



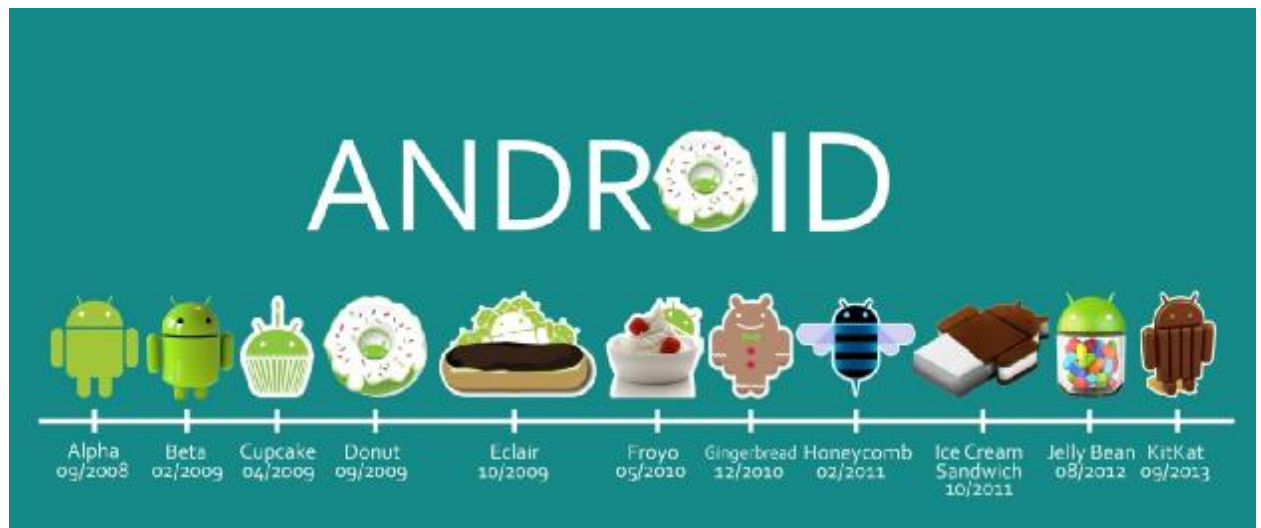
Τ.Ε.Ι ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΣΥΓΚΡΗΤΙΚΗ ΜΕΛΕΤΗ ΤΩΝ ΕΚΔΟΣΕΩΝ ΤΗΣ
ΠΛΑΤΦΟΡΜΑΣ ANDROID»



ΤΣΙΠΟΥΡΑ ΜΑΡΙΑ

ΜΕΛΕΣΑΝΑΚΗ ΜΑΡΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΣΤΑΜΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΑΤΡΑ 2014

Εκφράζουμε τις ευχαριστίες μας στον κύριο Στάμο Κωνσταντίνο, υπεύθυνο καθηγητή της πτυχιακής μας εργασίας, του οποίου η καθοδήγηση και το ενδιαφέρον έπαιξε σημαντικό ρόλο στην πορεία διεξαγωγής και ολοκλήρωσης του συγκεκριμένου έργου.

Επίσης στις οικογένειες μας οι οποίοι υπήρξαν αρωγοί και υποστηρικτές σε όλη την διάρκεια των προσωπικών μας προσπαθειών για την εκπόνηση της πτυχιακής μας εργασίας.

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία περιγράφονται όλες οι εκδόσεις του λειτουργικού συστήματος Android έως σήμερα, και αναλύονται με μια εφαρμογή, το σημειωματάριο. Το περιεχόμενο της εργασίας θα μπορούσε να χωριστεί σε 2 μέρη για καλύτερη κατανόηση, στο θεωρητικό και στο πρακτικό.

Στα πρώτα 5 κεφάλαια, στο θεωρητικό μέρος, αναφέρετε η ιστορία του Android, η δομή κατασκευής τους, αναλύονται όλες οι εκδόσεις και τέλος αποδίδουμε ενδεικτικά εφαρμογές για Android αλλά και οι χρήσεις που έχει το λειτουργικό σύστημα πέραν των κινητών τηλεφώνων.

Στο επόμενο κεφάλαια, στο πρακτικό μέρος, περιγράφονται αναλυτικά με εικόνες πώς έγινε η εγκατάσταση των εικονικών τηλεφώνων στους υπολογιστές μας, αναλύονται τα περιβάλλοντα των κινητών με την χρήση εικόνων, γίνετε ανάλυση των αρχείων και των φακέλων του Android που χρειάζονται για την δημιουργία μιας εφαρμογής και τέλος με την βοήθεια του κώδικα για σημειωματάριο αναλύσαμε τις διαφορές στην κάθε έκδοση.

Στο παράρτημα της εργασίας βρίσκετε ολόκληρος ο κώδικας java που χρησιμοποιήθηκε για την εφαρμογή σε όλες τις εκδόσεις.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	3
Κεφάλαιο 1 Ιστορία του Android	8
1.1 Χρονολογικά δεδομένα	8
Open Handset Alliance	8
1.2 Τι ακριβώς είναι το Android	9
1.3 Επίπεδα API	10
1.4 Android SDK	12
1.4.1 Γλώσσα Προγραμματισμού	12
Κεφάλαιο 2 Δομή των Android.....	13
2.1 Δομή των Android.....	13
2.2 Πυρήνας Λειτουργικού Linux	14
2.3 Android Runtime.....	15
2.3.1 Βιβλιοθήκες	15
2.3.1.1 Βιβλιοθήκη Συστήματος C	15
2.3.1.2 LibWebCore- Βιβλιοθήκη WEBKIT	16
2.3.1.3 Media Libraries	16
2.3.1.4 Surface Manager.....	16
2.3.1.5 SQLite.....	17
2.3.1.6 Βιβλιοθήκες τριών διαστάσεων (3D).....	17
2.3.1.7 Scalable Graphics Library (SGL)	17
2.3.2 Εικονική Μηχανή Dalvik (Dalvik Virtual Machine).....	18
2.4 Δομή εφαρμογής.....	19

2.5 Εφαρμογή	20
Κεφάλαιο 3 - Εκδόσεις Android.....	20
3.1 Android Beta	20
3.2 Android 1.0	20
3.3 Android 1.1	21
3.4 Android 1.5 (Cupcake)	23
3.5 Android 1.6 (Donut).....	24
3.6 Android 2.0 και 2.1 (Éclair)	26
3.7 Android 2.2 Froyo	29
2.8 Android 2.3 (Gingerbread).....	31
3.9 Android 3.0 (Honeycomb)	34
3.10 Android 4.0 (Ice cream sandwich)	35
Android Beam	37
3.11 Android 4.1 (Jelly Bean)	38
3.12 Android 4.2-4.3 (Jelly Bean).....	40
3.13 Android 4.4 (Kit Kat)	42
Κεφάλαιο 4 Έξυπνες Συσκευές	43
4.1 Smartwatch.....	44
4.2 Αυτοκίνητα	45
4.3 Ψηφιακές φωτογραφικές μηχανές	46
4.4 Κονσόλες παιχνιδιών	47
4.5 Smart glasses	48
4.6 Οικιακές συσκευές	49

4.7 Smart TVs.....	50
4.8 Καθρέπτες.....	51
Κεφάλαιο 5 Εφαρμογές Android	52
5.1 Σταυρόλεξα.....	53
5.2 Ελληνικό Εορτολόγιο/Εορτοlogio.....	54
5.3 Meteo.gr	55
5.4 Viber.....	56
5.5 Smart Tools.....	57
Κεφάλαιο 6 Περιβάλλον ανάπτυξης εφαρμογών	58
6.1 Εγκατάσταση των αρχείων από το Internet.....	58
6.2 Εγκατάσταση του εργαλείου Java.....	62
6.3 Εγκατάσταση του εργαλείου ANDROID SDK.....	64
6.4 Εγκατάσταση του εργαλείου ECLIPSE	70
Κεφάλαιο 7 : Περιβάλλον κάθε έκδοσης Android.....	76
7.1 Περιβάλλον Android 1.5.....	76
7.2 Περιβάλλον Android 1.6.....	78
7.3 Περιβάλλον Android 2.1.....	80
7.4 Περιβάλλον Android 2.2.....	81
7.5 Περιβάλλον Android 2.3.3.....	83
7.6 Περιβάλλον Android 3.0, 3.1, 3.2.....	84
7.7 Περιβάλλον Android 4.0, 4.0.3.....	85
7.9 Περιβάλλον Android 4.2, 4.3.....	89
7.10 Περιβάλλον Android 4.4.....	91

Κεφάλαιο 8 Ανάλυση στοιχείων Android.....	92
Κεφάλαιο 9: Σύγκριση μιας εφαρμογής στις διάφορες εκδόσεις Android.....	96
9.1 Android 1.5 (Cupcake).....	96
9.2 Android 1.6 (Donut).....	105
9.3 Android 2.1 (Éclair)	108
9.4 Android 2.2 (Froyo).....	118
9.5 Android 2.3.3 (Gingerbread).....	118
9.6 Android 3.0, 3.1, 3.2 (Honeycomb).....	126
9.7 Android 4.0, 4.0.3 (Ice cream sandwich)	134
9.8 Android 4.1 (Jelly Bean), Android 4.2, Android 4.3, Android 4.4	137
ΣΥΜΠΕΡΑΣΜΑ.....	139
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	141
ΠΑΡΑΡΤΗΜΑ.....	143

Κεφάλαιο 1 Ιστορία του Android

1.1 Χρονολογικά δεδομένα

Τον Οκτώβριο του 1993, στο Palo Alto της California ιδρύθηκε μία μικρή εταιρία η Android Inc..Ιδρυτές της ήταν ο Andy Rubin, ο Rich Miner και ο Chris White και βασικός στόχος της ήταν η δημιουργία λειτουργικών συστημάτων για κινητά τηλέφωνα.

Τον Αύγουστο του 2005, και αφού είχε γίνει μία παρουσίαση του Android από τον Andy Rubin αναφέροντας και τονίζοντας τις σταθερές συνήθειες του αγοραστικού κοινού, η Google αγόρασε εξ' ολοκλήρου την Android Inc. κρατώντας τους ιδρυτές της ως βασικά στελέχη της εταιρίας

Με την κίνηση αυτή η Google θέλησε να εισέλθει στον χώρο της αγοράς της κινητής τηλεφωνίας.

Open Handset Alliance

Το φθινόπωρο του 2005 η Intel και αρκετές ακόμη εταιρίες ένωσαν το δυναμικό τους με την Google για να δημιουργήσουν μία πλατφόρμα ανοιχτού κώδικα, με ενσωματωμένο λογισμικό Linux για κινητά τηλέφωνα. Αργότερα η Motorola, η LG, η HTC και άλλες, ανακοίνωσαν πως έχουν πρόθεση να δώσουν προς πώληση smartphones με λειτουργικό σύστημα Android με τα οποία ο καταναλωτής θα μπορεί να χρησιμοποιεί πολλές εφαρμογές. Με αυτόν τον τρόπο δημιουργήθηκε μία μεγάλη συμμαχία που αποτελείται από πολλές εταιρίες του τηλεπικοινωνιακού κλάδου, την Open Handset Alliance. Αυτή είχε ως στόχο την προώθηση του Android και των συμφερόντων τους.

Αργότερα τον Νοέμβριο του 2007, τέθηκε σε κυκλοφορία το Android beta SDK. Αυτό έδινε μία πλατφόρμα για να φτιαχτούν νέες εφαρμογές και παιχνίδια. Έτσι ο καθένας αγοραστής με κατάλληλη γνώση μπορούσε να δημιουργήσει ότι εφαρμογή ήθελε.

Στις 23 Σεπτεμβρίου 2008 , η συσκευή κινητού τηλεφώνου HTC Dream G1 λάνσαρε το νέο λειτουργικό Android 1.0 και έπειτα ανακοινώθηκε η πρώτη αναβάθμιση σε Android 1.1 που αναφερόταν μόνο σε συσκευές T-Mobile G1.

Τον Δεκέμβριο του 2008 ενσωματώθηκαν στην Open Handset Alliance δεκατέσσερα νέα μέλη όπως η Sony-Ericsson και η Vodafone.

1.2 Τι ακριβώς είναι το Android

Το Android είναι μία ενότητα λογισμικού ανοιχτού κώδικα, ολοκληρωμένη και ελεύθερη, για συσκευές κινητού τηλεφώνου, και διαθέτει ένα λειτουργικό σύστημα(Operating System)¹, τις κύριες εφαρμογές μαζί με ένα σύνολο από βιβλιοθήκες API για την κατασκευή εφαρμογών.

Το λογισμικό αυτό δίνει την δυνατότητα στους κατασκευαστές να γράψουν κώδικες στην java, ελέγχοντας έτσι το τηλέφωνο μέσω των βιβλιοθηκών της java που είναι κατασκευασμένες από την Google.

¹ Λειτουργικό σύστημα (Operating System-OS) είναι το λογισμικό μιας ηλεκτρονικής συσκευής που είναι υπεύθυνο για την διαχείριση και την ρύθμιση τις συχνότητας των εργασιών, καθώς επίσης και για την διανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα έχει πρωταρχική σημασία καθώς είναι ο συνδεδετικός κρίκος μεταξύ λογισμικού και υλικού.

1.3 Επίπεδα API

Τα επίπεδα API είναι μία ακέραια τιμή που προσδιορίζει μοναδικά την ταυτότητα της δομής μίας βελτιωμένης πλατφόρμας του Android. Με απλά λόγια είναι ένα χαρακτηριστικό που επιτρέπει στο σύστημα να προσδιορίσει σωστά εάν μία εφαρμογή είναι συμβατή με το σύστημα, πριν από την αίτηση για την εγκατάστασή του.

Η πλατφόρμα Android εξασφαλίζει την δομή του API που οι εφαρμογές μπορούν να χρησιμοποιούν για να αλληλεπιδρούν με την βάση, δηλαδή το σύστημα Android. Η δομή αυτή αποτελείται από:

- Ένα σετ από στοιχεία και χαρακτηριστικά γνωρίσματα XML² για την δήλωση ενός αρχείου.
- Ένα σετ από στοιχεία και χαρακτηριστικά γνωρίσματα XML για την δήλωση και την πρόσβαση πηγών-πόρων
- Ένα σετ από άδειες που μπορεί να ζητήσουν οι εφαρμογές, όπως για παράδειγμα άδεια για την εκτέλεση ενός αρχείου του συστήματος
- Ένα σετ από πακέτα και κλάσεις

Κάθε επιτυχημένη έκδοση από τις πλατφόρμες Android μπορεί να συμπεριλάβει ενημερώσεις στις εφαρμογές της δομής του API.

Οι ενημερώσεις στην δομή του API είναι σχεδιασμένες έτσι ώστε το νέο API να παραμένει συμβατό με τις παλιότερες εκδόσεις. Αυτό σημαίνει πως οι αλλαγές που γίνονται είναι αθροιστικές, νέες ή διορθώνουν με ενημερώσεις παλιότερες.

Ο παρακάτω πίνακας μας δείχνει τα επίπεδα του API που υποστηρίζει κάθε έκδοση από τις πλατφόρμες Android.

² Extensible Markup Language (xml) είναι μία γλώσσα σήμανσης που ορίζει μία σειρά κανόνων για την κωδικοποίηση εγγράφων που μπορεί να διαβαστεί από τον άνθρωπο και από έναν ηλεκτρονικό υπολογιστή.

Πλατφόρμα	Επίπεδο API	Κωδική ονομασία
Android 4.4	19	Kit Kat
Android 4.3	18	Jelly Bean
Android 4.2	17	Jelly Bean
Android 4.1	16	Jelly Bean
Android 4.0.3	15	Ice cream sandwich MR1
Android 4.0.2	14	Ice cream sandwich
4.0.1		
4.0		
Android 3.2	13	Honeycomb MR2
Android 3.1.x	12	Honeycomb MR1
Android 3.0.x	11	Honeycomb
Android 2.3.4	10	Gingerbread MR1
2.3.3		
Android 2.3.2	9	Gingerbread
2.3.1		
2.3		
Android 2.2.x	8	Froyo
Android 2.1.x	7	Éclair MR1
Android 2.0.1	6	Éclair 0.1
Android 2.0	5	Éclair
Android 1.6	4	Donut
Android 1.5	3	Cupcake
Android 1.1	2	Petit Four
Android 1.0	1	Base

1.4 Android SDK

Η ανάπτυξη λογισμικού Android (Android software development) είναι μία διεργασία με την οποία δημιουργούνται νέες εφαρμογές για το λειτουργικό σύστημα του Android. Οι εφαρμογές αναπτύσσονται κυρίως στην γλώσσα προγραμματισμού Java χρησιμοποιώντας το Android SDK και άλλα εργαλεία ανάπτυξης που είναι διαθέσιμα.

Το Android SDK (System Development kit) είναι ένα υποχρεωτικό εργαλείο για τους προγραμματιστές του Android. Περιλαμβάνει ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, κώδικες, έγγραφα, προγράμματα εκμάθησης καθώς και ένα εξομοιωτή. Στις μέρες μας οι υποστηριζόμενες πλατφόρμες ανάπτυξης περιλαμβάνουν και υπολογιστές με λειτουργικό σύστημα Linux, Mac OS x 10.5.8 ή Windows XP καθώς και τις νεότερες εκδόσεις τους.

1.4.1 Γλώσσα Προγραμματισμού

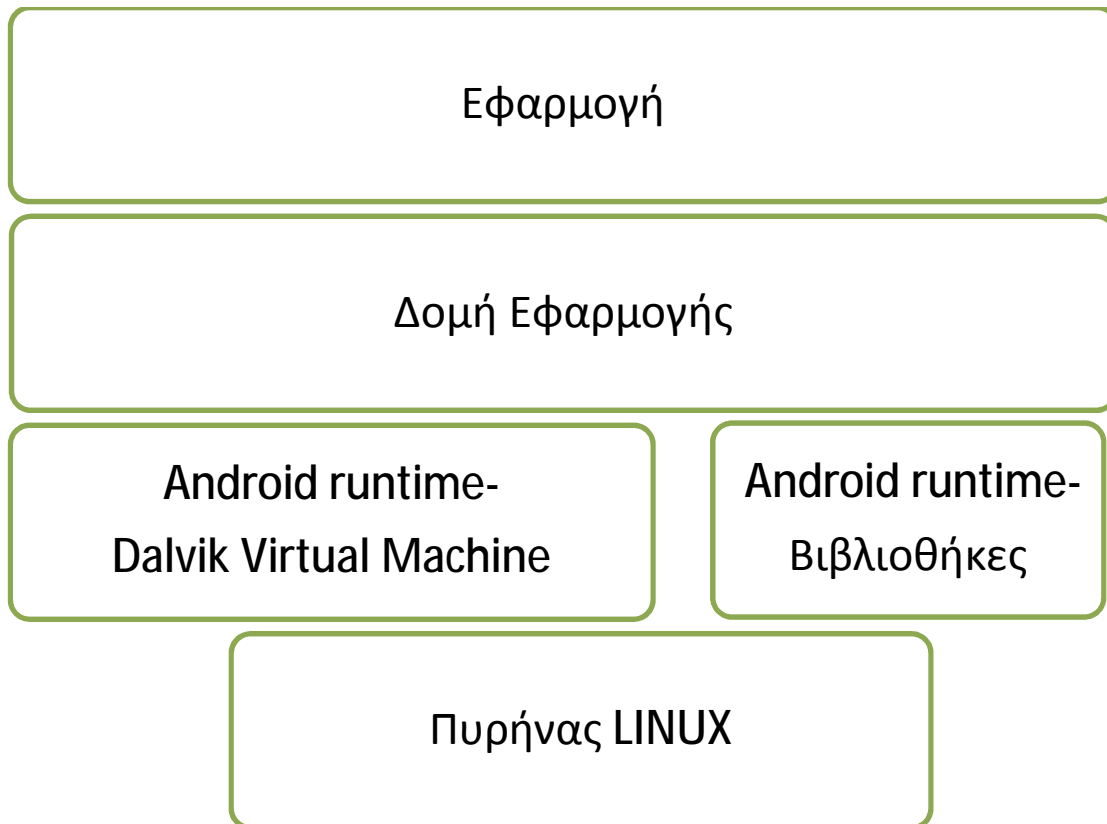
Το SDK χρησιμοποιεί γλώσσα προγραμματισμού Java που έχει ονομασία Java Android Library.

Η Java παίρνει ένα μέρος της σύνταξης της από τις γλώσσες C και C++ αλλά έχει τις λιγότερες χαμηλού επιπέδου εγκαταστάσεις (low-level programming language). Αυτό σημαίνει ότι οι εφαρμογές τις Java μπορούν να τρέξουν σε οποιαδήποτε εικονική μηχανή Java (JVM), ανεξάρτητα από την αρχιτεκτονική της συσκευής. Ουσιαστικά ο κώδικας της που τρέχει σε μια πλατφόρμα δε χρειάζεται να αναθεωρηθεί για να τρέξει σε μία άλλη.

Αυτό είναι ένα μεγάλο πλεονέκτημα για τους προγραμματιστές, καθώς όλοι είναι εξοικειωμένοι με την χρήση των γλωσσών προγραμματισμού που ανήκουν στην οικογένεια C.

Κεφάλαιο 2 Δομή των Android

2.1 Δομή των Android



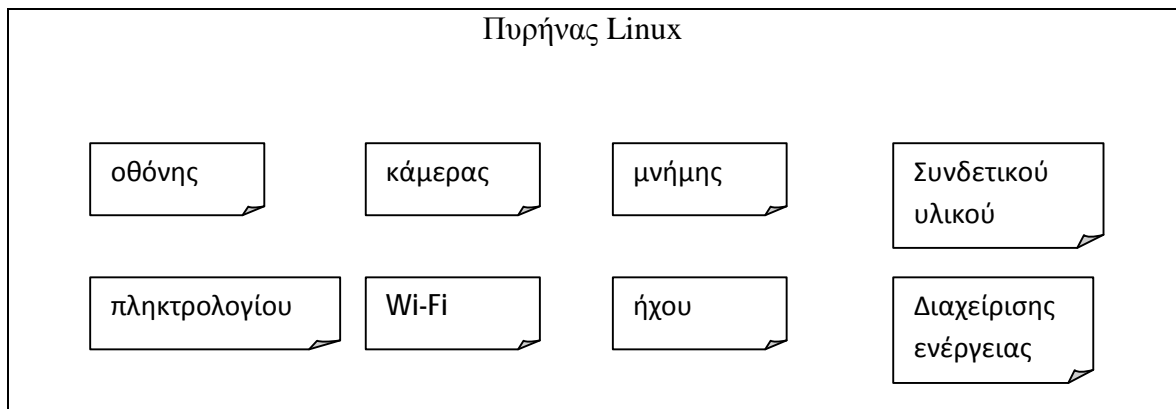
Οι Android εφαρμογές είναι χωρισμένες σε τέσσερα διαφορετικά επίπεδα και σε πέντε ομάδες. Αυτές είναι: πυρήνας Linux, βιβλιοθήκες και Android runtime, εικονική μηχανή Dalvik και Android runtime, δομή εφαρμογής και τέλος την εφαρμογή του Android.

Παρακάτω θα αναλύσουμε το κάθε επίπεδο και ομάδα ξεχωριστά.

2.2 Πυρήνας Λειτουργικού Linux³

Στο κατώτερο επίπεδο βρίσκετε ο πυρήνας του λειτουργικού Linux στον οποίο είναι βασισμένα τα Android. Ο πυρήνας του Linux θεωρείται σήμερα από τα πιο ευρέως διαδεδομένα ελεύθερα λογισμικά.

Ο πυρήνας Linux περιλαμβάνει προγράμματα οδήγησης για το hardware (υλικό εξοπλισμό), για την δικτύωση καθώς και για την πρόσβαση στο αρχεία του συστήματος. Διαγραμματικά ο πυρήνας περιλαμβάνει τα εξής προγράμματα οδήγησης (drivers) :



³ Linux είναι ένας κοινός όρος που μεταχειρίζεται λειτουργικά συστήματα που είναι βασισμένα στον πυρήνα του Linux. Η δομή του Linux είναι βασισμένη στο Unix κάποιο άλλο λειτουργικό σύστημα. Το Linux χρησιμοποιείται ευρέως, από κινητά τηλέφωνα έως σε μεγάλους ηλεκτρονικούς υπολογιστές.

2.3 Android Runtime

Το Android Runtime περιέχει ένα σύνολο από βασικές βιβλιοθήκες που χρησιμοποιούν οι προγραμματιστές για να γράψουν δικιές τους εφαρμογές, χρησιμοποιώντας γλώσσα Java.

2.3.1 Βιβλιοθήκες

Οι βιβλιοθήκες Android C/C++ χρησιμοποιούνται για τις υπηρεσίες του λειτουργικού συστήματος. Οι βιβλιοθήκες αυτές έχουν δυνατότητες που τις εκμεταλλεύονται οι προγραμματιστές με το πλαίσιο εφαρμογών του Android. (Android application framework)

Διαθέτουν όλο τον κώδικα που έχει το Android Operating System. Το Android OS είναι ένα λογισμικό που κατασκευάζεται ειδικά για κινητά τηλέφωνα και υποστηρίζεται από το λογισμικό Linux. Ο κώδικας αυτού του λογισμικού είναι ανοιχτός και μπορούν να γίνουν αλλαγές με την σύμφωνη γνώμη των προγραμματιστών του Android.

Οι βιβλιοθήκες που διαθέτουν όλες οι εφαρμογές είναι αρκετές και η κάθε μία έχει την δική της χρήση. Παρακάτω αναλύονται μερικές βασικές βιβλιοθήκες που περιλαμβάνονται στο λειτουργικό σύστημα του Android.

2.3.1.1 Βιβλιοθήκη Συστήματος C

Ονομάζεται και bionic lib. Η bionic LIBC είναι ένα παρακλάδι του κώδικα BDS⁴ και έχει αναπτυχθεί ειδικά για τα συστήματα των Android από την Google. Η βιβλιοθήκη αυτή έχει πολλά κοινά χαρακτηριστικά με τον κώδικα του Linux όμως η ανάπτυξη και η βελτίωση της γίνεται ανεξάρτητα από τις βάσεις του κώδικα. Οι καθορισμένοι στόχοι για την βιβλιοθήκη bionic είναι: Η άδεια BSD: η Google, επειδή το Android

⁴ Ο κώδικας BDS είναι ένας μεταγλωττιστής για τις υποκατηγορίες της γλώσσας προγραμματισμού C.

χρησιμοποιεί τον πυρήνα του Linux που υπάγεται στην GPL⁵, θέλησε να αποκλείσει κάθε επαφή των εφαρμογών των Android από τις επιδράσεις του GPL. Μικρό μέγεθος: η bionic είναι μικρότερη από την βιβλιοθήκη glibc και από την uclibc. Ταχύτητα : η bionic έχει σχεδιαστεί για τις κεντρικές μονάδες επεξεργασίας σε χαμηλές συχνότητες όπως εκείνες ενός ρολογιού.

2.3.1.2 LibWebCore- Βιβλιοθήκη WEBKIT

Η βιβλιοθήκη LibWebCore είναι μία μηχανή δρομολογητών που εξουσιάζει τα προγράμματα περιήγησης του Android καθώς και το ιστορικό των περιηγήσεων. Βασίζεται κυρίως στην WEBKIT. Η βιβλιοθήκη αυτή είναι μία διάταξη μίας μηχανής σχεδιασμού για να επιτρέπει στους δρομολογητές (browsers) να εμφανίζει ιστοσελίδες. Η WebKit ασκεί εξουσία σε προγράμματα περιήγησης όπως το Google Chrome και το Apple Safari. Χρησιμοποιείται ακόμα σαν βάση για τον πειραματικό δρομολογητή που συμπεριλαμβάνεται στο Amazon Kindle ebook reader, καθώς και ως προεπιλεγμένο πρόγραμμα περιήγησης στο iOS, Android και στις ταμπλέτες OS των BlackBerry.

2.3.1.3 Media Libraries

Αυτό το Open core packet video βασίζεται σε βιβλιοθήκες που παρέχουν δυνατότητες για πολυμέσα. Τέτοια είναι οι εγγραφή και αναπαραγωγή βίντεο και ήχου (MPEG4,H.264,MP3) καθώς και αρχεία εικόνας (JPG,PNG).Ο πυρήνας (core) υποστηρίζει υπηρεσίες πολυμέσων (LIVE TV, Internet Radio).

2.3.1.4 Surface Manager

Αυτή η βιβλιοθήκη παρέχει στον χρήστη πρόσβαση στο υποσύστημα της οθόνης.

⁵ (General Public License) Άδεια χρήσης του ελεύθερου λογισμικού

2.3.1.5 SQLite

Η βιβλιοθήκη αυτή είναι κατά κύριο λόγο αυτόνομη και δεν απαιτεί υποστήριξη από το λογισμικό ή από άλλες βιβλιοθήκες, έτσι γίνεται ιδανική για φορητές συσκευές που δεν έχουν υποδομή υποστήριξης όπως ένας επιτραπέζιος ηλεκτρονικός υπολογιστής. Στην SQLite μία διαδικασία που θέλει να έχει πρόσβαση στην βάση των δεδομένων, διαβάζει και γράφει αμέσως από τα αρχεία της βάσης που υπάρχουν στο δίσκο. Δηλαδή δεν υπάρχει κάποιος διαμεσολαβητής.

2.3.1.6 Βιβλιοθήκες τριών διαστάσεων (3D)

Αυτές οι βιβλιοθήκες των Android βασίζονται σε OpenGL ES⁶ 1,0 API. Οι βιβλιοθήκες αυτές μπορούν να χρησιμοποιούν είτε το επίπεδο επιτάχυνσης υλικού 3D είτε να χρησιμοποιούν το λογισμικό rasterizer⁷.

2.3.1.7 Scalable Graphics Library (SGL)

Είναι ένα υποσύστημα γραφικών που χρησιμοποιούν τα λειτουργικά συστήματα Android. Η SGL είναι χαμηλού επιπέδου βιβλιοθήκη γραφικών και ουσιαστικά είναι η βασική βιβλιοθήκη για τα γραφικά δυο διαστάσεων-μήκος, πλάτος.(2D)

Λειτουργεί σε συνδυασμό με άλλες βιβλιοθήκες όπως η Surface Manager καθώς και με το Window Manager⁸ για την γενική υλοποίηση των γραφικών στα Android.

⁶ Open Graphics Language είναι μία βασική προδιαγραφή που καθορίζεται από διάφορες γλώσσες προγ/μου για να δημιουργεί εφαρμογές που αναπαράγουν 2D/3D γραφικά υπολογιστών

⁷ Είναι ένα λογισμικό που μετατρέπει της εικόνες- σχήματα σε εικόνες με pixel με σκοπό την έξοδο από ένα βίντεο ή την αποθήκευση της σε αρχείο bitmap

⁸ Είναι ένα λειτουργικό σύστημα που ελέγχει την εμφάνιση και την τοποθέτηση παράθυρων σε ένα γραφικό περιβάλλον.

2.3.2 Εικονική Μηχανή Dalvik (Dalvik Virtual Machine)

Η Dalvik Virtual Machine (VM) είναι ένα πρωτόκολλο που βασίζεται σε μία εικονική μηχανή και είναι σχεδιασμένη από τον Bornstein και από προγραμματιστές της Google για το πλαίσιο των Android.

Η εικονική αυτή μηχανή είναι βελτιστοποιημένη ώστε να έχει την μικρότερη δυνατή απαίτηση σε μνήμη και να επιτρέπει να τρέχουν πολλές VM ταυτόχρονα. Είναι δηλαδή μία εφαρμογή της Google, με βιβλιοθήκες της Java και σχεδιάστηκε για κινητές συσκευές, που έχουν μικρή μνήμη και ισχύ. Όλος ο κώδικας που γράφετε για τα Android είναι γραμμένος σε γλώσσα Java και τρέχει εντός της μηχανής Dalvik. Η εικονική αυτή μηχανή επιτρέπει σε κάθε εφαρμογή να τρέξει την δική της εργασία μαζί με μία δική της ξεχωριστή Dalvik VM.

Η Dalvik VM συχνά γράφετε και αναφέρεται ως Java VM, όμως αυτό δεν ισχύει. Στην ουσία υπάρχει ένα εργαλείο με την ονομασία DX το οποίο υπάρχει στο Android SDK και κάνει το διαχωρισμό.

Η διαφορές που έχει με την κανονική- παραδοσιακή Java είναι:

- ο Ο πυρήνας των βιβλιοθηκών της Java που είναι μαζί με το σύστημα Android είναι διαφορετικός και από τις 2 άλλες εκδόσεις τις Java, την βιβλιοθήκη Java Standard Edition και την βιβλιοθήκη Java Mobile Edition.
- ο Επίσης η εικονική μηχανή τρέχει αρχεία τύπου .dex, τα οποία είναι αρχεία τύπου .class ή .jar και κατά την μεταγλώττιση έγιναν τύπου .dex που είναι πιο συμπαγή και πιο αποτελεσματικά σε σχέση με την παλιά τους μορφή, όσον αφορά την περιορισμένη μνήμη και την κατανάλωση μπαταρίας στις συσκευές.

2.4 Δομή εφαρμογής

Στο πλαίσιο εφαρμογής υπάρχουν όλα τα απαραίτητα δομικά στοιχεία που χρειάζονται για μία εφαρμογή. Αυτό σημαίνει πως οι προγραμματιστές μπορούν εύκολα να γράψουν τις δικές τους εφαρμογές για τις συσκευές τους.

Η δομή αυτή περιέχει εργαλεία για τον σχεδιασμό του περιβάλλοντος της επιφάνειας εργασίας του χρήστη όπως κουμπιά, πεδία για κείμενο και εργαλεία του συστήματος. Η δομή αυτή συνήθως είναι προεγκατεστημένη σε κάθε συσκευή Android όμως ο προγραμματιστής-χρήστης μπορεί να το επεκτείνει προσθέτοντας δικά του πλαίσια.

Στην ουσία, μία εφαρμογή Android αποτελείται από δραστηριότητες (activities) με τις οποίες ο χρήστης αλληλεπιδρά, από υπηρεσίες (services) προγράμματα δηλαδή που τρέχουν στο παρασκήνιο ή απλά υποβοηθούν κάποιες άλλες εφαρμογές, και τέλος από δέκτες μετάδοσης (broadcast receivers) προγράμματα που βρίσκουν πληροφορίες σχετικά με τις εφαρμογές.

Υπάρχουν κάποια βασικά στοιχεία σε κάθε πλαίσιο, που είναι τα εξής:

- Activity Manager (διαχειριστής δραστηριότητας): αυτός ελέγχει τον κύκλο ζωής των εφαρμογών. Δηλαδή όταν ο χρήστης εκτελεί μία εφαρμογή το σύστημα την φέρνει και την εκτελεί στο προσκήνιο. Στην συνέχεια ο χρήστης ίσως χρειαστεί μία δεύτερη εφαρμογή ή ακόμα και μία τρίτη. Όλες αυτές οι εφαρμογές που είναι ανοιχτές στο προσκήνιο αποθηκεύονται σε μία στοίβα της εφαρμογής από τον Διαχειριστή Δραστηριότητας. Έτσι ανά πάσα στιγμή ο χρήστης μπορεί να πατήσει το κουμπί 'πίσω' και να επιστρέψει στην προηγούμενη οθόνη, επίσης η στοίβα αυτή χρησιμεύει και ως ιστορικό όταν ο χρήστης περιηγείται στο διαδίκτυο.
- Notification Manager (διαχειριστής ειδοποιήσεων): παρουσιάζει τα γεγονότα στον χρήστη όπως μηνύματα, συναντήσεις, κ.τ.λ.
- Location Manager (διαχειριστής περιοχής): μία συσκευή Android ξέρει πάντα που βρίσκετε!

- ο Content Providers (υπηρεσίες παροχής περιεχομένου): τα αντικείμενα αυτά ενσωματώνουν στοιχεία που πρέπει να μοιράζονται μεταξύ των εφαρμογών για παράδειγμα οι επαφές

2.5 Εφαρμογή

Το υψηλότερο επίπεδο της δομής των Android είναι η εφαρμογή. Μόνο οι τελικοί χρήστες βλέπουν αυτό το επίπεδο, αγνοώντας τα προηγούμενα επίπεδα.

Εδώ υπάρχουν όλες οι εφαρμογές που είναι εγκατεστημένες στις συσκευές όπως οι επαφές, η μουσική και τα μηνύματα. Επίσης εδώ αποθηκεύονται και οι εφαρμογές που ο χρήστης εγκαθιστά από μόνος του, όπως παιχνίδια.

Κεφάλαιο 3 - Εκδόσεις Android

3.1 Android Beta

Το Android Beta ανακοινώθηκε και τέθηκε σε κυκλοφορία στις 7 Νοεμβρίου του 2007, ενώ παράλληλα το SDK (software developer's kit), η εργαλειοθήκη δηλαδή για τους προγραμματιστές, ανακοινώθηκε στις 12 Νοεμβρίου του ίδιου έτους.

3.2 Android 1.0

Η πρώτη συσκευή κινητού τηλεφώνου που έτρεξε με Android 1.1 ήταν η G1 Dream της εταιρίας HTC. Είχε επίπεδο API 1. Διέθετε οθόνη αφής 3,2'' και είχε πρόσβαση σε Gmail, YouTube, ημερολόγιο(calendar) Google, Google maps. Επίσης είχε κάρτα μνήμης Micro SD καθώς και κάμερα 3,2 Megapixels.

Αναλυτικότερα τα χαρακτηριστικά του είναι:

- Δρομολογητές (web browser) για να ανοίγουν, να κάνουν μεγέθυνση και να κάνουν λήψη σε ιστοσελίδες HTML και XHTML. Επίσης δύνεται η δυνατότητα να ανοίγονται πολλές σελίδες σαν καρτέλες
- Η Google μπορεί να συγχρονίζετε με διάφορες εφαρμογές.
- Υποστηρίζει το «κατέβασμα» και αναβαθμίσεις μέσω του Android Market
- Διαθέτει Bluetooth, GPS, και σύνδεση με Wi-Fi
- Στο κεντρικό μενού διαθέτει ξυπνητήρι- υπενθύμιση με ήχο, δόνηση και φωτισμό LED
- Φωνητική κλήση χωρίς να χρειάζεται να πληκτρολογήσεις αριθμό ή όνομα
- Google maps με γεωγραφικά μήκη και πλάτη, και οδηγίες για οδήγηση μέσω συστήματος GPS.
- Άλλες εφαρμογές που διαθέτει είναι : ξυπνητήρι, κομπιουτεράκι, τηλεφωνικός κατάλογος, επιφάνεια εργασίας, εικόνες από τον φάκελο Έκθεση καθώς και ρυθμίσεις.

3.3 Android 1.1

Το Android 1.1 παρέδωσε μία νέα βελτιωμένη έκδοση της οργάνωσης του επιπέδου API. Η κωδική του ονομασία είναι Petit Four, αλλά δεν χρησιμοποιήθηκε επίσημα. Το επίπεδο API του Android 1.1 είναι 2. Αυτή η αναβάθμιση έγινε αποκλειστικά και μόνο για την συσκευή G1 T-Mobile. Με την ανανέωση αυτή λύθηκαν κάποια προβλήματα με το μικρόφωνο του τηλεφώνου, και επιπλέον πρόσθεσαν κάποιες εφαρμογές .

Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

- Χάρτες: όταν ο χρήστης κάνει κλικ σε μία επιχείρηση προσθέτει σχόλια και λεπτομέρειες
- Κατά την κλήση προσθέτει ένα κουμπί για εμφάνιση ή απόκρυψη του πληκτρολογίου της οθόνης
- Προσφέρει υποστήριξη για αποθήκευση από συνημμένα MMS

- Όταν ο χρήστης μιλάει από το μεγάφωνο κατά την κλήση, υπάρχει χρονικό όριο φωτισμού οθόνης.

Η εφαρμογές της έκδοσης είναι η εξής:

- Χάρτες και StreetView (Όταν ο χρήστης ψάχνει για επιχειρήσεις στο χάρτη (Google Maps) να είναι διαθέσιμες κάποιες λεπτομέρειες, καθώς και να έχει μνήμη για τις τελευταίες επιλογές του χρήστη.)
- Μεγαλύτερος χρόνος αναμονής στις κλήσεις
- Ξυπνητήρι
- Αριθμομηχανή
- Browser
- Μηνύματα
- E-mail
- Μουσική
- Εικόνες
- Ρυθμίσεις
- Επικοινωνία
- Κάμερα
- Εργαλεία προγραμματιστών
- Τηλεφωνητής

3.4 Android 1.5 (Cupcake)



Η πλατφόρμα του Android 1.5 εισάγει πολλές νέες δυνατότητες για τους χρήστες αλλά και για τους προγραμματιστές. Έχει επίπεδο API 3.

Αναλυτικότερα οι νέες δυνατότητες είναι:

Σύστημα

- Νέος πυρήνας Linux έκδοσης 2.6.27
- Εφαρμογή για την κάρτα SIM με εργαλειοθήκη (toolkit) έκδοσης 1,0
- Κάρτα αρχείου SD με αυτό-έλεγχο και επιδιόρθωση

Οθόνη

- Το πληκτρολόγιο είναι αφής δηλαδή βρίσκετε πάνω στην οθόνη
- Διαθεσιμότητα δύο προσανατολισμών- οριζόντιου και κάθετου
- Λεξιικό για πιο εύκολη συγγραφή μηνυμάτων
- Υπάρχουν διάφορα γραφικά στοιχεία στην οθόνη όπως αναλογικό ρολόι και αναζήτηση

Κάμερα

- Πρόγραμμα εναλλαγής λειτουργιών μεταξύ κάμερας- βιντεοκάμερας
- Λήψη βίντεο και αναπαραγωγή του σε μορφή MPEG-4
- Ταχύτερη κάμερα όσον αφορά το άνοιγμα της και την λήψη φωτογραφιών

Επαφές

- Παρουσιάζει την εικόνα χρήστη για τα αγαπημένα
- Παρέχει αρχείο κλήσεων με ώρα και ημερομηνία

Πρόγραμμα περιήγησης-εφαρμογές

- Δυνατότητα αντιγραφής- επικόλλησης στην αναζήτηση
- Παγκόσμιο πλαίσιο αναζήτησης
- Άμεση φόρτωση των βίντεο μας στο YouTube και στο Picasa
- Δυνατότητα ενεργειών στο Gmail όπως αρχειοθέτηση επαφών ή μηνυμάτων, διαγραφή κτλ.
- Ανανεωμένο Google Play με περιήγηση ανά κατηγορίες και φίλτρα αναζήτησης.
- Ταχύτερη απόκτηση θέσης στο GPS βάσει συστήματος SUPL AGPS⁹
- Πιο γρήγορη συνομιλία στο Gmail
- Πιο εύκολη κύλιση της σελίδας στο πρόγραμμα περιήγησης

3.5 Android 1.6 (Donut)

Η νέα αυτή πλατφόρμα εισάγει νέα χαρακτηριστικά για τους χρήστες αλλά και για τους προγραμματιστές. Έχει επίπεδο API 4. Τα νέα χαρακτηριστικά είναι:



Σύστημα

- Νέος πυρήνας Linux έκδοσης 2.6.29
- Υποστηρίζονται οθόνες αναλύσεων WVGA¹⁰

⁹ Σύστημα που μπορεί να βελτιώσει την απόδοση εκκίνησης κάτω από ορισμένες συνθήκες, με ένα δορυφορικό σύστημα εντοπισμού θέσης.

