



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Εκπαιδευτικό quiz για φορητές συσκευές</b>
Όνοματεπώνυμο Φοιτητή	<b>Μεσσάρη Ειρήνη</b>
Πατρώνυμο	<b>Σπυρίδων</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/12034</b>
Επιβλέπων	<b>Αλέπης Ευθύμιος, Λέκτορας</b>

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

Βίρβου Μαρία  
Καθηγήτρια

(υπογραφή)

Τσιχριτζής Γεώργιος  
Καθηγητής

(υπογραφή)

Αλέπης Ευθύμιος  
Λέκτορας

## ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας εφαρμογής Android. Η εφαρμογή επιτρέπει στο κάτοχο κινητού τηλεφώνου με το λειτουργικό σύστημα Android να παίζει ένα παιχνίδι ερωταπαντήσεων για να εξετάσει τις γνώσεις του.



# Πίνακας Περιεχομένων

## Κεφάλαιο 1: Εισαγωγή

1.1 Αντικείμενο διπλωματικής.....	9
1.2 Δομή κεφαλαίων.....	9

## Κεφάλαιο 2: Το λειτουργικό σύστημα Android

2.1 Τι είναι το Android .....	11
2.2 Εκδόσεις του Android .....	14
2.2.1. Android 1.5 ( Cupcake ).....	15
2.2.2. Android 1.6 (Donut).....	15
2.2.3. Android 2.0 (Éclair) .....	16
2.2.4. Android 2.2 (Froyo) .....	16
2.2.5. Android 2.3 (Gingerbread).....	17
2.2.6. Android 3.0 (Honeycomb) .....	18
2.2.7. Android 4.0 (Ice Scream Sandwich) .....	18
2.2.8. Android 4.1 (Jelly Bean).....	19
2.2.9. Android 4.4 (Kit kat).....	19
2.2.10. Οι εκδόσεις που χρησιμοποιούνται περισσότερο.....	20
2.3 Αρχιτεκτονική Android .....	22
2.3.1. Επίπεδο Εφαρμογών (Blue layer-Application) .....	22
2.3.2. Επίπεδο Πλαισίου Εφαρμογών (Application Framework).....	23
2.3.3. Επίπεδο Βιβλιοθηκών (Green layer-Libraries) .....	26
2.3.4. Επίπεδο Εκτέλεσης (Yellow layer- Runtime Android).....	27
2.3.5. Ο πυρήνας του Linux (Red layer-Kernel).....	27
2.4. Συνιστώσες εφαρμογής Android.....	28
2.4.1. Δραστηριότητες(activities).....	28
2.4.2. Υπηρεσίες(services).....	28
2.4.3. Πάροχοι περιεχομένου (content providers).....	28
2.4.4. Προθέσεις (Intents).....	29
2.4.5. Καθολικοί παραλήπτες μηνυμάτων (broadcast receivers).....	29

2.5. Διεπαφή χρήστη (user interface).....	30
2.5.1. Διάταξη γραφικών στοιχείων οθόνης (layout).....	30
2.5.2. Παράθυρα διαλόγου(dialogs).....	35
2.5.2.1. Alert Dialog.....	35
2.5.2.2. ProgressDialog .....	36
2.5.3. Ειδοποιήσεις χρήστη(notifications).....	37
2.5.3.1. Toast notification.....	37
2.5.3.1. Status bar notification.....	38

### **Κεφάλαιο 3: Προγραμματισμός σε περιβάλλον Android**

3.1 Εργαλεία προγραμματισμού.....	39
3.1.1. Δημιουργία ενός Android Development Environment.....	39
3.1.2. Δημιουργία εικονικής συσκευής Android .....	41
3.1.3. Δημιουργία Android εφαρμογής.....	44

### **Κεφάλαιο 4: Η εφαρμογή**

4.1 Περιγραφή.....	47.
4.2 Δομή.....	47.
4.2.1. Φάκελος gen /.....	47
4.2.2. Φάκελος res /.....	48
4.2.3. Φάκελος bin /.....	61
4.2.4. Φάκελος src /.....	62
4.2.5. Το αρχείο AndroidManifest.xml.....	89
4.2.6. Το αρχείο project.properties.....	90
4.2.7. Φάκελος libs /.....	90

### **Κεφάλαιο 5: Συμπεράσματα**

5.1 Συμπεράσματα.....	91
5.2 Μελλοντικές εξελίξεις.....	91

Βιβλιογραφία.....	92
Ηλεκτρονική βιβλιογραφία.....	93

## Κατάλογος Εικόνων

### Κεφάλαιο 2

Εικόνα 2.1. Λογότυπο Android .....	11
Εικόνα 2.2. Open Handset Alliance (OHA).....	12
Εικόνα 2.3. Χρονοδιάγραμμα των εκδόσεων του Android .....	14
Εικόνα 2.4 Το λογότυπο του Cupcake .....	15
Εικόνα 2.5. Το λογότυπο του Donut .....	15
Εικόνα 2.6. Το λογότυπο του Éclair .....	16
Εικόνα 2.7. Το λογότυπο του Froyo.....	16
Εικόνα 2.8. Το λογότυπο του Gingerbread .....	17
Εικόνα 2.9. Το λογότυπο του Honeycomb .....	18
Εικόνα 2.10. Το λογότυπο του Ice Cream Sandwich.....	18
Εικόνα 2.11. Το λογότυπο του Jelly Bean .....	19
Εικόνα 2.12 .Το λογότυπο του Kit kat .....	19
Εικόνα 2.13. Αρχιτεκτονική Android .....	22
Εικόνα 2.14. Παράδειγμα LinearLayout.....	24
Εικόνα 2.15. Παράδειγμα Relative Layout.....	31
Εικόνα 2.16. Παράδειγμα Frame Layout .....	32
Εικόνα 2.17.Παράδειγμα Table Layout.....	33
Εικόνα 2.18Παράδειγμα Alert Dialog.....	34
Εικόνα 2.19 Παράδειγμα ProgressDialog.....	36
Εικόνα 2.20. Παράδειγμα toastnotification.....	36
Εικόνα 2.21. Παράδειγμα status bar notification.....	37

