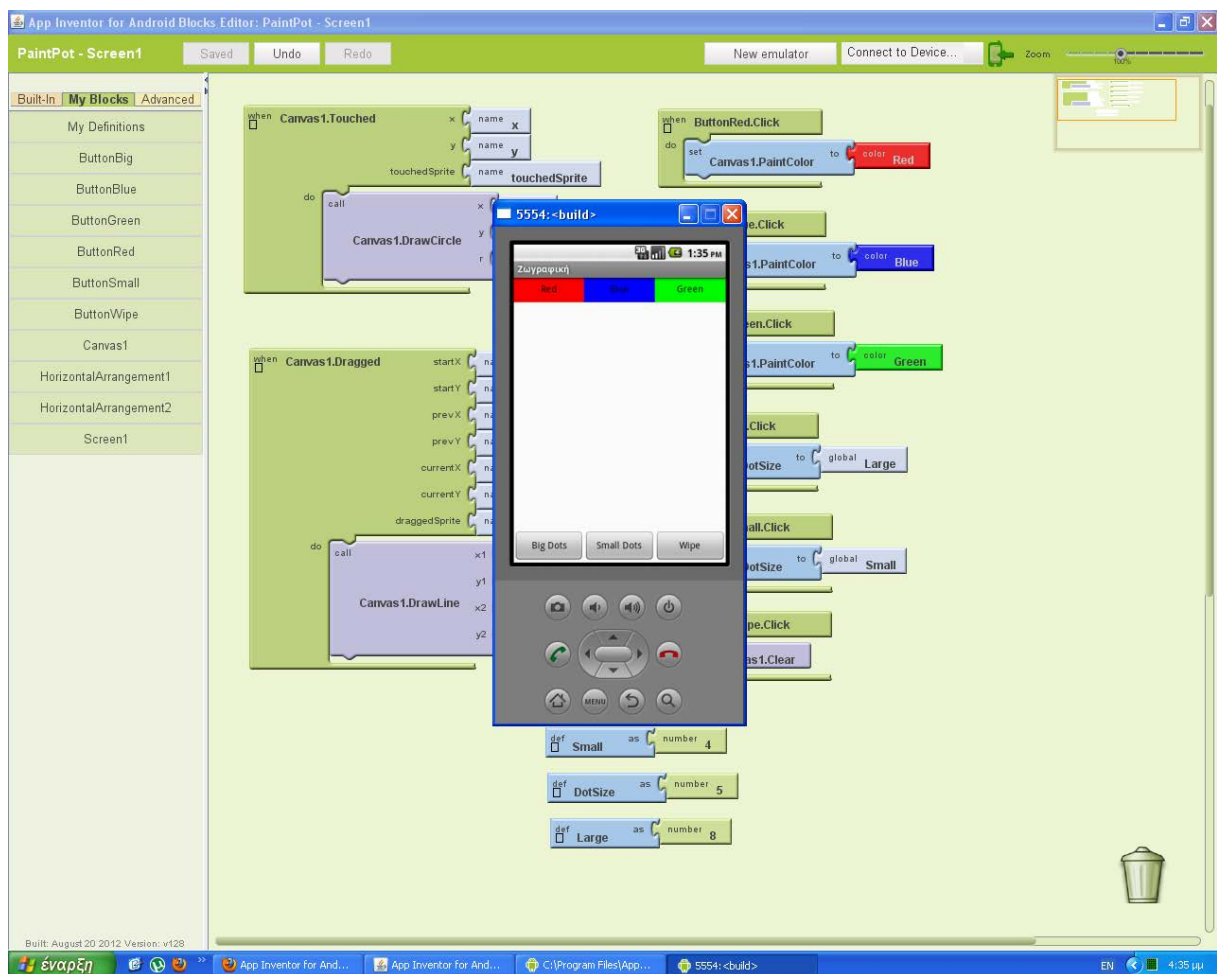
Εικόνα 59 Η εφαρμογή φορτωμένη στον Emulator.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 60 Test εφαρμογής.

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Εικόνα 61 Test εφαρμογής.

Κεφάλαιο 8

8.1 Συμπεράσματα

Στο κεφάλαιο αυτό συνοψίζονται σύντομα κάποια χρήσιμα συμπεράσματα της εφαρμογής που αναπτύχθηκε καθώς και κάποιες κατευθύνσεις για μελλοντικές επεκτάσεις της.