¹⁰ Ανάλυση οθόνης 800x480 pixels

Πλαίσιο αναζήτησης

- Ένα πλαίσιο αναζήτησης υπάρχει στην αρχική οθόνη για πιο γρήγορη αναζήτηση
- Πιο αποτελεσματικό διότι το σύστημα «μαθαίνει» ποια αποτελέσματα αναζήτησης είναι πιο συναφείς με ότι έχουμε πατήσει
- Η αναζήτηση γίνεται σε διάφορες πηγές όπως οι σελιδοδείκτες του προγράμματος περιήγησης ή το ιστορικό τους
- Φωνητική αναζήτηση

Κάμερα – βιντεοκάμερα

- Επιλογή πολλαπλής διαγραφής από την συλλογή
- Βελτίωση της ταχύτητας της κάμερας κατά 40% από την προηγούμενη έκδοση, καθώς και 30% βελτίωση στο χρόνο από την μία λήψη στην άλλη.
- Διαθέτει βιντεοκάμερα για λήψη βίντεο αλλά και προγράμματα αναπαραγωγής τους

Μπαταρία

- Η αρχική οθόνη διαθέτει εικονίδιο χρήσης μπαταρίας που επιτρέπει στον χρήστη να δει ποιες υπηρεσίες και εφαρμογές καταναλώνουν την ενέργεια της μπαταρίας.
- Ο χρήστης μπορεί να επέμβει μέσω των ρυθμίσεων χρήσης μπαταρίας για την εξοικονόμηση της.

Πρόγραμμα περιήγησης-εφαρμογές

- Άμεση αποστολή φωτογραφιών και βίντεο στο Picasa και στο YouTube
- Αναζήτηση και αντιγραφή – επικόλληση μέσα στο πρόγραμμα περιήγησης.
- Υποστήριξη Bluetooth
- Υπάρχει νέο πλαίσιο πρόσβασης έτσι ώστε όταν οι χρήστες κατεβάσουν νέες εφαρμογές να τους επιτρέπονται οι Ρυθμίσεις.

Ενσωματωμένες εφαρμογές

- Ξυπνητήρι
- Κάμερα – βιντεοκάμερα
- Ημερολόγιο
- E-mail
- Δρομολογητές
- Μουσική
- Μηνύματα
- Ρυθμίσεις
- IME¹¹ για την ιαπωνική γραφή
- Προσαρμογή ρυθμίσεων (για προγραμματιστές)
- Ανταλλακτικά (για προγραμματιστές)
- Εργαλεία DEV (για προγραμματιστές)

3.6 Android 2.0 και 2.1 (Éclair)



Η έκδοση αυτή αντιστοιχεί σε 3 πλατφόρμες.

Η πρώτη είναι η Android 2.0 η οποία εξελίχτηκε σε Android 2.0.1 και στην συνέχεια ξεπεράστηκαν όταν βγήκε σε κυκλοφορία η έκδοση Android 2.1. Όλες αυτές οι εκδόσεις έχουν κοινό επίπεδο API 7. Όμως στην τελευταία υπάρχουν κάποιες βελτιώσεις και διορθώσεις, και μάλιστα από την έκδοση αυτή θα πρέπει μόνο η Android 2.1 να λειτουργεί σε κινητά τηλέφωνα.

¹¹ Input method editor: αρχεία που χρειάζεται ο χρήστης για να την δημιουργία και επεξεργασία κειμένου στην κινεζική, ιαπωνική ή κορεατική γλώσσα

Τα χαρακτηριστικά των τριών αυτών πλατφορμών είναι τα ίδια.. Τα χαρακτηριστικά των εκδόσεων είναι τα εξής:

Πλαίσια Μέσων Ενημέρωσης

- Επανασχεδιασμένη αρχιτεκτονική γραφικών για ταχύτερη επίδοση του συστήματος
- Η εφαρμογή MediaScanner παράγει μικρογραφίες των εικόνων μας, όταν εισάγονται στο MediaStore

Επαφές

- Νέες επαφές APIs που επιτρέπουν δεδομένα από πολλούς χρήστες
- Γρήγορη επικοινωνία μέσω ενός παράθυρου με λίστα που επιτρέπει στους προγραμματιστές να επιλέξουν τον τρόπο που θα επικοινωνήσουν με ένα πρόσωπο
- Μπορούν να αποθηκευτούν πολλοί λογαριασμοί σε μία e-mail
- Γρήγορη πρόσβαση σε πληροφορίες επαφών καθώς και τρόπους επικοινωνίας

Bluetooth

- Υποστηρίζει Bluetooth 2,1
- Ενέργειες αυτόματου εντοπισμού
- Σύνδεση με απομακρυσμένες συσκευές κάνοντας χρήση του RFCOMM¹²

Κάμερα

- Ψηφιακό ζουμ
- Ενσωματωμένο φλας
- Νέο περιβάλλον εργασίας με την εφαρμογή Zoom Callback που πραγματοποιεί ενέργειες όταν το zoom έχει αλλάξει
- Νέες ρυθμίσεις για εφέ χρώματος, περιστροφή , ισορροπία λευκού και άλλες

¹² Radio Frequency Communication (ραδιοσυχνότητες επικοινωνίας) σύνολο πρωτοκόλλων Bluetooth

Προγράμματα περιήγησης- Εφαρμογές

- Υποστηρίζει την HTML 5
- Υποστηρίζει ζουμ εικόνας κάνοντας διπλό πάτημα
- Διαθέτει σελιδοδείκτες με μικρογραφία της ιστοσελίδας
- `windowManager.LayoutParams`: αυτή η εφαρμογή επιτρέπει σε ένα παράθυρο να ανοίξει την οθόνη όταν εκείνη είναι κλειδωμένη ή κλειστή, όπως για παράδειγμα το ξυπνητήρι.
- `WallpaperManager`: δίνει νέες ταπετσαρίες και επιτρέπει σε εφαρμογές να ζητούν την άδεια για ρύθμιση της ταπετσαρίας.
- `MotionEvent`: μπορούν να αναφέρουν ταυτόχρονα έως 3 πληροφορίες αφής, εάν το υποστηρίζει η συσκευή.
- Νέα στοιχεία APIs που ελέγχουν την κατάσταση σύνδεσης της συσκευής και επιτρέπουν σε εφαρμογές να ξεκινήσουν ειδικές δραστηριότητες για παράδειγμα όταν η συσκευή βρίσκεται μέσα σε αυτοκίνητο.
- Κεντρικός διαχειριστής λογαριασμού API για ασφαλή αποθήκευση και χρήση κωδικών πρόσβασης

Πληκτρολόγιο

- Κουμπιά αφής ή αλλιώς εικονικά πλήκτρα. Αυτά εκτελούν τις εξής εντολές: αρχική σελίδα (home), μενού (menu) , πίσω (back) , αναζήτηση (search).
- Διαθέτει ένα κουμπί για πάνω-κάτω (βοηθάει στην αποφυγή λάθος επιλογών)
- Ορθότερη διάταξη πληκτρολογίου άρα και βελτίωση της ταχύτητας δακτυλογράφησης.
- «Έξυπνο λεξικό» καθώς μαθαίνει από την χρήση των λέξεων και περιλαμβάνει αυτόματα ονόματα επαφών

Ενσωματωμένες εφαρμογές

- Ξυπνητήρι
- Δρομολογητής
- Ημερολόγιο

- Ρυθμίσεις
- Μηνύματα
- Επαφές
- Αριθμομηχανή
- Κάμερα-βιντεοκάμερα
- Μουσική
- Έκθεση
- IME για ιαπωνικούς χαρακτήρες
- E-mail
- Προσαρμογή ρυθμίσεων (για προγραμματιστές)
- Ανταλλακτικά (για προγραμματιστές)
- Εργαλεία DEV (για προγραμματιστές)

3.7 Android 2.2 Froyo

Το Android 2.2 αντιστοιχεί στην κωδική ονομασία Froyo. Έχει επίπεδο API 8, και εισάγει νέες δυνατότητες για τους χρήστες και τους προγραμματιστές. Οι εφαρμογές μπορούν να αποθηκεύονται τώρα και σε εξωτερικές μνήμες όπως micro SD.



Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

Πυρήνας Linux

- Αναβάθμιση πυρήνα σε 2.6.32
- Υποστήριξη μνήμης RAM μεγαλύτερη από 256 MB
- Προγραμματισμός SDIO¹³ και βελτιώσεις BT¹⁴

¹³ Ασφαλές Ψηφιακά δεδομένα εισόδου (Secure digital input/output)

¹⁴ Αναβάθμιση του πυρήνα

Bluetooth

- Φωνητική κλήση μέσω Bluetooth
- Δυνατότητα σύνδεσης με Bluetooth στο αυτοκίνητο

Υπηρεσίες υποστήριξης

- Οι προγραμματιστές μπορούν να γίνουν διαχειριστές της συσκευής και έτσι να μπορούν να ελέγχουν τα χαρακτηριστικά ασφαλείας της συσκευής
- Οι διαχειριστές των υπηρεσιών μπορούν να επαναφέρουν τις εργοστασιακές ρυθμίσεις στο κινητό, για την ασφάλεια των δεδομένων όταν κλαπούν ή χαθούν
- Μεγαλύτερη ασφάλεια καθώς το κλείδωμα γίνεται και με αριθμό PIN
- Υποστήριξη στα E-mail καθώς παρέχεται παγκόσμια λίστα αναζήτησης που επιτρέπει στους χρήστες να αποθηκεύουν τα ονόματα των παραληπτών από τον κατάλογο

Οθόνη

- Παρέχει ανανεωμένη αρχική σελίδα με διάφορα γραφικά στοιχεία που βοηθούν τον χρήστη να ρυθμίσει την συσκευή του (ώρα, ημερομηνία κ.α.)
- Διαθέτει 5 αρχικές οθόνες
- Διαθέτει μικρογραφίες των εφαρμογών που θέλουμε να μεταφέρουμε στην αρχική οθόνη για περισσότερη ευκολία

Κάμερα

- Η έκθεση δείχνει της φωτογραφίες σε στοίβες για πιο γρήγορη αναζήτηση
- Στην λειτουργία κάμερας υπάρχουν εικονικά κουμπιά στην οθόνη για ευκολία στις ρυθμίσεις (π.χ. ισορροπία λευκού)
- Διαθέτει φλας LED για τις νυχτερινές λήψεις και βίντεο
- Υποστηρίζει κατακόρυφο προσανατολισμό

Πληκτρολόγιο

- Οι χρήστες μπορούν να προσθέτουν διάφορες γλώσσες για την συγγραφή των μηνυμάτων τους

Φορητό ενεργό σημείο (hotspot)

- Ορισμένες συσκευές μπορούν να εκπέμπουν σήμα Wi-Fi και μπορούν να συνδεθούν ως 8 συσκευές
- Δυνατότητα χρήσης του κινητού για 3G συνδέσεις και μέσω καλωδίων USB να μεταφέρουμε την σύνδεση στον υπολογιστή μας

2.8 Android 2.3 (Gingerbread)

Το Android 2.3 αντιστοιχεί στην κωδική ονομασία Gingerbread. Έχει επίπεδο API 9, όμως στις εκδόσεις 2.3.3 και μετά έχει επίπεδο 10.



Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

Πυρήνας Linux :

- Αναβάθμιση στην έκδοση 2.6.35

Dalvik Runtime

- Strict Mode: σύστημα εντοπισμού σφαλμάτων όσο αφορά θέματα μνήμης
- Βελτιωμένη επαλήθευση κωδικού
- Προσθέτει παραπάνω βελτιστοποιήσεις JIT¹⁵

¹⁵ Just in time: είναι ένα στοιχείο λογισμικού που παίρνει τον κώδικα από μία εφαρμογή, τον αναλύει και τον επεξεργάζεται και τον μεταφράζει σε μία μορφή που κάνει την εφαρμογή να τρέχει πιο γρήγορα, παράλληλα όμως η εφαρμογή συνεχίζει να λειτουργεί

Βιβλιοθήκες πυρήνα

- Εκτεταμένη υποστήριξη H.264
- Βελτιωμένα αρχεία ελέγχου διαβάσματος και γραψίματος
- Βελτιωμένα δίκτυα APIs
- Νέες συλλογές και χρησιμότητες APIs
- Οι απαντήσεις σε HTTP είναι εξ' ορισμού συμπιεσμένες έως και 60%

Πλαίσια ενημέρωσης

- Ένα νέο πλαίσιο αντικαθιστά πλήρως των ανοιχτό κώδικα (open core) διατηρώντας όμως την προηγούμενη υποστήριξη με τον κωδικοποιητή – αποκωδικοποιητή για την κωδικοποίηση και την αποκωδικοποίηση
- ¹⁶Προσθέτει την κωδικοποίηση AAC και την ευρείας ζώνης κωδικοποίηση AMR
- Διαθέτει ενσωματωμένη υποστήριξη για τη VP8¹⁷ ανοιχτή τηλεοπτική μορφή συμπίεσης

Πληκτρολόγιο

- Επανασχεδιασμένο και βελτιστοποιημένο για ταχύτερη εισαγωγή κειμένου
- Δυνατότητα διόρθωσης λέξεων από τις προτάσεις που υπάρχουν στο λεξικό
- Νέο κουμπί με πολλαπλές επιλογές για την εισαγωγή αριθμών και συμβόλων
- Με το πάτημα ενός κουμπιού κάνει αντιγραφή- επικόλληση από ιστοσελίδες ή από την εισαγωγή κειμένου

Οθόνη

- Νέα βελτιωμένη και απλούστερη στην χρήση
- Ζωντανά χρώματα και αντιθέσεις στην γραμμή των ειδοποιήσεων

¹⁶ Υποστήριξη αρχείων πολυμέσων (εικόνες, ήχος, βίντεο)

¹⁷ Ελεύθερη μορφή συμπίεσης βίντεο

- Αλλαγές στο μενού και στις ρυθμίσεις για πιο εύκολη πλοήγηση και για τον έλεγχο των λειτουργιών του κινητού και του συστήματος

Έλεγχος εφαρμογών

- Εμφάνιση συντόμευσης της Διαχείρισης Εφαρμογών για πιο εύκολο έλεγχο των εφαρμογών
- Από την καρτέλα διαχείρισης εφαρμογών βλέπουμε ποιες είναι εν ενεργεία και πόση μνήμη χρησιμοποιούν

Μπαταρία

- Εμφάνιση στο χρήστη την ισχύ που καταναλώνουν οι εφαρμογές που εκτελούνται
- Ακριβής εικόνα για το πως χρησιμοποιείται η ενέργεια από κάθε εφαρμογή

Κλήση Internet

- Δυνατότητα φωνητικής κλήσης μέσω internet με άλλους χρήστες που διαθέτουν λογαριασμό SIP
- Το SIP καθώς και οι κλήσεις μέσω internet υποστηρίζονται από συγκεκριμένες συσκευές

Διαχείριση λήψεων

- Εύκολη πρόσβαση σε οποιοδήποτε αρχείο που μεταφορτώνεται από το πρόγραμμα περιήγησης ή από κάποια άλλη εφαρμογή
- Οι λήψεις βασίζονται σε μία νέα εφαρμογή διαχείρισης λήψεων στο σύστημα και ο χρήστης μπορεί εύκολα να διαχειρίζεται και να αποθηκεύει τη λήψη

3.9 Android 3.0 (Honeycomb)



Το Android 3.0 αντιστοιχεί στην κωδική ονομασία Honeycomb. Έχει εκδώσει τρεις πλατφόρμες την 3.0, την 3.1 και την 3.2, για το λόγο αυτό έχει και επίπεδα APIs 10,11,12 αντίστοιχα. Η πλατφόρμα αυτή είναι ειδικά σχεδιασμένη για συσκευές με μεγάλες οθόνες όπως τα tablets. Τα χαρακτηριστικά των εκδόσεων είναι τα εξής:

Σύστημα

- ο Αναβάθμιση πυρήνα στην έκδοση 2.6.36
- ο Για την συνολική εικόνα του συστήματος και για τις ειδοποιήσεις υπάρχει μία ράβδος ελέγχου στο κάτω μέρος της οθόνης
- ο Για τον έλεγχο μίας εφαρμογής, όταν αυτή χρησιμοποιείτε, υπάρχει μία ράβδος δράσης που διαθέτει επιλογές και ρυθμίσεις ανάλογα με την εφαρμογή

Οθόνη

- ο Κάθε συσκευή διαθέτει 5 αρχικές οθόνες
- ο Στην οθόνη εμφανίζονται μικρογραφίες των εγκατεστημένων εφαρμογών, για πιο εύκολη εκκίνηση

Προγράμματα περιήγησης- Εφαρμογές

- ο Στην γραμμή συστήματος υπάρχει μία λίστα με τις πιο πρόσφατες εφαρμογές καθώς και εκείνες που είναι σε λειτουργία
- ο Αντικατάσταση πολλών παραθύρων με πολλαπλές καρτέλες
- ο Κατάσταση ανώνυμης περιήγησης "incognito"
- ο Οι χρήστες επιλέγουν για την αυτόματη εγγραφή σε ιστότοπους της google όπως το Gmail

Πληκτρολόγιο

- ο Γρήγορη και ακριβείς εισαγωγή κειμένου με νέο πληκτρολόγιο διαμορφωμένο ειδικά για μεγάλες οθόνες
- ο Παρατεταμένα πατημένα πλήκτρα εμφανίζουν σύμβολα και αριθμούς

- Από την γραμμή του συστήματος αλλάζει η είσοδος δεδομένων, είτε γραπτά είτε προφορικά
- Αντιγραφή και επικόλληση με ρύθμιση της περιοχής επιλογής με σύρσιμο μεταξύ δύο βελών

Κάμερα και Έκθεση

- Μεγαλύτερη ωφέλεια λόγω της μεγαλύτερης οθόνης για πιο γρήγορη πρόσβαση καθώς και για τις ρυθμίσεις της κάμερας
- Δυνατότητα εμφάνισης άλμπουμ και άλλων συλλογών σε πλήρη οθόνη

3.10 Android 4.0 (Ice cream sandwich)



Η πλατφόρμα της Android 4.0 υποστηρίζει επίπεδο API 14.

Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

Σύστημα

- Αναβάθμιση πυρήνα στην έκδοση 3.0.1

Οθόνη

- Εξελιγμένη, πιο εύχρηστη και υψηλής ανάλυσης αρχική οθόνη
- Εικονικά κουμπιά για εύκολη περιήγηση στο μενού και στις εφαρμογές
- Εμφάνιση φακέλων στην αρχική οθόνη για την ομαδοποίηση εφαρμογών, όπως αρέσει στον χρήστη.
- Στις εισερχόμενες κλήσεις υπάρχει δυνατότητα άμεσης αποστολής μηνύματος χωρίς να απαντηθεί η κλήση
- Ξεκλείδωμα οθόνης βάση προσώπου: με μία κάμερα στο μπροστινό μέρος της συσκευής με την τεχνολογία αναγνώρισης προσώπου. Υπάρχει και το κλείδωμα με αριθμό PIN

Πληκτρολόγιο

- Γρήγορη και ακριβείς πληκτρολόγηση κειμένου
- Νέα λεξικά που διορθώνουν λάθη και προτείνουν λέξεις
- Προσθήκη αρθρογράφου για έλεγχο της ορθογραφίας
- Φωνητική εγγραφή, ο χρήστης υπαγορεύει σε όποια γλώσσα θέλει και η συσκευή αυτόματα γράφει, εντοπίζοντας παράλληλα λάθη και προτείνει προτάσεις ή λέξεις από το λεξικό
- Οι χρήστες μπορούν να αυξήσουν το μέγεθος της γραμματοσειράς, πράμα που θα γίνει σε ολόκληρο το σύστημα

Προγράμματα περιήγησης-εφαρμογές

- Μεγάλη χρήση των δεδομένων του δικτύου για το κατέβασμα εφαρμογών, για τον συγχρονισμό δεδομένων κ.α.
- Στις Ρυθμίσεις υπάρχουν ενδείξεις για την συνολική χρήση και το ποσό των δεδομένων που χρησιμοποιούνται ανάλογα με τον τύπο του δικτύου (τα Wi-Fi ή τα κινητά)
- Δυνατότητα χρήσης από χρήστες με προβλήματα όρασης καθώς για όλες τις κινήσεις των χρηστών διαθέτει και φωνητική λειτουργία
- Οι χρήστες μπορούν να αποθηκεύουν ένα αντίγραφο ενός εγγράφου για να το διαβάσουν εκτός δικτύου
- Σελιδοδείκτες με οπτική λίστα για γρηγορότερη πρόσβαση
- Εύκολη αποστολή e-mail καθώς δίνει την δυνατότητα αυτόματης συμπλήρωσης παραλήπτη
- Κουμπί πρόσφατων εφαρμογών που εμφανίζει μικρογραφίες των πρόσφατα χρησιμοποιημένων εφαρμογών

Επαφές

- Στην αρχική οθόνη υπάρχει μία νέα εφαρμογή newPeople που δίνει την δυνατότητα στο χρήστη να συνδέσει όλες τις πληροφορίες και τις εικόνες που αφορούν σε κάθε μία επαφή του

- Υπάρχει επίσης μία νέα εφαρμογή Me που ο χρήστης έχει όλα τα προσωπικά του στοιχεία σε αυτή

Κάμερα

- Ταχύτερη λήψη φωτογραφιών και μηδενική καθυστέρηση κλείστρου
- Μετάβαση στην κάμερα και στις ειδοποιήσεις χωρίς ξεκλείδωμα της οθόνης
- Κατά την εγγραφή βίντεο δίνετε η δυνατότητα λήψης φωτογραφίας χωρίς να επηρεάσει την ροή της εγγραφής
- Μετά την εγγραφή βίντεο ή την λήψη φωτογραφιών οι χρήστες μπορούν εύκολα με το πάτημα ενός κουμπιού να τις μοιράζονται μέσω κοινωνικών δικτύων, Bluetooth κ.α.
- Βελτιωμένη Έκθεση με δυνατότητα επεξεργασίας των φωτογραφιών
- Δυνατότητα προσθήκης ψεύτικων εφέ κατά την διάρκεια εγγραφής βίντεο για αστεία αποτελέσματα

Android Beam

Το Android Beam είναι μία καινοτομία που βοηθάει τον χρήστη στην άμεση ανταλλαγή εικόνων, εφαρμογών, αρχείων ήχου κ.α. είναι υπερβολικά εύκολο στην χρήση καθώς λειτουργεί σαν ένα σύγχρονο και εξελιγμένο Bluetooth.

Αυτό σημαίνει πώς για παράδειγμα για την κοινή χρήση των εφαρμογών η μία συσκευή ωθεί ένα σύνδεσμο προς την σελίδα εφαρμογών του Google Play και η άλλη συσκευή με την εφαρμογή Google Play αρχίζει να φορτώνει την σελίδα με τις λεπτομέρειες για την λήψη της εφαρμογής.



3.11 Android 4.1 (Jelly Bean)

Η πλατφόρμα της Android 4.1 υποστηρίζει επίπεδο API 16. Είναι σαφώς γρηγορότερη από τις προηγούμενες εκδόσεις και έχει υποστεί βελτιώσεις σε ολόκληρη την πλατφόρμα. Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

Οθόνη

- Μεγαλύτερα εικονίδια διαλόγων για πιο εύκολη επιλογή τις προτιμώμενης εφαρμογής
- Το ξεκλείδωμα της οθόνης βάσει προσώπου (Face Unlock) είναι πιο εύχρηστο και ακριβές
- Επίσης γίνεται τώρα πιο «έξυπνο» καθώς προαιρετικά απαιτεί ένα βλεφαρισμό για να βεβαιωθεί ότι το πρόσωπο δεν είναι απλά μία φωτογραφία.

Επαφές

- Εισάγει μία νέα ειδοποίηση, για μία κλήση που χάθηκε, που επιτρέπει την κλήση ή την αποστολή μηνύματος με ένα απλό άγγιγμα
- Δυνατότητα μεταφοράς επαφών από το αρχείο κλήσεων στις επαφές, και είναι μόνο για ανάγνωση
- Στην καρτέλα Αγαπημένες επαφές, δίνετε η δυνατότητα συχνής διαγραφής της λίστας των επαφών
- Μία ειδοποίηση επιτρέπει το κλείσιμο της κλήσης που βρίσκετε σε εξέλιξη με ένα άγγιγμα
- Τα voicemails (μηνύματα φωνητικού ταχυδρομείου) διαβάζονται απλά, με το πάτημα της ειδοποίησης

Πληκτρολόγιο

- Υπάρχει μεγαλύτερη ακρίβεια στα λεξικά
- Διαθέτει μία νέα εφαρμογή για φωνητική εγγραφή, ακόμη και όταν δεν υπάρχει σύνδεση στο διαδίκτυο
- 18 νέες γλώσσες εισόδου όπως τα Χίντι, Ινδικά Μαγιάντα κ.α.
- Διαθέτει πλήκτρο αλλαγής γλωσσών

- Επιλογή μεταξύ διαφόρων ειδών πληκτρολογίων, όπως το κλασσικό qwerty ή άλλα όπως το azerty και το colemak, επίσης επιλογή μεταξύ 20 στυλ εισαγωγής των χαρακτήρων

Κάμερα

- Με μία κίνηση εναλλάσσονται η μπροστινή με την πίσω κάμερα
- Κατά την εστίαση σε ένα αντικείμενο, μας δίνει την οπτική ανάδραση σχετικά με την κατάσταση της εστίασης
- Στην έκθεση φωτογραφιών μπορούμε με ένα άγγιγμα να διαγράψουμε μία φωτογραφία ή ακόμη και να αναιρέσουμε την διαγραφή αυτή

Προγράμματα περιήγησης- εφαρμογές

- Τα προγράμματα περιήγησης έχουν καλύτερη απόδοση, μεγαλύτερη μνήμη καθώς και βελτιωμένο επεξεργαστή CPU
- Διαθέτει ενημερωμένο JavaScript Engine V8 για καλύτερη απόδοση σε κινούμενα σχέδια και πιο γρήγορη φόρτωση σελίδων
- Καλύτερη υποστήριξη των βίντεο σε HTML5 και εύκολη μετάβαση από μικρή σε πλήρη οθόνη
- Το ημερολόγιο γίνεται πιο χρηστικό με χρώματα και κινούμενα σχέδια. Επίσης διαχειρίζεται ειδοποιήσεις για επερχόμενα γεγονότα αλλά διαθέτει και επιλογή για την δημιουργία σύντομου μηνύματος για τους επισκέπτες ενός γεγονότος.

3.12 Android 4.2-4.3 (Jelly Bean)

Η επόμενη εκδόσεις του Android κράτησαν την ίδια ονομασία με την προηγούμενη έκδοση δηλαδή Jelly Bean. Η έκδοση 4.2 έχει API 17, ενώ η έκδοση 4.3 έχει αριθμό API 18. Οι δυνατότητες παραμένουν ίδιες, όμως παραθέτουμε κάποια νέα χαρακτηριστικά που έχουν προστεθεί. Οι νέες δυνατότητες της έκδοσης είναι οι εξής:

Οθόνη

- Η εναλλαγή μεταξύ των αρχικών οθόνων δίνει την αίσθηση ενός ξεφυλλίσματος ενός βιβλίου
- Υπάρχει δυνατότητα στην κλειδωμένη οθόνη να προσθέσεις εφαρμογές όπως ημερολόγιο, παγκόσμιο ρολόι, Gmail, κάμερα και μηνύματα για γρηγορότερη πρόσβαση
- Με την νέα λειτουργία εμφάνισης 'Day Dream' υπάρχει δυνατότητα οι εφαρμογές να εμφανίζουν διαδραστικά εικόνες, όταν το κινητό δεν χρησιμοποιείται όπως για παράδειγμα ένα άλμπουμ από την έκθεση

Κάμερα

- Άμεση πρόσβαση στην κάμερα ακόμη και από κλειδωμένη οθόνη
- Διαθέτει μία νέα εφαρμογή κάμερας που παρέχει νέα εργαλεία και χαρακτηριστικά για την λήψη φωτογραφιών
- Όταν χτυπάμε για να εστιάσει σε μία φωτογραφία υπάρχουν νέες κινήσεις για να μας πουν τον βαθμό της εστίασης
- Με την εφαρμογή Photo Sphere μπορεί να γίνει λήψη πανοραμικών φωτογραφιών με 360°
- Με το Photo Sphere μπορούν να κοινοποιηθούν φωτογραφίες στην τοποθεσία που λήφθηκαν, πάνω στο χάρτη της Google.
- Νέο πρόγραμμα για την προσαρμογή των φωτογραφιών με νέα φίλτρα και εργαλεία

Διεθνοποίηση και Εντοπισμός

- Υποστηρίζει γλώσσες που διαβάζονται από δεξιά προς αριστερά όπως τα αραβικά σε εφαρμογές και στην αρχική οθόνη
- Πιο ακριβής τοποθέτηση στους τόνους της ινδικής, αραβικής και της εβραϊκής γλώσσας

Ρολόι

- Επιλογή ανάμεσα σε ψηφιακό και αναλογικό ρολόι. Υπάρχουν νέες δυνατότητες όπως αλλαγή μεγέθους του ψηφιακού ρολογιού για να εμφανιστούν και ώρες από πόλεις σε όλον τον κόσμο
- Διαθέτει νέο χρονόμετρο με ικανότητες παρακολούθησης προκαθορισμένων χρόνων, έναρξη πολλαπλών χρονομέτρων ταυτόχρονα και άλλα

Σύστημα - Εφαρμογές

- Στην πληκτρολόγηση μία νέα εφαρμογή κάνει την εμφάνιση της το Gesture Typing, βάσει αυτού σέρνοντας το δάκτυλο από γράμμα σε γράμμα και με την επιλογή των αυτόματων διαστημάτων η πληκτρολόγηση θα είναι ευκολότερη
- Δυνατότητα πολλαπλών χρηστών στα tablets. Δίνει τη δυνατότητα για 3 ταυτόχρονα ενεργούς χρήστες.
- Υποστηρίζει την τεχνολογία Miracast™ μέσω Wi-Fi που επιτρέπει την μεταφορά αρχείων από το κινητό ή το tablet σε τηλεοράσεις με εξοπλισμό HDTV
- Εφαρμόζει μία νέα πολιτική επαλήθευσης, με αυξημένη ασφάλεια, των εφαρμογών που κατεβάζει ο χρήστης από το Google Play.
- Όταν μία εφαρμογή προσπαθεί να στείλει μηνύματα με χρέωση εν αγνοία του χρήστη, ειδοποιεί τον χρήστη πριν την αποστολή

3.13 Android 4.4 (Kit Kat)



Η τελευταία έκδοση του Android έχει κωδική ονομασία Kit Kat και API 19.

Τα χαρακτηριστικά της έκδοσης είναι τα εξής:

Σύστημα

- Χρησιμοποιεί 16% λιγότερη μνήμη από την έκδοση Jelly Bean, που σημαίνει ότι υπάρχουν περισσότεροι διαθέσιμοι πόροι για το τρέξιμο εφαρμογών.
- Διαθέτει μνήμη RAM 512 MB επομένως ο χειρισμός του είναι εύκολος σε πολλές ταυτόχρονες διεργασίες και η ταχύτητα θα είναι μεγαλύτερη

Οθόνη - Προσβασιμότητα

- Στην αρχική οθόνη ή στο Google Play αρκεί να πεις “Okay google” και αυτόματα θα ξεκινήσει φωνητική αναζήτηση, αναπαραγωγή τραγουδιού ακόμη και αποστολή μηνύματος
- Τα κουμπιά στην γραμμή κατάστασης και πλοήγησης κρύβονται με ένα τράβηγμα κατά την ανάγνωση ή την αναπαραγωγή βίντεο
- Διαθέτει βελτιωμένο λογισμικό για ταχύτητα και ακρίβεια στο άγγιγμα της οθόνης

Κάμερα

- Η κυριότερη εφαρμογή στην κάμερα είναι η λειτουργία του HDR+ (High Dynamic Range). Με αυτή την εφαρμογή οι φωτογραφίες έχουν μεγάλη αντίθεση και πλούσια χρώματα.
- Οι ρυθμίσεις της κάμερας δίνουν την δυνατότητα για την ρύθμιση της φωτεινότητας, την ισορροπία του λευκού καθώς και ενός χρονομέτρου αντίστροφης μέτρησης

Πληκτρολόγιο

- Νέα emoticons στην σύνταξη των μηνυμάτων

Επαφές

- Το αρχείο κλήσεων και οι επαφές οργανώνονται αλφαβητικά αλλά επίσης και με βάση τους αριθμούς που γίνονται κλήσεις συχνότερα
- Υπάρχει δυνατότητα να εμφανίζονται πληροφορίες για τις κοντινότερες επιχειρήσεις από την ίδια μπάρα των επαφών

Προγράμματα περιήγησης- εφαρμογές

- Υπάρχει η δυνατότητα να διαχειριστείς ένα άλμπουμ τραγουδιών σε κλειδωμένη οθόνη κατά την αναπαραγωγή τους
- Με την εφαρμογή Google cloud printers, η εκτύπωση γίνεται πιο εύκολη
- Εφαρμογές όπως eBooks ή ταινίες εμφανίζονται σε πλήρη οθόνη αυτόματα με απόκρυψη των κουμπιών και της μπάρας πλοήγησης

Κεφάλαιο 4 Έξυπνες Συσκευές

Οι περισσότερες εφαρμογές προς το παρόν αναφέρονται σε κινητές συσκευές, tablets ή notebooks. Στην πραγματικότητα όμως έχουν δημιουργηθεί νέα καινοτόμα προϊόντα που αξίζει να αναφέρουμε την συμβατότητα τους με τα Android. Στο κεφάλαιο αυτό θα δούμε μερικές από τις χρήσεις που έχει ένα λειτουργικό σύστημα Android σε άλλου είδους συσκευές.

4.1 Smartwatch

ή αλλιώς το «έξυπνο ρολόι». Ένας μικρός υπολογιστής που φοριέται στον καρπό του χρήστη και έχει ίδιες δυνατότητες με ένα Smartphone. Το ρολόι αυτό μπορεί να συνδεθεί με το κινητό και παρέχει λειτουργίες όπως μηνύματα, κλήσεις, κοινωνικά δίκτυα, εφαρμογές του διαδικτύου, συντεταγμένες GPS και άλλα. Επίσης οι ειδοποιήσεις της συσκευής, έρχονται και στο ρολόι. Υπάρχουν εφαρμογές για το αξεσουάρ αυτό και τις βρίσκουμε στο Google play. Μερικές από τις εταιρίες εταιρείες που παράγουν το έξυπνο ρολόι είναι η Sony, η LG, η Apple και η Samsung.



4.2 Αυτοκίνητα

Αρκετές βιομηχανίες αυτοκινήτων έχουν δημιουργήσει συστήματα ψυχαγωγίας και ενημέρωσης που στηρίζονται στο Android. Ενδεικτικά να αναφέρουμε την Saab, με το iQon, και την Ford, με το πρόγραμμα OpenXC¹⁸, που έχουν ήδη υλοποιήσει τέτοιο λογισμικό το οποίο δίνει στον οδηγό ένα ολοκληρωμένο σύστημα από ψυχαγωγία-διασκέδαση αλλά και ένα αρκετά καλό σύστημα GPS το οποίο συνδέεται με τις λειτουργίες και τους αισθητήρες του αμαξιού.



¹⁸ Απευθύνεται σε προγραμματιστές και αφορά την σύνδεση του Android με περαιτέρω λειτουργίες του αυτοκινήτου

4.3 Ψηφιακές φωτογραφικές μηχανές

Παρόλο που τα τελευταία χρόνια οι κάμερες στα κινητά τηλέφωνα έχουν εξελιχθεί τόσο, όσο να μην χρειαστεί ο κάτοχος να αγοράσει φωτογραφική μηχανή, οι κατασκευαστές τους δεν πτοούνται. Για παράδειγμα θα αναφέρουμε την κάμερα της που εάν κάποιος έβλεπε την πίσω όψη της μηχανής, ίσως να το πέρναγε για ένα ακόμη κινητό τηλέφωνο. Το μοντέλο αυτό διαθέτει Wi-Fi, Bluetooth, ραδιόφωνο και GPS. Τον ίδιο δρόμο έχει ακολουθήσει και η Nikon με το Coolpix S800c.



4.4 Κονσόλες παιχνιδιών

Πέρα από τις κλασσικές παιχνιδοκονσόλες υπάρχει και η ouya. Η ouya είναι μία κονσόλα που βασίζεται στην δική της εκδοχή του λειτουργικού συστήματος των Android. Είναι μία ανοιχτού κώδικα παιχνιδομηχανή, διαθέτει δωρεάν παιχνίδια, έχει ένα κιτ¹⁹ για σχεδιασμό νέων παιχνιδιών (αφορά μόνο τους προγραμματιστές) και είναι διαθέσιμη σε όλο το ευρύ κοινό.



¹⁹ Διαθέτει 8 GB μνήμης, 1 GB RAM, Android 4.0, Ethernet, Bluetooth, HDMI έξοδο για την τηλεόραση και κάρτα γραφικών NVIDIA Tegra 3

4.5 Smart glasses

Η Google έχει σχεδιάσει ένα καινοτόμο ζευγάρι γυαλιών, που ακόμη βρίσκετε σε πειραματικά στάδια. Στην ουσία είναι γυαλιά τα οποία συνδέονται με το κινητό τηλέφωνο και εκείνος που τα φοράει βλέπει μπροστά του ειδοποιήσεις, αποστάσεις, κλήσεις ή ότι άλλο θα μπορούσε να δει στην συσκευή του. Ακόμη θα υποστηρίζει φωνητική εγγραφή για άμεση απάντηση σε μηνύματα.

Υπάρχει επίσης ένα διαφορετικό μοντέλο έξυπνων γυαλιών που αφορούν τους αθλητές. Αυτά, μαζί με ένα λογισμικό το Android HUD εμφανίζει το υψόμετρο, την θερμοκρασία και τις συντεταγμένες σε μία μικρή οθόνη που έχει ενσωματωθεί στα γυαλιά.



4.6 Οικιακές συσκευές

Μεγάλες εταιρίες όπως η LG, η Samsung και η Panasonic έχουν ήδη κατασκευάσει ηλεκτρικές οικιακές συσκευές με οθόνες αφής που είναι συμβατές με το λειτουργικό σύστημα των Android. Αυτές οι συσκευές προς το παρόν είναι πλυντήρια και ψυγείοκαταψύκτες που με την σύνδεση στη συσκευή μπορούμε να συνδεθούμε στο διαδίκτυο αλλά και να μάθουμε τον χρόνο που απομένει για την ολοκλήρωση της πλύσης, συνταγές μαγειρικής ή υπόλοιπο τροφίμων.



4.7 Smart TVs

Με λίγα λόγια σαν μεγάλος υπολογιστής με τηλεχειριστήριο. Οι τηλεοράσεις αυτές διαθέτουν Wi-Fi και κατά συνέπεια εξυπηρετεί την ίδια δουλειά με την οθόνη ενός Η/Υ. Συνδέονται στο διαδίκτυο, διαθέτουν ενσωματωμένες κάμερες για βιντεοκλήσεις (Skype), τρέχουν εφαρμογές και τέλος μπορούν να συνδεθούν με το κινητό, το tablet ή το laptop ασύρματα.



4.8 Καθρέπτες

Ο cyberecture mirror είναι ένας απλός καθρέπτης. Με το πάτημα ενός κουμπιού όμως ζωντανεύει. Δεν είναι αφής, διαθέτει ένα χειριστήριο, όμως μπορεί να ελεγχθεί και από το Android κινητό(ή το Ipad) με την κατάλληλη εφαρμογή. Μπορούμε να φτιάξουμε το προφίλ μας ή ακόμη το προφίλ όλης της οικογένειας και έτσι για κάθε άνοιγμα του καθρέπτη απαιτείται και ο κωδικός του προφίλ. Υποστηρίζει Wi-Fi συνδέσεις, επομένως και συνδέσεις με όλα τα κοινωνικά δίκτυα. Άλλες υπηρεσίες που διαθέτει ο καθρέπτης αυτός είναι η διασκέδαση (για παράδειγμα να δούμε ένα πρόγραμμα στην τηλεόραση), η ενημέρωση (είτε για παράδειγμα για τον καιρό, είτε για τα updates των φίλων μας στα κοινωνικά δίκτυα), να ελέγξουμε την φυσική μας κατάσταση (για παράδειγμα με μία ειδική ζυγαριά να μετρήσει το βάρος, το ύψος, την μυϊκή μάζα μας και όλα αυτά τα κρατάει στην μνήμη του), και να έχει ρυθμιζόμενο φωτισμό (μεταξύ 3 ή 6 λάμπες).



Κεφάλαιο 5 Εφαρμογές Android

Σε αυτό το κεφάλαιο θα δείξουμε μερικές εφαρμογές Android που υπάρχουν στην αγορά και την χρήση τους. Σκοπός μας είναι να δώσουμε μια εικόνα για το πώς είναι μια εφαρμογή, σε τι χρησιμεύει και πώς μπορούμε να βρούμε μια.

Η πιο γνωστή ιστοσελίδα για να κατεβάσει κάποιος μια εφαρμογή είναι της Google, η Google Play (play.google.com). Σε αυτήν την σελίδα υπάρχουν πάρα πολλές εφαρμογές οι οποίες είτε είναι δωρεάν είτε επί πληρωμή. Επίσης, όποιος θέλει μπορεί να ανεβάσει τη δική του εφαρμογή για να την πουλήσει ή να είναι δωρεάν σε χρήστες.

Βέβαια, δεν είναι μόνο η Google Play η οποία μας δίνει την δυνατότητα να βρούμε εφαρμογές, για παράδειγμα υπάρχουν διάφορα forums στα οποία μπορούμε να βρούμε αυτό που θέλουμε.

Παρακάτω υπάρχουν μερικές ενδεικτικές εφαρμογές για Android.

5.1 Σταυρόλεξα

Η εφαρμογή Σταυρόλεξα διατίθεται δωρεάν και είναι κλασικό παιχνίδι μυαλού και γνώσεων. Είναι διασκεδαστική και εκπαιδευτική εφαρμογή με την οποία μπορεί κανείς να εμπλουτίσει τις γνώσεις του.