### Κεφάλαιο 3

Εικόνα 3.1.JDK .....	39
Εικόνα 3.2.Eclipse IDE.....	40
Εικόνα 3.3.SDK Manager.....	40
Εικόνα 3.4.Επιλογή χώρου εργασίας.....	41
Εικόνα 3.5.AVD Manager.....	42
Εικόνα 3.6.Παράθυρο δημιουργίας νέας Εικονικής μηχανής.....	43
Εικόνα 3.7.Εικονική συσκευή.....	44
Εικόνα 3.8.Τα στάδια δημιουργίας ενός νέου project.....	44
Εικόνα 3.9 .Android Virtual Device ή AVD.....	46
Εικόνα 3.10. Λογότυπο Eclipse.....	46

### Κεφάλαιο 4

Εικόνα 4.1.Δομή της εφαρμογής Spiel.....	47
Εικόνα 4.2.Περιεχόμενα φακέλου gen/.....	48
Εικόνα 4.3.Περιεχόμενα φακέλου res / .....	48
Εικόνα 4.4 .background.xml.....	49
Εικόνα 4.5.button-selector.xml.....	49
Εικόνα 4.6.Περιεχόμενα φακέλου layout.....	50
Εικόνα 4.7.Περιεχόμενα υποφακέλου welcome.xml φακέλου layout.....	52
Εικόνα 4.8. Περιεχόμενα υποφακέλου question.xml φακέλου layout.....	54
Εικόνα 4.9. Περιεχόμενα υποφακέλου endgame.xml φακέλου layout.....	57
Εικόνα 4.10.Σωστές απαντήσεις.....	58
Εικόνα 4.11.Περιεχόμενα υποφακέλου setting.xml φακέλου layout.....	59
Εικόνα 4.12Περιεχόμενα υποφακέλου rules.xml φακέλου layout.....	60
Εικόνα 4.13.Περιεχόμενα φακέλου values.....	61
Εικόνα 4.14.Περιεχόμενα υποφακέλου endgame.xml φακέλου values.....	61
Εικόνα 4.15. Περιεχόμενα φακέλου src /.....	62
Εικόνα 4.16. .Περιεχόμενα φακέλου AndroidManifest.xml.....	89

## **Κατάλογος Πινάκων**

Κεφάλαιο 4

Πίνακας 4.1. Icon size

## **Κατάλογος Σχημάτων**

Κεφάλαιο 2

Σχήμα 2.1. Κατανομή των εκδόσεων του Android (Μάιος)

Σχήμα 2.2. Κατανομή των εκδόσεων του Android (Αύγουστος)

Σχήμα 2.3. Ο κύκλος ζωής μιας δραστηριότητας Android

Σχήμα 2.4. Παράδειγμα διαχείρισης πόρων



# ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

## 1.1 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας έξυπνης εφαρμογής παιχνιδιού για το λειτουργικό σύστημα Android. Η εφαρμογή αυτή θα επιτρέψει στο χρήστη

## 1.2 Δομή κεφαλαίων

Η διπλωματική αυτή εργασία αναλύεται σε κεφάλαια. Παρακάτω αναφέρεται περιληπτικά τι περιγράφεται σε κάθε ένα κεφάλαιο.

### Κεφάλαιο 1

Στο κεφάλαιο αυτό παρουσιάζεται το αντικείμενο που πραγματεύεται η διπλωματική εργασία και αναφέρονται τα περιεχόμενα κάθε κεφαλαίου.

### Κεφάλαιο 2

Στο κεφάλαιο αυτό θα περιγραφεί το λειτουργικό σύστημα Android ,τα βασικά στοιχεία της αρχιτεκτονικής του καθώς και κάποια βασικά χαρακτηριστικά παλαιότερων εκδόσεων του λειτουργικού συστήματος έως και τη σημερινή έκδοση,οι συνιστώσες που αποτελούν μια τέτοια εφαρμογή καθώς και τη διεπαφή χρήστη.

### Κεφάλαιο 3

Στο κεφάλαιο αυτό θα περιγραφούν τα εργαλεία που χρησιμοποιούνται για τη δημιουργία μιας εφαρμογής για android

### Κεφάλαιο 4

Στο κεφάλαιο αυτό θα περιγραφούν τα δομικά στοιχεία της εφαρμογής,τα layouts,οι κλάσεις της java.

### Κεφάλαιο 5

Στο κεφάλαιο αυτό θα αναφερθούν τα συμπεράσματα καθώς και ορισμένες μελλοντικές βελτιώσεις του παιχνιδιού

Τέλος,παρουσιάζεται η **βιβλιογραφία** (έντυπη και ηλεκτρονική) που χρησιμοποιήθηκε για τη συγγραφή της εργασίας.



## ΚΕΦΑΛΑΙΟ 2: ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID

### **2.1 Τι είναι το Android**

Ο όρος Android έχει ελληνική προέλευση, καθώς προέρχεται από την λέξη Andro-«ανθρώπινη» + eides-«μορφή, σχήμα». Συνεπώς, η έννοια που δίνεται στην λέξη Android είναι το «Ανδροειδής» και συμβολίζεται από το ρομπότ με την μορφή ανθρώπου, όπως εμφανίζεται και στην Εικόνα 1. Το Android είναι μια στοίβα λογισμικού για κινητές συσκευές που τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα το πρόγραμμα ανάπτυξης του Android, συνεχίστηκε με την συνεργασία της εταιρίας αυτής με την Open Handset Alliance (OHA). Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA(εικόνα 2) , μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας.Μερικά μέλη του οργανισμού αυτού ,είναι οι Sprint Nextel, T-Mobile,Motorola, Samsung Electronics,LG Electronics,Dell,Sony Ericsson, Vodafone,Google,Intel,Texas Instruments κ.α.Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android, υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.