Πλέον το Android έχει καταφέρει να εισβάλλει για τα καλά στην παγκόσμια αγορά.

Κυριαρχεί σχεδόν σε όλες τις κατασκευάστριες εταιρείες κινητών τηλεφώνων και όχι μόνο που το επιλέγουν ως κύριο λειτουργικό σύστημα για τις φορητές συσκευές τους. Υπάρχουν χιλιάδες δωρεάν και επί πληρωμή εφαρμογές διαθέσιμες για όλους τους χρήστες του Android.

Όπως είδαμε ο σχεδιασμός μιας Android εφαρμογής και ιδιέτερα με το MIT APP Inventor είναι μια σχετικά εύκολη και ευχάριστη διαδικασία. Σ' αυτό έχουν συνεισφέρει πολύ τα έτοιμα εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google και μπορεί να τα βρει κανείς εντελώς δωρεάν στο Διαδίκτυο.

Γενικά η φιλοσοφία αυτής της πτυχιακής στηρίζεται στην ιδέα του προγραμματίζω παίζοντας το οποίο βρίσκει όλο και πιο πολλούς φανατικούς θαυμαστές τελευταία καθώς δεν χρειάζεται να έχεις κανείς ιδιαίτερες γνώσεις προγραμματισμού για να φτιάξει τη δική του εφαρμογή χρησιμοποιώντας το APP Inventor της Google.

Η εφαρμογή έχει ακόμα πολλά περιθώρια ανάπτυξης. Μερικές κατευθύνσεις για μελλοντική επέκτασή της είναι:

- Προσθήκη παλέτας χρωμάτων.
- Προσθήκη σβήστρας η οποία θα σβήνει κομμάτια του σχεδίου που θέλει ο χρήστης και όχι όλο το σχέδιο όπως είναι τώρα.
- Προσθήκη TextBox στο οποίο θα καθορίζει ο χρήστης το πάχος της γραμμής ή της κουκίδας.
- Προσθήκη κουμπιού αποθήκευσης έργου.

Βιβλιογραφία

- [1] Android Developers:
<http://developer.android.com/index.html>
- [2] Mobile Phone Development
<http://www.mobilephonedevlopment.com/>
- [3] Mobile operating system
http://en.wikipedia.org/wiki/Mobile_operating_system
- [4] Η ιστορία της κινητής τηλεφωνίας
<http://www.sansimera.gr/articles/241>
- [5] Open Handset Alliance
<http://www.openhandsetalliance.com/index.html>
- [6] Techcrunch
<http://techcrunch.com/2012/02/28/forget-about-android-4-0-google-exec-hints-android-5-0-will-launch-in-fall-of-2012/>
- [7] App Inventor
<http://appinventor.mit.edu/explore/>
<http://appinventor.mit.edu/teach/resources.html>
- [8] HyperNext Android Creator (HAC)
<http://www.hypernextandroid.com/>
<http://www.androidtools.org/native/tag/hypernext-android-creator>
- [9] Developer Android
<http://developer.android.com/sdk/index.html>
- [10] Reuse existing C code with the Android NDK
<http://www.ibm.com/developerworks/opensource/tutorials/os-androidndk/os-androidndk-pdf.pdf>
- [11] Android Magazine
<http://www.androidmag.com/>
- [12] Πληροφορίες για τις διάφορες εκδόσεις και των API της πλατφόρμας του Android
<http://developer.android.com/resources/dashboard/platform-versions.html>
- [13] Περιγραφή μεθόδων και κύκλου ζωής του Activity
<http://developer.android.com/reference/android/app/Activity.html>
<http://developer.android.com/reference/android/app/Activity.html>

Πτυχιακή Εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων

- [14] Το λειτουργικό σύστημα Android
<http://www.android.com/>
- [15] Το Logo του Android
http://en.wikipedia.org/wiki/File:Android_logo.svg
- [16] Πληροφορίες για χαρακτηριστικά και αρχιτεκτονική του Android
<http://developer.android.com/guide/basics/what-is-android.html>
- [17] NDK
<http://www.ibm.com/developerworks/opensource/tutorials/os-androidndk/os-androidndk-pdf.pdf>
- [18] Android SDK
<http://developer.android.com/sdk/index.html>
- [19] Context
<http://developer.android.com/reference/android/content/Context.html>
- [20] manifest
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [21] Professional Android™ Application Development (Book)
- [22] App Inventor for Android: Build Your Own Apps (Book)
- [23] Android™ Wireless Application Development (Book)