Η εφαρμογή "Σταυρόλεξα" για Android προσφέρει :

- Πολλές λέξεις μαζί με εγκυκλοπαιδικά στοιχεία για αυτές
- Υποστηρίζει όλα τα κινητά τηλέφωνα και tablet PCs
- Πολύ μικρό μέγεθος, μικρότερο από 1Mb
- Δυνατότητα μεταφοράς στην κάρτα SD
- Συνεχείς ενημερώσεις με νέες λέξεις
- Είναι κατάλληλη, τόσο για μικρά παιδιά, όσο και για ενήλικες



5.2 Ελληνικό Εορτολόγιο/Eortologio

Η εφαρμογή αυτή μας προσφέρει τις εξής λειτουργίες:

- Ενημέρωση για εορτάζουσες επαφές (ονομαστική εορτή/γενέθλια)
- Κλήση ή αποστολή μηνύματος στις εορτάζουσες επαφές
- Αργίες
- Προσθήκη/Τροποποίηση/Διαγραφή εορτών, γενεθλίων και αργιών
- Ρυθμιζόμενο θέμα, χρώμα και μέγεθος γραμματοσειράς
- Δυνατότητα απενεργοποίησης ειδοποίησης
- Προβολή εορτών και γενεθλίων των φίλων από τα κοινωνικά δίκτυα καθώς και δυνατότητα δημοσίευσης στον τοίχο τους
- Ανανεωμένη αναζήτηση βάσει ονόματος και ημερομηνίας
- Επιλογή δημιουργίας και επαναφοράς αντίγραφου ασφαλείας (backup)



5.3 Meteo.gr

Τι περιλαμβάνει η εφαρμογή:

- Πρόγνωση καιρού (για 6 ημέρες)
- Webcams για παρατήρηση σε πραγματικό χρόνο
- Παρατήρηση Καιρού (ο καιρός τώρα)
- My Meteo (οι αγαπημένες μου περιοχές)
- Πρόγνωση κατά Περιοχή
- Χειμερινούς Προορισμούς
- Χιονοδρομικά Κέντρα
- Προγνώσεις Θαλασσών
- Οδικό Δίκτυο
- Καθημερινές ενημερώσεις για τον καιρό (επιλογή "Νέα")

Ο χρήστης έχει τη δυνατότητα να κάνει ζουμ πάνω στο χάρτη ή να αναζητήσει κάποια περιοχή από αλφαβητική λίστα (εφόσον δεν γνωρίζει που βρίσκεται η περιοχή στο χάρτη). Επιπλέον, οι επιλογές του μενού της βασικής μπάρας προσαρμόζονται ανάλογα με τις ανάγκες του χρήστη. Για παράδειγμα, κάποιος που ασχολείται με χειμερινά σπορ μπορεί να έχει στη βασική μπάρα τα «Χιονοδρομικά Κέντρα» κτλ.



5.4 Viber

Με την εφαρμογή Viber μπορούμε να επικοινωνήσουμε με κλήσεις, βιντεοκλήσεις, μηνύματα και φωτογραφίες με άτομα από όλο τον κόσμο, με μόνη προϋπόθεση να υπάρχει σύνδεση με Wi-Fi ή με 3G. Η εφαρμογή είναι δωρεάν και είναι συμβατή με πολλά έξυπνα τηλέφωνα και tablets. Απαιτεί έκδοση λειτουργικού συστήματος Android 2,2,x και πάνω.

Τι περιλαμβάνει η εφαρμογή:

- Αποστολή μηνυμάτων κειμένου
- Δωρεάν κλήσεις με ποιότητας ήχου υψηλής πιστότητας
- Αποστολή φωτογραφιών, βίντεο, ηχητικών μηνυμάτων και τοποθεσιών
- Άμεση απάντηση σε μηνύματα μέσω της λειτουργίας «γρήγορη απάντηση»
- Αποστολή φωτογραφιών και βίντεο απευθείας από την συλλογή της συσκευής
- Σχεδιάστηκε βάσει του Native Android UI
- Λειτουργία προώθησης ειδοποιήσεων για να μην χάνονται μηνύματα ή κλήσεις ακόμη και όταν το Viber είναι απενεργοποιημένο
- Υπηρεσία υποστήριξης για την εφαρμογή Viber Desktop σε Windows και Mac



5.5 Smart Tools

Η εφαρμογή αυτή περιλαμβάνει ένα σύνολο με πέντε ομάδες εφαρμογών για μετρήσεις διαφόρων μεγεθών. Απαιτεί έκδοση λειτουργικού συστήματος Android 2,1 και πάνω. Την εφαρμογή μπορούμε να την βρούμε με μικρή χρέωση. Τα σύνολα αυτές διακρίνονται σε:

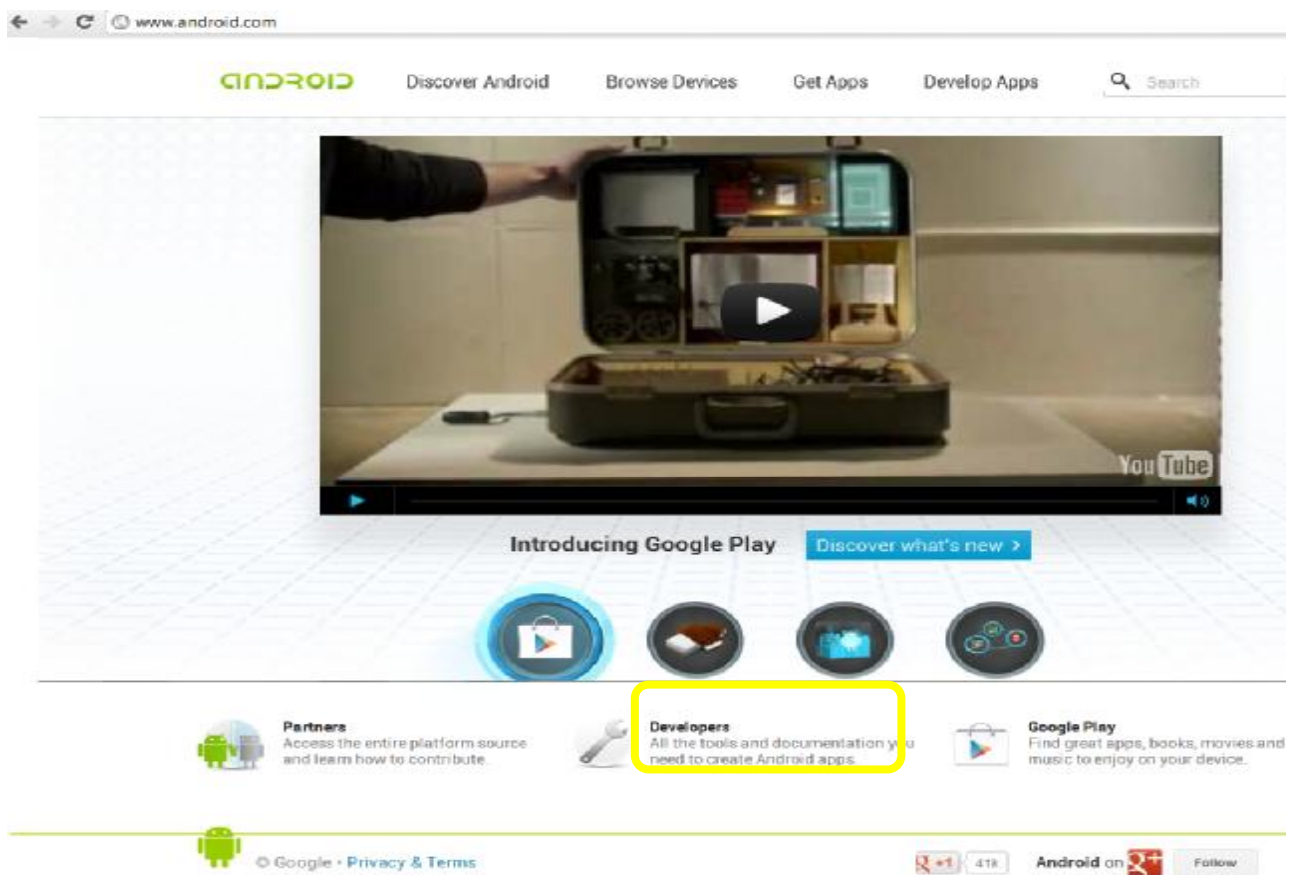
- Πρώτη ομάδα: μήκος, γωνία, κλίση, επίπεδο και χάρακας
- Δεύτερη ομάδα: ύψος, πλάτος και απόσταση
- Τρίτη ομάδα: ανιχνευτής μετάλλων, GPS και πυξίδα
- Τέταρτη ομάδα: μετρητής θορύβου και μετρητής ηχητικών δονήσεων
- Πέμπτη ομάδα: φακός, μεγεθυντικός φακός και καθρέπτης



Κεφάλαιο 6 Περιβάλλον ανάπτυξης εφαρμογών

Σε αυτό το σημείο θα δείξουμε πώς αναπτύσσουμε μια εφαρμογή android. Αρχικά θα πρέπει να προετοιμάσουμε το περιβάλλον ανάπτυξης εφαρμογών android, το οποίο αποτελείται από 3 βασικά εργαλεία: το Java, το Eclipse και το Android. Η εγκατάσταση αυτών πρέπει να γίνει με την σειρά και τον τρόπο που έχουμε παρακάτω.

6.1 Εγκατάσταση των αρχείων από το Internet



Βήμα 1: Μπαίνουμε στην σελίδα www.android.com και επιλέγουμε Developers.

Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the Android SDK and AVD Manager, rather than downloading a new SDK starter package. See [Adding SDK Components](#).

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r18-windows.zip	37448775 bytes	afbf0fb2d086cc2a621544d708fa3b
	installer_r18-windows.exe (Recommended)	37456234 bytes	48b1f71431ab6b9c6952b75d089f
Mac OS X (Intel)	android-sdk_r18-macosx.zip	3933706 bytes	8320e6d531c9d09f1a0261cb97a0ff
Linux (64)	android-sdk-linux.tgz	29721463 bytes	8cd716d9e04424b805fec3c25c3485c

Here's an overview of the steps you must follow to set up the Android SDK:

1. Prepare your development computer and ensure it meets the system requirements.
2. Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)
3. Install the ADT Plugin for Eclipse (if you'll be developing in Eclipse).
4. Add Android platforms and other packages to your SDK.
5. Explore the contents of the Android SDK (optional).

To get started, download the appropriate package from the table above, then read the guide to [Installing the SDK](#).

Βήμα 2: Επιλέγουμε το installer_r18-windows.exe και κατεβάζουμε το αρχείο.

Installing the SDK

If you encounter any problems during installation, see the [Troubleshooting](#) section at the bottom of this page.

Updating?

If you already have an Android SDK, use the Android SDK Manager tool to install updated tools and new Android platforms into your existing environment. For information about how to do that, see [Adding SDK Packages](#).

Step 1. Preparing Your Development Computer

Before getting started with the Android SDK, take a moment to confirm that your development computer meets the [System Requirements](#). In particular, you might need to install the [JDK](#), if you don't have it already.

If you will be developing in Eclipse with the Android Development Tools (ADT) Plugin—the recommended path if you are new to Android—make sure that you have a suitable version of Eclipse installed on your computer as described in the [System Requirements](#) document. If you need to install Eclipse, you can download it from this location:

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

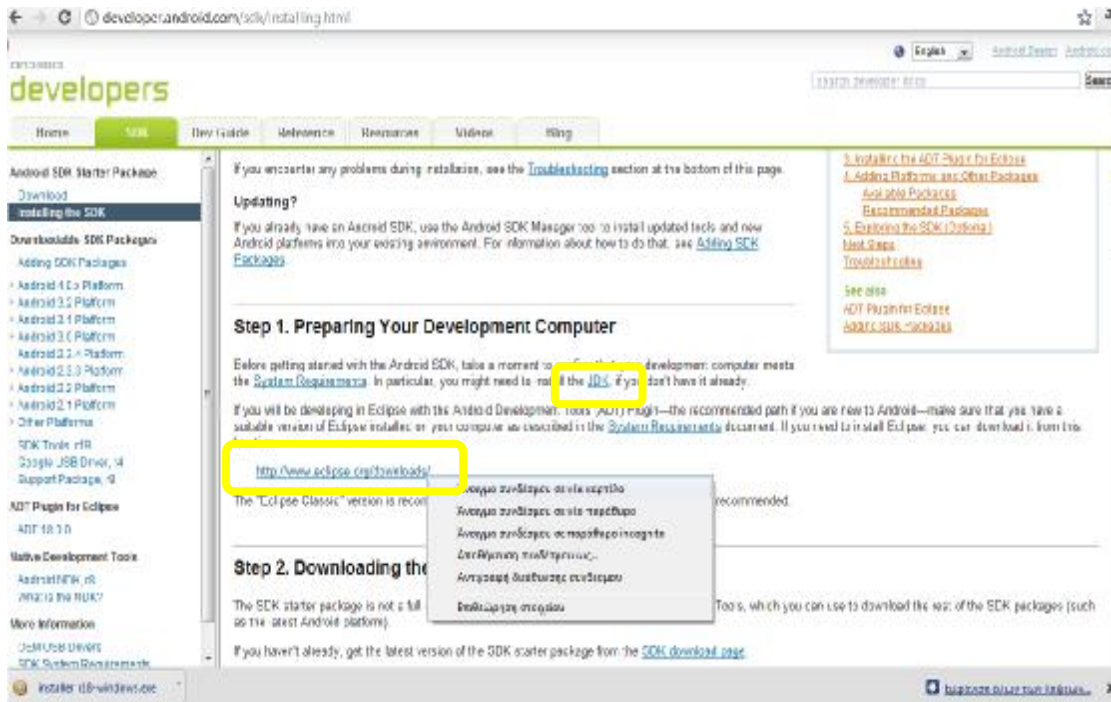
The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

Step 2. Downloading the SDK Starter Package

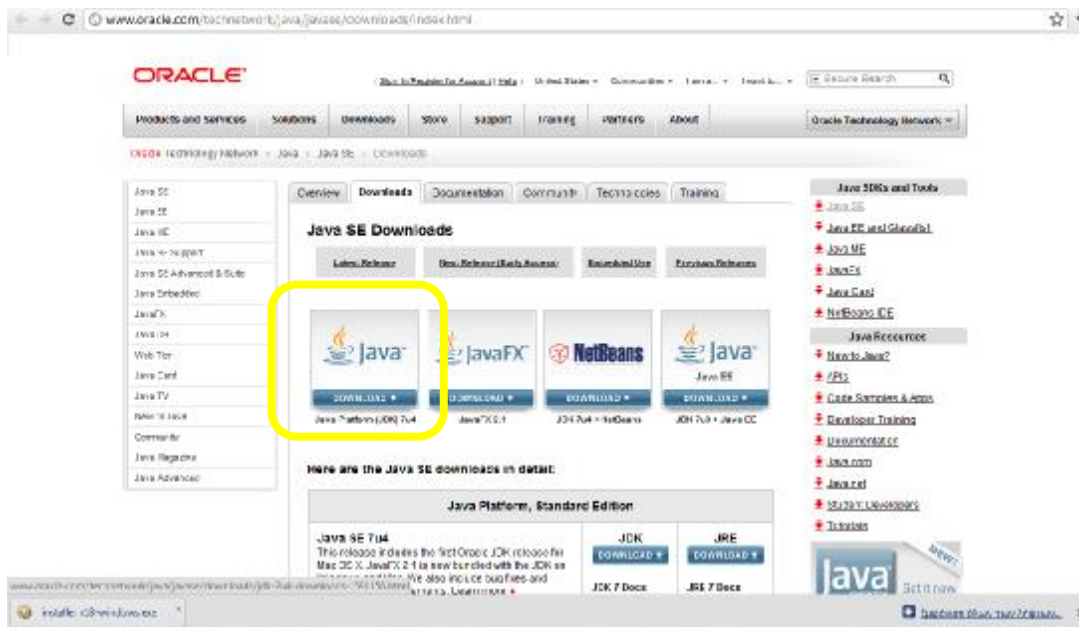
The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK packages (such as the latest Android platform).

If you haven't already, get the latest version of the SDK starter package from the [SDK download page](#).

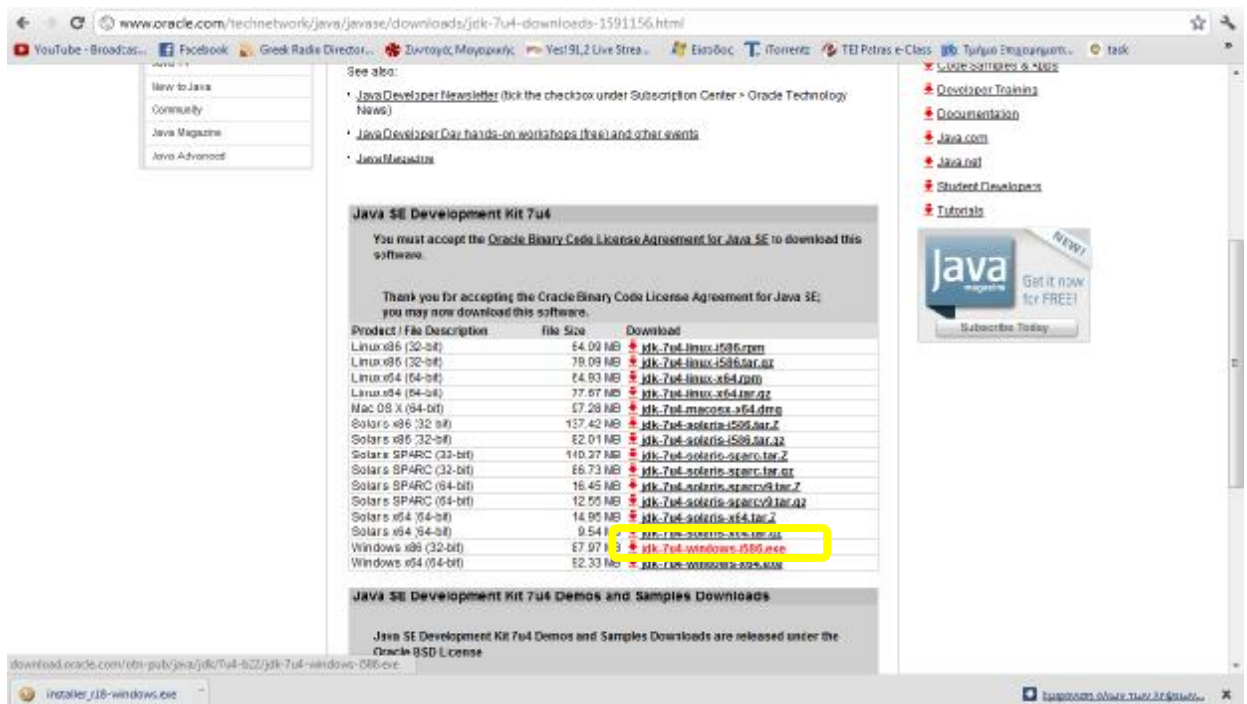
Βήμα 3: Πηγαίνουμε στην καρτέλα Installing the SDK που βρίσκεται στο μενού αριστερά της οθόνης.



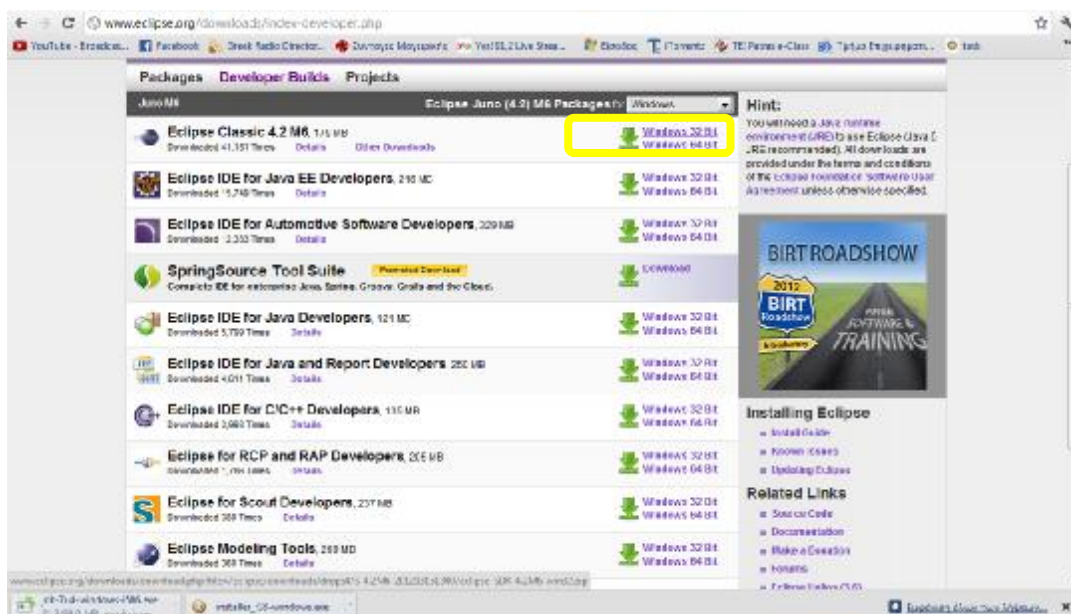
Βήμα 4: Πατάμε δεξί κλικ στον σύνδεσμο που βλέπουμε (<http://www.eclipse.org/downloads/>) και επιλέγουμε να ανοίξει σε νέα καρτέλα. Το ίδιο κάνουμε και για τον σύνδεσμο JDK.



Βήμα 5: Πηγαίνουμε στην δεύτερη καρτέλα που ανοίξαμε (JDK) και πατάμε το Java Platform(JDK) 7u4.



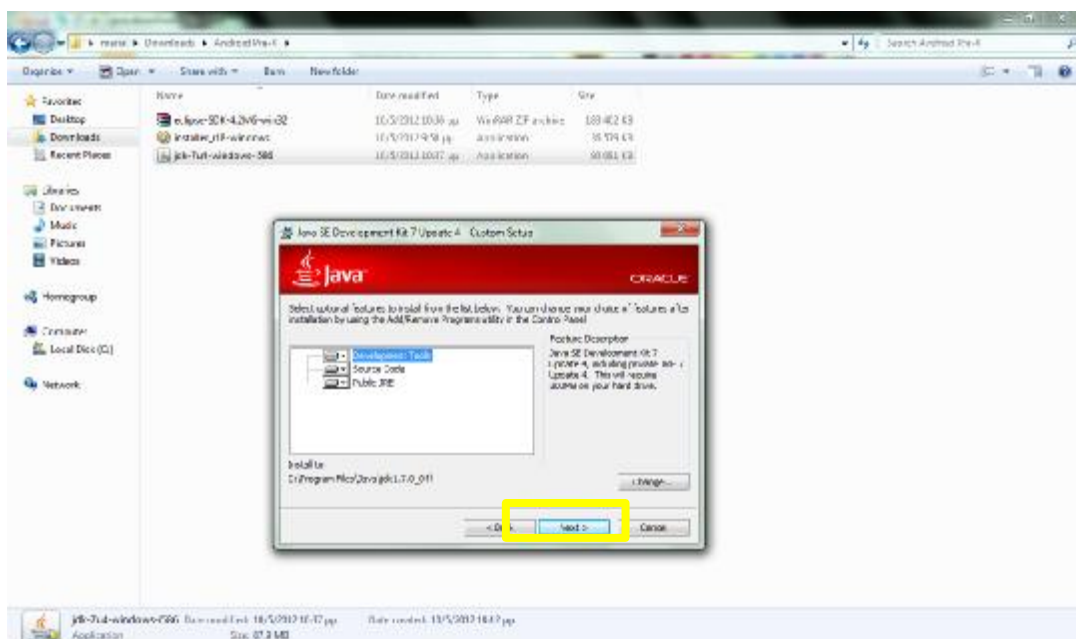
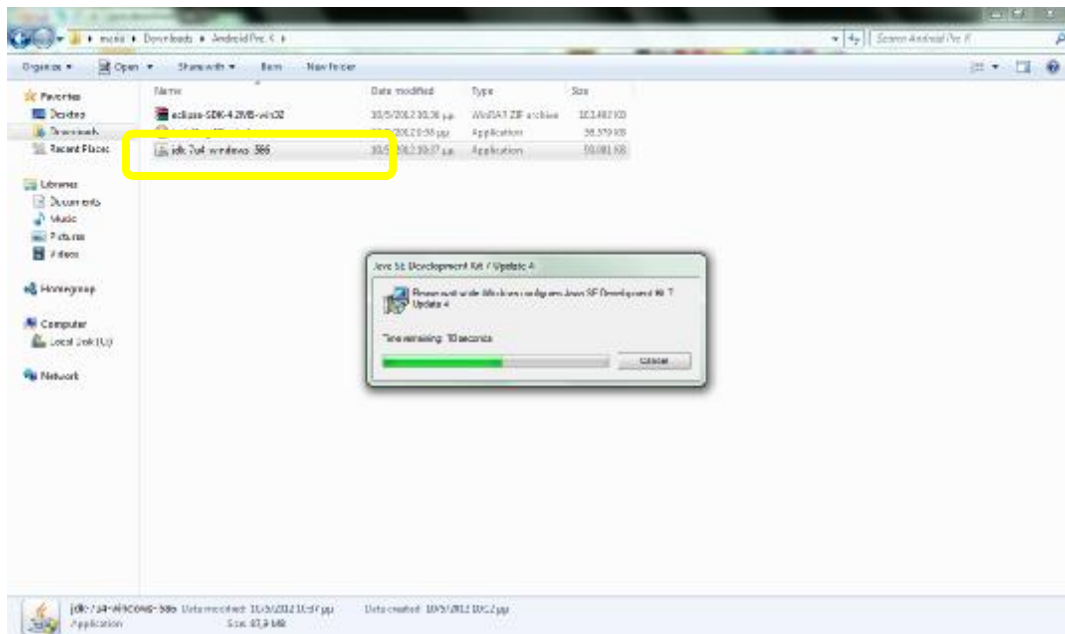
Βήμα 6: Επιλέγουμε το αρχείο για τα Windows x86 (32-bit) και κατεβάζουμε το αρχείο μας. Στην περίπτωση που κάποιος έχει άλλο λειτουργικό σύστημα επιλέγει ανάλογα.



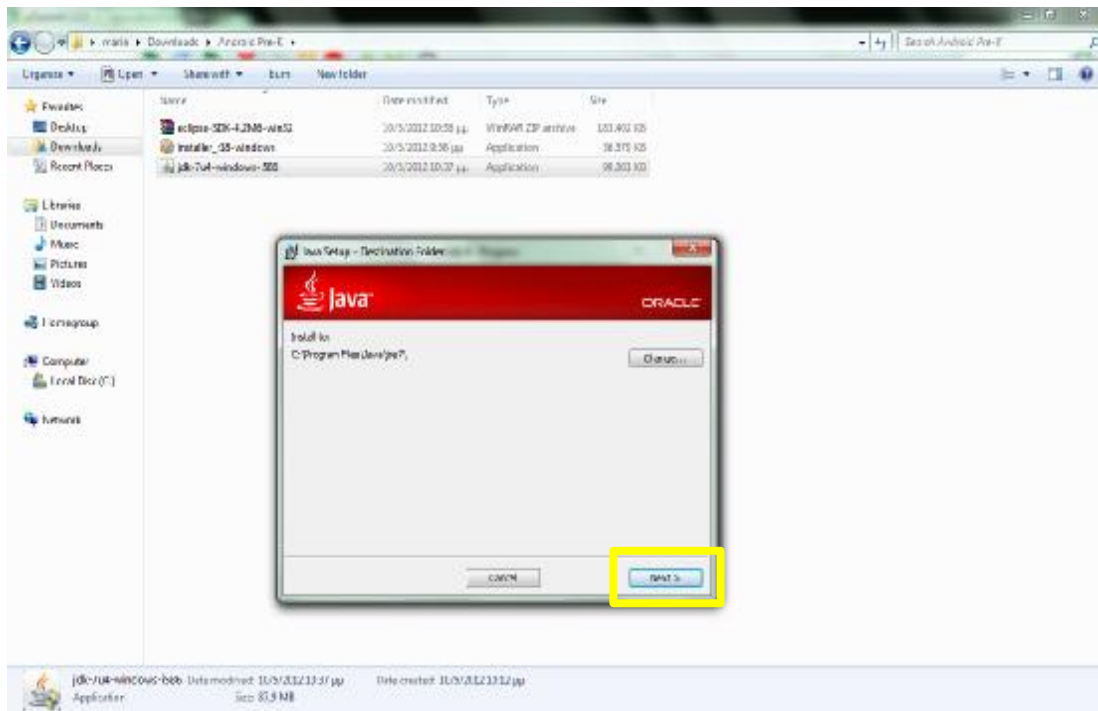
Βήμα 7: Πηγαίνουμε στην πρώτη καρτέλα που είχαμε ανοίξει για το eclipse και επιλέγουμε το Eclipse Classic 4.2 M6 για τα Windows 32 Bit και κατεβάζουμε και το 3^ο αρχείο μας.

6.2 Εγκατάσταση του εργαλείου Java

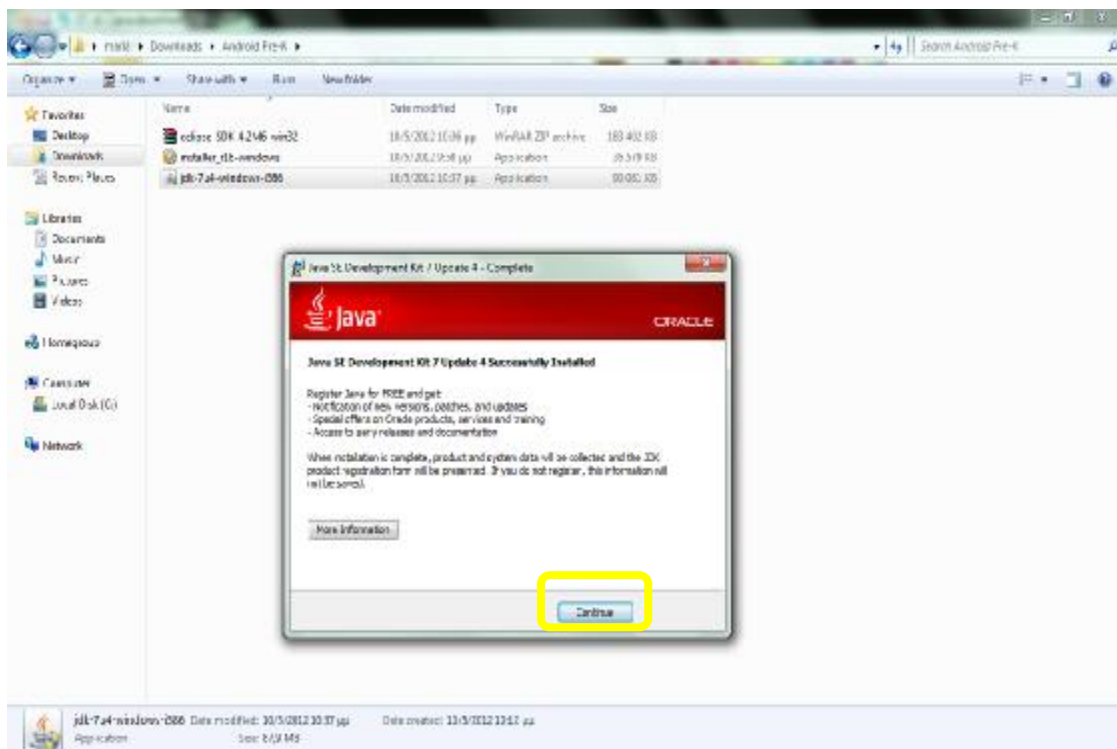
Αφού κατέβουν και τα 3 αρχεία πηγαίνουμε στις λήψεις μας και ανοίγουμε πρώτα το jdk-7u4-windows-i586 για να το εγκαταστήσουμε.



Πατάμε next.



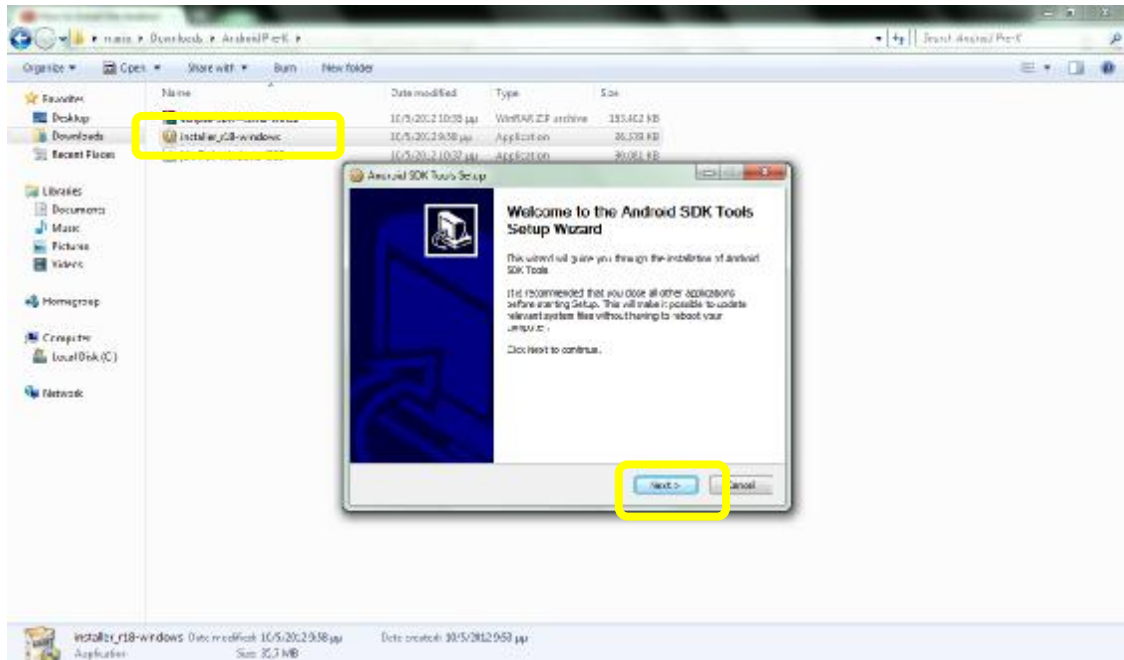
Πατάμε next.



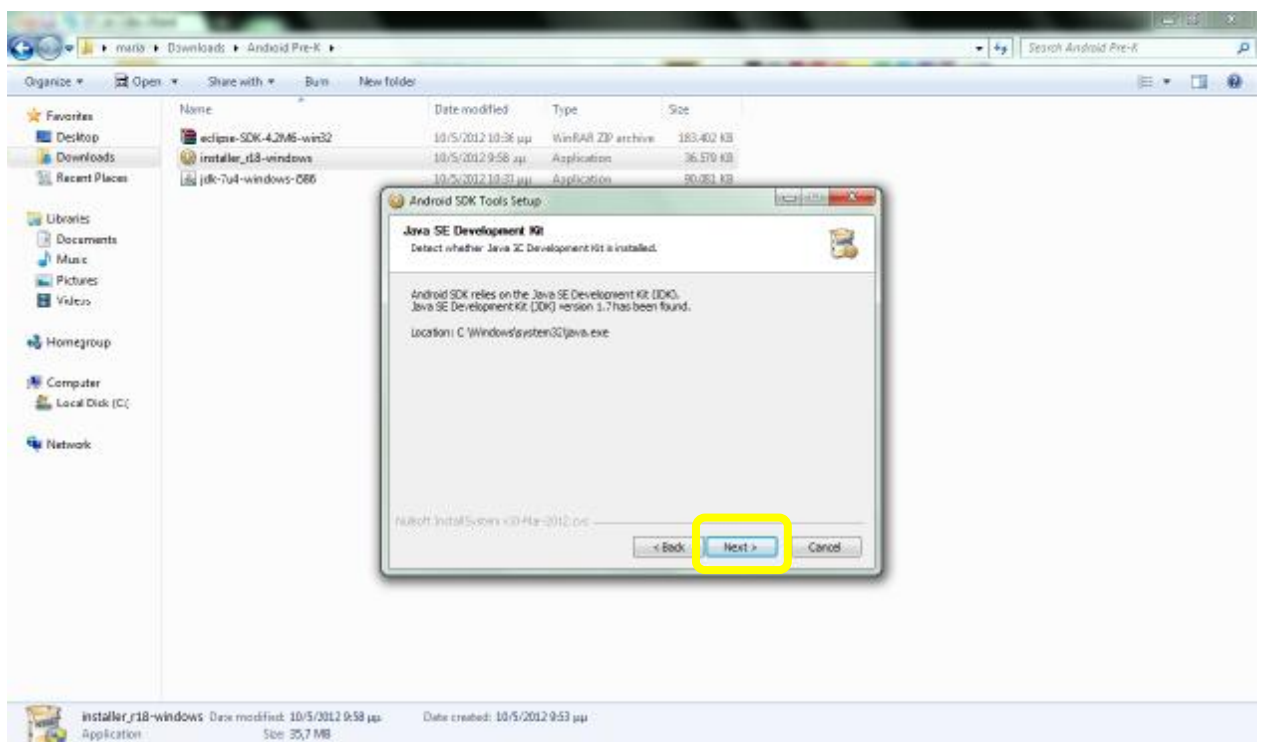
Πατάμε Continue.

6.3 Εγκατάσταση του εργαλείου ANDROID SDK

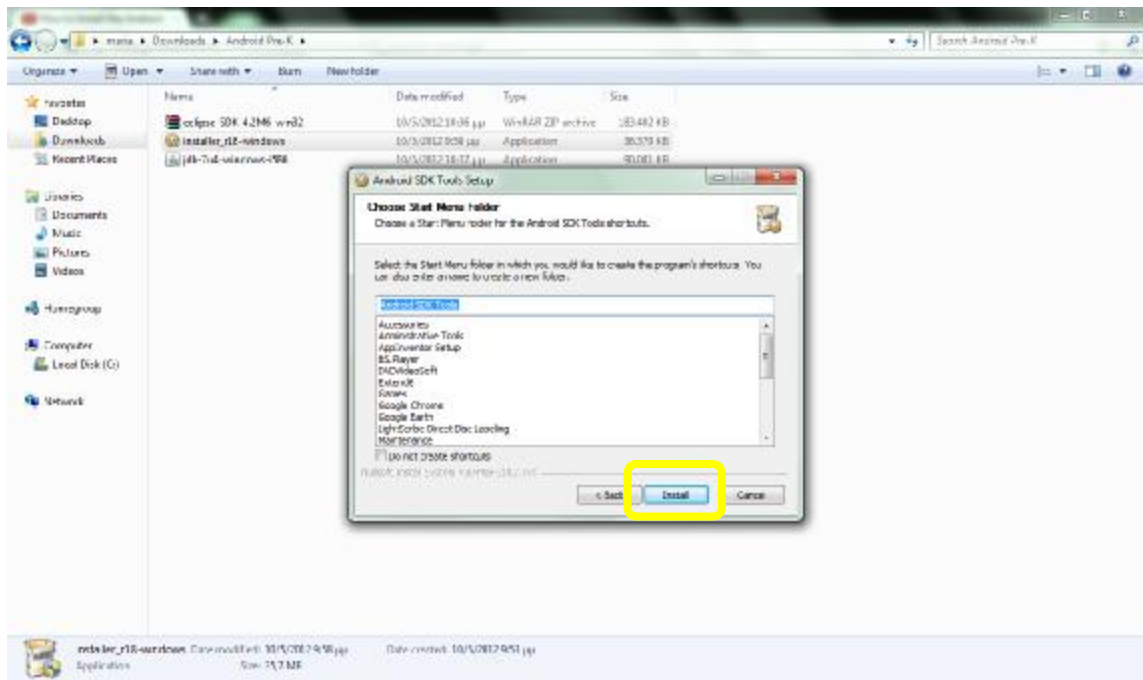
Έπειτα επιλέγουμε να εγκαταστήσουμε το αρχείο installer_r18-windows.



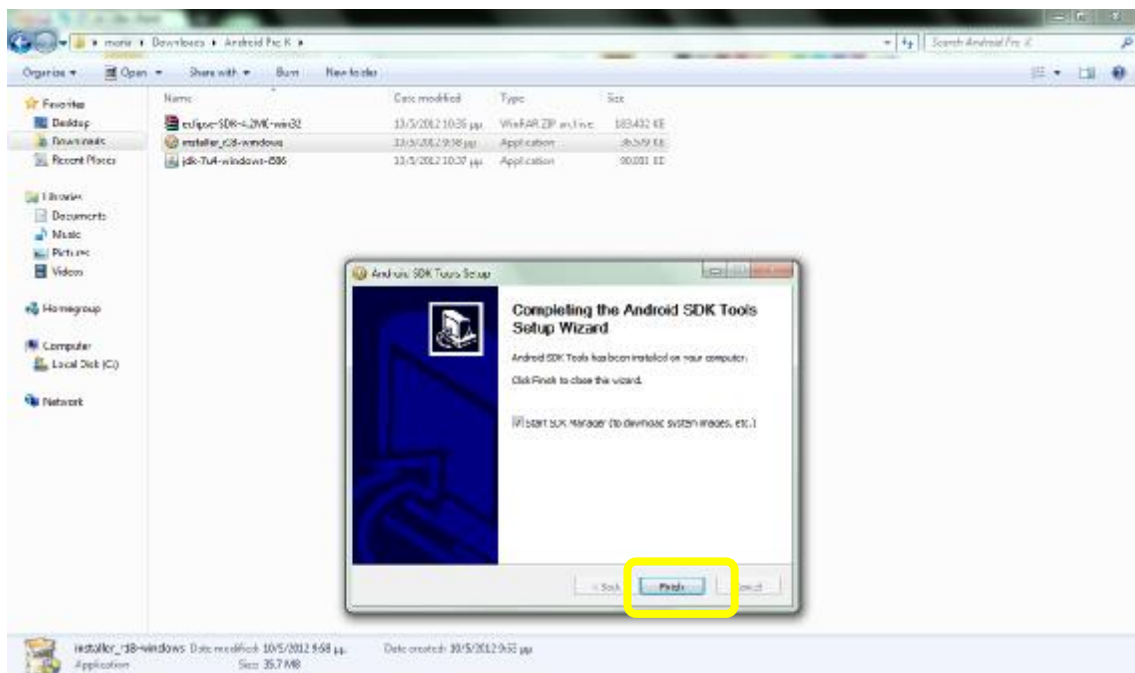
Πατάμε next.



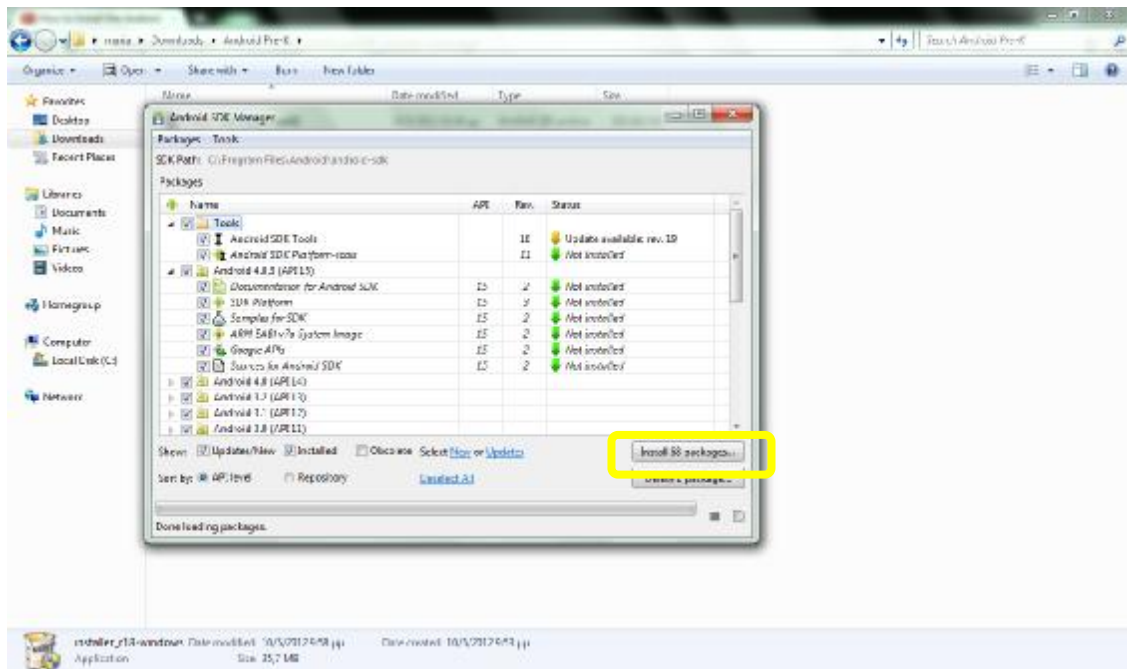
Πατάμε next.