Εικόνα 2.1. Λογότυπο Android



Εικόνα 2.2. Open Handset Alliance (OHA)

Το Android είναι ένα ανοιχτού κώδικα λειτουργικό σύστημα, ενδιάμεσο λογισμικό και τις βασικές εφαρμογές. Το Android SDK παρέχει τα εργαλεία και τα απαραίτητα APIs για να είναι εφικτή η ανάπτυξη εφαρμογών για την πλατφόρμα Android, χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Το Android είναι ένα ελεύθερο λογισμικό ανοιχτού κώδικα (free and open source software). Αυτό σημαίνει ότι ο κάθε προγραμματιστής (developer) είναι ικανός να τροποποιεί και να προσαρμόζει το λειτουργικό σύστημα για καθεμία συσκευή ξεχωριστά. Έτσι, διαφορετικές κινητές συσκευές που χρησιμοποιούν το λογισμικό Android είναι δυνατό να έχουν διαφορετικά γραφικά για τις διεπαφές χρήστη, παρόλο που χρησιμοποιούν το ίδιο λειτουργικό σύστημα.

Το Android έχει διάφορα χαρακτηριστικά, όπως :

- **Λειτουργίες Οθόνης:** Η πλατφόρμα είναι προσαρμόσιμη για οθόνες μεγέθους από 2.3 ίντσες μέχρι 4.3 ίντσες, διαθέτουν VGA ανάλυση, διδιάστατες ψηφιακές γραφικές βιβλιοθήκες καθώς και τρισδιάστατα γραφικά βασισμένα στα χαρακτηριστικά του OpenGL ES 2.0 [21].
- **Αποθήκευση Δεδομένων:** Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης
- **Συνδεσιμότητα:** Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC και WiMAX.
- **Αποστολή μηνυμάτων:** SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
- **Περιήγηση στον Ιστό:** Για την περιήγηση στον ιστό το Android είναι βασισμένο στην open source τεχνολογία WebKit [20] σε συνδυασμό με την μηχανή JavaScript Chrome V8.
- **Υποστήριξη Java:** Το λογισμικό είναι γραμμένο σε Java, όμως δεν υπάρχει Java Virtual Machine στην πλατφόρμα, ενώ – κατά τα γνωστά – ο Java byte code δεν είναι εκτελέσιμος. Παρόλα αυτά, οι κλάσεις Java μεταγλωττίζονται σε εκτελέσιμα αρχεία Dalvik και τρέχουν στην εικονική μηχανή Dalvik, η οποία είναι

- μια εξειδικευμένη υλοποίηση εικονική μηχανής, σχεδιασμένη για χρήση σε φορητές συσκευές, παρόλο που δεν είναι μια πρότυπη εικονική μηχανή Java.
- **Υποστήριξη Πολυμέσων:** Το λειτουργικό Android υποστηρίζει τις ακόλουθες μορφές ήχου, στατικής και κινούμενης εικόνας: WebM, H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.
  - **Επιπλέον υποστήριξη hardware:** Το λειτουργικό Android μπορεί να υποστηρίξει με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, αισθητήρες εγγύτητας, αισθητήρες πίεσης, γυροσκόπιο, θερμόμετρο, διδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.
  - **Περιβάλλον Ανάπτυξης Λογισμικού:** Περιλαμβάνει έναν προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα add on για το Eclipse IDE.
  - **Αγορά και Εγκατάσταση Εφαρμογών:** Παρόμοια με το App Store του iPhone OS, το Android Market είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και να εγκατασταθούν στην συσκευή άμεσα μέσω WiFi, χωρίς τη χρήση υπολογιστή. Αρχικά μόνο δωρεάν εφαρμογές ήταν διαθέσιμες ενώ από τις 19 Φεβρουαρίου 2009 στο Android Market υπάρχουν εφαρμογές που διατίθενται είτε δωρεάν είτε επί πληρωμή
  - **Οθόνη Αφής Πολλαπλών Σημείων:** Το λειτουργικό Android είχε εξ ορισμού υποστήριξη για οθόνες πολλαπλών σημείων αλλά η δυνατότητα αυτή έχει κλειδωθεί σε επίπεδο πυρήνα (πιθανόν για αποφυγή παραβιάσεων των πατεντών λογισμικού από την Apple στις τεχνολογίες οθονών αφής).

Γιατί Android;

- Είναι μια πραγματικά ανοικτή πλατφόρμα προγραμματισμού που βασίζεται στο Linux.
- Έχει αρχιτεκτονική βασισμένη σε στοιχεία που εμπνεύστηκε από το Internet, δηλαδή κομμάτια μια εφαρμογής μπορούν να χρησιμοποιηθούν με τρόπους που οι αρχικοί προγραμματιστές της δεν είχαν φανταστεί. Μπορεί ακόμα ο προγραμματιστής μιας εφαρμογής να αλλάξει έτοιμα ενσωματωμένα στοιχεία με δικές του βελτιωμένες εκδόσεις.
- Έχει πάρα πολλές έτοιμες υπηρεσίες όπως εντοπισμός θέσης με χρήση τριγωνοποίησης κυψελών ή με χρήση δέκτη GPS. Επίσης έχει πλήρη συνεργασία με βάσεις δεδομένων SQL.
- Παρέχει αυτοματοποιημένη διαχείριση του κύκλου ζωής μιας εφαρμογής.
- Τα προγράμματα είναι απομονωμένα το ένα από το άλλο από πολλαπλά επίπεδα ασφαλείας πράγμα που δεν έχει εφαρμοστεί ξανά σε πλατφόρμες έξυπνων συσκευών και προσδίδει μεγάλη αξιοπιστία.
- Υποστηρίζονται πολύ καλής ποιότητας γραφικά και ήχοι κάνοντας χρήση του προτύπου OpenGL και κωδικοποιητών ήχου και βίντεο.
- Τέλος υπάρχει μεγάλη φορητότητα καθώς υποστηρίζονται αρκετοί τύποι υλικού. Όλα τα προγράμματα είναι γραμμένα σε γλώσσα προγραμματισμού Java και εκτελούνται από την εικονική μηχανή του Android που ονομάζεται Dalvik. Επίσης υποστηρίζονται πολύ τύποι εισόδου όπως πληκτρολόγια, αφή, και οπτικός κέρσορας.

## 2.2. Εκδόσεις του Android



Εικόνα 2.3. Χρονοδιάγραμμα των εκδόσεων του Android

Μέχρι σήμερα το Android έχει εξελιχθεί αρκετές φορές δημιουργώντας μια πληθώρα εκδόσεων διαθέσιμες στο ευρύ κοινό. Βέβαια δεν ενδείκνυται όλες για οποιαδήποτε συσκευή έχει κανείς. Παρακάτω θα αναφέρουμε όλες τις υπάρχουσες εκδόσεις.

Η πρώτη έκδοση του Android SDK που εμφανίστηκε τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση.

Έτσι, η πρώτη εμπορική έκδοση του λογισμικού, κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008. Η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0.

Η πρώτη συσκευή Android, το HTC Dream, ενσωματώνει τα ακόλουθα χαρακτηριστικά του Android 1.0.

- Την εφαρμογή Android Market όπου ο χρήστης μπορεί να κατεβάσει νέες εφαρμογές και να αναβαθμίσει αυτές που ήδη έχει.
- Browser για την διαχείριση ιστοσελίδων
- Υποστήριξη κάμερας
- Πρόσβαση σε email-servers, υποστηρίζοντας τα πρωτόκολλα POP-3, IMAP4 και SMTP.
- Ενσωμάτωσε τα Google Contacts, Calendar, Maps, Search, Talk.
- Υποστήριξη Wi-Fi και Bluetooth.

Στις 9 Φεβρουαρίου 2009, το Android 1.1 κυκλοφόρησε, αρχικά μόνο για το T-Mobile G.. Η ενημέρωση έλυσε σφάλματα, άλλαξε το API και πρόσθεσε μια σειρά από άλλα χαρακτηριστικά:

- Δυνατότητα να αποθηκεύσετε τα συνημμένα σε μηνύματα
- Λεπτομέρειες και κριτικές όταν ένας χρήστης ψάχνει για τις επιχειρήσεις στο Google Maps
- Προστέθηκε υποστήριξη για μαρκίζα σε διατάξεις του συστήματος.

Από τον Απρίλιο του 2009, κάθε έκδοση του Android έχει πάρει την κωδική ονομασία της από το όνομα ενός επιδορπίου. Αυτές οι εκδόσεις έχουν κυκλοφορήσει με αλφαβητική σειρά: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb και Ice Cream Sandwich, Jelly Bean, Kit kat. Οι προηγούμενες εκδόσεις του Android είχαν βαπτιστεί Astro και Bender αλλά δεν μπορούσαν να χρησιμοποιηθούν για εμπορικούς λόγους, για αυτό το λόγο έχουν μείνει χωρίς κωδικές ονομασίες.

### 2.2.1. Android 1.5 ( Cupcake )



Εικόνα 2.4. Το λογότυπο του Cupcake

Τον Απρίλιο του 2009 κυκλοφόρησε η έκδοση Android 1.5, ονόματι 'Cupcake' το λογότυπο της οποίας φαίνεται και στην Εικόνα 5 . Το 'Cupcake' εισάγει κάποια καινούργια χαρακτηριστικά και ανανεώσεις στην διεπιφάνεια χρήστη (User Interface):

- Υποστήριξη για Widgets – μικροσκοπική θεώρηση της εφαρμογής που μπορεί να ενσωματωθεί σε άλλες εφαρμογές (όπως στην αρχική οθόνη) και να λαμβάνει περιοδικές ενημερώσεις.
- Έξυπνο πληκτρολόγιο
- Προστίθενται χαρακτηριστικά αντιγραφής και επικόλλησης στο πρόγραμμα περιήγησης στο Web.
- Δυνατότητα να ανεβάσετε βίντεο στο YouTube.
- Υποστήριξη Bluetooth A2DP και AVRCP
- Ικανότητα για καταγραφή και παρακολούθηση βίντεο μέσα από την λειτουργία της βιντεοκάμερας, μεταφόρτωση βίντεο στο YouTube και φωτογραφιών στο Picasa απευθείας από το τηλέφωνο, καινούργιο μαλακό πληκτρολόγιο (αφής) με πρόβλεψη κειμένου

### 2.2.2. Android 1.6 ( Donut )

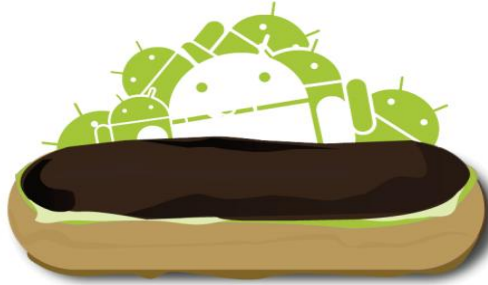


Εικόνα 2.5. Το λογότυπο του Donut

Τον Σεπτέμβριο του 2009 κυκλοφόρησε η έκδοση Android 1.6, ονόματι 'Donut' το λογότυπο της οποίας φαίνεται και στην Εικόνα 6. Στα νέα χαρακτηριστικά που εισήγαγε ήταν:

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

### 2.2.3. Android 2.0 ECLAIR



Εικόνα 2.6. Το λογότυπο του Éclair

Τον Νοέμβριο του 2009 κυκλοφόρησε η έκδοση Android 2.0, ονόματι 'Éclair' το λογότυπο της οποίας φαίνεται και στην Εικόνα 7 . Στη συνέχεια έχουμε τις επανεκδόσεις του σε Android 2.0.1 ή Éclair 0.1 το Δεκέμβριο του 2009 και τον Ιανουάριο του 2010 το Android 2.1 ή Éclair MR1.Ενσωματώθηκαν οι παρακάτω αλλαγές :

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

### 2.2.4. Android 2.2.x ( Froyo )



Εικόνα 2.7.Το λογότυπο του Froyo

Τον Μάιο του 2010 κυκλοφόρησε η έκδοση Android 2.2, ονόματι 'Froyo' το λογότυπο της οποίας φαίνεται και στην Εικόνα 8 . Η έκδοση FROYO, ανάμεσα σε άλλες αλλαγές, περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα του λειτουργικού συστήματος, στην μνήμη και στην απόδοση



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

### 2.2.5. Android 2.3 ( Gingerbread )



Εικόνα 2.8. Το λογότυπο του Gingerbread

Το Δεκέμβριο του 2010 κυκλοφόρησε η έκδοση Android 2.3, ονόματι 'Gingerbread' το λογότυπο της οποίας φαίνεται και στην Εικόνα 9. Έχει τις εξής επανεκδόσεις, Android 2.3.3 (Φλεβάρη 2011), Android 2.3.4 (Μάιος 2011), Android 2.3.5 (Ιούλιος 2011), Android 2.3.6, Android 2.3.7 .

Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:

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

### 2.2.6. Android 3.x (Honeycomb )



Εικόνα 2.9. Το λογότυπο του Honeycomb

Το Φεβρουάριο του 2011 κυκλοφόρησε η έκδοση Android 3.0, ονόματι 'Honeycomb' το λογότυπο της οποίας φαίνεται και στην Εικόνα 10 . Μερικά από τα χαρακτηριστικά αυτής της έκδοσης είναι :

- Υποστήριξη διπύρηνων και τετραπύρηνων επεξεργαστών (multi-core)
- Καλύτερη υποστήριξη για tablets με ένα νέο εικονικό και «ολογραμμικό» user interface.
- Αναβαθμισμένο 3D UI
- Βίντεο μέσω του Google Talk
- Ιδιωτική περιήγηση
- Υποστήριξη για χειριστήρια παιχνιδιών
- Συνδεσιμότητα για αξεσουάρ USB.
- Υποστήριξη HTTP proxy για κάθε συνδεδεμένο σημείο Wi-Fi.
- Βελτιωμένη υποστήριξη Adobe Flash στον browser.  
Προστέθηκε μια μπάρα συστήματος, που διαθέτει γρήγορη πρόσβαση στις ειδοποιήσεις, η εμφάνιση της κατάστασης της συσκευής, και τα μαλακά πλήκτρα πλοήγησης, διαθέσιμα στο κάτω μέρος της οθόνης.
- Απλοποιήθηκε το multitasking, με την μπάρα συστήματος ο χρήστης μπορεί να δει όλες τις εφαρμογές που τρέχουν και να μεταβεί από την μια στην άλλη.
- Επιτάχυνση του hardware

### 2.2.7. Android 4.0 (Ice Cream Sandwich)



Εικόνα 2.10. Το λογότυπο του Ice Cream Sandwich

Τον Οκτώβριο του 2011 κυκλοφόρησε η έκδοση Android 4.0, ονόματι 'Ice Cream Sandwich' το λογότυπο της οποίας φαίνεται και στην Εικόνα 11. Αυτή η ενημέρωση παρουσιάζει πολλά νέα χαρακτηριστικά, όπως:

- Αναγνώριση προσώπου που επιτρέπει το ξεκλείδωμα της συσκευής (Face Unlock),
- Εφαρμογή Android Beam για μεταφορά δεδομένων μέσω NFC
- Καλύτερη αναγνώριση φωνής (λειτουργία Voice Typing)
- Νέος web browser, που επιτρέπει έως και 16 καρτέλες περιήγησης
- Βελτιστοποίηση ταχύτητας και απόδοσης
- Βελτιωμένη διόρθωση λάθους στο πληκτρολόγιο.
- Εικονικά κουμπιά στη διεπαφή χρήστη αντί για φυσικά κουμπιά
- Ανάλυση χρήσης δεδομένων

### 2.2.8. Android 4.1 (Jelly Bean)



Εικόνα 2.11. Το λογότυπο του Jelly Bean

Τον Ιούλιο του 2012 κυκλοφόρησε η έκδοση Android 4.1, ονόματι 'Jelly Bean' το λογότυπο της οποίας φαίνεται και στην Εικόνα 12 .

- Αυτόματη προσαρμογή widget σε κάθε οθόνη που περιέχει ήδη άλλα widgets και αφαίρεση του widget απλά "πετώντας" το εκτός οθόνης
- Νέος τρόπος λειτουργίας πληκτρολογίου κατά τον οποίο το σύστημα προβλέπει και προτείνει προς επιλογή την επόμενη λέξη που πιστεύει ότι θα πληκτρολογήσει ο χρήστης
- Η φωνητική πληκτρολόγηση δουλεύει πλέον και χωρίς σύνδεση Internet
- Περισσότερες επιλογές για A.M.E.A
- Πιο εύκολη πλοήγηση σε φωτογραφίες μέσα απο την εφαρμογή της κάμερας και το ίδιο σύστημα "πετάγματος" εκτός οθόνης για τη διαγραφή της
- Μικρές αλλαγές στο σύστημα ειδοποιήσεων με εμφάνιση του εξώφυλλου του άλμπουμ από τη μουσική που αναπαράγεται και ειδοποιήσεις που "ξεδιπλώνονται" ώστε να αποκαλύψουν ακόμα περισσότερη πληροφορία
- Νέο γραφικό περιβάλλον στα αποτελέσματα μιας αναζήτησης
- Πιο γρηγορή φωνητική πληκτρολόγηση για μια αναζήτηση, τα αποτελέσματα της οποίας μπορούν να εκφωνηθούν απο την εφαρμογή
- Τεχνολογία Google now η οποία εμφανίζει αποτελέσματα αναζήτησης ανάλογα με τη τοποθεσία του χρήστη και τα προσωπικά στοιχεία που έχει δηλώσει στο ημερολόγιο