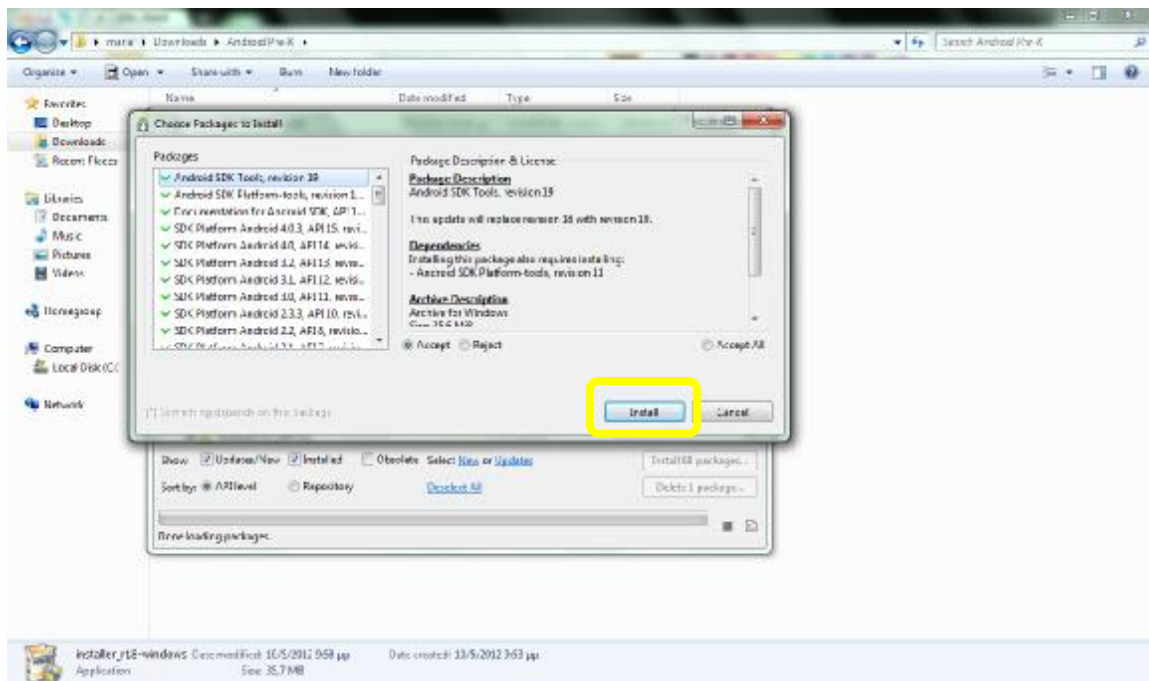
Πατάμε Install.



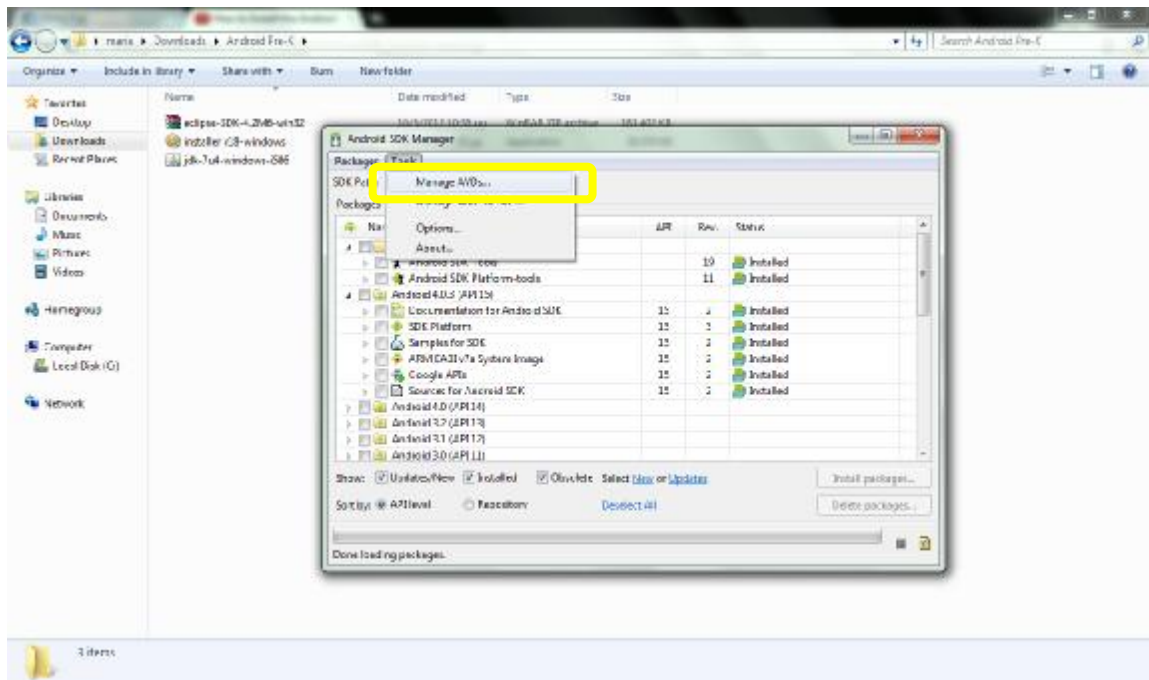
Πατάμε Finish.



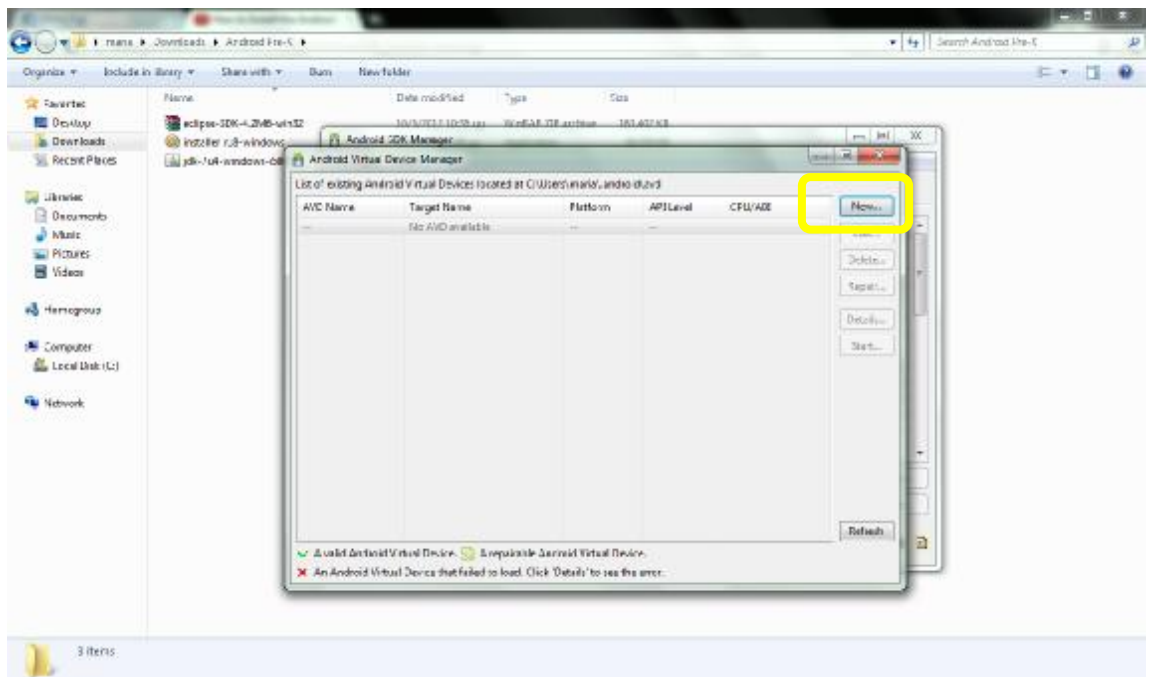
Ανοίγει το Android SDK Manager όπου θα κατεβάσουμε τις πλατφόρμες για κάθε έκδοση Android. Έτσι επιλέγουμε όλα τα αρχεία που εμφανίζονται και πατάμε install 68 packages.



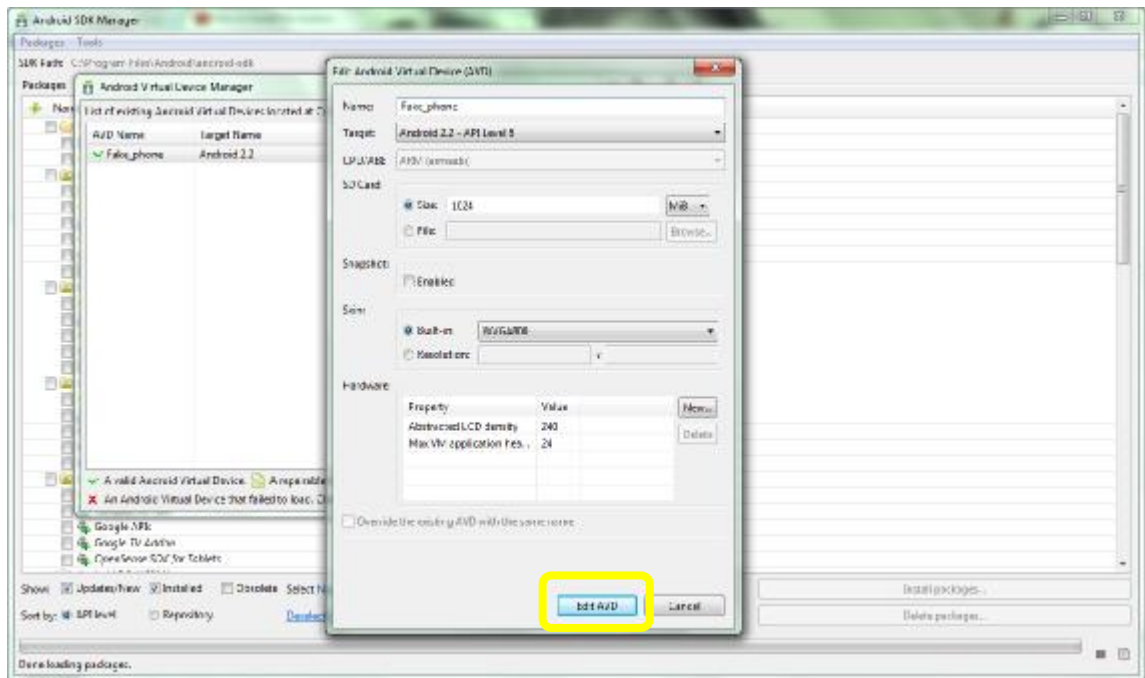
Πατάμε Accept All και μετά Install.



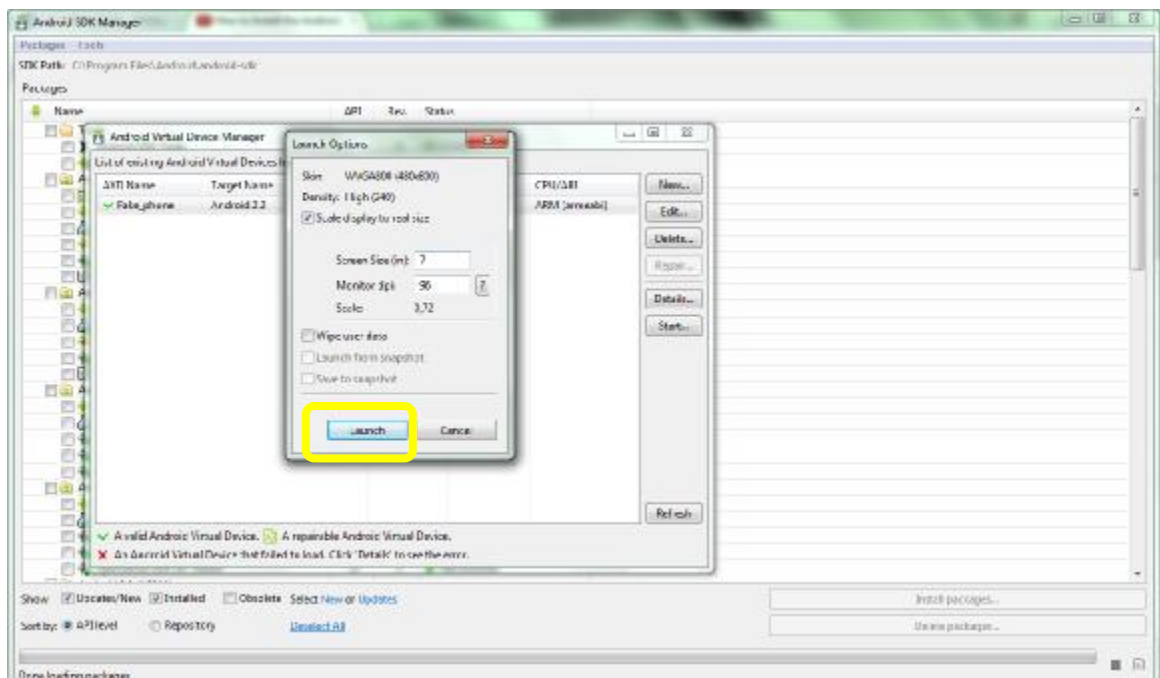
Αφού κατέβουν όλα πηγαίνουμε στην επιλογή Tools à Manage AVDs...



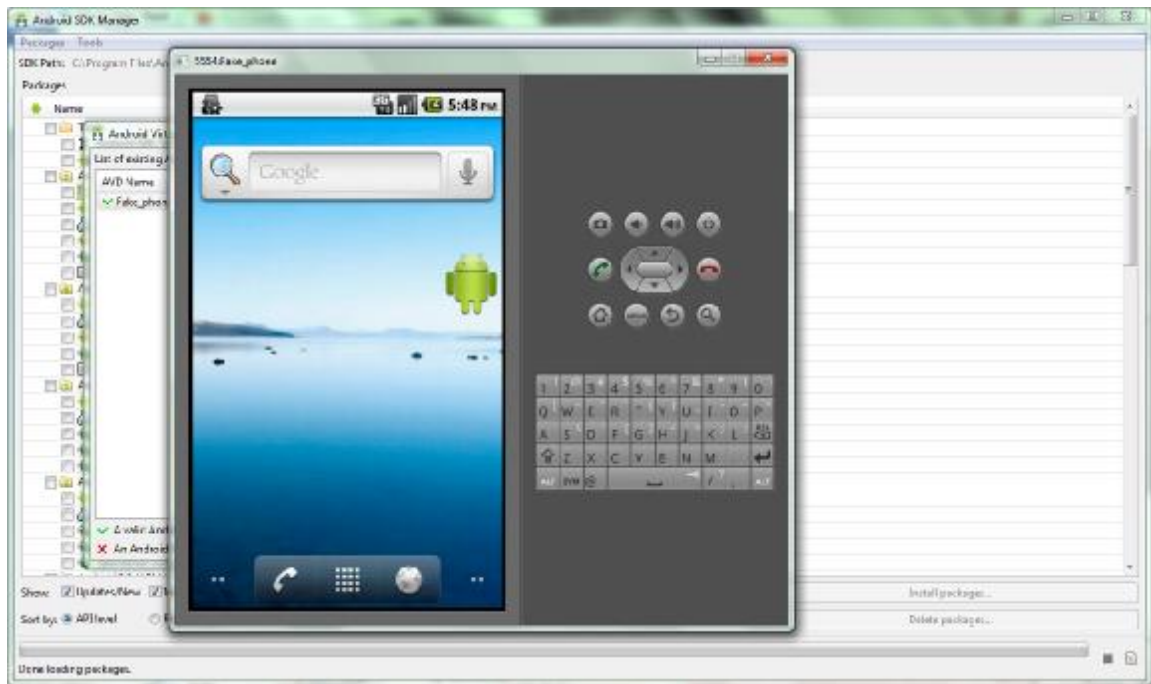
Σε αυτό το στάδιο δημιουργούμε τα εικονικά μας τηλέφωνα για κάθε έκδοση android ξεχωριστά έτσι ώστε να δείξουμε πως εμφανίζονται οι εφαρμογές στην πράξη. Οπότε πατάμε New.. για να ξεκινήσουμε.



Στο Name γράφουμε το όνομα της εικονικής συσκευής, εδώ Fake_phone, στο Target επιλέγουμε την έκδοση android που θα έχει η συσκευή και στο size έχουμε γράψει 1024 για το μέγεθος που θα έχει. Τέλος πατάμε Edit AVD.

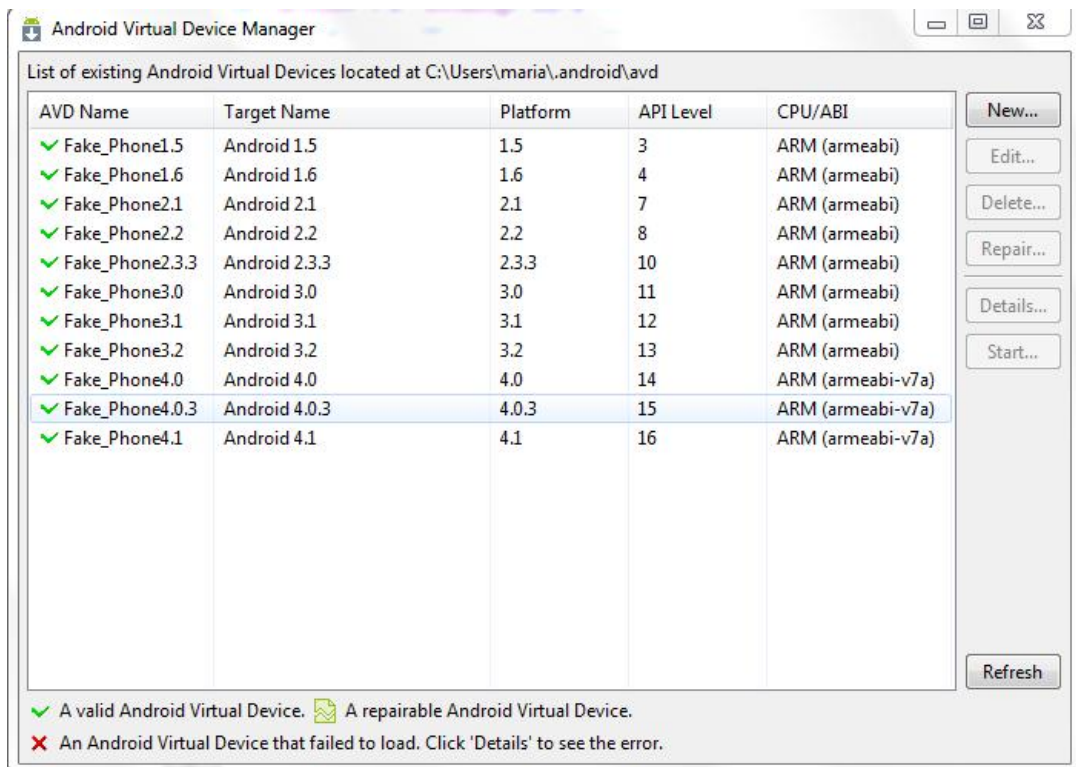


Εάν πατήσουμε Edit ρυθμίζουμε τις διαστάσεις της οθόνης και πατάμε launch για να ξεκινήσει.

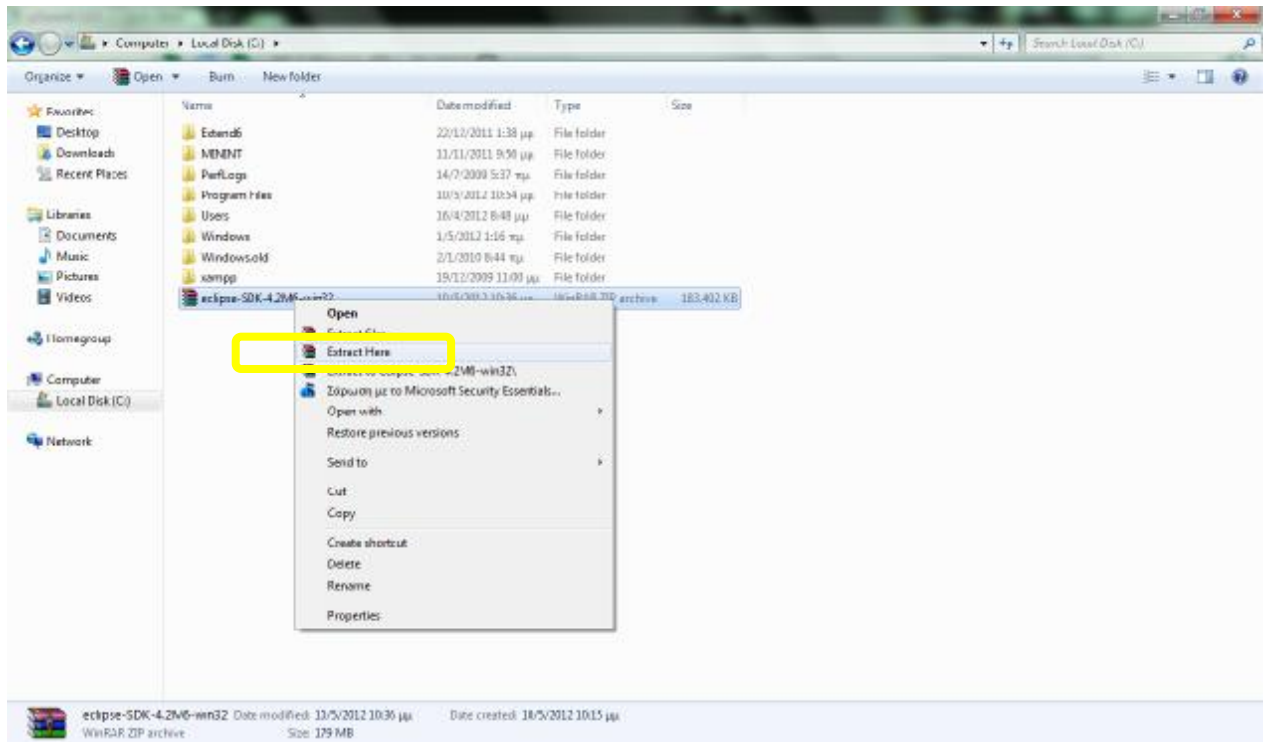


Το εικονικό τηλέφωνο είναι έτοιμο.

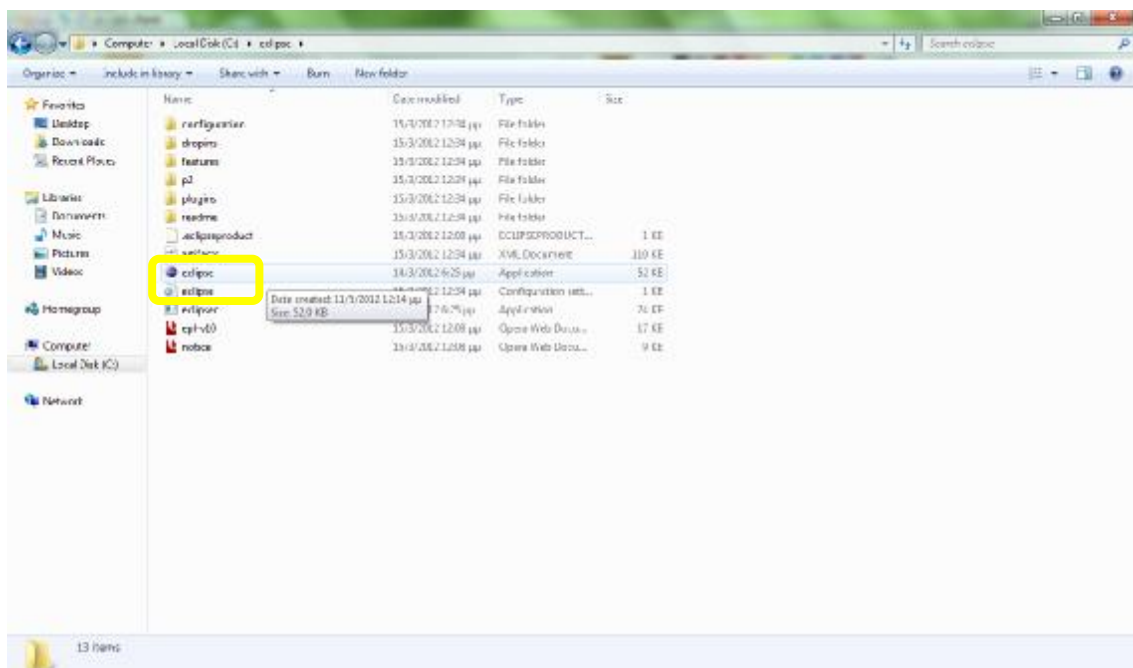
Στη συνέχεια δημιουργούμε και άλλα εικονικά τηλέφωνα για κάθε έκδοση android (στην επιλογή target επιλέγουμε την έκδοση android που θέλουμε και αλλάζουμε το όνομα της κάθε συσκευής) και έχουμε το παρακάτω αποτέλεσμα.



6.4 Εγκατάσταση του εργαλείου ECLIPSE

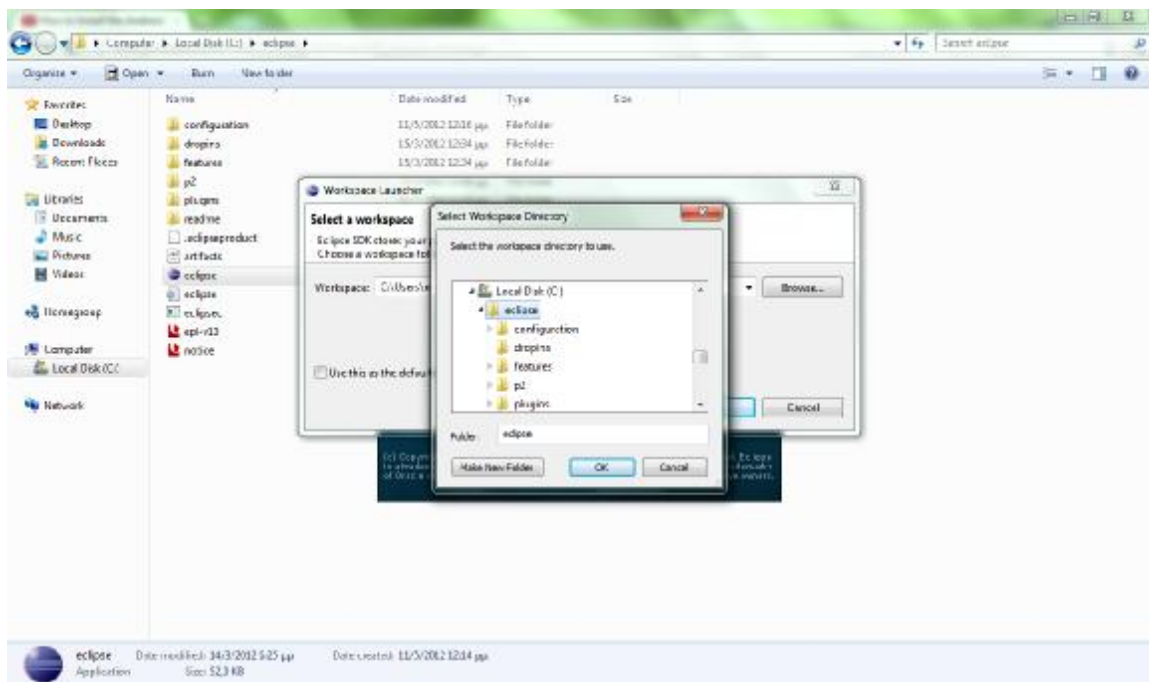


Στην συνέχεια αντιγράφουμε το αρχείο eclipse-SDK-4.2M6-win32 από τις λήψεις μας που είχαμε κατεβάσει και κάνουμε επικόλληση στον δίσκο C: του υπολογιστή μας. Έπειτα, πατάμε δεξί κλικ πάνω του και αποσυμπιέζουμε το αρχείο.

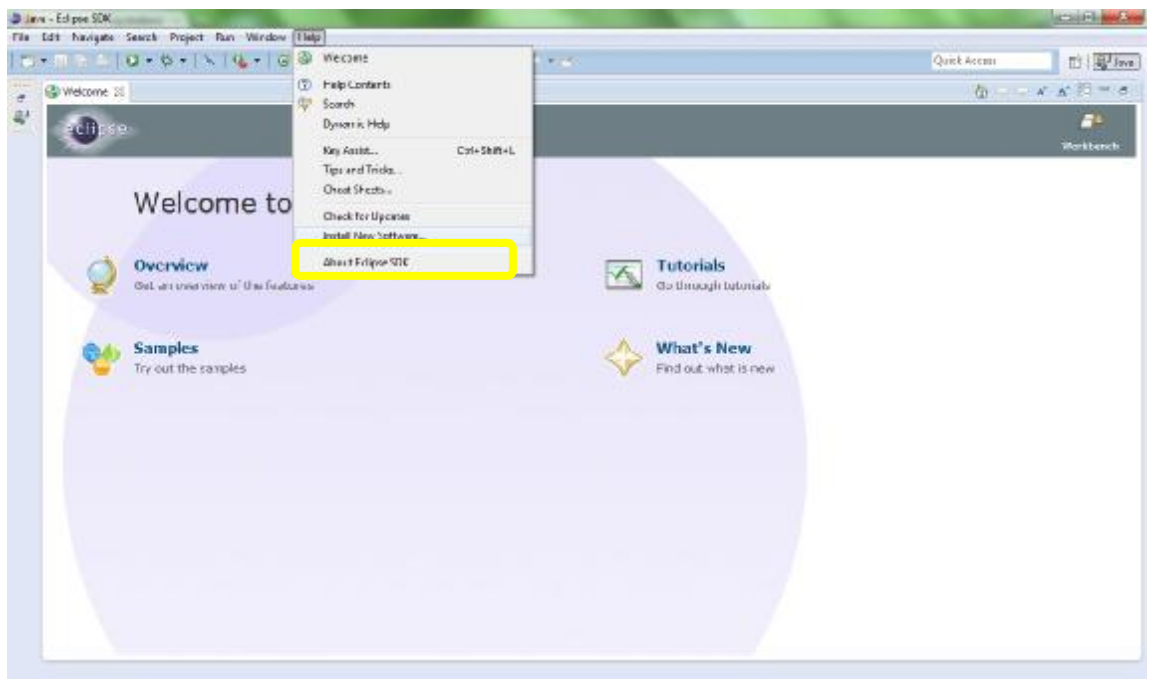


Στον φάκελο που δημιουργήθηκε (αν δεν δημιουργήθηκε φάκελος φτιάχνουμε

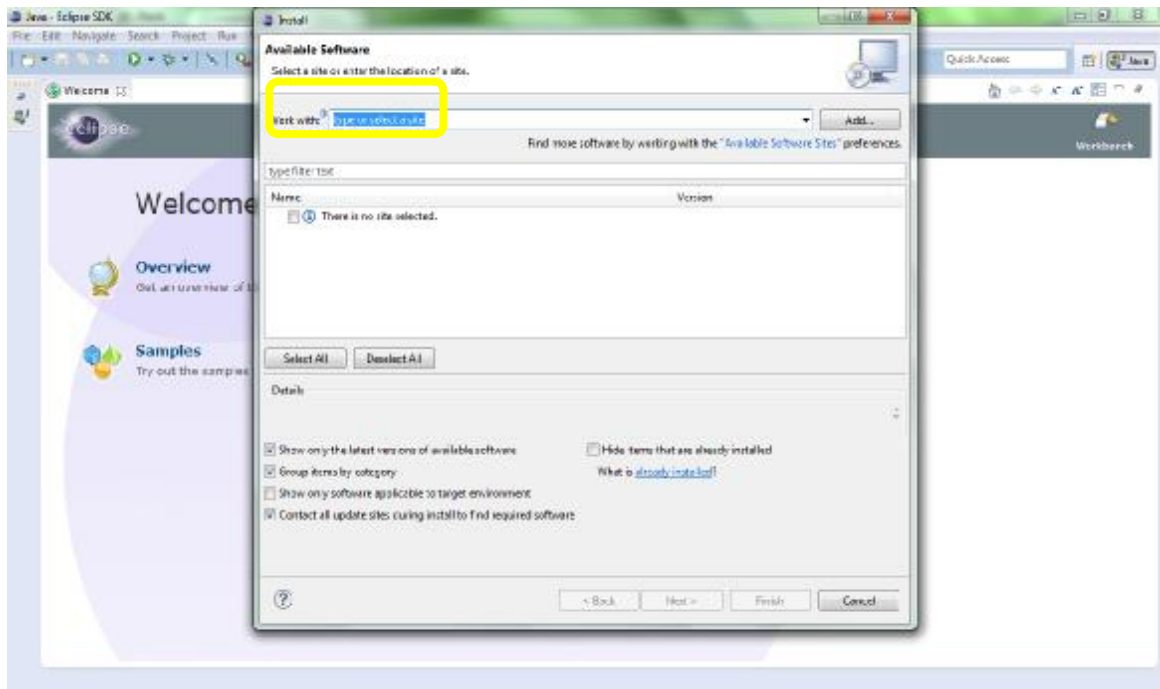
έναν με το όνομα eclipse και βάζουμε όλα τα αρχεία που εμφανίστηκαν για να τα έχουμε συγκεντρωμένα) πατάμε διπλό κλικ στο eclipse.



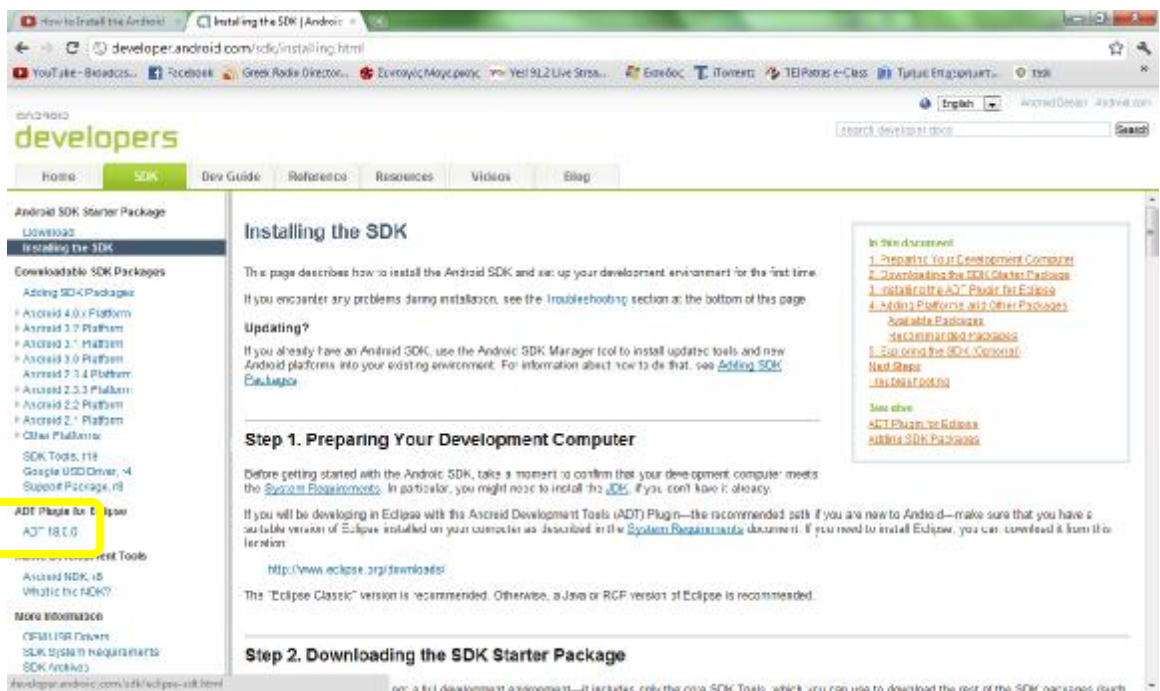
Επιλέγουμε τον φάκελο που θέλουμε να αποθηκεύονται τα αρχεία που θα δημιουργούμε στο eclipse.



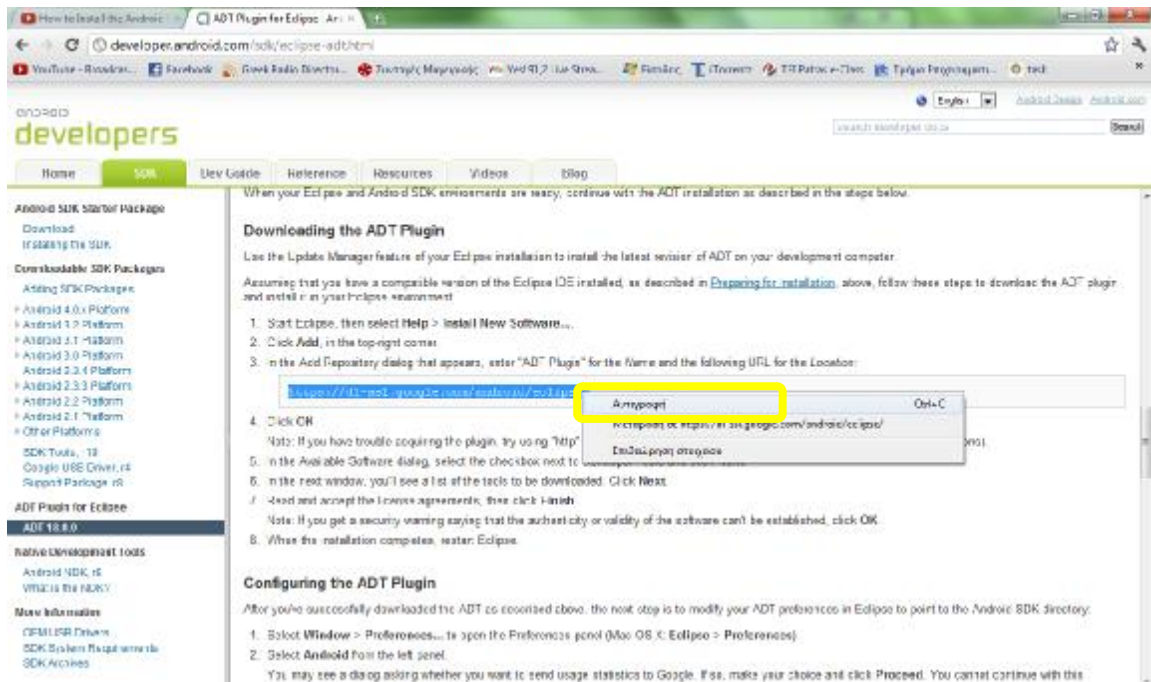
Όταν ανοίξει το eclipse πατάμε Help → Install New Software...



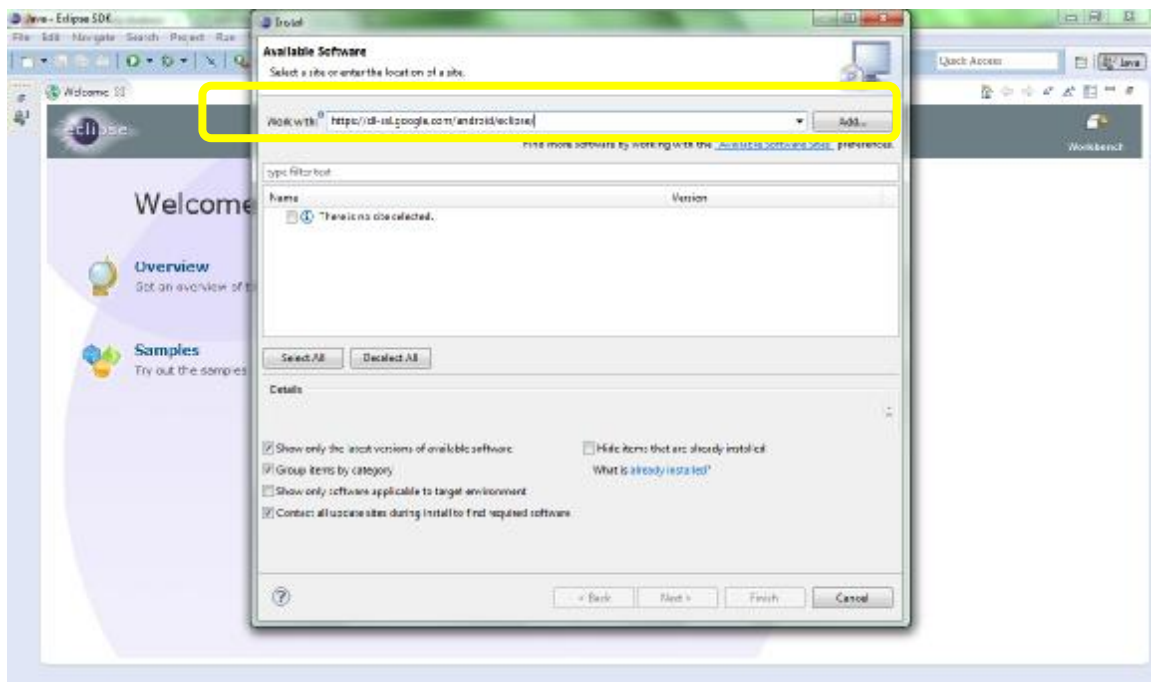
Εμφανίζεται αυτή η καρτέλα όπου θέλει να εισάγουμε μια διεύθυνση.



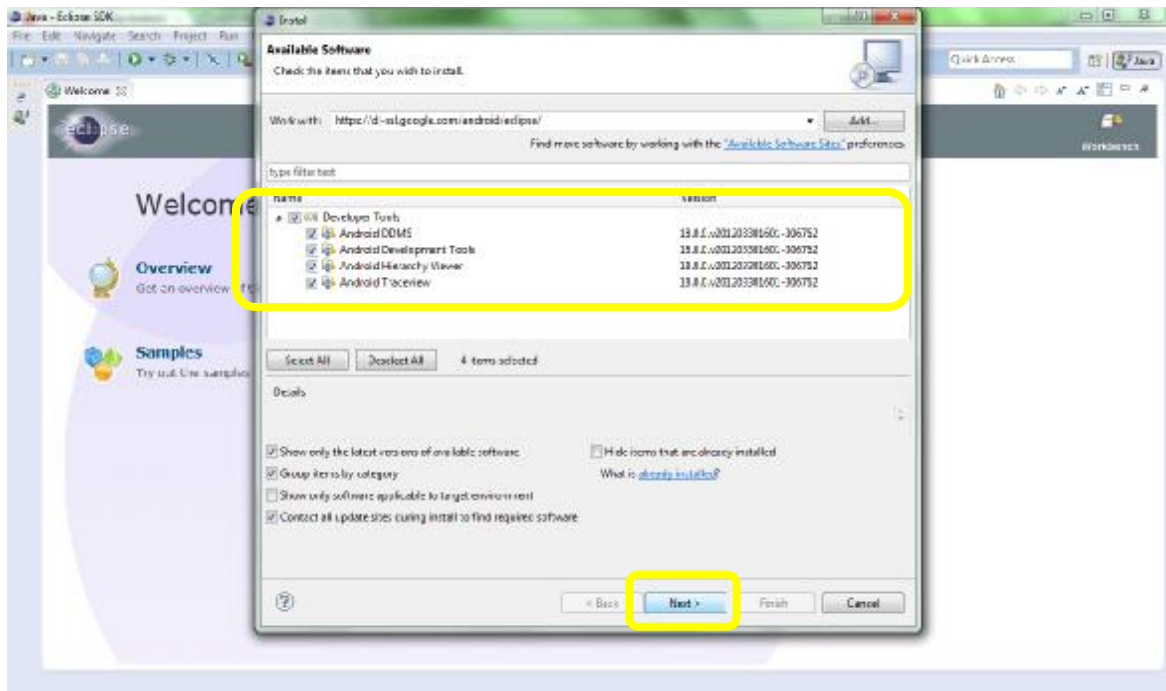
Έτσι πηγαίνουμε στην σελίδα developer.android.com/sdk/installing.html και πατάμε το ADT 18.0.0 που βρίσκεται στα αριστερά της οθόνης.