### 2.2.9. Android 4.4 (Kit kat)



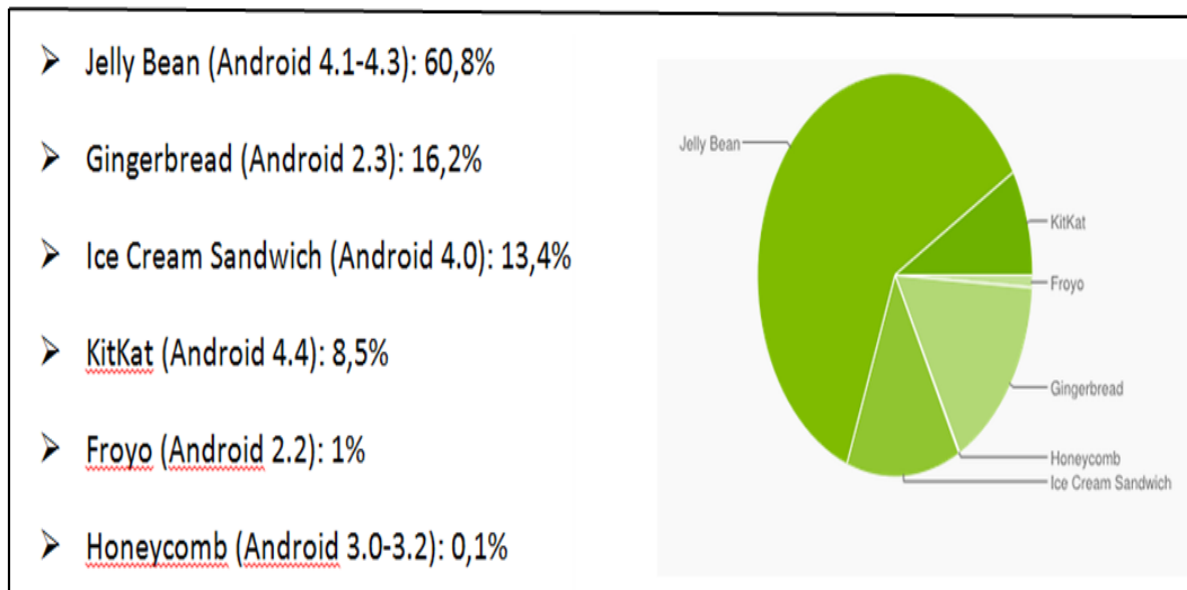
Εικόνα 2.12. Το λογότυπο του Kit kat

Τον Οκτώβριο του 2013 κυκλοφόρησε η έκδοση Android 4.4, ονόματι 'Kit kat' το λογότυπο της οποίας φαίνεται και στην Εικόνα 13.

- Διόρθωση για τις τακτικές αποσυνδέσεις από το διαδίκτυο όταν χρησιμοποιούμε τα δεδομένα
- Διόρθωση στην εστίαση στη κάμερα
- Διόρθωση στην οθόνη Power Manager
- Διορθώσεις στο Bluetooth
- Διόρθωση για τις τυχαίες επανεκκινήσεις
- Οι συντομεύσεις εφαρμογών δεν αφαιρούνται όταν γίνονται ενημερώσεις στις εφαρμογές
- Βελτιωμένη ασφάλεια στο USB debugging
- Διορθώσεις στις συντομεύσεις των εφαρμογών
- Διόρθωση στην αυτόματη σύνδεση του Wi-Fi
- Διορθώσεις στη κάμερα και στα MMS, Email/Exchange, Calendar, People/Dialer/Contacts, DSP, IPv6, VPN
- Διόρθωση ενός προβλήματος που κολλούσε το κινητό στην οθόνη καλωσορίσματος
- Διόρθωση στο LED όταν υπάρχει αναπάντητη κλήση
- Διορθώσεις στους υπότιτλους Διόρθωση στο γράφημα των δεδομένων

### 2.2.10.Οι εκδόσεις που χρησιμοποιούνται περισσότερο

Τον Μάιο του 2014, η Google έδωσε στη δημοσιότητα τα χαρακτηριστικά χρήσης των εκδόσεων του Android. Στο σχήμα 3, παρατηρούμε την κατανομή των εκδόσεων του Android στις ενεργές συσκευές. Η έκδοση Jelly Bean (Android 4.1, 4.2, και 4.3) είναι πλέον εγκατεστημένη σε πάνω από τις μισές (60,8%) ενεργές συσκευές Android. Ακολουθεί το Gingerbread με ποσοστό χρήσης 16,2% και το Ice Cream Sandwich με 13,4%. Το KitKat (Android 4.4) σημειώνει μεγάλη άνοδο καθώς όλο και περισσότερες συσκευές αναβαθμίζονται σε αυτήν την έκδοση. Ενδεικτικά αναφέρουμε ότι το ποσοστό του KitKat έχει φτάσει στο 8,5%, όταν το Μάρτιο του 2014 βρισκόταν μόλις στο 2,5%.