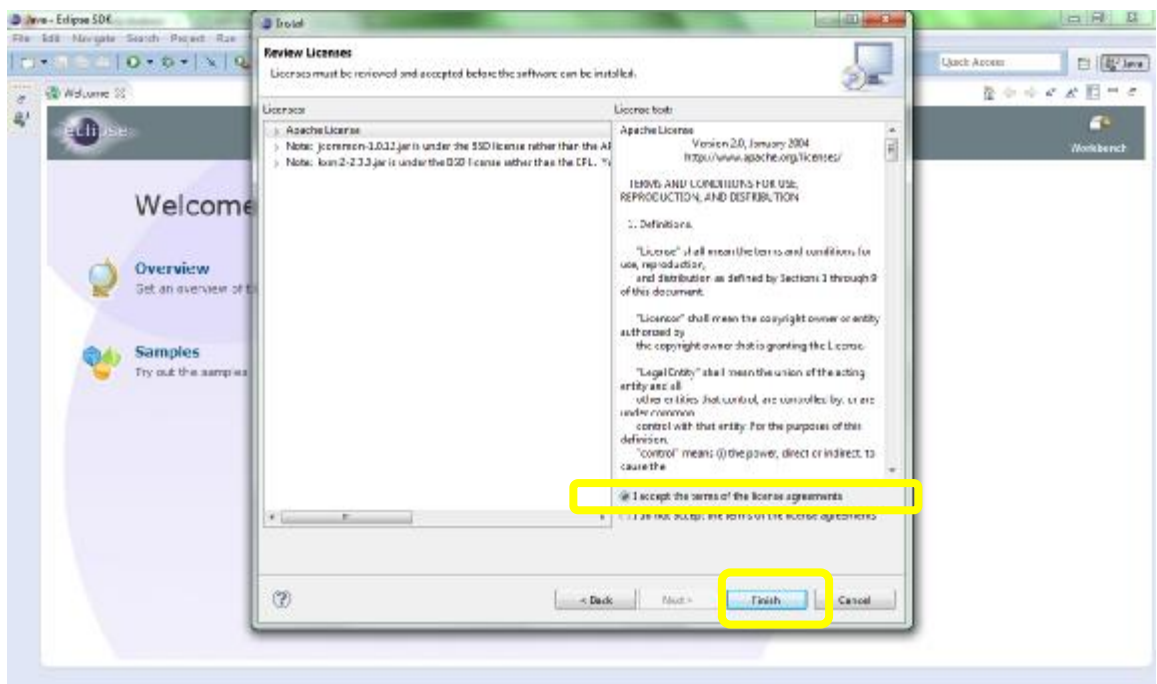
Αντιγράφουμε το url που βλέπουμε



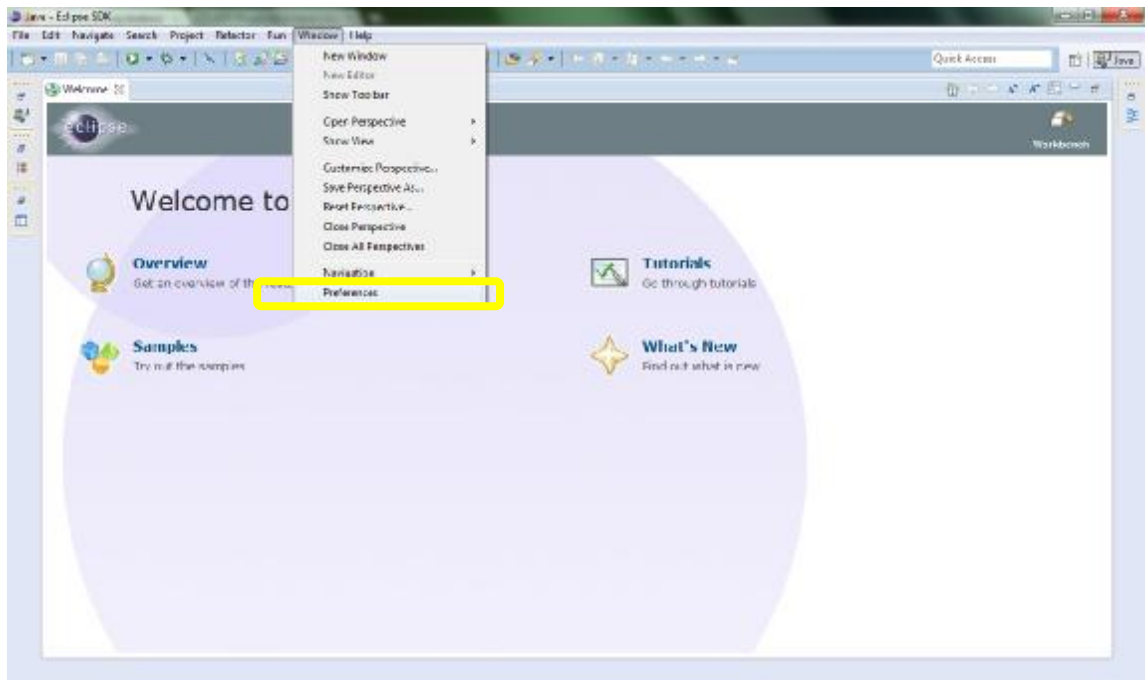
Και κάνουμε επικόλληση στο πεδίο που ζητάγε την διεύθυνση στο eclipse και πατάμε Add.



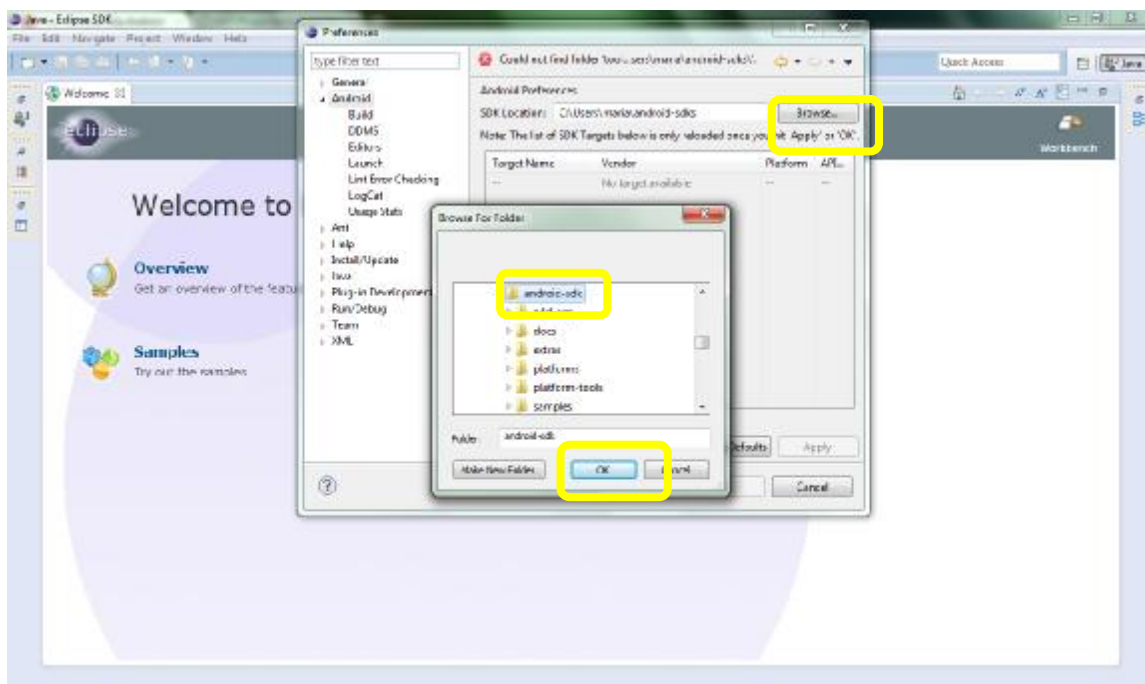
Επιλέγουμε όλα τα αρχεία που μας έχει και μετά next



Επιλέγουμε *I accept the terms of the license agreements* και μετά *Finish*.



Μετά πηγαίνουμε στην επιλογή Window à Preferences

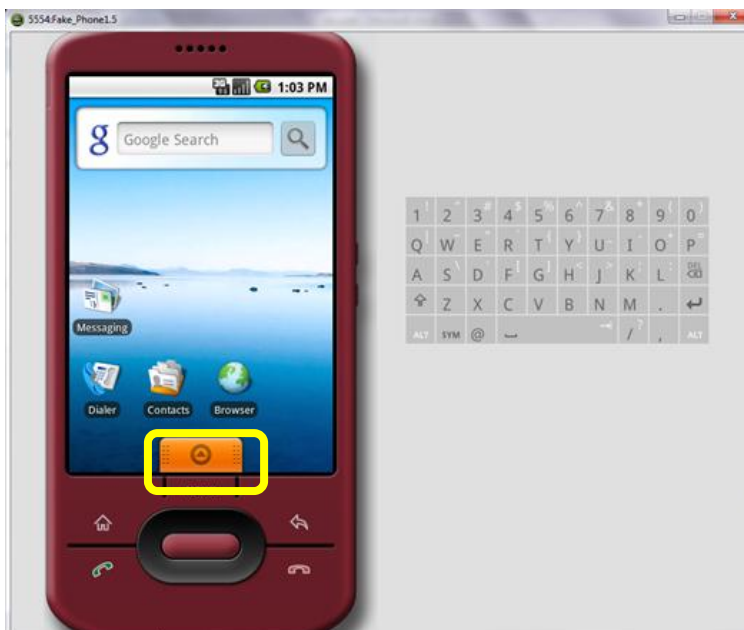


Στην καρτέλα Android πατάμε *Browse* και επιλέγουμε τον φάκελο *android-sdk* και μετά *OK*.

Κεφάλαιο 7 : Περιβάλλον κάθε έκδοσης Android

7.1 Περιβάλλον Android 1.5

Ανοίγοντας την συσκευή του Android 1.5 βλέπουμε την αρχική οθόνη που αποτελείται από την μηχανή αναζήτησης της Google, την εφαρμογή Messaging για την αποστολή μηνυμάτων, την εφαρμογή Dialer όπου πληκτρολογούμε αριθμούς τηλεφώνων, την εφαρμογή Contacts με τις επαφές του κινητού και την εφαρμογή Browser όπου μπορούμε να μπούμε σε όποια σελίδα του διαδικτύου θέλουμε. Για να μπούμε σε όλες τις επιλογές που μας παρέχονται πατάμε το βελάκι που βρίσκεται στο κάτω μέρος της οθόνης.



Όπως βλέπουμε υπάρχουν πολλές εφαρμογές εγκατεστημένες στο κινητό μαζί με την δικιά μας εφαρμογή Note pad.

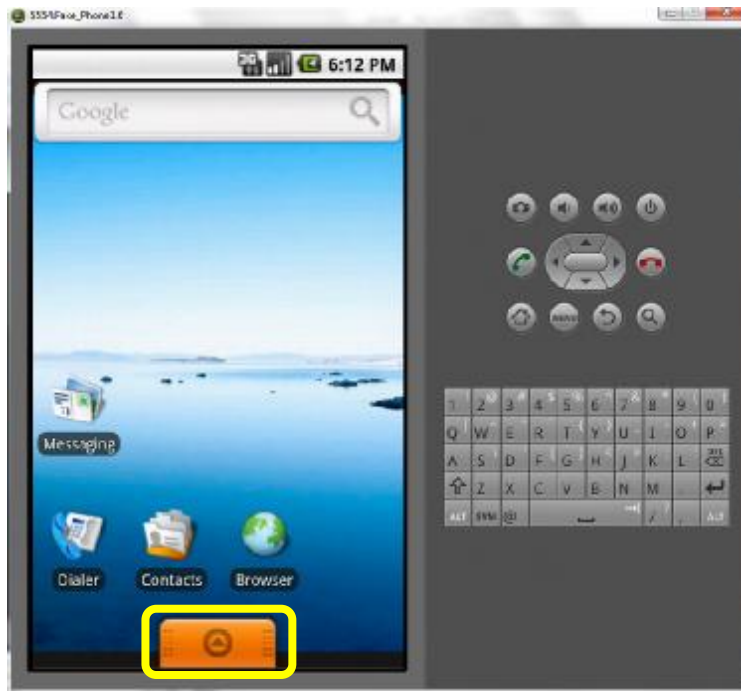


Γυρνώντας στην αρχική οθόνη υπάρχει και η επιλογή MENU όπου μπορούμε να προσθέσουμε εφαρμογές (Add), να αλλάξουμε το φόντο μας (Wallpaper), να αναζητήσουμε κάτι στο κινητό(Search), να δούμε τις ειδοποιήσεις(Notifications) και να πάμε στις ρυθμίσεις(Settings).

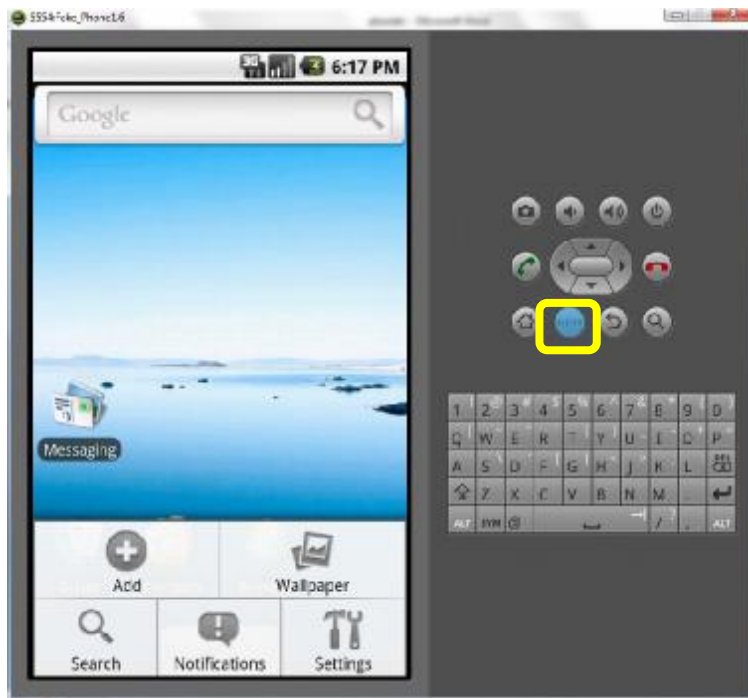


7.2 Περιβάλλον Android 1.6

Πάμε να δούμε τώρα πώς φαίνεται αυτή η έκδοση και να βρούμε πρόσθετα στοιχεία σε σχέση με την προηγούμενη. Όπως βλέπουμε η αρχική οθόνη δεν αλλάζει. Μπαίνουμε στις εφαρμογές πατώντας το βελάκι.



Επίσης από την αρχική οθόνη πατάμε MENU για να δούμε τις επιλογές μας.



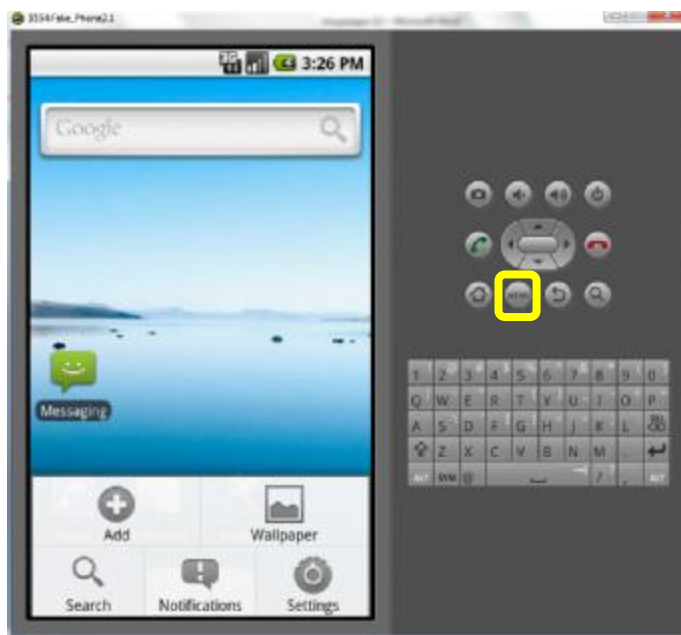
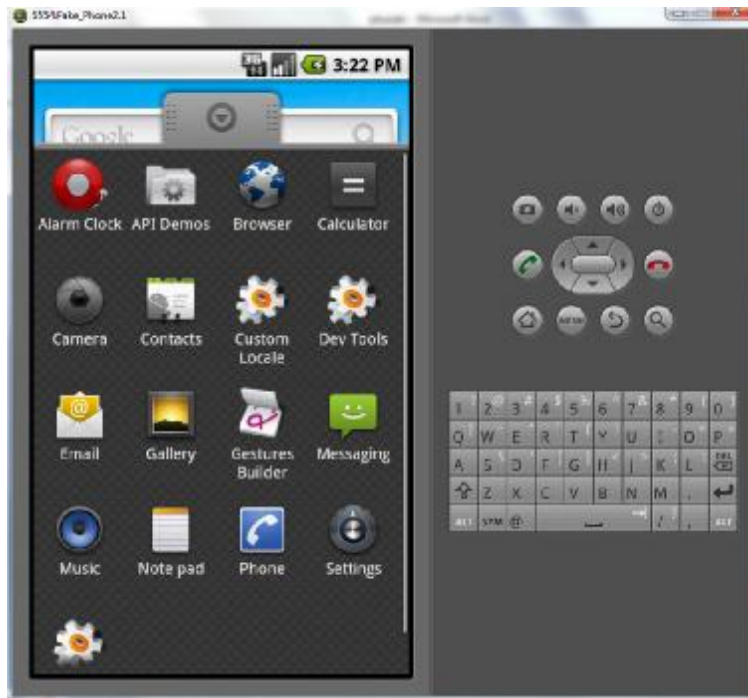
Εδώ έχουμε τις ίδιες επιλογές που είχαμε και στην προηγούμενη έκδοση με λίγο καλύτερη ποιότητα στα γραφικά.

7.3 Περιβάλλον Android 2.1

Το περιβάλλον σε αυτή την έκδοση δεν αλλάζει πολύ από το προηγούμενο. Η αρχική οθόνη έχει τις ίδιες εφαρμογές με διαφορετικές εικόνες. Πατάμε το βελάκι για να μπούμε σε όλες τις εφαρμογές του τηλεφώνου.



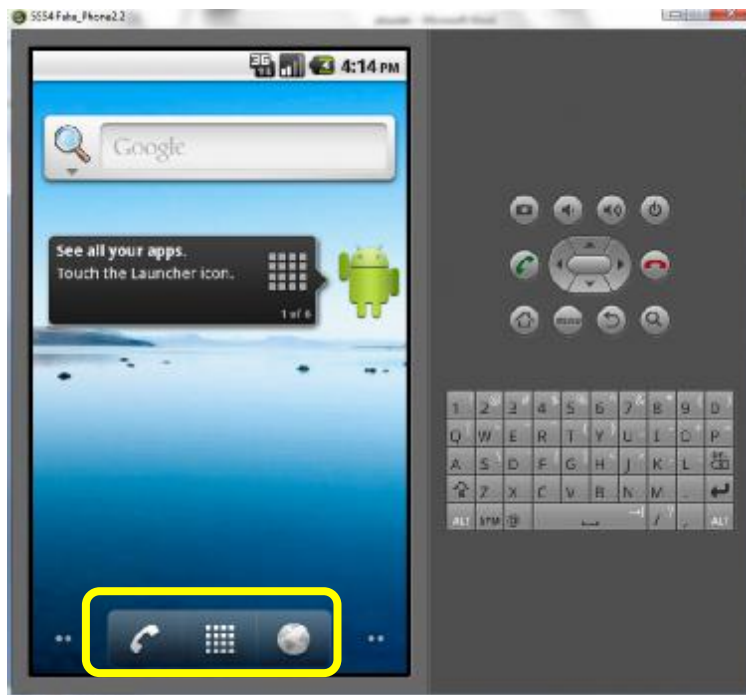
Βλέπουμε ότι δεν έχει αλλάξει κάτι εκτός από τα γραφικά στοιχεία.



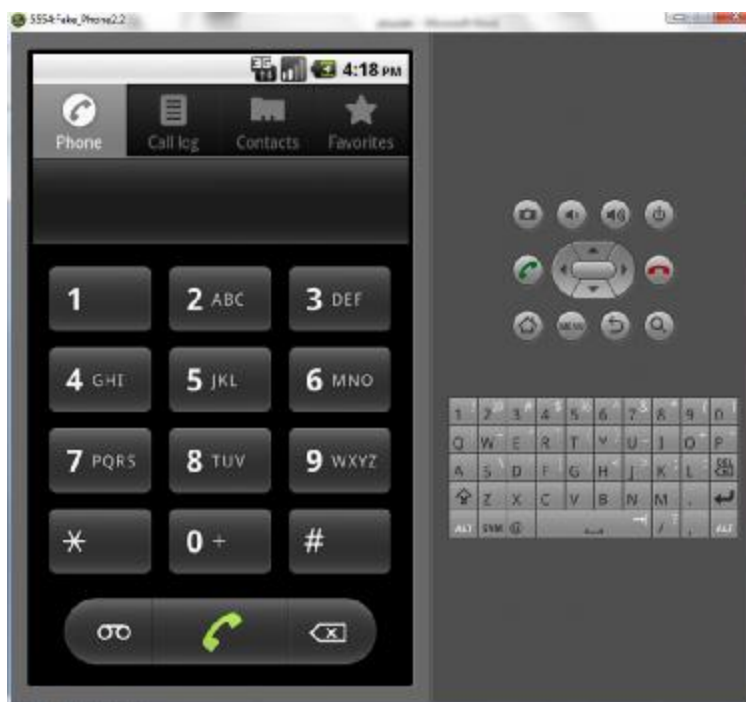
Και το μενού παραμένει το ίδιο.

7.4 Περιβάλλον Android 2.2

Το περιβάλλον σε αυτή την έκδοση έχει αλλάξει όπως φαίνεται στην παρακάτω εικόνα. Οι εφαρμογές που βρίσκονταν στην αρχική οθόνη (Dialer, Browser) βρίσκονται τώρα μαζί στο μενού επιλογών.



Το πληκτρολόγιο έχει αλλάξει όσον αφορά τα γραφικά του στοιχεία.

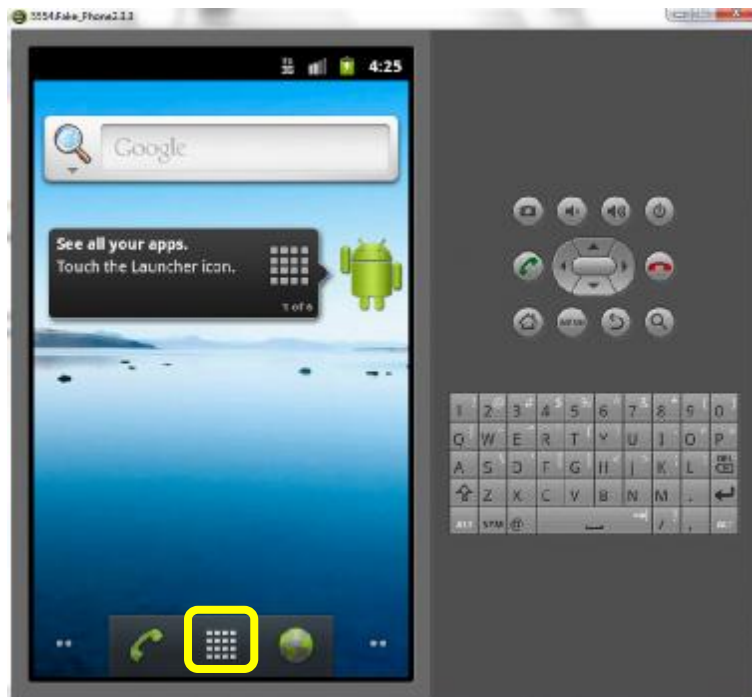


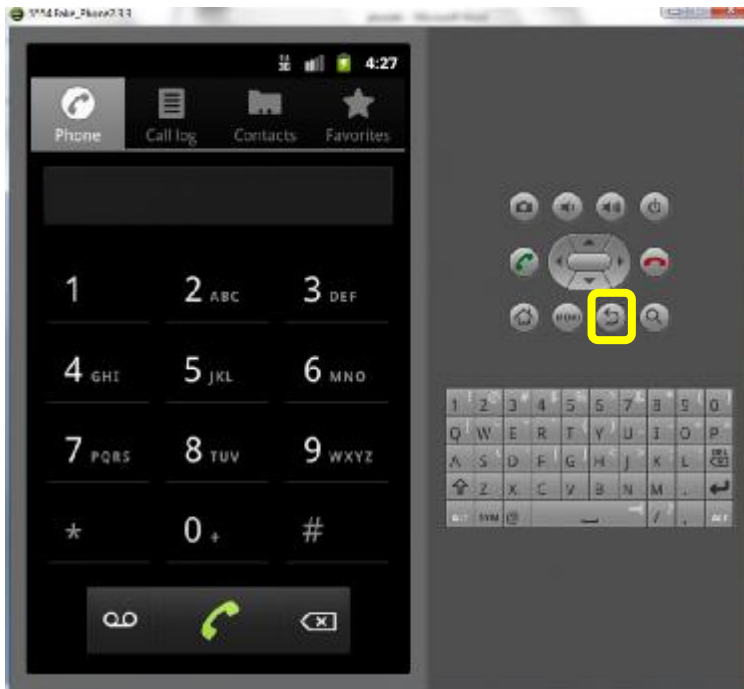
Το μενού των εφαρμογών είναι σχεδόν ίδιο με τα προηγούμενα.



7.5 Περιβάλλον Android 2.3.3

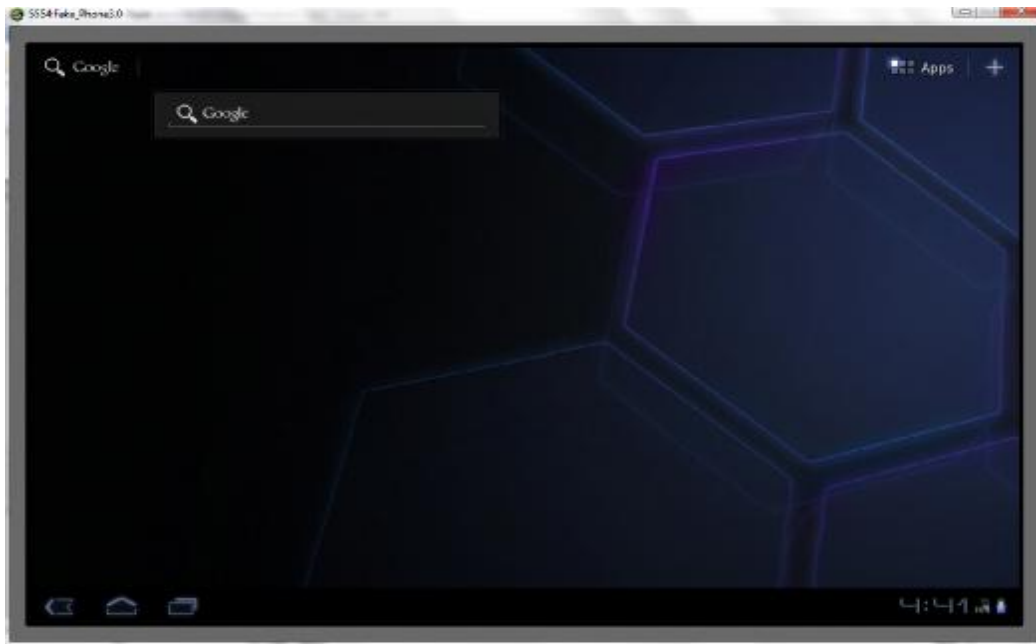
Στην έκδοση αυτή έχουν αλλάξει λίγο τα γραφικά όσον αφορά τα χρώματα. Επίσης παρατηρούμε πιο κάτω ότι έχει αλλάξει και η μορφή του πληκτρολογίου.





7.6 Περιβάλλον Android 3.0, 3.1, 3.2

Όπως βλέπουμε στις εικόνες το περιβάλλον των εκδόσεων αυτών είναι αρκετά διαφορετικό από τα προηγούμενα γιατί αφορά tablets κυρίως. Το μενού των εφαρμογών βρίσκεται πάνω δεξιά [Apps] και η προσθήκη εφαρμογών στην αρχική οθόνη βρίσκεται δίπλα με την μορφή [+]. Πατώντας το [Apps] βλέπουμε όλες τις εγκατεστημένες εφαρμογές.

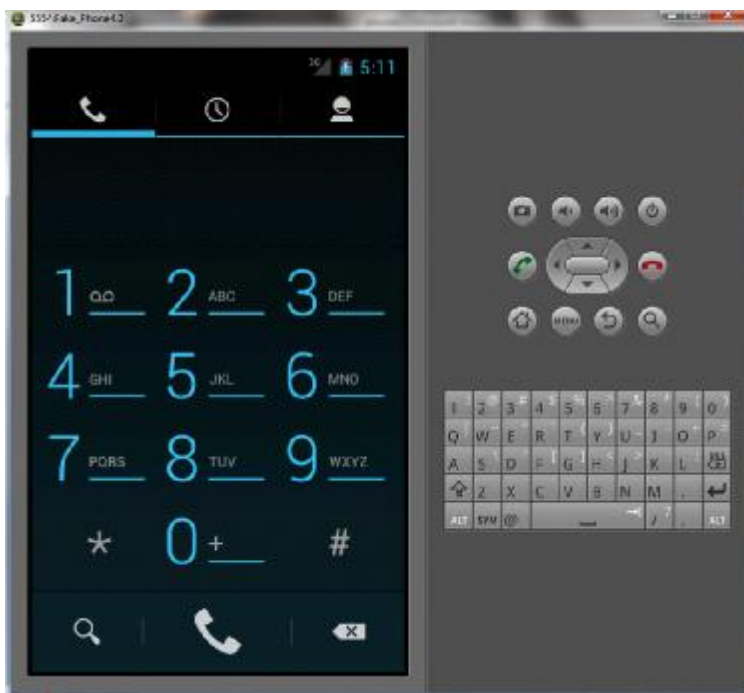


7.7 Περιβάλλον Android 4.0, 4.0.3

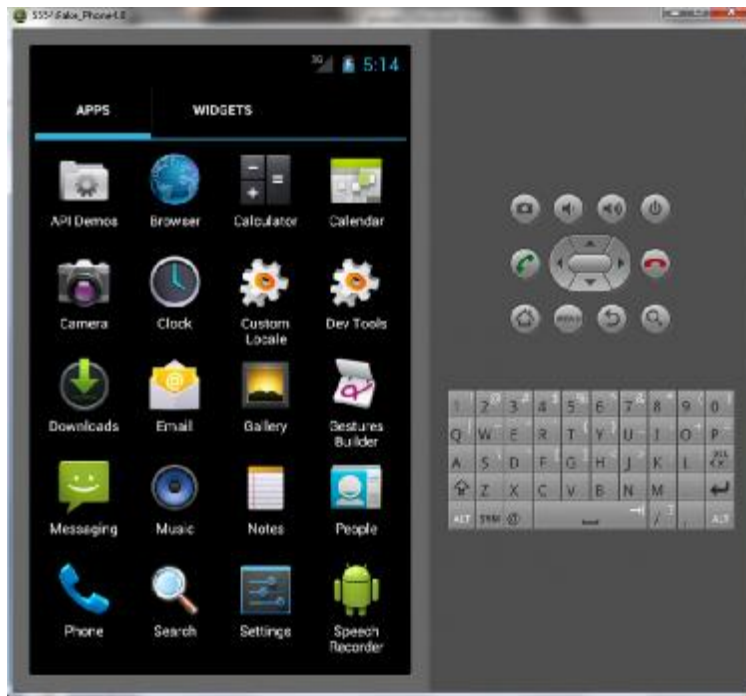
Εδώ το περιβάλλον έχει αλλάξει πολύ σε σχέση με προηγούμενες εκδόσεις. Έχουμε πιο ωραία γραφικά στοιχεία και οι βασικές εφαρμογές όπως dialer, μηνύματα, επαφές και browser βρίσκονται στο κάτω μέρος της οθόνης μαζί με το μενού.



Το πληκτρολόγιο έχει αλλάξει και σε αυτήν την έκδοση



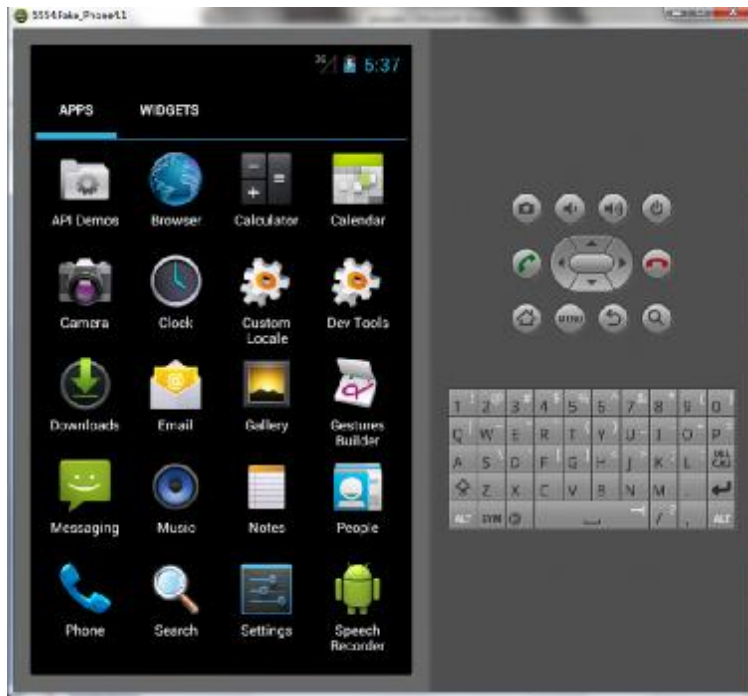
Το μενού εφαρμογών είναι παρόμοιο με το μενού των εκδόσεων 3.0, 3.1, 3.2



7.8 Περιβάλλον Android 4.1

Από ότι διακρίνουμε η έκδοση αυτή είναι ίδια όσον αφορά την εμφάνιση με τις 4.0, 4.0.3.

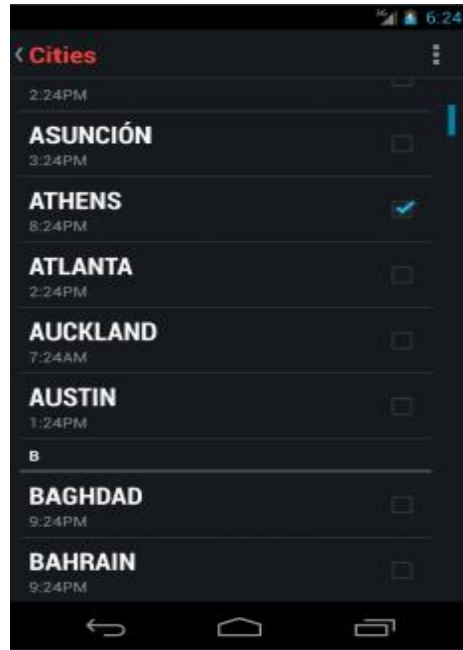




7.9 Περιβάλλον Android 4.2, 4.3

Στην έκδοση 4.2, και 4.3 δεν έχουν αλλάξει πολλά πράγματα, παρακάτω σας δείχνουμε ενδεικτικά την εφαρμογή του ρολογιού που μας δείχνει ώρες από πόλεις σε όλον τον κόσμο και πιο κάτω δείχνουμε το μενού με τις πρόσφατες εφαρμογές που έχουμε ανοίξει όπου από εκεί μπορούμε να πάμε αυτόματα στην εφαρμογή που ήμασταν ή να την διακόψουμε έτσι ώστε να μην λειτουργεί από πίσω. Αυτό μπορούμε να το κάνουμε σέρνοντας την εφαρμογή που θέλουμε να διακόψουμε προς τα δεξιά.





7.10 Περιβάλλον Android 4.4

Στην έκδοση αυτή δεν έχουμε κάτι καινούριο να δείξουμε που έχει αλλάξει από την προηγούμενη έκδοση.



Κεφάλαιο 8 Ανάλυση στοιχείων Android

Όταν θα δημιουργήσουμε ένα νέο project χρησιμοποιώντας τα εργαλεία ανάπτυξης του Android (development tools), θα δημιουργηθούν για μας τα βασικά αρχεία και οι φάκελοι που θα δούμε στη περιοχή Package Explorer του Eclipse. Κάποια από τα αρχεία και τους φακέλους που θα δημιουργηθούν εξαρτώνται από το αν χρησιμοποιήθηκε το plug in του Eclipse ή το Android tools για τη δημιουργία του project. Επίσης, όσο η εφαρμογή που προγραμματίζουμε αρχίζει και γίνεται πιο πολύπλοκη μπορεί να χρειαστεί να δημιουργήσουμε νέα είδη πόρων, καταλόγων και αρχείων. Παρακάτω θα εξηγήσουμε αναλυτικότερα τους φακέλους και τα αρχεία αυτά.

Φάκελος src

Ο φάκελος αυτός προέρχεται από την λέξη source και περιέχει όλα τα αρχεία τύπου . Java ή . Aidl .

Φάκελος gen

Στο φάκελο αυτό περιέχονται τα αρχεία που παράγονται από το ADT (Android Development Tools) όπως το αρχείο R.java και οι διάφορες διασυνδέσεις που δημιουργούνται από τα AIDL αρχεία. Τα αρχεία αυτά αυτό δημιουργούνται και προσθέτουν από μόνα τους τα στοιχεία που χρειάζονται και είναι καλό να μη τα επεξεργαζόμαστε.

Φάκελος res

Θα πρέπει πάντα να εξωτερικεύουμε τους πόρους, όπως εικόνες και συμβολοσειρές από τον κώδικα της εφαρμογής μας, ώστε να μπορούμε να τους διατηρήσουμε ανεξάρτητα. Εξωτερικεύοντας τους πόρους μας, μας επιτρέπει επίσης να παρέχουμε εναλλακτικές πηγές που υποστηρίζουν συγκεκριμένες ρυθμίσεις της συσκευής, όπως διαφορετικές γλώσσες ή μεγέθη οθόνης, η οποία γίνεται όλο και πιο σημαντική, καθώς όλες οι συσκευές Android είναι διαθέσιμες με διαφορετικές συνθέσεις. Προκειμένου να παρέχουμε συμβατότητα με διαφορετικές συνθέσεις, θα πρέπει να οργανώσουμε τους πόρους στον κατάλογο RES / του έργου μας, χρησιμοποιώντας διάφορους υπό-καταλόγους ανάλογα τον τύπο και τη διαμόρφωση. Περιέχει τους πόρους της εφαρμογής, όπως σχεδιάσιμα

αρχεία, αρχεία διάταξης, καθώς και τις τιμές συμβολοσειράς. Πιο αναλυτικά περιέχει τους φακέλους:

- Anim, που περιέχει αρχεία xml που αναφέρονται σε αντικείμενα κινουμένων σχεδίων.
- Color, που περιέχει αρχεία xml που αναφέρονται σε λίστες περιγραφής χρωμάτων.
- Drawable, που περιέχει αρχεία bitmap (png, jpeg ή gif), αρχεία εικόνας 9-Patch και αρχεία xml που περιγράφουν σχεδιάσιμα σχήματα ή σχεδιάσιμα αντικείμενα. Έχει να κάνει με τα γραφικά και τις κινήσεις καρέ-καρέ. Ο φάκελος αυτός υπάρχει τέσσερις φορές μέσα στο res αλλά με διαφορετική ονομασία (drawable-hdpi, drawable-mdpi, drawable-ldpi, drawable-xhdpi). Αυτό συμβαίνει γιατί δε μπορεί η ίδια εικόνα παραδείγματος χάρη να χρησιμοποιηθεί σε όλων των ειδών τις οθόνες γιατί σύμφωνα με την ανάλυση της οθόνης η εικόνα θα φαίνεται μικρότερη ή μεγαλύτερη. Για να το αποφύγουμε αυτό λοιπόν, δημιουργούμε για κάθε οθόνη την αντίστοιχη εικόνα και τη τοποθετούμε στον αντίστοιχο φάκελο.
- Layout, περιέχει xml αρχεία που αναφέρονται στο τρόπο διάταξης της οθόνης. Καλό θα είναι να δημιουργήσουμε και άλλους layout φακέλους (layout-small, layout-long, layout-large) και να σχεδιάσουμε το καθένα ξεχωριστά έτσι ώστε η εφαρμογή μας να είναι ομοιόμορφη σε όλα τα μεγέθη οθονών.
- Menu, περιέχει αρχεία xml που έχουν να κάνουν με το μενού της εφαρμογής μας.
- Raw, περιέχει ακατέργαστα αρχεία όπως και στο φάκελο assets. Η διαφορά εδώ είναι ο τρόπος με τον οποίο έχουμε πρόσβαση σε αυτά. Για παράδειγμα, αυτό είναι ένα καλό μέρος για τα αρχεία τύπου mp3, mp4, txt, xml, κ.α.
- Values, περιέχει αρχεία xml που έχουν να κάνουν με: τιμές boolean, χρώματα, διαστάσεις, ακέραιους, συστοιχίες ακεραίων, συστοιχίες μεικτού τύπου, απλά στοιχεία που μπορούν να σχεδιαστούν, συμβολοσειρές, συστοιχία συμβολοσειρών, πληθυντικός συμβολοσειρών, στυλ και θέματα.

- ο Xml, περιέχει διάφορα αρχεία xml που διαμορφώνουν τα στοιχεία της εφαρμογής.

Βιβλιοθήκες

Στη συνέχεια παρατηρούμε δύο βιβλιοθήκες (Android και Android Dependencies) που περιέχουν αρχεία με τη κατάληξη .jar δηλαδή αρχεία Java. Τα αρχεία αυτά ορίζουν ένα μορφότυπο συνάθροισης πολλών αρχείων σε ένα, και χρησιμοποιούνται για την διανομή κλάσεων Java.

Φάκελος assets

Ακολουθεί ο φάκελος assets ο οποίος αρχικά είναι άδειος. Εδώ μπορούμε να αποθηκεύσουμε αρχεία και καταλόγους της εφαρμογής τους οποίους όμως δε θέλουμε να διαχειριστούμε σαν πόρους της εφαρμογής. Παραδείγματος χάρι, στο φάκελο αυτό μπορούμε να δημιουργήσουμε ένα άλλο φάκελο με το όνομα Fonts και να τοποθετήσουμε μέσα αρχεία γραμματοσειρών που θα χρησιμοποιήσουμε στην εφαρμογή μας.

Φάκελος bin

Μέσα στο φάκελο bin βρίσκουμε το τελικό αρχείο .apk που θα χρειαστούμε για να εκτελέσουμε την εφαρμογή μας σε κάποια συσκευή Android, καθώς επίσης και άλλους εκτελέσιμους πόρους.

Φάκελος libs

Στο φάκελο αυτό βρίσκονται οι ιδιωτικές βιβλιοθήκες άμα υπάρχουν.

Φάκελος jni

Ο φάκελος αυτός περιέχει εγγενή κώδικα που αναπτύχθηκε με τη χρήση του NDK (Native Development Kit).

Αρχείο AndroidManifest.xml

Κάθε εφαρμογή πρέπει να έχει ένα αρχείο AndroidManifest.xml (ακριβώς με αυτό το όνομα) στον αρχικό κατάλογο (/root). Το αρχείο αυτό, δημιουργείται από μόνο του κατά τη δημιουργία του project και είναι απαραίτητο αρχείο της εφαρμογής γιατί παρέχει όλες τις πληροφορίες που χρειάζεται το σύστημα του Android για να

τρέξει οποιοδήποτε κομμάτι του κώδικα. Πιο αναλυτικά το αρχείο αυτό κάνει τα εξής:

- Ονομάζει το πακέτο Java της εφαρμογής ώστε να μπορεί το πακέτο να έχει το όνομα ως ένα μοναδικό αναγνωριστικό για την εφαρμογή.
- Περιγράφει τα στοιχεία της εφαρμογής – τις δραστηριότητες (activities), τις υπηρεσίες (services), τους δέκτες (broadcast receivers) και τους παρόχους περιεχομένου από τους οποίους αποτελείται η εφαρμογή. Κατονομάζει τις κλάσεις που υλοποιεί καθένα από τα στοιχεία και δημοσιεύει τις ικανότητες του. Για παράδειγμα, ποια μηνύματα προθέσεων (Intents messages) μπορούν να χειριστούν. Αυτές οι δηλώσεις επιτρέπουν στο σύστημα Android να γνωρίζει ποιά είναι τα στοιχεία αυτά και υπό ποιες συνθήκες μπορούν να ενεργοποιηθούν.
- Καθορίζει ποιες διαδικασίες θα φιλοξενήσουν τα στοιχεία της εφαρμογής.
- Δηλώνει τι δικαιώματα πρέπει να έχει η εφαρμογή για να αποκτήσει πρόσβαση σε προστατευόμενες περιοχές της API και να μπορεί να αλληλεπιδράσει με άλλες εφαρμογές.
- Επίσης, δηλώνει τα δικαιώματα που πρέπει να έχουν οι άλλοι για να αλληλεπιδρούν με τα στοιχεία της εφαρμογής.
- Όσο η εφαρμογή εκτελείται, παραθέτει τις Instrumentation κλάσεις οι οποίες παρέχουν το προφίλ και άλλες πληροφορίες. Αυτές οι δηλώσεις εμφανίζονται μόνο κατά τη διάρκεια που η εφαρμογή είναι υπό ανάπτυξη και δοκιμή, και αφαιρούνται πριν την δημοσίευση της εφαρμογής.
- Δηλώνει το ελάχιστο επίπεδο API που απαιτεί η εφαρμογή και,
- Παραθέτει τις βιβλιοθήκες με τις οποίες πρέπει να συνδέεται η εφαρμογή.

Αρχείο project.properties

Το αρχείο αυτό περιέχει τις ρυθμίσεις του project και είναι αναπόσπαστο κομμάτι του.

Κεφάλαιο 9: Σύγκριση μιας εφαρμογής στις διάφορες εκδόσεις Android

Στο σημείο αυτό θα πάρουμε μια εφαρμογή από τα παραδείγματα του eclipse για να δείξουμε στην πράξη ποιες είναι οι διαφορές που έχουν οι εκδόσεις των android μεταξύ τους. Διαλέξαμε μια εφαρμογή απλή την οποία θα μπορούμε να δείξουμε σε κάθε έκδοση. Αυτή η εφαρμογή είναι ένα απλό σημειωματάριο στο οποίο δημιουργούμε και επεξεργαζόμαστε σημειώσεις. Παρακάτω θα δείξουμε ένα κομμάτι κώδικα και την υλοποίηση της εφαρμογής για κάθε έκδοση έτσι ώστε να επισημάνουμε τις διαφορές και στα δύο μέρη της εφαρμογής.

9.1 Android 1.5 (Cupcake)

Βασικός κώδικας της εφαρμογής

- ο *Αρχείο NoteEditor.java*

Παρακάτω εμφανίζεται ένα μέρος του κώδικα στο οποίο έχουμε την εισαγωγή και επεξεργασία της σημείωσης. Ελέγχονται οι περιπτώσεις: ο χρήστης να έχει επιλέξει την επεξεργασία ή την εισαγωγή σημειώματος, αν το σημείωμα είναι κενό τότε να εμφανίζεται μήνυμα ότι είχαμε αποτυχία να εισάγουμε νέα σημείωση και να τερματίζεται, και τέλος αν έχουμε άγνωστη ενέργεια να εμφανίζεται μήνυμα και να τερματίζεται. Στο τέλος της διαδικασίας επιστρέφονται οι ενέργειες έτσι ώστε να φαίνονται στον χρήστη. Αν δεν έχουν γίνει αλλαγές φαίνεται το αρχικό κείμενο.


```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    final Intent intent = getIntent();

    final String action = intent.getAction();

    if (Intent.ACTION_EDIT.equals(action)) {

        mState = STATE_EDIT;

        mUri = intent.getData();

    } else if (Intent.ACTION_INSERT.equals(action)) {

        mState = STATE_INSERT;

        mUri = getContentResolver().insert(intent.getData(), null);

        if (mUri == null) {

            Log.e(TAG, "Failed to insert new note into " + getIntent().getData());

            finish();

            return;

        }

        setResult(RESULT_OK, (new Intent()).setAction(mUri.toString()));

    } else {

        Log.e(TAG, "Unknown action, exiting");

        finish();

        return;

    }

    setContentView(R.layout.note_editor);

    mText = (EditText) findViewById(R.id.note);

    mCursor = managedQuery(mUri, PROJECTION, null, null, null);

    if (savedInstanceState != null) {

        mOriginalContent = savedInstanceState.getString(ORIGINAL_CONTENT);

    }
}

```

Σημαντική ενέργεια επίσης είναι και η επεξεργασία του τίτλου της σημείωσης. Όπως βλέπουμε αν δεν είχαμε πρόβλημα να εισάγουμε δεδομένα τότε μπορούμε να επεξεργαστούμε ή να δημιουργήσουμε νέο τίτλο, αλλιώς εμφανίζεται μήνυμα ότι υπάρχει πρόβλημα.

```
protected void onResume() {  
  
    super.onResume();  
  
    if (mCursor != null) {  
  
        mCursor.moveToFirst();  
  
        if (mState == STATE_EDIT) {  
  
            setTitle(getText(R.string.title_edit));  
  
        } else if (mState == STATE_INSERT) {  
  
            setTitle(getText(R.string.title_create));  
  
        }  
  
        String note = mCursor.getString(COLUMN_INDEX_NOTE);  
  
        mText.setTextKeepState(note);  
  
        if (mOriginalContent == null) {  
  
            mOriginalContent = note;  
  
        }  
  
    } else {  
  
        setTitle(getText(R.string.error_title));  
  
        mText.setText(getText(R.string.error_message));  
  
    }  
  
}
```

- Αρχείο *NotePad.java*

Σε αυτό το αρχείο γίνεται η δήλωση των βασικών στηλών της εφαρμογής δηλαδή ο τύπος του περιεχομένου, ο τίτλος, η σημείωση, η ημερομηνία δημιουργίας και η ημερομηνία αλλαγής

- Αρχείο *NotePadProvider.java*

Αυτό το αρχείο μας παρέχει πρόσβαση στην βάση δεδομένων των σημειώσεων. Επιπλέον εδώ γίνεται η αποθήκευση των τίτλων, σημειώσεων και αλλαγών μαζί με τις ημερομηνίες δημιουργίας και αλλαγής.

```
private static class DatabaseHelper extends SQLiteOpenHelper {  
  
    DatabaseHelper(Context context) {  
  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
  
    }  
  
    @Override  
  
    public void onCreate(SQLiteDatabase db) {  
  
        db.execSQL("CREATE TABLE " + NOTES_TABLE_NAME + " ("  
  
            + Notes._ID + " INTEGER PRIMARY KEY,"  
  
            + Notes.TITLE + " TEXT,"  
  
            + Notes.NOTE + " TEXT,"  
  
            + Notes.CREATED_DATE + " INTEGER,"  
  
            + Notes.MODIFIED_DATE + " INTEGER"  
  
            + ");");  
  
    }  
  
}
```

- ο *Αρχείο NotesList.java*

Στο αρχείο αυτό δημιουργείται η λίστα των σημειώσεων. Πιο αναλυτικά, αρχικά καθορίζεται τι θα φαίνεται στην λίστα αν δεν υπάρχει κάποια σημείωση το οποίο θα είναι το προεπιλεγμένο περιεχόμενο. Έπειτα δημιουργούνται αντικείμενα μενού για την εισαγωγή νέας σημείωσης(Add), την επεξεργασία(Edit) και την διαγραφή(Delete). Τέλος, γίνεται η διαδικασία της επιλογής μιας σημείωσης με την επιστροφή του περιεχομένου της στον χρήστη.

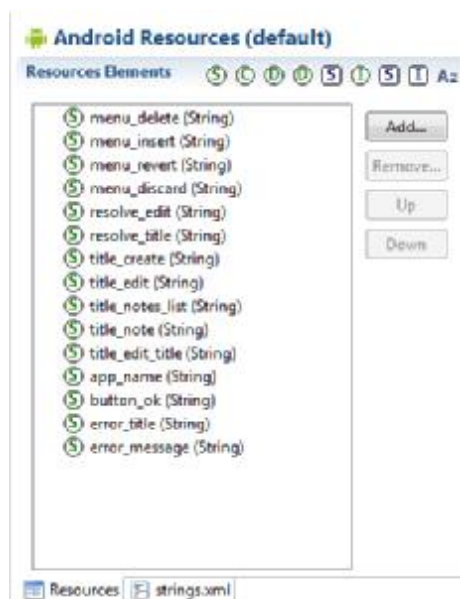
- ο *Αρχείο TitleEditor.java*

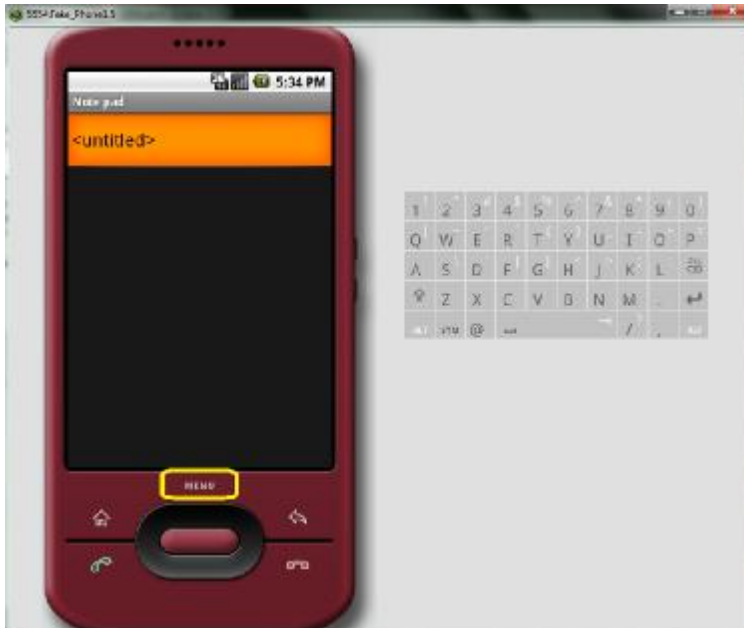
Εδώ γίνεται η επεξεργασία του τίτλου της σημείωσης. Όλος ο κώδικας υπάρχει στο ΠΑΡΑΡΤΗΜΑ.

Εμφάνιση της εφαρμογής

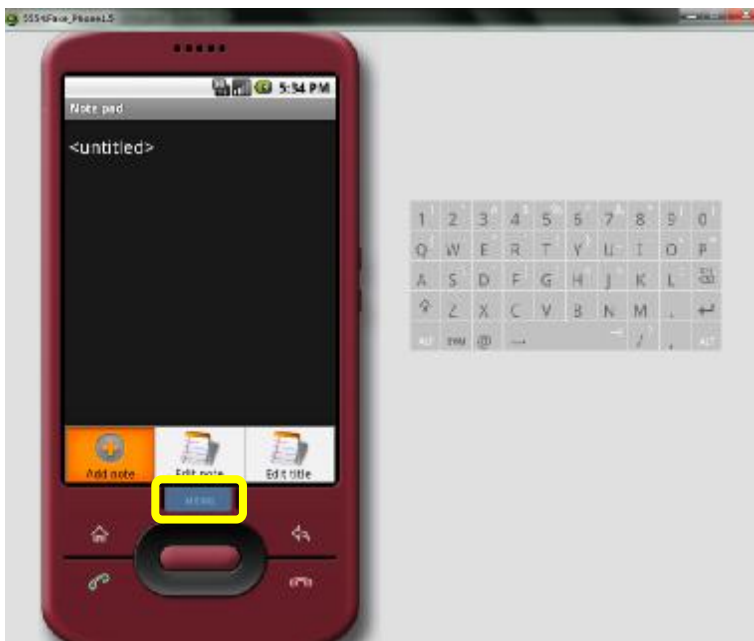


- Ø Στον φάκελο drawable έχουμε την εικόνα της εφαρμογής:
- Ø Στον φάκελο layout έχουμε τις ρυθμίσεις της εμφάνισης της εφαρμογής κατά την λειτουργία της.
- Ø Στον φάκελο values έχουμε τη δήλωση μεταβλητών που υπάρχουν στην εφαρμογή:

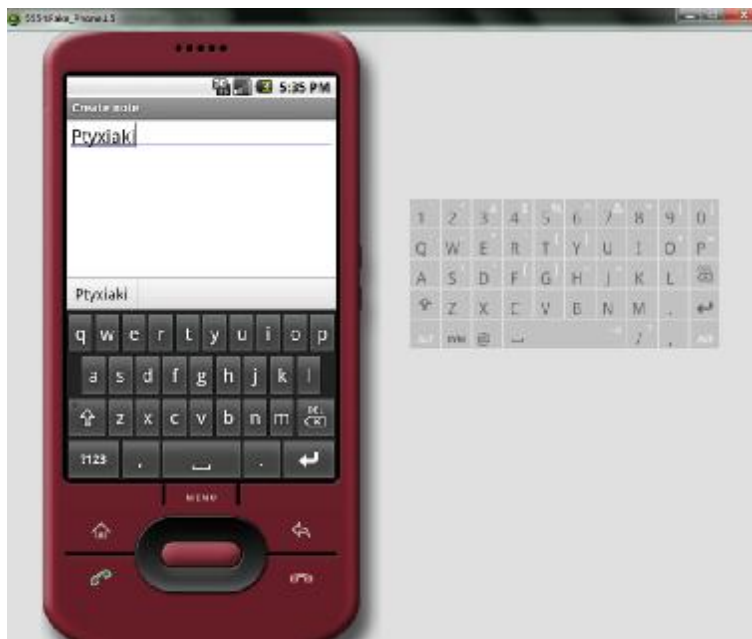




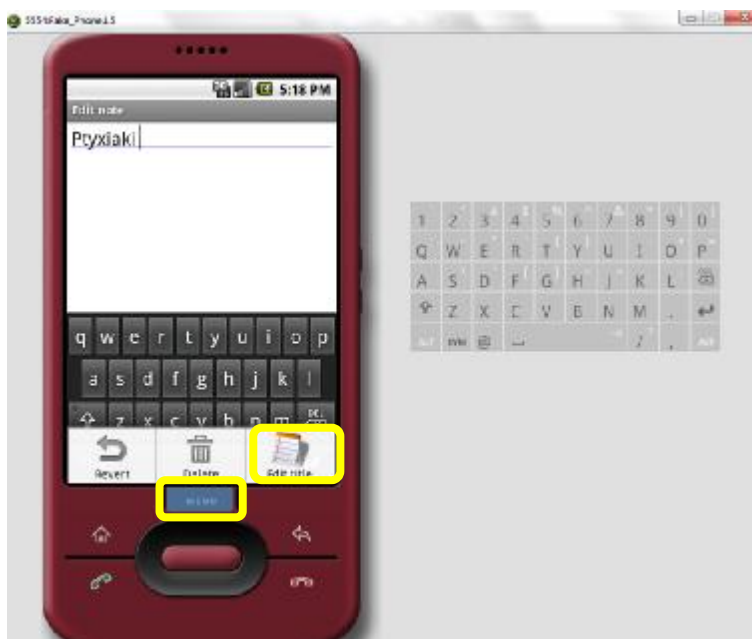
Εδώ βλέπουμε το μενού μας και τις επιλογές που έχουμε: Add note, Edit note και Edit title. Επιλέγουμε Add note για να εισάγουμε μια σημείωση.



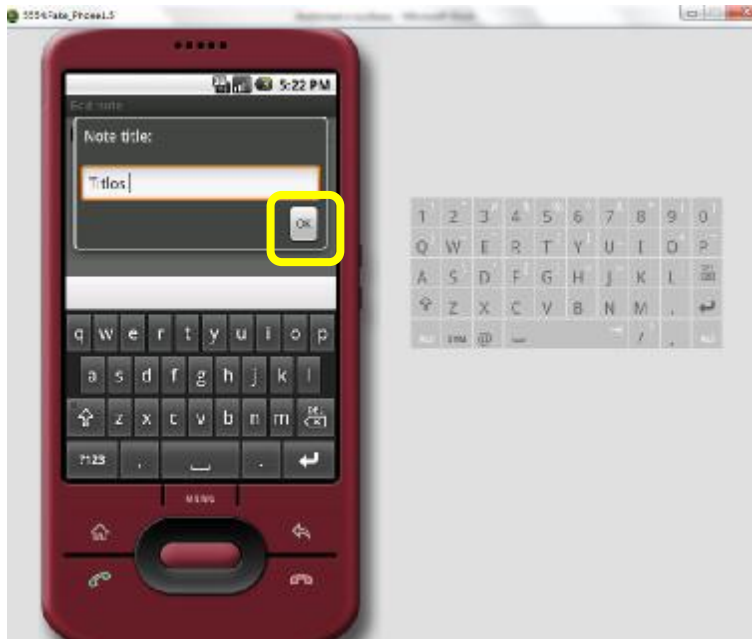
Γράφουμε την σημείωση μας:



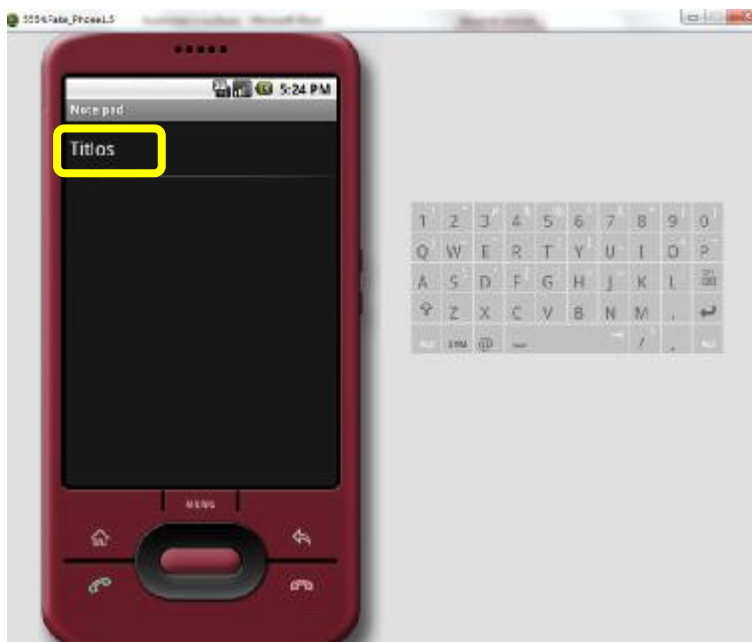
Επιλέγουμε MENU και βλέπουμε ότι μπορούμε να επεξεργαστούμε τον τίτλο της σημείωσης.



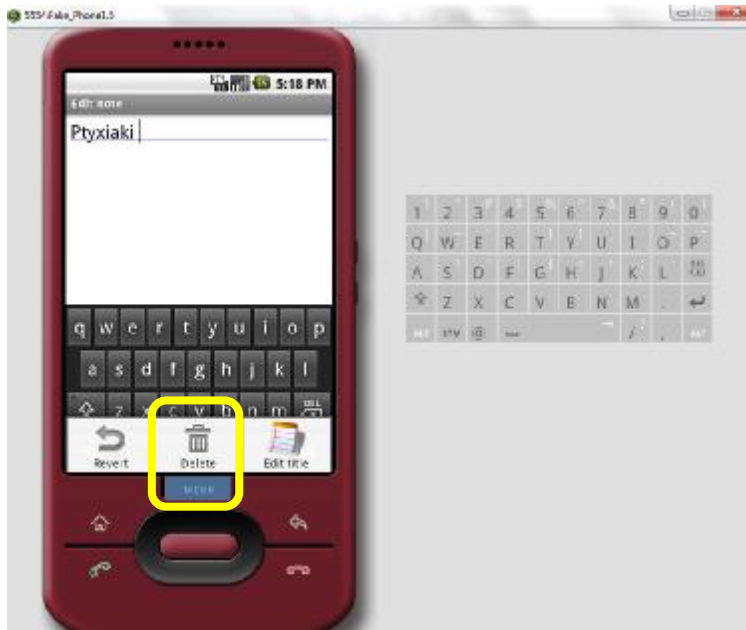
Γράφουμε τον τίτλο μας και πατάμε OK.



Η νέα σημείωση έχει προστεθεί στην λίστα των σημειώσεων. Αν θέλουμε να προσθέσουμε κι άλλη σημείωση πατάμε MENU και Add note. Για να δούμε την σημείωση πατάμε πάνω της.



Αν θέλουμε να την διαγράψουμε πατάμε MENU και Delete.



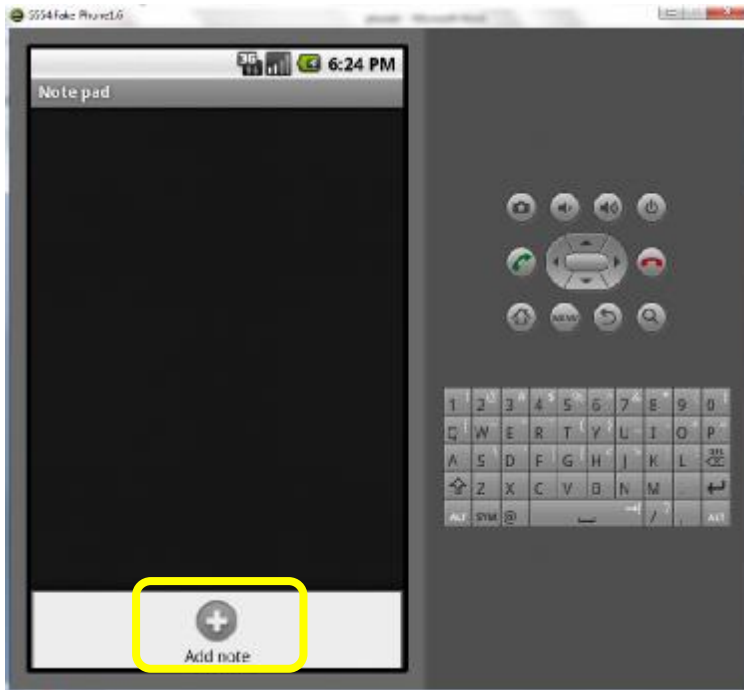
9.2 Android 1.6 (Donut)

Κώδικας της εφαρμογής

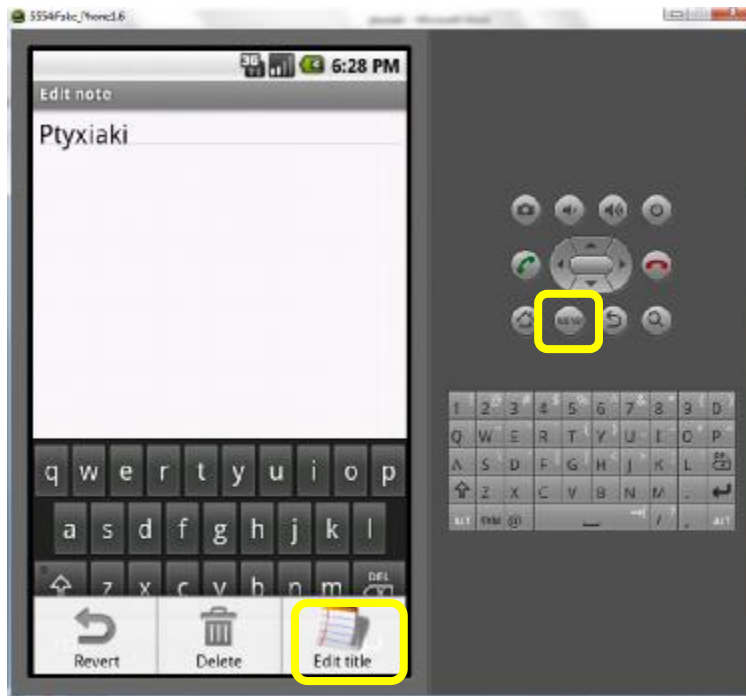
Ο κώδικας και οι ρυθμίσεις της εφαρμογής είναι ίδια με αυτά της προηγούμενης έκδοσης με αποτέλεσμα να μην αλλάζουν οι λειτουργίες της εφαρμογής.

Εκτέλεση της εφαρμογής

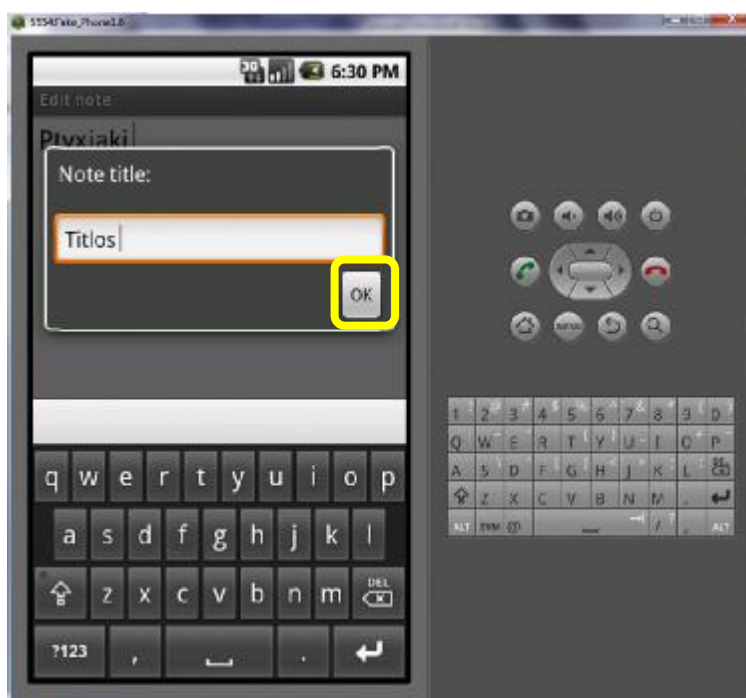
Τρέχουμε την εφαρμογή όπως δείξαμε πριν και εισάγουμε και επεξεργαζόμαστε την εφαρμογή με τα ίδια βήματα όπως φαίνεται στις παρακάτω εικόνες.



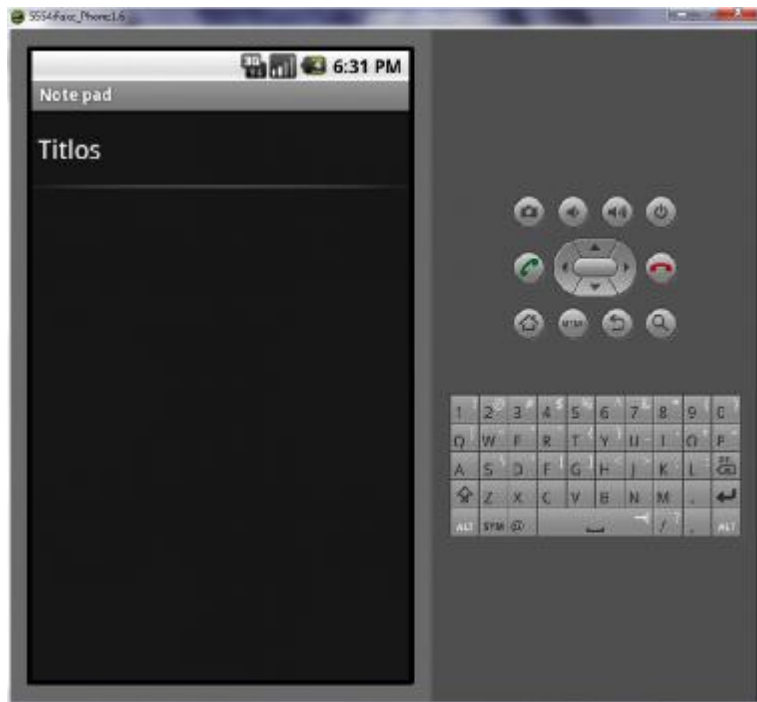
Πληκτρολογούμε την σημείωση



Πατάμε MENU και έπειτα Edit title για να επεξεργαστούμε τον τίτλο της σημείωσης μας.



Πληκτρολογούμε το όνομα της σημείωσης και πατάμε OK



Η σημείωσή μας έχει δημιουργηθεί.

9.3 Android 2.1 (Éclair)

Κώδικας της εφαρμογής

Στην έκδοση αυτή εντοπίζουμε επιπλέον κώδικα. Τα αρχεία NoteEditor.java, NotePad.java, NotesList.java και TitleEditor.java δεν αλλάζουν καθόλου σε σχέση με τον αρχικό κώδικα. Αντίθετα το αρχείο NotePadProvider.java αλλάζει και επιπλέον προστίθεται άλλο ένα αρχείο java το NotesLiveFolder.java. Παρακάτω έχουμε πιο αναλυτικά τα 2 αυτά αρχεία και την εξήγησή τους.

ο *Αρχείο NotePadProvider.java*

Στο αρχείο αυτό προστέθηκε μια επιπλέον λειτουργία των εφαρμογών Android η οποία μας δίνει την δυνατότητα να εμφανίσουμε την εφαρμογή στην αρχική οθόνη μας χωρίς να χρειαστεί να ξεκινήσουμε την εφαρμογή. Η λειτουργία αυτή είναι το Live Folder. Παρακάτω έχουμε τον κώδικα όπως είναι αλλαγμένος σε σχέση με τον αρχικό. Επιλογές λίστας σημειώσεων ανάλογα με το τι θα επιλέξει ο χρήστης.

```

public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs,
    String sortOrder) {
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
    qb.setTables(NOTES_TABLE_NAME);
    switch (sUriMatcher.match(uri)) {
    case NOTES:
        qb.setProjectionMap(sNotesProjectionMap);
        break;
    case NOTE_ID:
        qb.setProjectionMap(sNotesProjectionMap);
        qb.appendWhere(Notes._ID + "=" + uri.getPathSegments().get(1));
        break;
    case LIVE_FOLDER_NOTES:
        qb.setProjectionMap(sLiveFolderProjectionMap);
        break;
    default:
        throw new IllegalArgumentException("Unknown URI " + uri);
    }
}

```

Ανάλογα με την επιλογή σημείωσης φαίνεται και το αντίστοιχο περιεχόμενο στον χρήστη.

```

public String getType(Uri uri) {
    switch (sUriMatcher.match(uri)) {
    case NOTES:
    case LIVE_FOLDER_NOTES:
        return Notes.CONTENT_TYPE;
    }
}

```

```

case NOTE_ID:

    return Notes.CONTENT_ITEM_TYPE;

default:

    throw new IllegalArgumentException("Unknown URI " + uri);

}

}

```

- *Αρχείο NotesLiveFolder.java*

Για να μπορούμε να εμφανίσουμε τον φάκελο που αναφέραμε πριν πρέπει να δημιουργηθεί. Παρακάτω δηλώνεται το όνομα του φακέλου, η εικόνα που θα φαίνεται, η λίστα με τις σημειώσεις και η επιλογή προσθήκης μιας σημείωσης.

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    final Intent intent = getIntent();

    final String action = intent.getAction();

    if (LiveFolders.ACTION_CREATE_LIVE_FOLDER.equals(action)) {

        final Intent liveFolderIntent = new Intent();

        liveFolderIntent.setData(CONTENT_URI);

        liveFolderIntent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_NAME,

            getString(R.string.live_folder_name));

        liveFolderIntent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_ICON,

            Intent.ShortcutIconResource.fromContext(this,

                R.drawable.live_folder_notes));

        liveFolderIntent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_DISPLAY_MODE,

            LiveFolders.DISPLAY_MODE_LIST);

        liveFolderIntent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_BASE_INTENT,




            new Intent(Intent.ACTION_EDIT, NOTE_URI));
    }
}

```

```
    setResult(RESULT_OK, liveFolderIntent);  
  
} else {  
    setResult(RESULT_CANCELED);  
}  
  
finish(); }
```

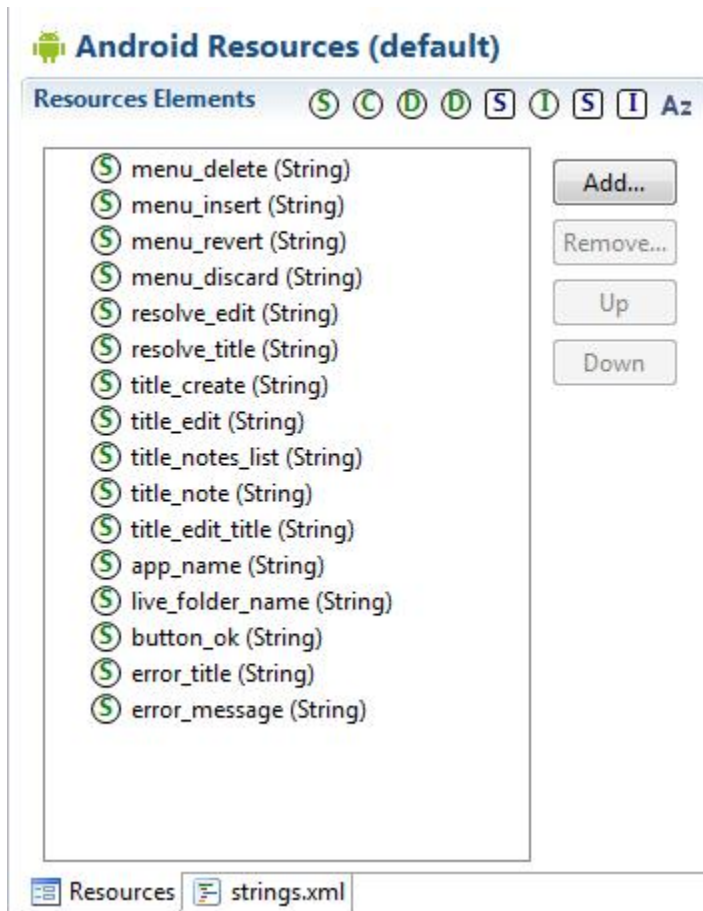
Εμφάνιση της εφαρμογής

∅ Στον φάκελο `res\drawable` έχουμε αυτές τις εικόνες:

-  εικόνα εφαρμογής
-  εικόνα προσθήκης σημείωσης
-  εικόνα live folder για την αρχική οθόνη

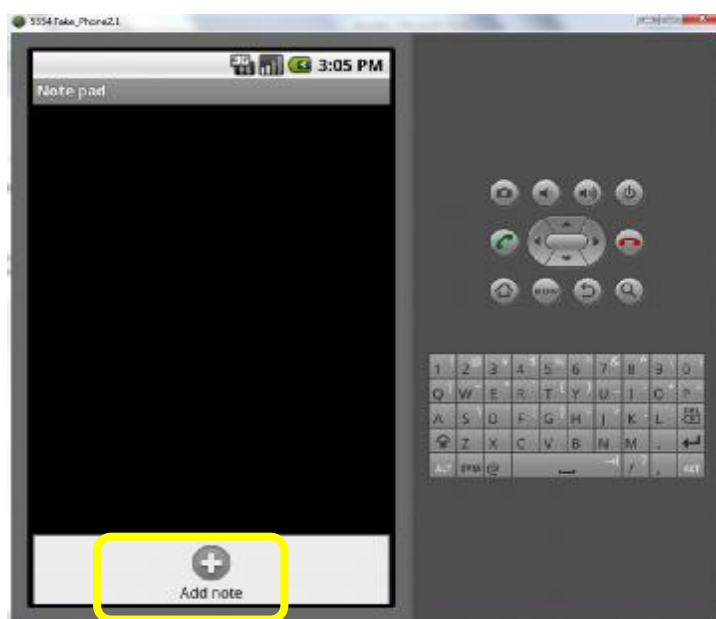
∅ Στον φάκελο `layout` δεν αλλάζει κάτι σε σχέση με τον αρχικό κώδικα.

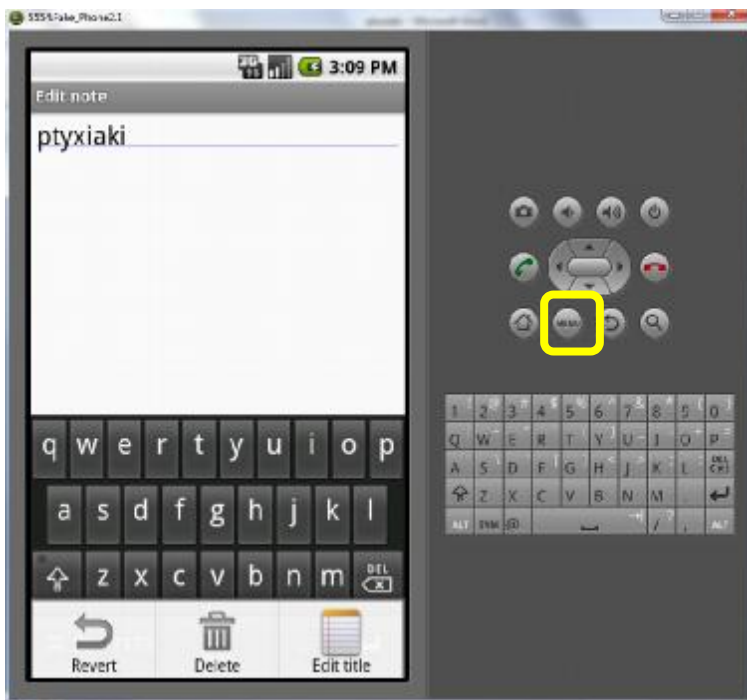
∅ Στον φάκελο `values` έχουμε τη δήλωση μεταβλητών που υπάρχουν στην εφαρμογή:

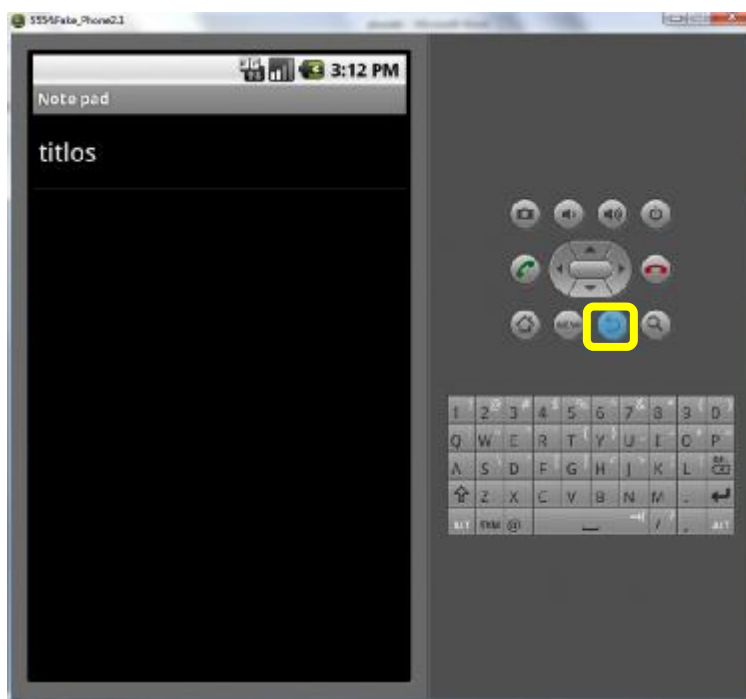
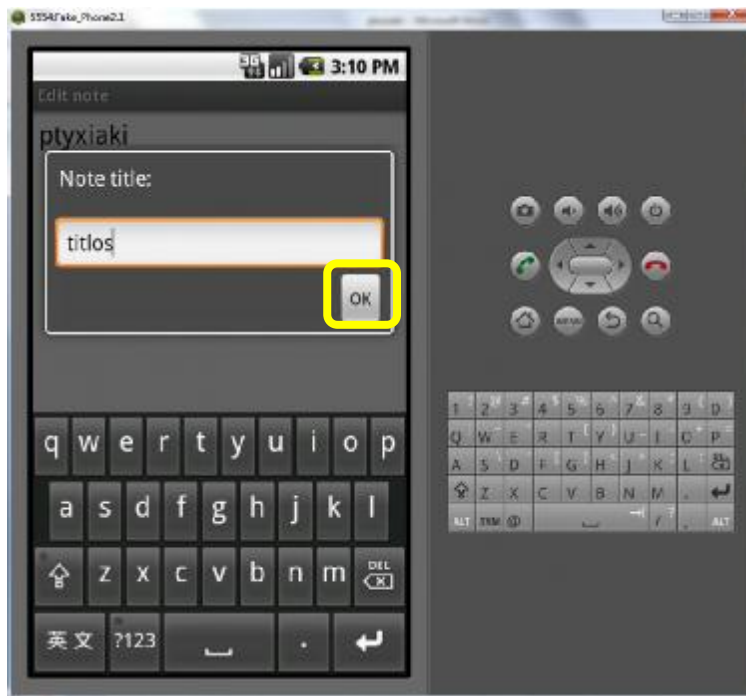


Εκτέλεση της εφαρμογής

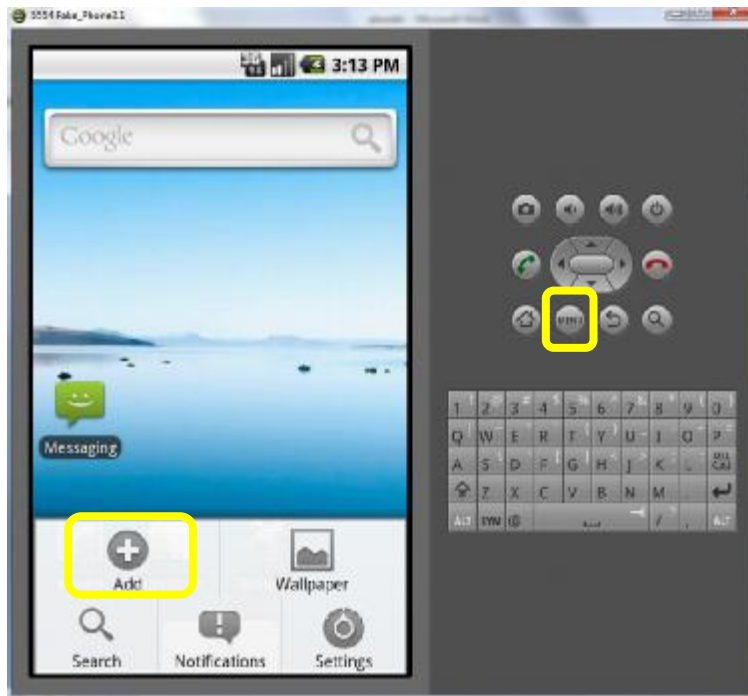
Για την εκτέλεση της εφαρμογής κάνουμε ότι κάναμε και στις προηγούμενες εκδόσεις.



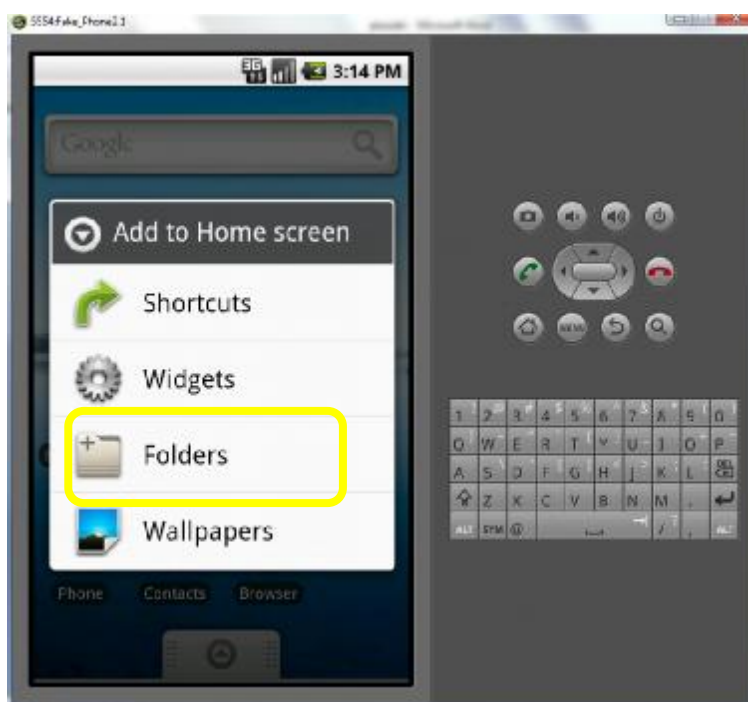




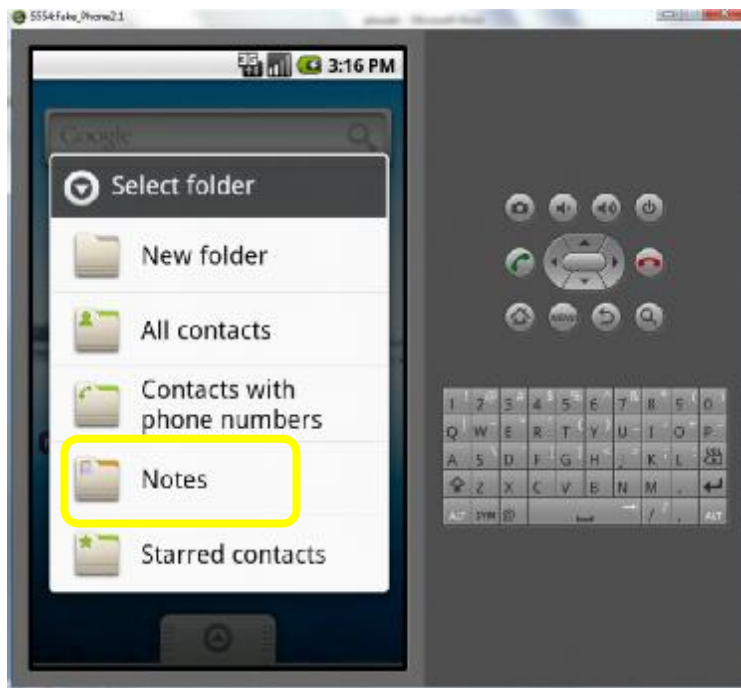
Σε αυτό το σημείο θα δείξουμε τι επιπλέον δυνατότητα έχουμε σε αυτή την εφαρμογή με την προσθήκη του Live Folders που εξηγήσαμε στον κώδικα. Πηγαίνοντας στην αρχική οθόνη πατάμε MENU και μετά Add.



Βλέπουμε τις επιλογές που έχουμε για να εισάγουμε στην αρχική οθόνη. Εμείς θέλουμε τους φακέλους οπότε πατάμε Folders.



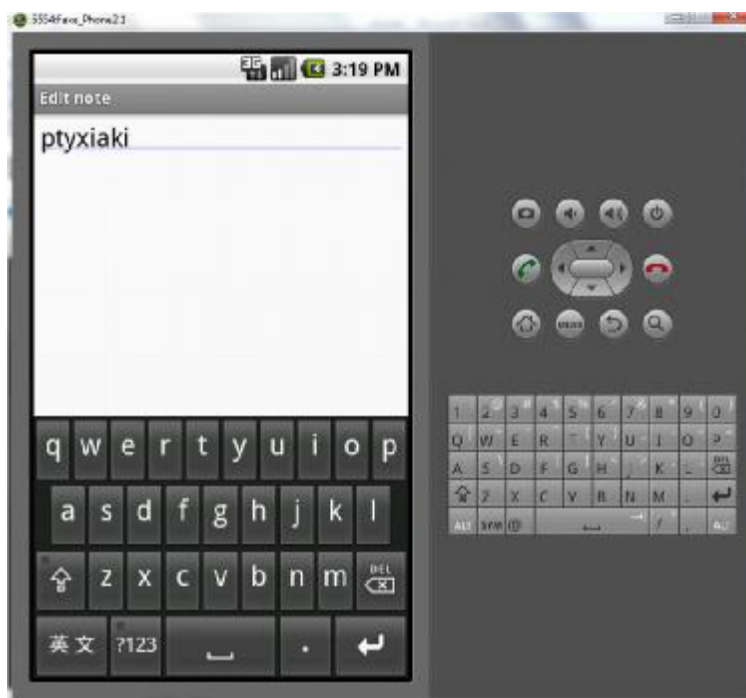
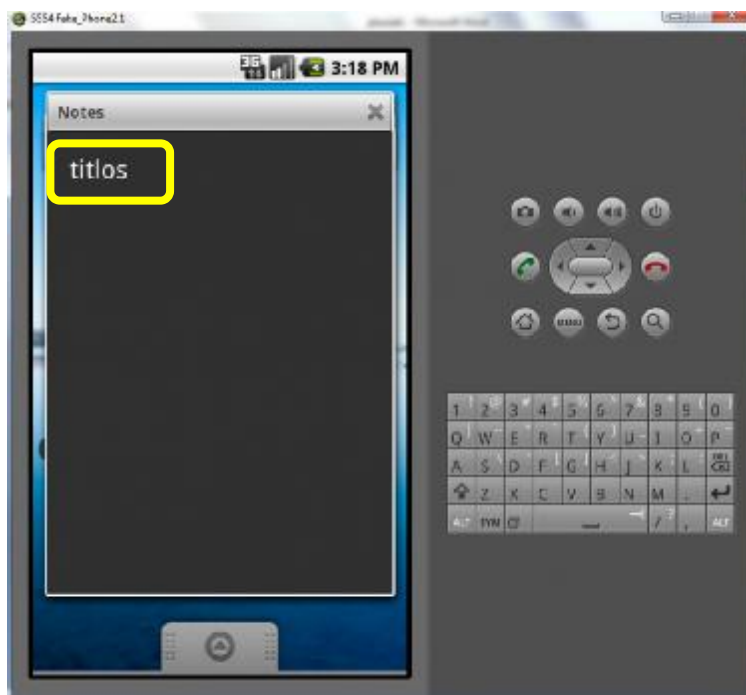
Μας εμφανίζονται διάφοροι φάκελοι οι οποίοι είναι εγκατεστημένοι στο κινητό και ο δικός μας φάκελος που δημιουργήθηκε στον κώδικα. Οπότε πατάμε Notes.



Ο φάκελος μπαίνει αυτόματα στην αρχική οθόνη. Πατάμε πάνω του για να ξεκινήσουμε την εφαρμογή.



Εδώ φαίνονται όλες οι σημειώσεις που έχουμε δημιουργήσει. Πατώντας στον τίτλο της σημείωσης θα μπούμε στην εφαρμογή και στις δυνατότητες που έχει αυτή.



9.4 Android 2.2 (Froyo)

Κώδικας της εφαρμογής

Ο κώδικας και οι ρυθμίσεις της εφαρμογής είναι ίδια με αυτά της προηγούμενης έκδοσης με αποτέλεσμα να μην αλλάζουν οι λειτουργίες της εφαρμογής.

Εκτέλεση της εφαρμογής

Και στην εκτέλεση δεν αλλάζει απολύτως τίποτα. Είναι ακριβώς ίδια με την έκδοση 2.1.

9.5 Android 2.3.3 (Gingerbread)

Κώδικας της εφαρμογής

- ο *Αρχείο NoteEditor.java*

Εδώ προστίθενται 2 επιπλέον στήλες οι NOTE και TITLE. Στις προηγούμενες εκδόσεις είχαμε μόνο την ID και την NOTE.