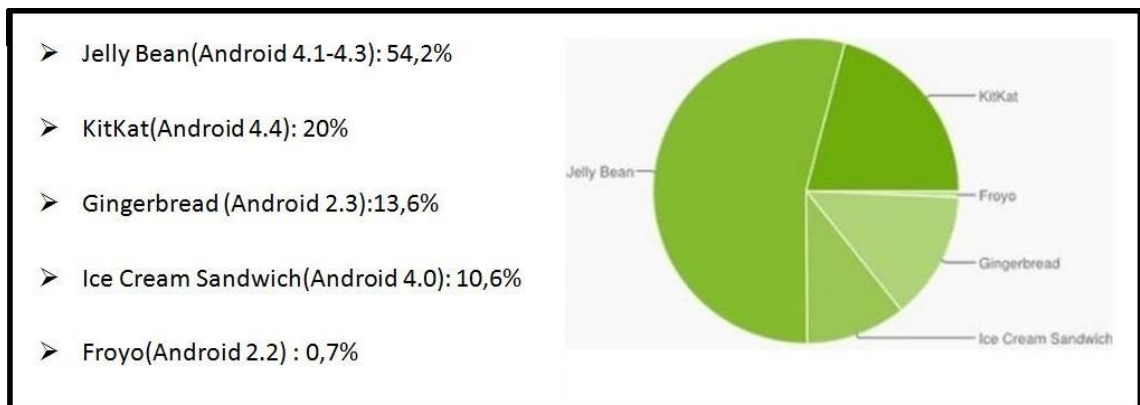


Σχήμα 2.1. Κατανομή των εκδόσεων του Android στις ενεργές συσκευές (Μάιος 2014)

Τον Αύγουστο του ίδιου έτους , η Google έδωσε στη δημοσιότητα τα χαρακτηριστικά χρήσης των εκδόσεων του Android. Η τελευταία έκδοση του Android, KitKat "ανέβηκε" κι άλλο, και βρίσκεται εγκατεστημένη στο 20,9% των συσκευών, σε σχέση με το 17,9% του προηγούμενου μήνα και με το 8,5% του Μαΐου.

Οι τρεις εκδόσεις του Android Jelly Bean πλέον κατέχουν το 54,2% της "πίτας", ποσοστό μειωμένο σε σχέση με το 60,8% του Μαΐου.

Πτωτική ήταν και η πορεία του Android 4.0 ICS (Ice Cream Sandwich), ενώ μειώθηκαν και τα ποσοστά του Android 2.2 (Froyo) και 2.3 (Gingerbread) .



Σχήμα 2.2. Κατανομή των εκδόσεων του Android στις ενεργές συσκευές (Αύγουστος 2014)

## 2.3.Αρχιτεκτονική Android

Η πλατφόρμα του Android αποτελείται από μια στοίβα λογισμικού(εικόνα 3),τα επίπεδα της οποίας ,απο το υψηλότερο στο χαμηλότερο είναι το επίπεδο των εφαρμογών(applications),το επίπεδο πλαισίου των εφαρμογών(application framework),οι βιβλιοθήκες(libraries),το επίπεδο Εκτέλεσης(android runtime) και στο τελευταίο επίπεδο είναι ο πυρήνας του Linux.



Εικόνα 2.13 .Αρχιτεκτονική Android

### 2.3.1.Επίπεδο Εφαρμογών (Blue layer-Application)

Στο επίπεδο αυτό υπάρχουν οι βασικές εφαρμογές οι οποίες έχουν γραφτεί σε Java. Μερικές από τις βασικές εφαρμογές είναι το email, ο web browser, το πρόγραμμα που κάνει τις τηλεφωνικές κλήσεις ,πρόγραμμα SMS,ημερολόγιο,χάρτες(Google Maps)και άλλα. Ο μέσος χρήστης κάνει χρήση μόνο αυτού του layer (στρώμα) χωρίς να γνωρίζει τη διεργασία που γίνεται στα χαμηλότερα επίπεδα, ενώ τα υπόλοιπα layers χρησιμοποιούνται από τους προγραμματιστές, τους υπεύθυνους για την ανάπτυξη του λογισμικού και τους κατασκευαστές υλικού.

### 2.3.2. Επίπεδο Πλαισίου Εφαρμογών (Blue layer- Application Framework)

Αυτός είναι ο σκελετός του android ή αλλιώς το πλαίσιο που θα ακολουθήσουν όλοι οι developers. Οι developers μπορούν να έχουν πρόσβαση σε όλα τα APIs αυτού του πλαισίου και μπορούν να διαχειρίζονται τις βασικές λειτουργίες του τηλεφώνου όπως την κατανομή των πόρων, τις τηλεφωνικές εφαρμογές και την παρακολούθηση της φυσικής θέσης του τηλεφώνου. Το application framework είναι ουσιαστικά ένα σύνολο βασικών εργαλείων με τα οποία ένας developer μπορεί να αναπτύξει πιο σύνθετα εργαλεία.

Τα σημαντικότερα δομικά στοιχεία του πλαισίου είναι:

- **Διαχειριστής Δραστηριοτήτων (Activity Manager):** Η κλάση αυτή αλληλεπιδρά με το σύνολο των δραστηριοτήτων που εκτελούνται στο σύστημα. Διαχειρίζεται τον κύκλο ζωής μιας εφαρμογής και παρέχει τη δυνατότητα μετάβασης σε προηγούμενες καταστάσεις τους. Δίνει πληροφορίες σχετικά με τη διαθέσιμη μνήμη του συστήματος (ActivityManager.MemoryInfo), με διεργασίες οι οποίες είναι σε λάθος κατάσταση (ActivityManager.ProcessErrorStateInfo), με εργασίες, τις οποίες, ο χρήστης έχει πρόσφατα ξεκινήσει ή επισκευθεί (ActivityManager.RecentTaskInfo), με τρέχουσες διεργασίες (ActivityManager.RunningAppProcessInfo), μια υπηρεσία που εκτελείται αυτή τη στιγμή στο σύστημα (ActivityManager.RunningServiceInfo) και με μια συγκεκριμένη εργασία που αυτή τη στιγμή τρέχει στο σύστημα (ActivityManager.RunningTaskInfo). Μια Activity ως ιδέα αντιπροσωπεύει την απλούστερη δραστηριότητα την οποία μπορεί να κάνει ένας χρήστης. Αποτελεί τη βάση για κάθε εφαρμογή. Από την πλευρά του ο προγραμματιστής έχει στα χέρια του μια οντότητα με την οποία μπορεί εύκολα να καθορίζει το γραφικό περιβάλλον και να διαχειρίζεται τον κύκλο ζωής της εφαρμογής του.