```
public class NoteEditor extends Activity {  
  
    private static final String TAG = "NoteEditor";  
  
    private static final String[] PROJECTION = new String[] {  
        NoteColumns.ID, // 0  
        NoteColumns.NOTE, // 1  
        NoteColumns.TITLE, // 2  
    };  
  
    private static final int COLUMN_INDEX_NOTE = 1;  
  
    private static final int COLUMN_INDEX_TITLE = 2;
```

Αποθήκευση σημείωσης κατά την επεξεργασία της. Αποθηκεύονται οι νέες ενημερώσεις και η ημερομηνία επεξεργασίας και γίνεται η εισαγωγή του τίτλου.

```
private final void saveNote() {  
  
    if (mCursor != null) {  
  
        ContentValues values = new ContentValues();  
  
        values.put(NoteColumns.MODIFIED_DATE, System.currentTimeMillis());  
  
        String text = mText.getText().toString();  
  
        int length = text.length();  
  
        if (mState == STATE_INSERT) {  
  
            if (length == 0) {  
  
                Toast.makeText(this, R.string.nothing_to_save, Toast.LENGTH_SHORT).show();  
  
                return;  
  
            }  
  
            String title = text.substring(0, Math.min(30, length));  
  
            if (length > 30) {  
  
                int lastSpace = title.lastIndexOf(' ');  
  
                if (lastSpace > 0) {  
  
                    title = title.substring(0, lastSpace);  
  
                }  
  
            }  
  
            values.put(NoteColumns.TITLE, title);  
  
        }  
  
        values.put(NoteColumns.NOTE, text);  
  
        try {  
  
            getContentResolver().update(mUri, values, null, null);  
  
        } catch  
(NullPointerException e) { Log.e(TAG, e.getMessage()); } } }
```

Εμφάνιση της εφαρμογής

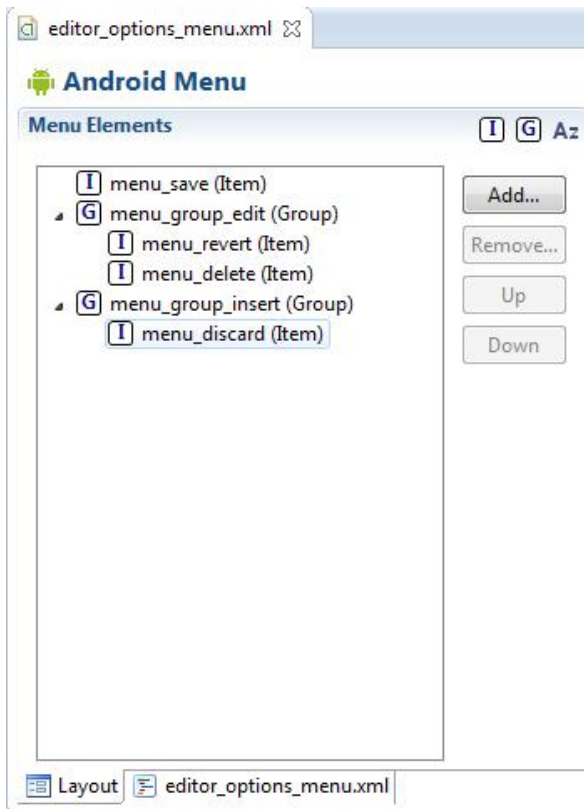
Ø Στον φάκελο `res\drawable` έχουμε αυτές τις εικόνες:

-  εικόνα εφαρμογής
-  προσθήκη σημείωσης
-  διαγραφή σημείωσης
-  έξοδος
-  επεξεργασία σημείωση
-  πίσω στα προηγούμενα
-  αποθήκευση σημείωσης
-  φάκελος αρχικής οθόνης(live folder)

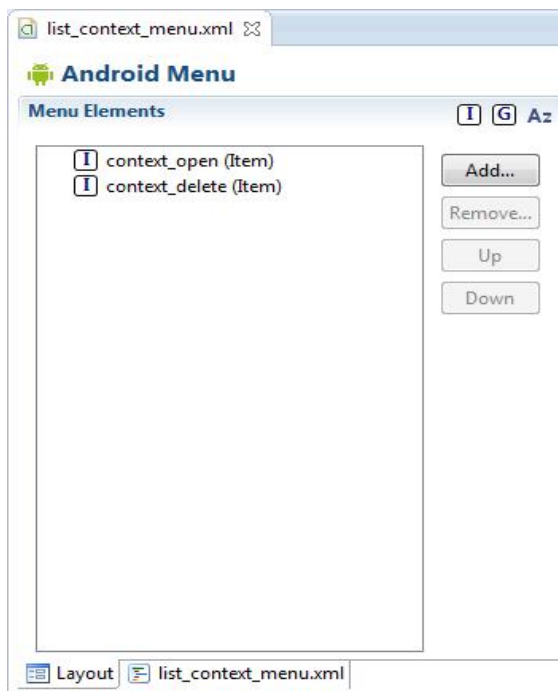
Ø Στον φάκελο `layout` δεν αλλάζει κάτι σε σχέση με τον αρχικό κώδικα.

Ø Σε αυτή την έκδοση έχουμε έναν επιπλέον φάκελο τον `menu`. Περιλαμβάνει τα αρχεία `editor_options_menu.xml`, `list_context_menu.xml` και `list_options_menu.xml`.

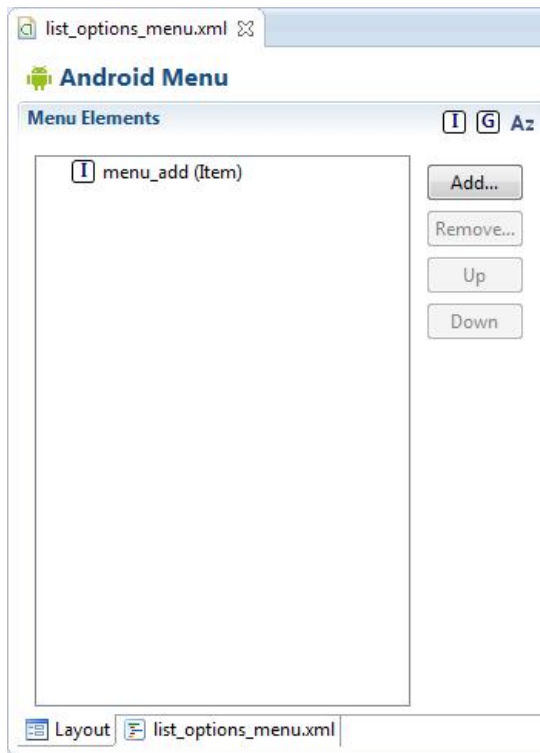
- Αρχείο *editor_options_menu.xml*



- Αρχείο *list_context_menu.xml*



- Αρχείο `list_options_menu.xml`

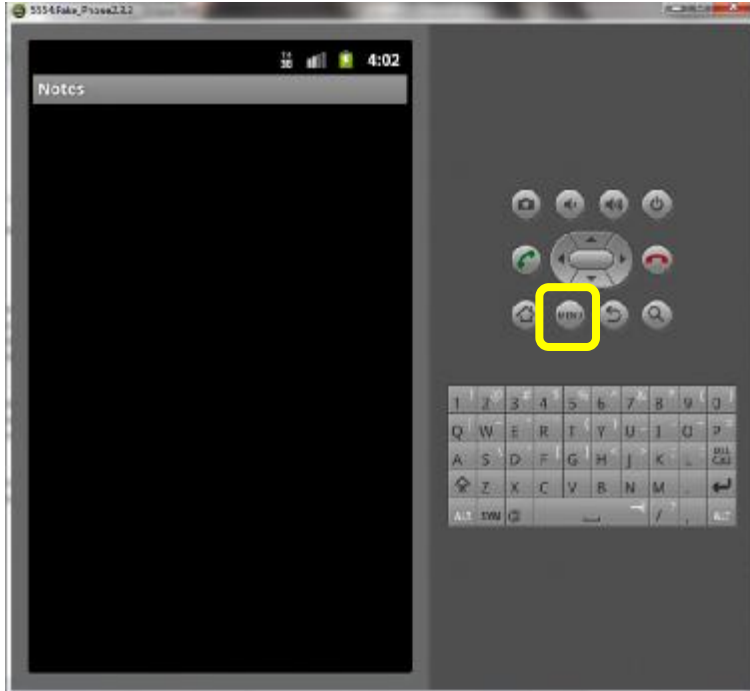


- Ø Στον φάκελο `values` έχουμε τη δήλωση μεταβλητών που υπάρχουν στην εφαρμογή:

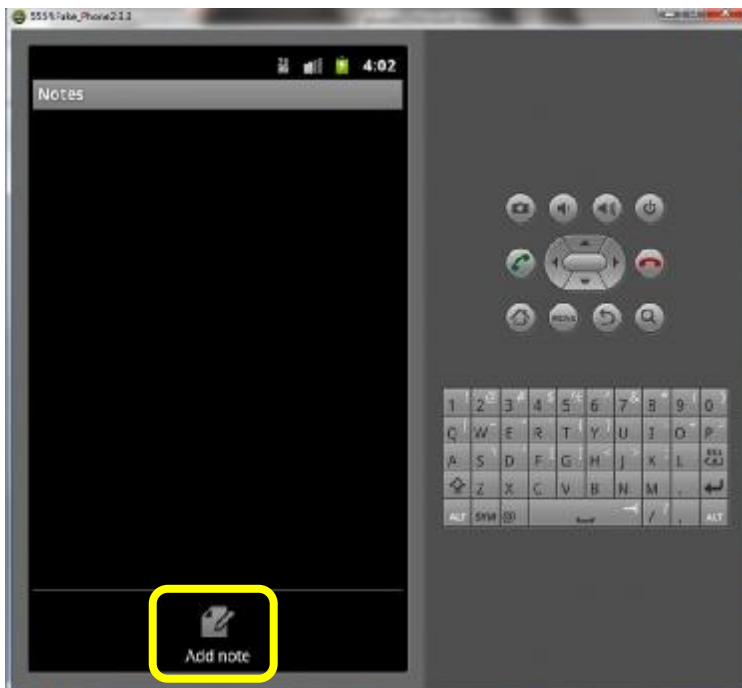


Εκτέλεση της εφαρμογής

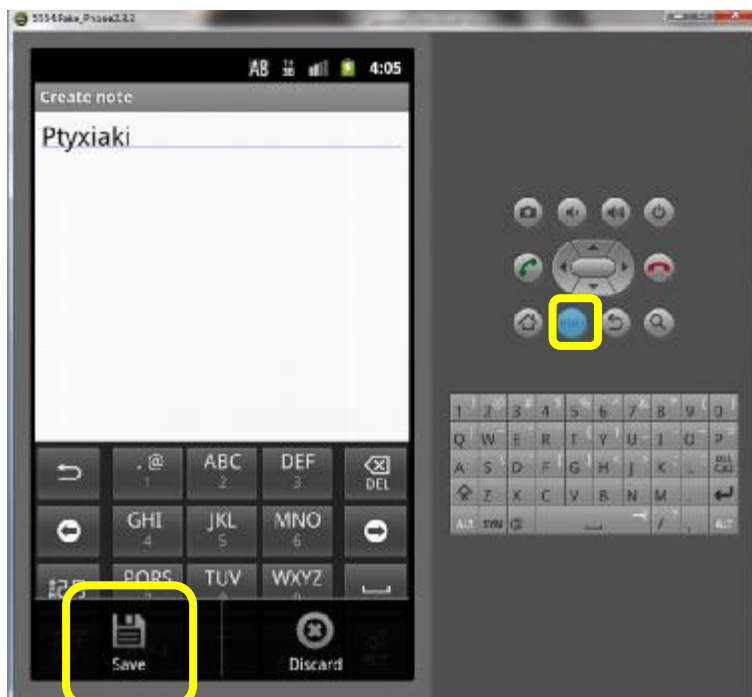
Ανοίγοντας την εφαρμογή επιλέγουμε MENU



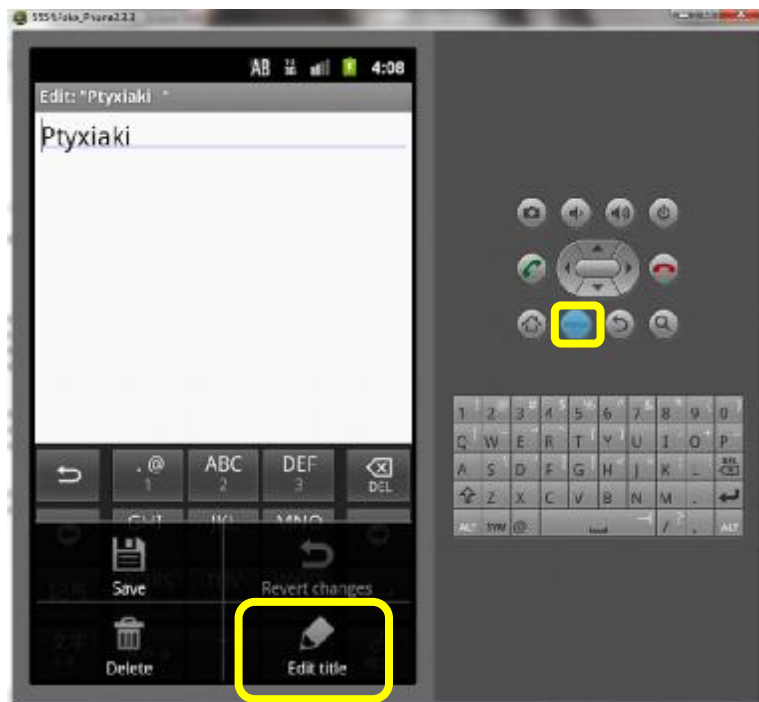
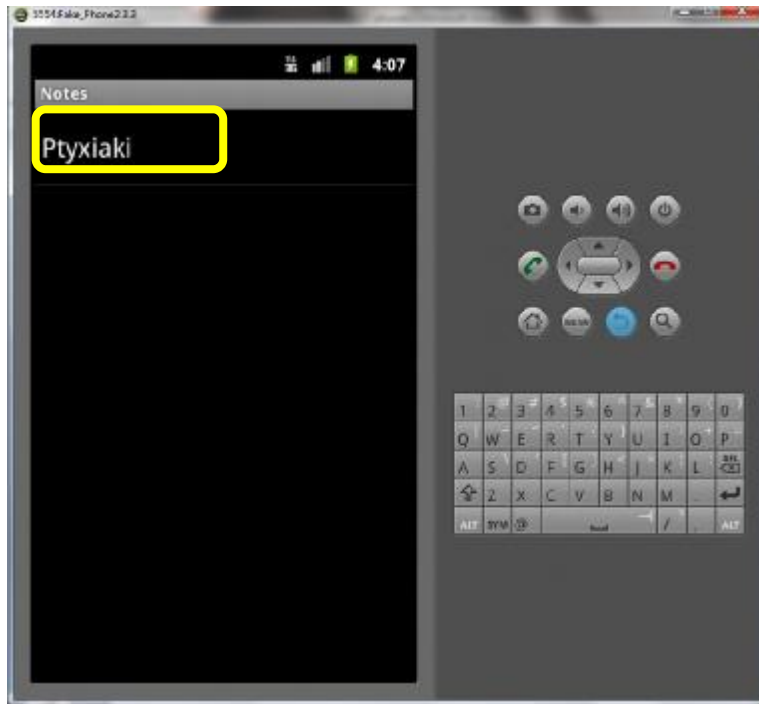
Επιλέγουμε Add note για να ξεκινήσουμε την σημείωση



Γράφουμε την σημείωση που θέλουμε και επιλέγουμε MENU και SAVE για να αποθηκευτεί, αλλιώς μπορούμε να επιλέξουμε Discard για να ακυρώσουμε την διαδικασία.



Πηγαίνοντας πίσω βλέπουμε ότι η σημείωση έχει αποθηκευτεί με όνομα ίδιο με αυτό που γράψαμε προηγουμένως αυτόματα. Για να επεξεργαστούμε τον τίτλο της σημείωσης επιλέγουμε πάλι την σημείωση και έπειτα επιλέγουμε menu και edit title, γράφουμε τον τίτλο μας και μετά OK.





9.6 Android 3.0, 3.1, 3.2 (Honeycomb)

Κώδικας της εφαρμογής

Τα αρχεία `NotesList.java`, `NotesLiveFolder.java` και `TitleEditor.java` δεν αλλάζουν από την προηγούμενη έκδοση. Οι αλλαγές που έχουν γίνει φαίνονται παρακάτω.

- Αρχείο `NoteEditor.java`

- Ø Σε σχέση με την προηγούμενη έκδοση προστίθεται μια ακόμα ενέργεια, η επικόλληση.

```

if (Intent.ACTION_EDIT.equals(action)) {

    mState = STATE_EDIT;
    mUri = intent.getData();

} else if (Intent.ACTION_INSERT.equals(action)
|| Intent.ACTION_PASTE.equals(action)) {

    mState = STATE_INSERT;
    mUri = getContentResolver().insert(intent.getData(),
null);

```

```

if (Intent.ACTION_PASTE.equals(action)) {
    performPaste();
    mState = STATE_EDIT;
}

```

∅ Η λειτουργία της επικόλλησης γίνεται ως εξής: Παίρνει βοήθεια από το Clipboard Manager όπου εκεί θα βρίσκονται τα δεδομένα για την επικόλληση. Βρίσκει ποια από αυτά είναι ίδιου τύπου με αυτά του Note pad provider έτσι ώστε να πάρει τα σωστά στοιχεία και παίρνει τη σημείωση και τον τίτλο της. Έπειτα ανανεώνει την τρέχουσα σημείωση με τα νέα στοιχεία.

∅ Τώρα χρειαζόμαστε και την λειτουργία της ανανέωσης της σημείωσης η οποία γίνεται ως εξής: Αρχικά δημιουργεί έναν χάρτη ο οποίος θα περιέχει μεταβλητές για ανανέωση. Μετά παίρνει τις αλλαγές που έχουν γίνει και τις αποθηκεύει πίσω στον χάρτη.

ο Αρχείο *NotePad.java*

```

public final class NotePad {
    public static final String AUTHORITY = "com.google.provider.NotePad";
    private NotePad() {
    }
    public static final class Notes implements BaseColumns {
        private Notes() {}
        public static final String TABLE_NAME = "notes";
        private static final String SCHEME = "content://";
        private static final String PATH_NOTES = "/notes";
        private static final String PATH_NOTE_ID = "/notes/";
        public static final int NOTE_ID_PATH_POSITION = 1;
        private static final String PATH_LIVE_FOLDER = "/live_folders/notes";
        public static final Uri CONTENT_URI = Uri.parse(SCHEME + AUTHORITY +
PATH_NOTES);
        public static final Uri CONTENT_ID_URI_BASE
            = Uri.parse(SCHEME + AUTHORITY + PATH_NOTE_ID);
        public static final Uri CONTENT_ID_URI_PATTERN
            = Uri.parse(SCHEME + AUTHORITY + PATH_NOTE_ID + "/"#");
        public static final Uri LIVE_FOLDER_URI
            = Uri.parse(SCHEME + AUTHORITY + PATH_LIVE_FOLDER);
        public static final String CONTENT_TYPE = "vnd.android.cursor.dir/vnd.google.note";
        public static final String CONTENT_ITEM_TYPE =
"vnd.android.cursor.item/vnd.google.note";
        public static final String DEFAULT_SORT_ORDER = "modified DESC";
        public static final String COLUMN_NAME_TITLE = "title";

```

```

    public static final String COLUMN_NAME_NOTE = "note";
    public static final String COLUMN_NAME_CREATE_DATE = "created";
    public static final String COLUMN_NAME_MODIFICATION_DATE = "modified";
}
}

```

ο *Αρχείο NotePadProvider.java*

Στο αρχείο αυτό έχουμε την ίδια δομή με την προηγούμενη έκδοση με μια επιπλέον διαδικασία την `writeDataToPipe` όπου γίνεται η μετατροπή των δεδομένων έτσι ώστε να μπορεί ο χρήστης να τα βλέπει.

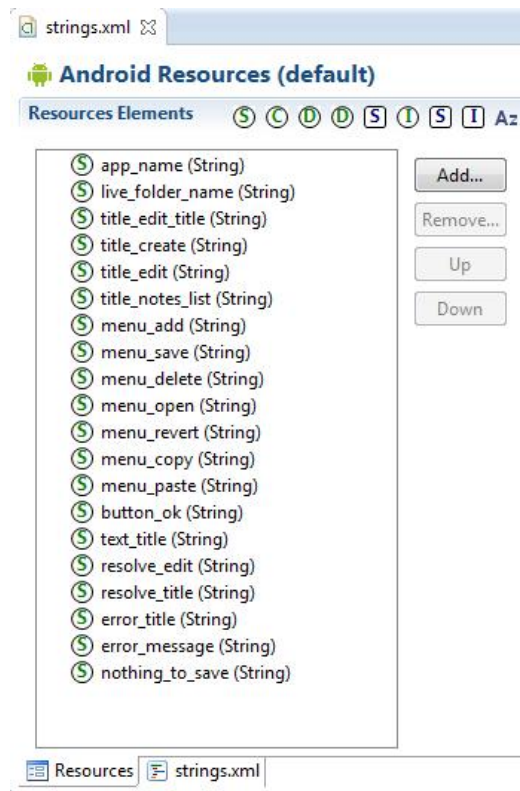
```

    public void writeDataToPipe(ParcelFileDescriptor output, Uri uri, String mimeType,
        Bundle opts, Cursor c) {
        FileOutputStream fout = new FileOutputStream(output.getFileDescriptor());
        PrintWriter pw = null;
        try {
            pw = new PrintWriter(new OutputStreamWriter(fout, "UTF-8"));
            pw.println(c.getString(READ_NOTE_TITLE_INDEX));
            pw.println("");
            pw.println(c.getString(READ_NOTE_NOTE_INDEX));
        } catch (UnsupportedEncodingException e) {
            Log.w(TAG, "Ooops", e);
        } finally {
            c.close();
            if (pw != null) {
                pw.flush();
            }
            try {
                fout.close();
            } catch (IOException e) {
            }
        }
    }
}

```

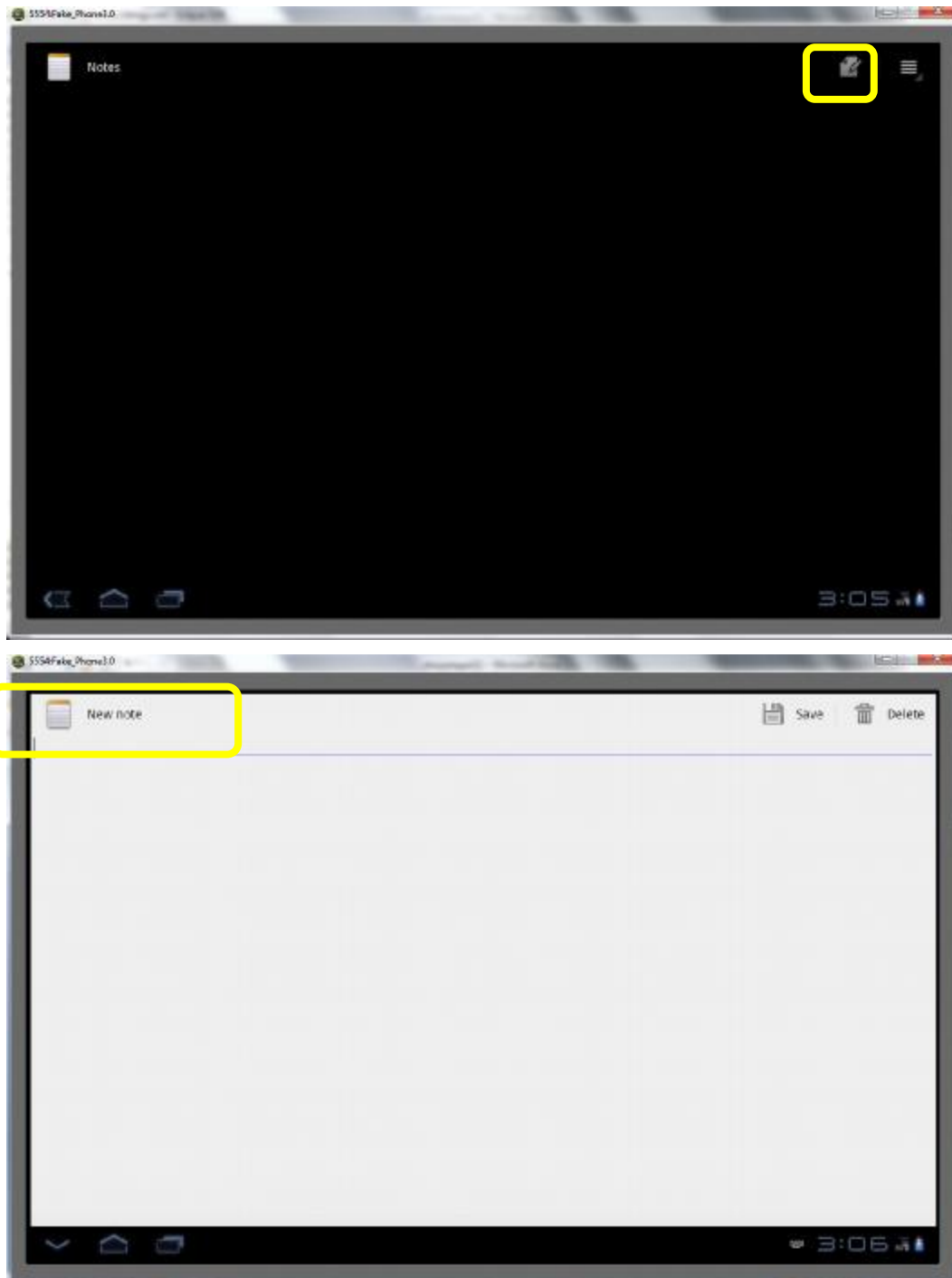

Εμφάνιση της εφαρμογής

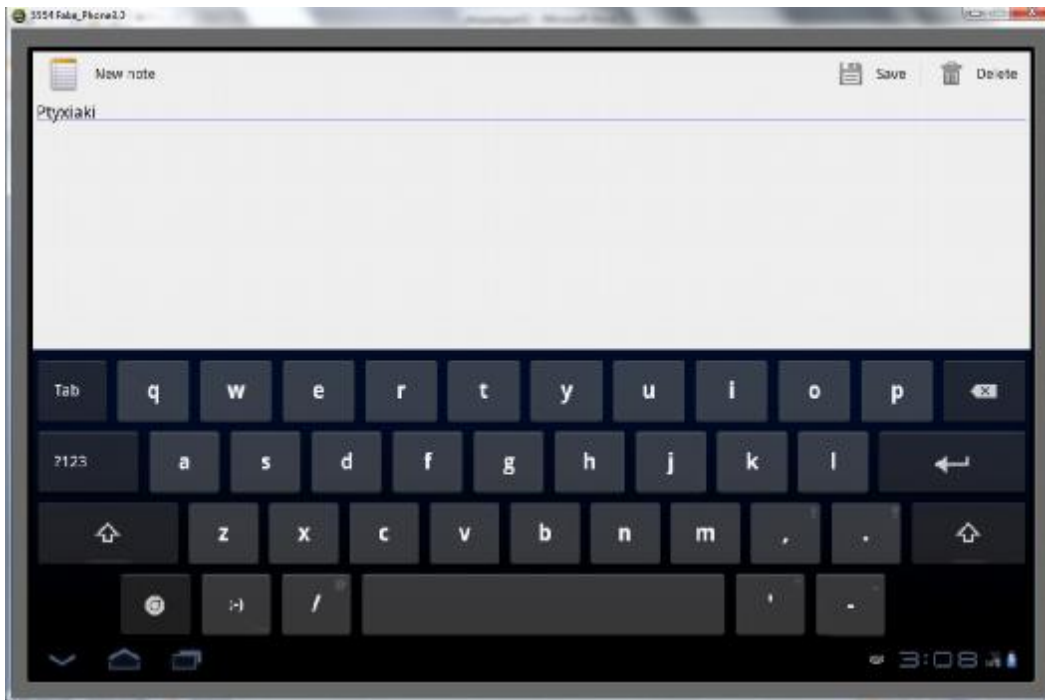
- Ø Στον φάκελο `res\drawable` έχουμε τις ίδιες εικόνες με την προηγούμενη έκδοση.
- Ø Στον φάκελο `layout` και `menu` δεν αλλάζει κάτι
- Ø Στον φάκελο `values` έχουμε τις μεταβλητές:



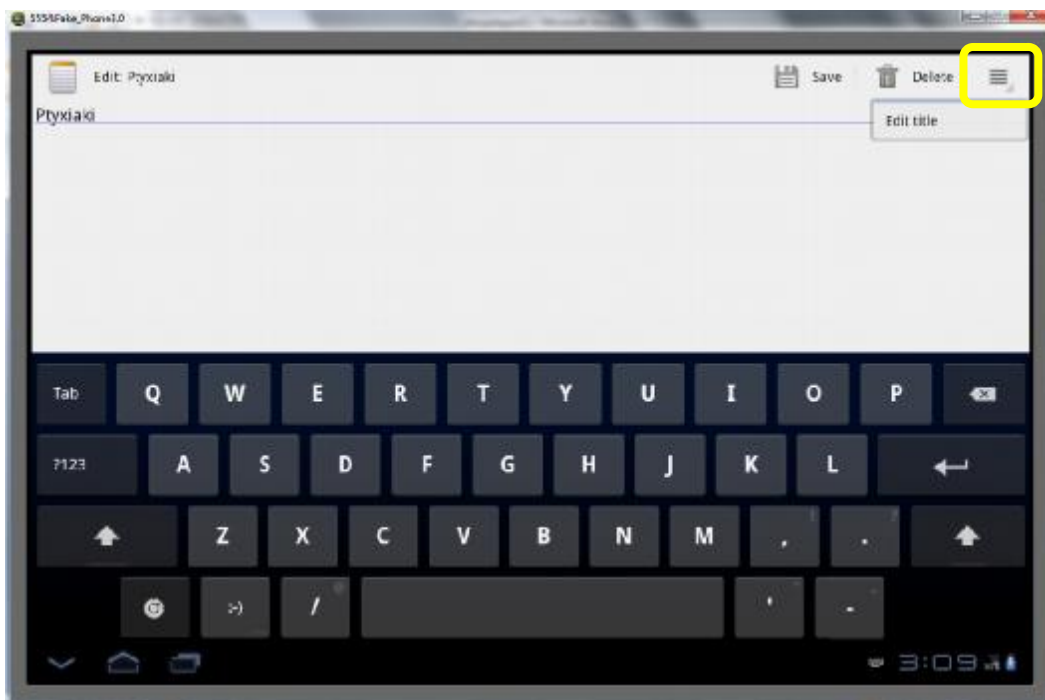
Εκτέλεση της εφαρμογής

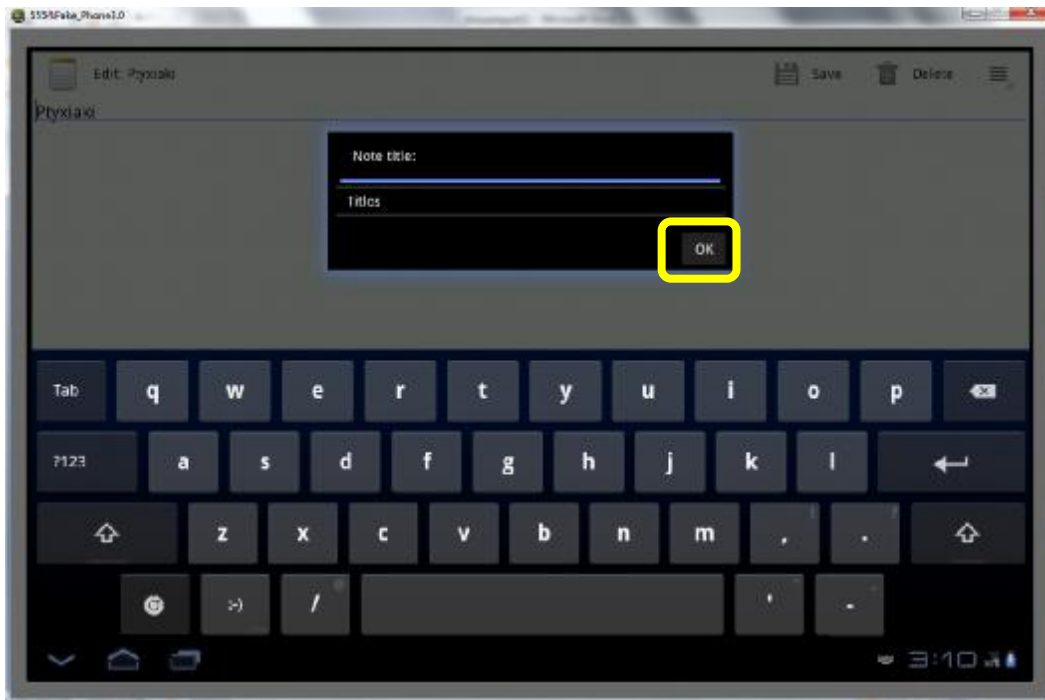
Όπως βλέπουμε παρακάτω στις εικόνες η εφαρμογή έχει αλλάξει και τα κουμπιά επιλογών εμφανίζονται στην οθόνη. Για να δημιουργήσουμε μια σημείωση επιλέγουμε το εικονίδιο πάνω δεξιά στην οθόνη μας που σημαίνει δημιουργία σημείωσης. Επιλέγουμε New note και γράφουμε την σημείωση μας.



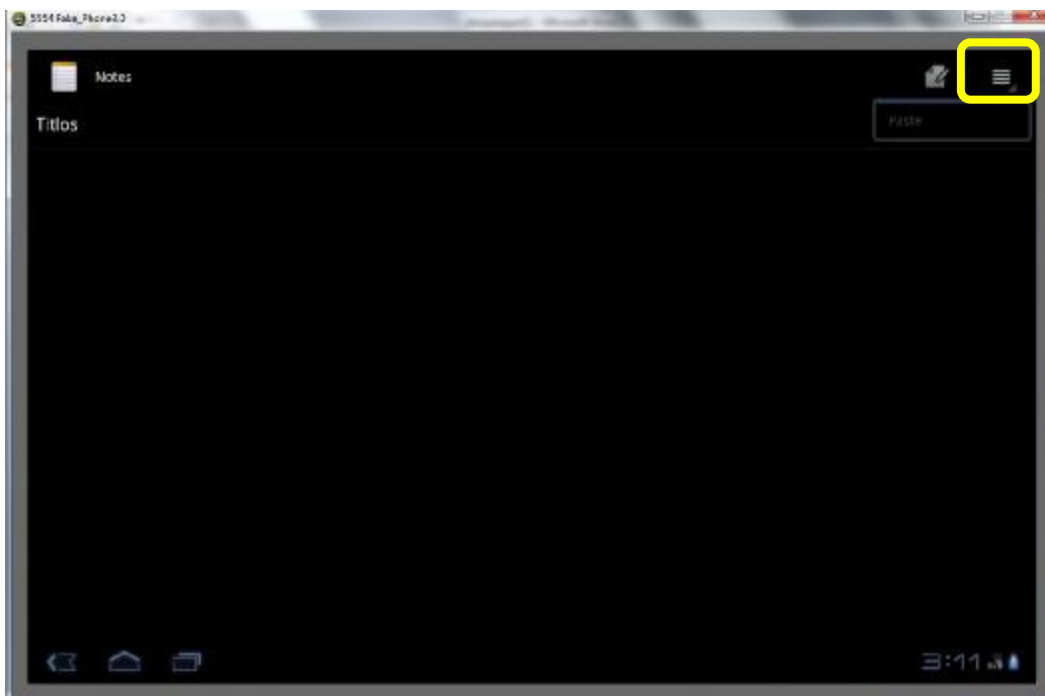


Για να επεξεργαστούμε τον τίτλο της σημείωσης σε αυτή την έκδοση είναι πιο εύκολο πατώντας το εικονίδιο μενού πάνω δεξιά στην οθόνη μας και επιλέγουμε Edit title, γράφουμε τον τίτλο και OK

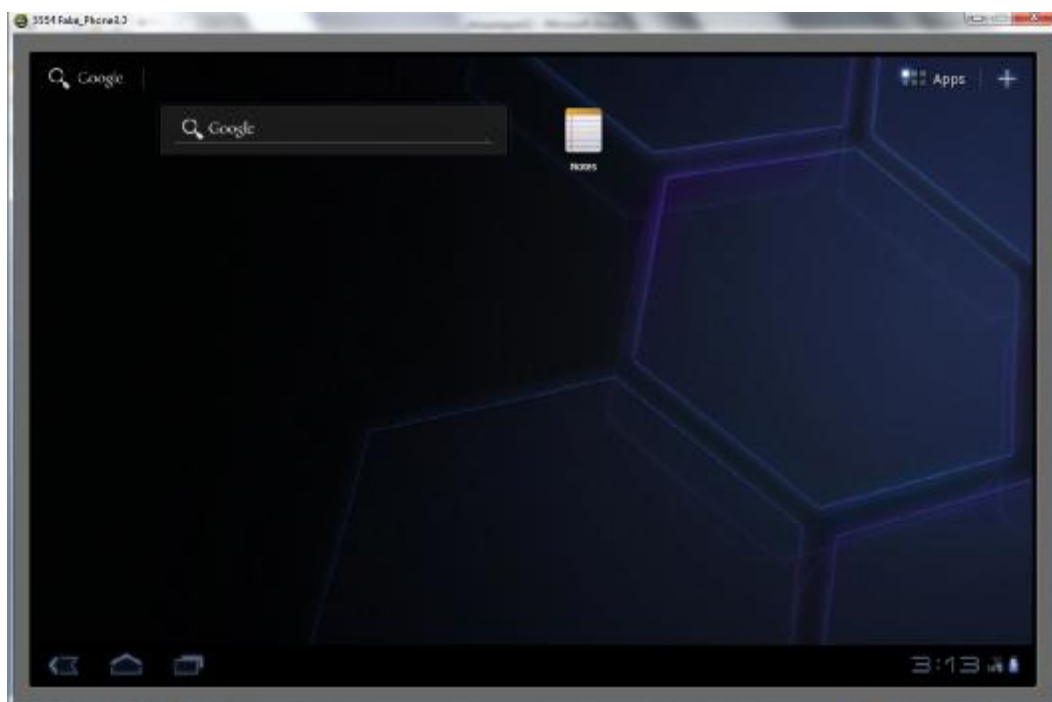




Μπορούμε να δούμε και την νέα προσθήκη στην εφαρμογή την επικόλληση. Αν είχαμε αντιγράψει ένα κείμενο από οπουδήποτε στην κονσόλα μας, πατώντας το paste θα μας έβγαζε το κείμενο σαν νέα σημείωση.



Βλέπουμε και την εφαρμογή πώς φαίνεται στην κονσόλα μας. Μπορούμε να την σύρουμε στην αρχική οθόνη για να την επιλέγουμε πιο εύκολα.



9.7 Android 4.0, 4.0.3 (Ice cream sandwich)

Κώδικας εφαρμογής

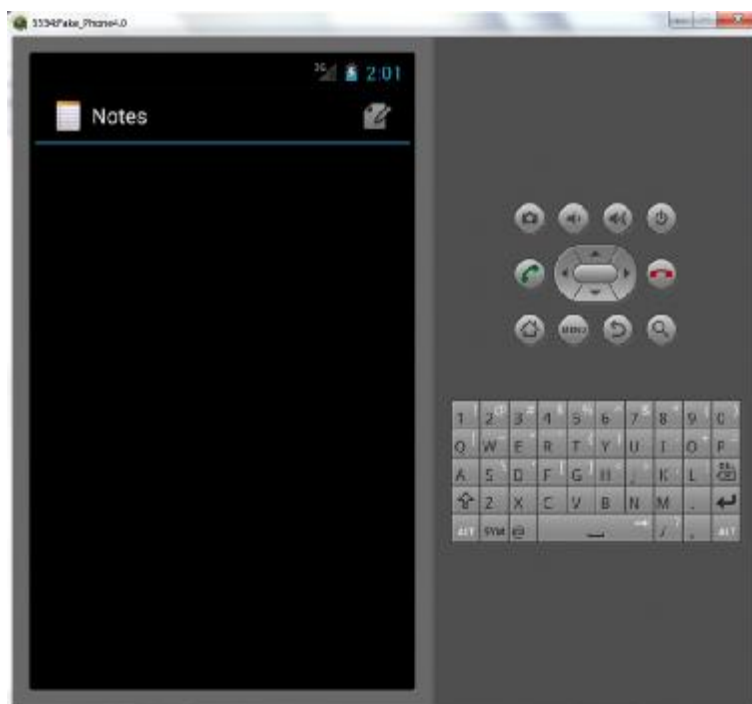
Ο κώδικας δεν αλλάζει καθόλου σε αυτή την έκδοση.

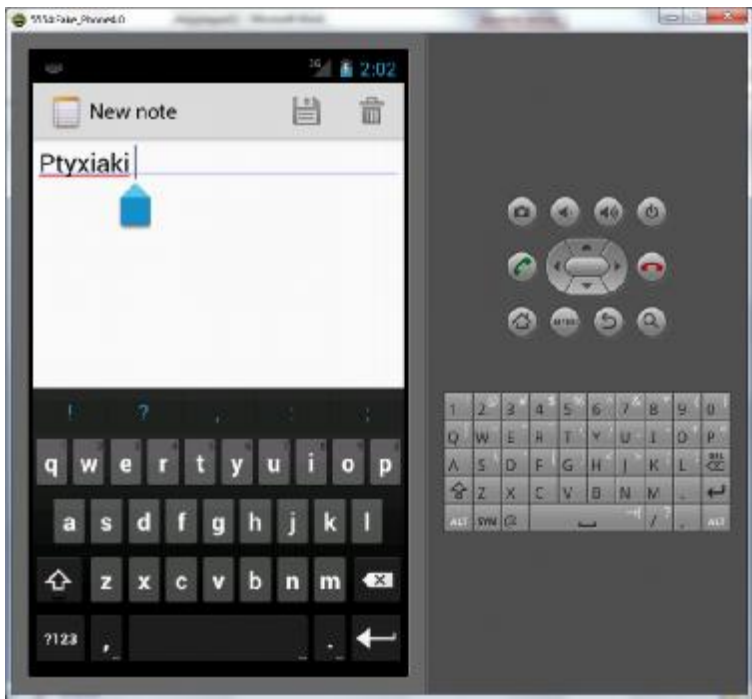
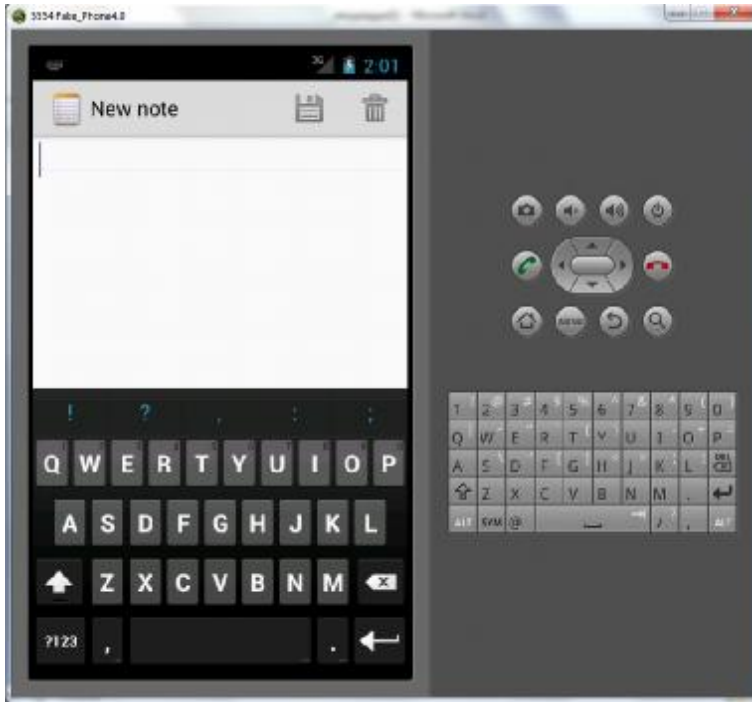
Εμφάνιση εφαρμογής

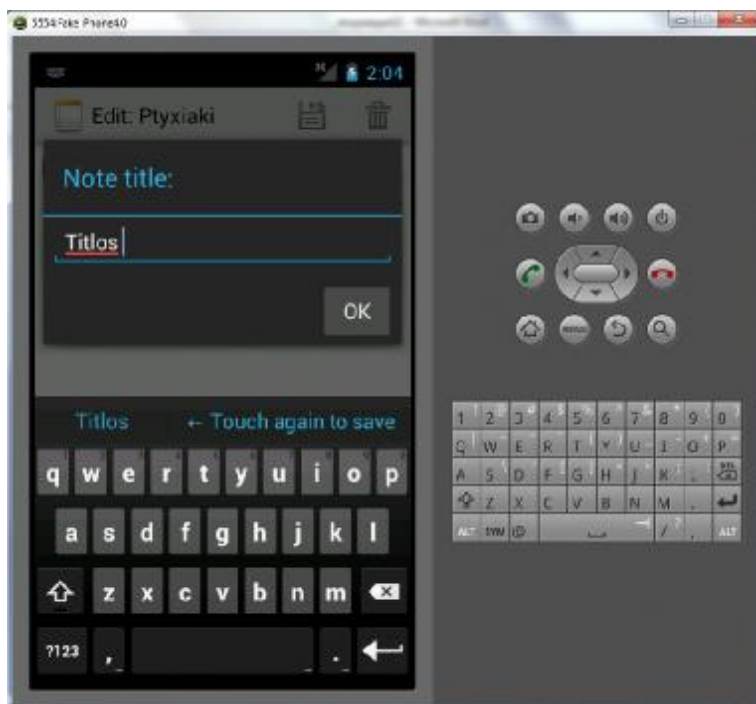
Ούτε στις ρυθμίσεις της εμφάνισης δεν αλλάζει κάτι. Πάμε να δούμε λοιπόν την εφαρμογή σε αυτή την έκδοση android.

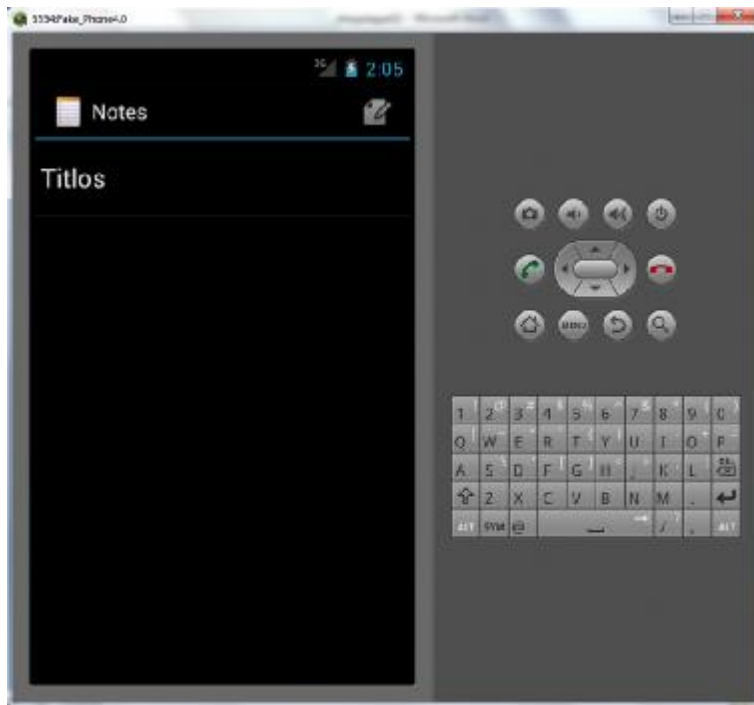
Εκτέλεση της εφαρμογής

Εδώ παρατηρούμε ότι η εφαρμογή φαίνεται ίδια όπως στην προηγούμενη έκδοση καθώς και τα βήματα που ακολουθούμε για να δημιουργήσουμε και να επεξεργαστούμε μια σημείωση.









9.8 Android 4.1 (Jelly Bean), Android 4.2, Android 4.3, Android 4.4

Κώδικας εφαρμογής

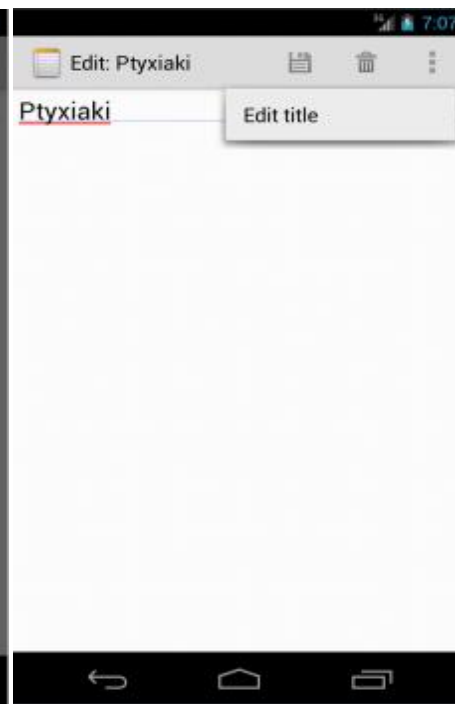
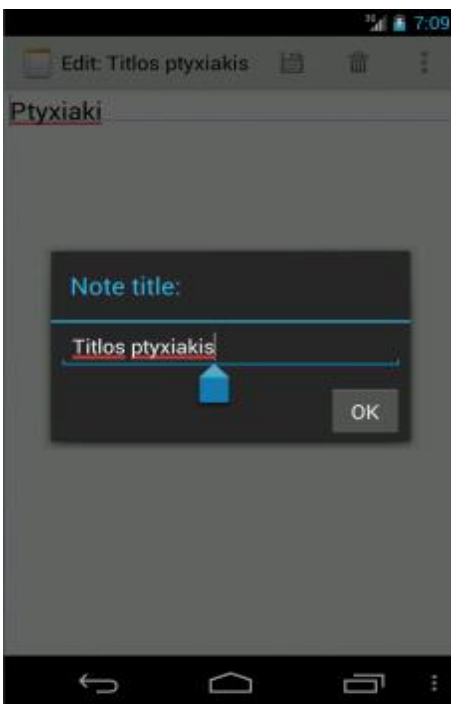
Ο κώδικας δεν αλλάζει καθόλου ούτε σε αυτήν την έκδοση.

Εμφάνιση της εφαρμογής

Ούτε στις ρυθμίσεις της εμφάνισης δεν αλλάζει κάτι. Πάμε να δούμε λοιπόν την εφαρμογή σε αυτή την έκδοση android.

Εκτέλεση της εφαρμογής

Η εφαρμογή έχει την ίδια εμφάνιση με την έκδοση 4.0 και 4.0.3. Δεν αλλάζει τίποτα και λειτουργεί ακριβώς με τον ίδιο τρόπο. Το μόνο που αξίζει να σημειωθεί είναι ότι το περιβάλλον έχει αλλάξει και έχουν βελτιωθεί τα γραφικά.



ΣΥΜΠΕΡΑΣΜΑ

Στην εργασία αυτή είχαμε στόχο να μελετήσουμε τις διάφορες εκδόσεις Android και να κάνουμε μια σύγκριση αυτών. Περιγράψαμε την κάθε έκδοση ξεχωριστά όσον αφορά τα χαρακτηριστικά τους και τις λειτουργίες τους και στην συνέχεια αναλύσαμε μια εφαρμογή, το σημειωματάριο, για να δείξουμε πρακτικά τις διαφορές που υπάρχουν ανά τις εκδόσεις. Από αυτή την ανάλυση καταλήξαμε σε κάποια συμπεράσματα τα οποία αναφέρουμε πιο κάτω.

Όπως είδαμε στην εξέλιξη της πτυχιακής μας εργασίας, η τεχνολογία των Android εξελίσσεται γρήγορα και από έκδοση σε έκδοση παρατηρούμε αλλαγές προς το καλύτερο. Παρατηρούμε ότι στην πρώτη έκδοση Android 1.0 σε σχέση με την Android 1.1 λύθηκαν κάποια προβλήματα με το μικρόφωνο του τηλεφώνου, και επιπλέον πρόσθεσαν κάποιες εφαρμογές. Η έκδοση 1.1 σε σχέση με την 1.5 διαφέρουν πολύ για το λόγο ότι ο πυρήνας Linux άλλαξε, το πληκτρολόγιο έγινε αφής και προστέθηκε λεξικό. Η έκδοση 1.5 με την 1.6 διαφέρουν και εδώ στον πυρήνα Linux, βελτιώνεται η ανάλυση της οθόνης, βελτίωση της ταχύτητας της κάμερας κατά 40% και προστίθεται η υποστήριξη Bluetooth. Η έκδοση 1.6 με την 2.0 και 2.1 διαφέρουν στα γραφικά στοιχεία τα οποία έχουν επανασχεδιαστεί για ταχύτερη επίδοση του συστήματος, βελτιώνεται η έκδοση του Bluetooth σε Bluetooth 2,1, καθώς και η γλώσσα προγραμματισμού σε HTML 5. Στην έκδοση 2.2 έχουμε νέες δυνατότητες για τους χρήστες και τους προγραμματιστές και οι εφαρμογές μπορούν να αποθηκεύονται τώρα και σε εξωτερικές μνήμες όπως micro SD. Στην έκδοση 2.3 αναβαθμίζεται ο πυρήνας και έχουμε ταχύτερη λειτουργία του συστήματος. Στην 3.0 η πλατφόρμα είναι σχεδιασμένη για συσκευές με μεγάλες οθόνες όπως τα tablets γι' αυτό και αλλάζει όλη η εικόνα του συστήματος σε σχέση με τις προηγούμενες εκδόσεις. Έχουμε 5 αρχικές οθόνες και στην οθόνη εμφανίζονται μικρογραφίες των εγκατεστημένων εφαρμογών για πιο εύκολη εκκίνηση. Στην έκδοση 4.0 έχουμε εξελιγμένη, πιο εύχρηστη και υψηλής ανάλυσης αρχική οθόνη καθώς και πιο γρήγορη λειτουργία των εφαρμογών και της κάμερας. Η έκδοση 4.1 παρατηρούμε ότι είναι πιο γρήγορη από τις προηγούμενες εκδόσεις και έχει βελτιωθεί ολόκληρη η πλατφόρμα. Οι εκδόσεις 4.2 και 4.3 έχουν τις ίδιες δυνατότητες με την 4.1 με λίγες προσθήκες όπως νέες δυνατότητες όπως

αλλαγή μεγέθους του ψηφιακού ρολογιού για να εμφανιστούν και ώρες από πόλεις σε όλον τον κόσμο, δίνει τη δυνατότητα για πολλαπλούς χρήστες στα tablets, και άλλα τα οποία έχουμε καταγράψει στο κεφάλαιο 3.12. Τέλος, στην έκδοση 4.4 έχουμε περισσότερους διαθέσιμους πόρους για το τρέξιμο των εφαρμογών και μεγαλύτερη ταχύτητα που δίνει την ευκολία για πολλές ταυτόχρονες διαδικασίες.

Συνοψίζοντας, παρατηρούμε ότι με την πάροδο των χρόνων η τεχνολογία εξελίσσεται και όσον αφορά τα Android συστήματα προστίθενται ολοένα και καινούρια χαρακτηριστικά και λειτουργίες για την διευκόλυνση του χρήστη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- ✓ Shane Conder-Lauren Darcey (2010): Learn Java for Android Development: Introduction to Java. <http://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604>
- ✓ Jeff Friesen (2010): Learn Java for Android Development
- ✓ Reto Meier(2009): Professional Android 4 Application Development

ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

- ✓ <http://el.wikipedia.org/wiki/Android>
- ✓ Android- discover android: <http://www.android.com/about/>
- ✓ Η ιστορία του android: <http://www.allaboutandroid.gr/?p=6362>
- ✓ Τι είναι το android; : <http://www.adds.gr/company/technology/what-is-android/>
- ✓ Ορολογία του android: <http://www.myphone.gr/forum/showthread.php?t=306146#post3645021>
- ✓ Τελευταία νέα και εξελίξεις για android: <http://tech.in.gr/tag-search/?tag=Android>
- ✓ History of android: <http://www.tech2crack.com/history-android/>
- ✓ The complete history of android (infographic): <http://www.bitrebels.com/technology/the-complete-android-history-timeline-infographic/>
- ✓ Google Android's internal structure: A look into google android's internal structure sim: <http://www.brighthub.com/mobile/google-android/articles/17822.aspx>
- ✓ Android application development: <http://skogberg.eu/android/#whatisandroid>

- ✓ The android source code | Android Developers:
<http://source.android.com/source/index.html>
- ✓ <uses-sdk> | Android Developers:
<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>
- ✓ What's new in android 4.1 Jelly Bean:
<http://lifehacker.com/5921761/whats-new-in-android-41-jelly-bean>
- ✓ Android everywhere: 10 types of devices that android is making better:
<http://www.androidauthority.com/android-everywhere-10-types-of-devices-that-android-is-making-better-57012/>
- ✓ Androids are everywhere: 15 Android-powered devices :
<http://www.foxnews.com/tech/slideshow/2012/09/26/15-android-powered-devices/#slide=15>
- ✓ What is Smartwatch? A definition from webopedia:
<http://www.webopedia.com/TERM/S/smartwatch.html>
- ✓ Android Kit Kat | Android Developers:
<http://developer.android.com/about/versions/kitkat.html>
- ✓ Android Kit Kat review: <http://www.techradar.com/reviews/pc-mac/software/operating-systems/android-4-4-kitkat-1214798/review#articleContent>
- ✓ Android update guide: all the features of every version | Review:
<http://reviews.cnet.co.uk/mobile-phones/android-updates-guide-all-the-features-of-every-version-50003779/>
- ✓ Dalvik Libraries: <http://www.milk.com/kodebase/dalvik-docs-mirror/docs/libraries.html>
- ✓ A blog for android: <http://blog4android.blogspot.in/>
- ✓ Android and Me: <http://androidandme.com/>
- ✓ <https://www.eclipse.org/downloads/>

ΠΑΡΑΡΤΗΜΑ

Κώδικας για την εφαρμογή NoteEditor

```
package com.example.android.notepad;

import com.example.android.notepad.NotePad.Notes;

import android.app.Activity;
import android.content.ComponentName;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Rect;
import android.net.Uri;
import android.os.Bundle;
import android.util.AttributeSet;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;

public class NoteEditor extends Activity {
    private static final String TAG = "Notes";
```

```

private static final String[] PROJECTION = new String[] {
    Notes._ID, // 0
    Notes.NOTE, // 1
};

private static final int COLUMN_INDEX_NOTE = 1;

private static final String ORIGINAL_CONTENT = "origContent";

private static final int REVERT_ID = Menu.FIRST;
private static final int DISCARD_ID = Menu.FIRST + 1;
private static final int DELETE_ID = Menu.FIRST + 2;
private static final int STATE_EDIT = 0;
private static final int STATE_INSERT = 1;

private int mState;

private boolean mNoteOnly = false;

private Uri mUri;

private Cursor mCursor;

private EditText mText;

private String mOriginalContent;

public static class LinedEditText extends EditText {
    private Rect mRect;

```



```

private Paint mPaint;

public LinedEditText(Context context, AttributeSet attrs) {
    super(context, attrs);

    mRect = new Rect();
    mPaint = new Paint();
    mPaint.setStyle(Paint.Style.STROKE);
    mPaint.setColor(0x800000FF);
}

@Override
protected void onDraw(Canvas canvas) {
    int count = getLineCount();
    Rect r = mRect;
    Paint paint = mPaint;

    for (int i = 0; i < count; i++) {
        int baseline = getLineBounds(i, r);

        canvas.drawLine(r.left, baseline + 1, r.right, baseline +
1, paint);
    }

    super.onDraw(canvas);
}
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    final Intent intent = getIntent();

    final String action = intent.getAction();
    if (Intent.ACTION_EDIT.equals(action)) {
        mState = STATE_EDIT;
        mUri = intent.getData();
    } else if (Intent.ACTION_INSERT.equals(action)) {
        // στον υποδοχέα.
        mState = STATE_INSERT;
        mUri = getResolver().insert(intent.getData(), null);

        if (mUri == null) {
            Log.e(TAG, "Failed to insert new note into " +
                getIntent().getData());
            finish();
            return;
        }

        setResult(RESULT_OK,
            Intent().setAction(mUri.toString()));
    }
}

```

```

    } else {
        Log.e(TAG, "Unknown action, exiting");
        finish();
        return;
    }

    setContentView(R.layout.note_editor);

    mText = (EditText) findViewById(R.id.note);

    mCursor = managedQuery(mUri, PROJECTION, null, null, null);

    if (savedInstanceState != null) {
        mOriginalContent =
savedInstanceState.getString(ORIGINAL_CONTENT);
    }
}

@Override
protected void onResume() {
    super.onResume();

    if (mCursor != null) {
        mCursor.moveToFirst();

        if (mState == STATE_EDIT) {
            setTitle(getText(R.string.title_edit));

```

```

    } else if (mState == STATE_INSERT) {
        setTitle(getText(R.string.title_create));
    }

    String note = mCursor.getString(COLUMN_INDEX_NOTE);
    mText.setTextKeepState(note);

    if (mOriginalContent == null) {
        mOriginalContent = note;
    }

} else {
    setTitle(getText(R.string.error_title));
    mText.setText(getText(R.string.error_message));
}

}

@Override
protected void onSaveInstanceState(Bundle outState) {
    // Save away the original text, so we still have it if the
    activity
        outState.putString(ORIGINAL_CONTENT, mOriginalContent);
}

@Override
protected void onPause() {
    super.onPause();
}

```

```

if (mCursor != null) {

    String text = mText.getText().toString();

    int length = text.length();

    if (isFinishing() && (length == 0) && !mNoteOnly) {

        setResult(RESULT_CANCELED);

        deleteNote();

    } else {

        ContentValues values = new ContentValues();

        if (!mNoteOnly) {

            values.put(Notes.MODIFIED_DATE,
System.currentTimeMillis());

            if (mState == STATE_INSERT) {

                String title = text.substring(0, Math.min(30,
length));

                if (length > 30) {

                    int lastSpace = title.lastIndexOf(' ');

                    if (lastSpace > 0) {

                        title = title.substring(0, lastSpace);

                    }

                }

                values.put(Notes.TITLE, title);

            }

        }

    }
}

```

```

        values.put(Notes.NOTE, text);

        getContentResolver().update(mUri, values, null, null);
    }
}
}

```

@Override

```

public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    if (mState == STATE_EDIT) {
        menu.add(0, REVERT_ID, 0, R.string.menu_revert)
            .setShortcut('0', 'r')
            .setIcon(android.R.drawable.ic_menu_revert);
        if (!mNoteOnly) {
            menu.add(0, DELETE_ID, 0, R.string.menu_delete)
                .setShortcut('1', 'd')
                .setIcon(android.R.drawable.ic_menu_delete);
        }
    } else {
        menu.add(0, DISCARD_ID, 0, R.string.menu_discard)
            .setShortcut('0', 'd')
            .setIcon(android.R.drawable.ic_menu_delete);
    }
    if (!mNoteOnly) {

```

```

        Intent intent = new Intent(null, getIntent().getData());
        intent.addCategory(Intent.CATEGORY_ALTERNATIVE);
        menu.addIntentOptions(Menu.CATEGORY_ALTERNATIVE, 0, 0,
            intent, 0, null);
        new ComponentName(this, NoteEditor.class), null,
    }

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case DELETE_ID:
            deleteNote();
            finish();
            break;
        case DISCARD_ID:
            cancelNote();
            break;
        case REVERT_ID:
            cancelNote();
            break;
    }
    return super.onOptionsItemSelected(item);
}

private final void cancelNote() {

```

```

if (mCursor != null) {
    if (mState == STATE_EDIT) {
        mCursor.close();
        mCursor = null;

        ContentValues values = new ContentValues();
        values.put(Notes.NOTE, mOriginalContent);
        getContentResolver().update(mUri, values, null, null);
    } else if (mState == STATE_INSERT) {

        deleteNote();
    }
}

setResult(RESULT_CANCELED);
finish();
}

private final void deleteNote() {
    if (mCursor != null) {
        mCursor.close();
        mCursor = null;

        getContentResolver().delete(mUri, null, null);
        mText.setText("");
    }
}
}

```

Κώδικας για την εφαρμογή NotePad


```

package com.example.android.notepad;

import android.net.Uri;
import android.provider.BaseColumns;

public final class NotePad {

    public static final String AUTHORITY = "com.google.provider.NotePad";

    private NotePad() {}

    public static final class Notes implements BaseColumns {

        private Notes() {}

        public static final Uri CONTENT_URI = Uri.parse("content://" +
AUTHORITY + "/notes");

        public static final String CONTENT_TYPE =
"vnd.android.cursor.dir/vnd.google.note";

        public static final String CONTENT_ITEM_TYPE =
"vnd.android.cursor.item/vnd.google.note";

        public static final String DEFAULT_SORT_ORDER = "modified DESC";

        public static final String TITLE = "title";

```

```

        public static final String NOTE = "note";

        public static final String CREATED_DATE = "created";

        public static final String MODIFIED_DATE = "modified";
    }
}

```

Κώδικας για την εφαρμογή NotePadProvider

```

package com.example.android.notepad;

import com.example.android.notepad.NotePad.Notes;

import android.content.ContentProvider;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.UriMatcher;
import android.content.res.Resources;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteQueryBuilder;
import android.net.Uri;
import android.text.TextUtils;

```

```

import android.util.Log;

import java.util.HashMap;

public class NotePadProvider extends ContentProvider {

    private static final String TAG = "NotePadProvider";

    private static final String DATABASE_NAME = "note_pad.db";
    private static final int DATABASE_VERSION = 2;
    private static final String NOTES_TABLE_NAME = "notes";

    private static HashMap<String, String> sNotesProjectionMap;

    private static final int NOTES = 1;
    private static final int NOTE_ID = 2;

    private static final UriMatcher sUriMatcher;

    private static class DatabaseHelper extends SQLiteOpenHelper {

        DatabaseHelper(Context context) {

            super(context, DATABASE_NAME, null, DATABASE_VERSION);

        }
    }

```

```

@Override

public void onCreate(SQLiteDatabase db) {

    db.execSQL("CREATE TABLE " + NOTES_TABLE_NAME + " ("

        + Notes._ID + " INTEGER PRIMARY KEY, "

        + Notes.TITLE + " TEXT, "

        + Notes.NOTE + " TEXT, "

        + Notes.CREATED_DATE + " INTEGER, "

        + Notes.MODIFIED_DATE + " INTEGER"

        + ");");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

    Log.w(TAG, "Upgrading database from version " + oldVersion +
" to "

        + newVersion + ", which will destroy all old data");

    db.execSQL("DROP TABLE IF EXISTS notes");

    onCreate(db);

}

}

private DatabaseHelper mOpenHelper;

@Override

public boolean onCreate() {

    mOpenHelper = new DatabaseHelper(getContext());

    return true;

```

```

    }

    @Override
    public Cursor query(Uri uri, String[] projection, String selection,
        String[] selectionArgs,
            String sortOrder) {
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

        switch (sUriMatcher.match(uri)) {
            case NOTES:
                qb.setTables(NOTES_TABLE_NAME);
                qb.setProjectionMap(sNotesProjectionMap);
                break;

            case NOTE_ID:
                qb.setTables(NOTES_TABLE_NAME);
                qb.setProjectionMap(sNotesProjectionMap);
                qb.appendWhere(Notes._ID + " = " +
                    uri.getPathSegments().get(1));
                break;

            default:
                throw new IllegalArgumentException("Unknown URI " + uri);
        }

        String orderBy;

        if (TextUtils.isEmpty(sortOrder)) {
            orderBy = NotePad.Notes.DEFAULT_SORT_ORDER;
        } else {

```

```

        orderBy = sortOrder;
    }

    SQLiteDatabase db = mOpenHelper.getReadableDatabase();

    Cursor c = qb.query(db, projection, selection, selectionArgs,
null, null, orderBy);

    c.setNotificationUri(getContext().getContentResolver(), uri);

    return c;
}

@Override

public String getType(Uri uri) {
    switch (sUriMatcher.match(uri)) {
        case NOTES:
            return Notes.CONTENT_TYPE;

        case NOTE_ID:
            return Notes.CONTENT_ITEM_TYPE;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
}

@Override

public Uri insert(Uri uri, ContentValues initialValues) {

```

```

if (sUriMatcher.match(uri) != NOTES) {
    throw new IllegalArgumentException("Unknown URI " + uri);
}

ContentValues values;

if (initialValues != null) {
    values = new ContentValues(initialValues);
} else {
    values = new ContentValues();
}

Long now = Long.valueOf(System.currentTimeMillis());

if (values.containsKey(NotePad.Notes.CREATED_DATE) == false) {
    values.put(NotePad.Notes.CREATED_DATE, now);
}

if (values.containsKey(NotePad.Notes.MODIFIED_DATE) == false) {
    values.put(NotePad.Notes.MODIFIED_DATE, now);
}

if (values.containsKey(NotePad.Notes.TITLE) == false) {
    Resources r = Resources.getSystem();

    values.put(NotePad.Notes.TITLE,
r.getString(android.R.string.untitled));
}

```

```

    if (values.containsKey(NotePad.Notes.NOTE) == false) {
        values.put(NotePad.Notes.NOTE, "");
    }

    SQLiteDatabase db = mOpenHelper.getWritableDatabase();
    long rowId = db.insert(NOTES_TABLE_NAME, Notes.NOTE, values);
    if (rowId > 0) {
        Uri noteUri =
ContentUris.withAppendedId(NotePad.Notes.CONTENT_URI, rowId);
        getContext().getContentResolver().notifyChange(noteUri,
null);
        return noteUri;
    }

    throw new SQLException("Failed to insert row into " + uri);
}

@Override
public int delete(Uri uri, String where, String[] whereArgs) {
    SQLiteDatabase db = mOpenHelper.getWritableDatabase();
    int count;
    switch (sUriMatcher.match(uri)) {
        case NOTES:
            count = db.delete(NOTES_TABLE_NAME, where, whereArgs);
            break;

        case NOTE_ID:
            String noteId = uri.getPathSegments().get(1);

```



```

        count = db.delete(NOTES_TABLE_NAME, Notes._ID + "=" + noteId
+ (!TextUtils.isEmpty(where) ? " AND (" + where + ')'"
: ""), whereArgs);

        break;

    default:

        throw new IllegalArgumentException("Unknown URI " + uri);

    }

    getContext().getContentResolver().notifyChange(uri, null);

    return count;

}

@Override

public int update(Uri uri, ContentValues values, String where,
String[] whereArgs) {

    SQLiteDatabase db = mOpenHelper.getWritableDatabase();

    int count;

    switch (sUriMatcher.match(uri)) {

        case NOTES:

            count = db.update(NOTES_TABLE_NAME, values, where,
whereArgs);

            break;

        case NOTE_ID:

            String noteId = uri.getPathSegments().get(1);

            count = db.update(NOTES_TABLE_NAME, values, Notes._ID + "=" +
noteId

```

```

        + (!TextUtils.isEmpty(where) ? " AND (" + where + ') '
: ""), whereArgs);
        break;

    default:
        throw new IllegalArgumentException("Unknown URI " + uri);
    }

    getContext().getContentResolver().notifyChange(uri, null);

    return count;
}

static {
    sUriMatcher = new UriMatcher(UriMatcher.NO_MATCH);

    sUriMatcher.addURI(NotePad.AUTHORITY, "notes", NOTES);
    sUriMatcher.addURI(NotePad.AUTHORITY, "notes/#", NOTE_ID);

    sNotesProjectionMap = new HashMap<String, String>();
    sNotesProjectionMap.put(Notes._ID, Notes._ID);
    sNotesProjectionMap.put(Notes.TITLE, Notes.TITLE);
    sNotesProjectionMap.put(Notes.NOTE, Notes.NOTE);
    sNotesProjectionMap.put(Notes.CREATED_DATE, Notes.CREATED_DATE);
    sNotesProjectionMap.put(Notes.MODIFIED_DATE,
Notes.MODIFIED_DATE);
}
}

```

Κώδικας για την εφαρμογή NotesList

```
package com.example.android.notepad;

import com.example.android.notepad.NotePad.Notes;

import android.app.ListActivity;
import android.content.ComponentName;
import android.content.ContentUris;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ContextMenu.ContextMenuInfo;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;

public class NotesList extends ListActivity {

    private static final String TAG = "NotesList";

    public static final int MENU_ITEM_DELETE = Menu.FIRST;
```

```

public static final int MENU_ITEM_INSERT = Menu.FIRST + 1;

private static final String[] PROJECTION = new String[] {

    Notes._ID, // 0

    Notes.TITLE, // 1

};

private static final int COLUMN_INDEX_TITLE = 1;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setDefaultCloseOperation(DEFAULT_KEYS_SHORTCUT);

    Intent intent = getIntent();

    if (intent.getData() == null) {

        intent.setData(Notes.CONTENT_URI);

    }

    listView().setOnCreateContextMenuListener(this);

    Cursor cursor = managedQuery(getIntent().getData(), PROJECTION,
null, null,

        Notes.DEFAULT_SORT_ORDER);

    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
R.layout.notes_list_item, cursor,

```

```

        new String[] { Notes.TITLE }, new int[] {
android.R.id.text1 });

        setListAdapter(adapter);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);

        menu.add(0, MENU_ITEM_INSERT, 0, R.string.menu_insert)
            .setShortcut('3', 'a')
            .setIcon(android.R.drawable.ic_menu_add);

        Intent intent = new Intent(null, getIntent().getData());
        intent.addCategory(Intent.CATEGORY_ALTERNATIVE);
        menu.addIntentOptions(Menu.CATEGORY_ALTERNATIVE, 0, 0,
            new ComponentName(this, NotesList.class), null, intent,
0, null);

        return true;
    }

    @Override
    public boolean onPrepareOptionsMenu(Menu menu) {
        super.onPrepareOptionsMenu(menu);

        final boolean haveItems = getListAdapter().getCount() > 0;

```

```

        if (haveItems) {

            Uri uri = ContentUris.withAppendedId(getIntent().getData(),
getSelectedItemId());

            Intent[] specifics = new Intent[1];

            specifics[0] = new Intent(Intent.ACTION_EDIT, uri);

            MenuItem[] items = new MenuItem[1];

            Intent intent = new Intent(null, uri);

            intent.addCategory(Intent.CATEGORY_ALTERNATIVE);

            menu.addIntentOptions(Menu.CATEGORY_ALTERNATIVE, 0, 0, null,
specifics, intent, 0,

                items);

            if (items[0] != null) {

                items[0].setShortcut('1', 'e');

            }

        } else {

            menu.removeGroup(Menu.CATEGORY_ALTERNATIVE);

        }

        return true;

    }

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

```

```

switch (item.getItemId()) {

    case MENU_ITEM_INSERT:

        startActivity(new Intent(Intent.ACTION_INSERT,
getIntent().getData()));

        return true;

    }

    return super.onOptionsItemSelected(item);

}

@Override

public void onCreateContextMenu(ContextMenu menu, View view,
ContextMenuInfo menuInfo) {

    AdapterView.AdapterContextMenuInfo info;

    try {

        info = (AdapterView.AdapterContextMenuInfo) menuInfo;

    } catch (ClassCastException e) {

        Log.e(TAG, "bad menuInfo", e);

        return;

    }

    Cursor cursor = (Cursor) getListAdapter().getItem(info.position);

    if (cursor == null) {

        return;

    }

    menu.setHeaderTitle(cursor.getString(COLUMN_INDEX_TITLE));

```

```

        menu.add(0, MENU_ITEM_DELETE, 0, R.string.menu_delete);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        AdapterView.AdapterContextMenuInfo info;

        try {
            info = (AdapterView.AdapterContextMenuInfo)
item.getMenuInfo();
        } catch (ClassCastException e) {
            Log.e(TAG, "bad menuInfo", e);
            return false;
        }

        switch (item.getItemId()) {
            case MENU_ITEM_DELETE: {
                Uri noteUri =
ContentUris.withAppendedId(getIntent().getData(), info.id);
                getContentResolver().delete(noteUri, null, null);
                return true;
            }
        }
        return false;
    }

    @Override

```



```

protected void onItemClick(ListView l, View v, int position, long
id) {

    Uri uri = ContentUris.withAppendedId(getIntent().getData(), id);

    String action = getIntent().getAction();

    if (Intent.ACTION_PICK.equals(action) ||
Intent.ACTION_GET_CONTENT.equals(action)) {

        setResult(RESULT_OK, new Intent().setData(uri));

    } else {

        startActivity(new Intent(Intent.ACTION_EDIT, uri));

    }

}
}
}
}

```

Κώδικας για την εφαρμογή TitleEditor

```

package com.example.android.notepad;

import com.example.android.notepad.NotePad.Notes;

import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

```

```

import android.widget.EditText;

public class TitleEditor extends Activity implements View.OnClickListener
{

    public static final String EDIT_TITLE_ACTION =
"com.android.notepad.action.EDIT_TITLE";

    private static final String[] PROJECTION = new String[] {

        NotePad.Notes._ID, // 0

        NotePad.Notes.TITLE, // 1

    };

    private static final int COLUMN_INDEX_TITLE = 1;

    private Cursor mCursor;

    /**

    private EditText mText;

    private Uri mUri;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.title_editor);

        mUri = getIntent().getData();

        mCursor = managedQuery(mUri, PROJECTION, null, null, null);

```

```

        mText = (EditText) this.findViewById(R.id.title);
        mText.setOnClickListener(this);

        Button b = (Button) findViewById(R.id.ok);
        b.setOnClickListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();

        if (mCursor != null) {
            mCursor.moveToFirst();
            mText.setText(mCursor.getString(COLUMN_INDEX_TITLE));
        }
    }

    @Override
    protected void onPause() {
        super.onPause();

        if (mCursor != null) {
            // Write the title back to the note
            ContentValues values = new ContentValues();

```

```
        values.put(Notes.TITLE, mText.getText().toString());
        getResolver().update(mUri, values, null, null);
    }
}

public void onClick(View v) {

    finish();
}
}
```