Το λειτουργικό σύστημα διαχειρίζεται τις activities με μια στοίβα. Στην κορυφή της στοίβας τοποθετείται η activity η οποία βρίσκεται στο προσκήνιο και με την οποία αλληλεπιδρά ο χρήστης. Κατά τη διάρκεια ζωής της μια activity περνάει από τρεις καταστάσεις με κριτήριο αν είναι πλήρως, μερικώς ή καθόλου ορατή από τον χρήστη:

- Έαν μια activity βρίσκεται στο προσκήνιο, δηλαδή είναι πλήρως ορατή από τον χρήστη, τότε χαρακτηρίζεται ως ενεργή ( active ).
- Αν βρίσκεται στο παρασκήνιο αλλά παραμένει μερικώς ορατή , τότε βρίσκεται σε κατάσταση αναμονής ( paused). Αυτό μπορεί να συμβεί όταν η ενεργή activity είναι ημιδιαφανής ή δεν καλύπτει πλήρως την οθόνη.
- Όταν μια activity έχει πλήρως επικαλυφθεί από κάποια άλλη τότε λέμε ότι είναι σταματημένη ( stopped) . Στην κατάσταση αυτή ο χρήστης δεν μπορεί να αλληλεπιδράσει μαζί της.

Και στις τρεις καταστάσεις μια activity είναι ζωντανή, δηλαδή διατηρεί τις μεταβλητές και μπορεί να κάνει ενέργειες (π.χ. να κάνει υπολογισμούς), να χρησιμοποιεί το διαδίκτυο και να επικοινωνεί με τους αισθητήρες. Σε περίπτωση που το λειτουργικό σύστημα χρειάζεται μνήμη αρχίζει να τερματίζει activities ξεκινώντας από τις σταματημένες και συνεχίζοντας με αυτές που βρίσκονται σε κατάσταση αναμονής.

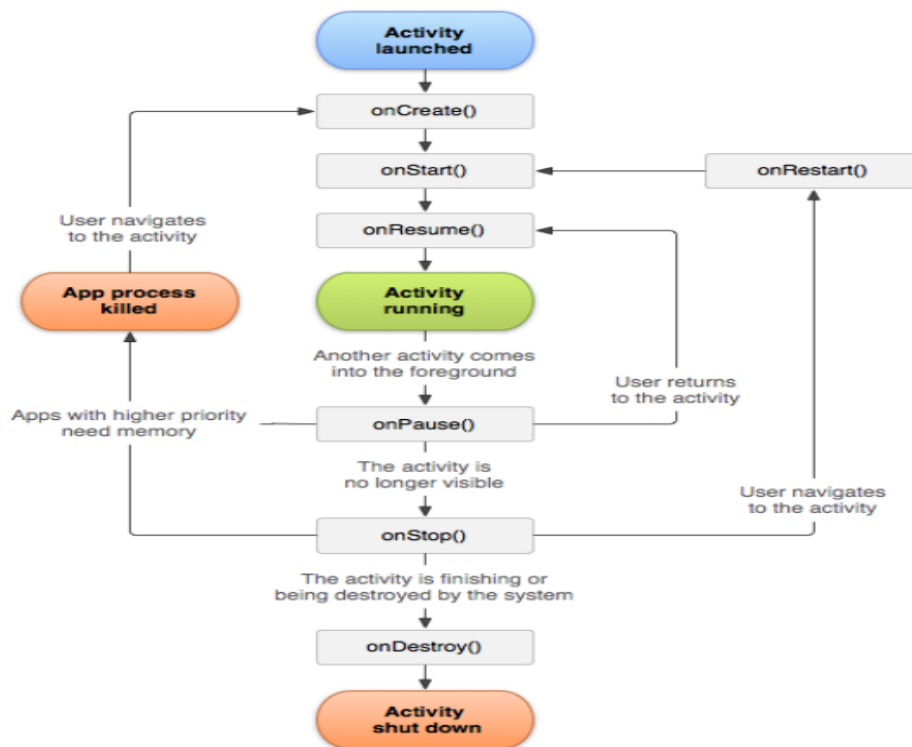
Το σχήμα 1 δείχνει τον κύκλο ζωής μιας Activity, στον οποίο μπορούμε να διακρίνουμε τρεις βασικούς υποκύκλους:

- Μια Activity ξεκινάει με την κλήση της onCreate() και τελειώνει την κλήση της onDestroy(). Στα πλαίσια της onCreate() πρέπει να γίνονται όλες οι βασικές αρχικοποιήσεις ενώ η onDestroy() πρέπει να ελευθερώνει όλους τους πόρους που έχει δεσμεύσει η Activity.

- Μια Activity είναι ορατή και μπορεί να αλληλεπιδράσει μ' αυτή ο χρήστης ανάμεσα στις κλήσεις των μεθόδων onStart() και onStop(). Μόνο κατά τη διάρκεια αυτού του υποκύκλου είναι απαραίτητη η δέσμευση πόρων που πρέπει να είναι ορατοί

στον χρήστη. Χαρακτηριστικό παράδειγμα τέτοιου πόρου είναι η εικόνα από την κάμερα της συσκευής.

- Μια Activity βρίσκεται στο προσκήνιο, δηλαδή μπροστά από τις υπόλοιπες activities, ανάμεσα στις κλήσεις των μεθόδων `onResume()` και `onPause()`.

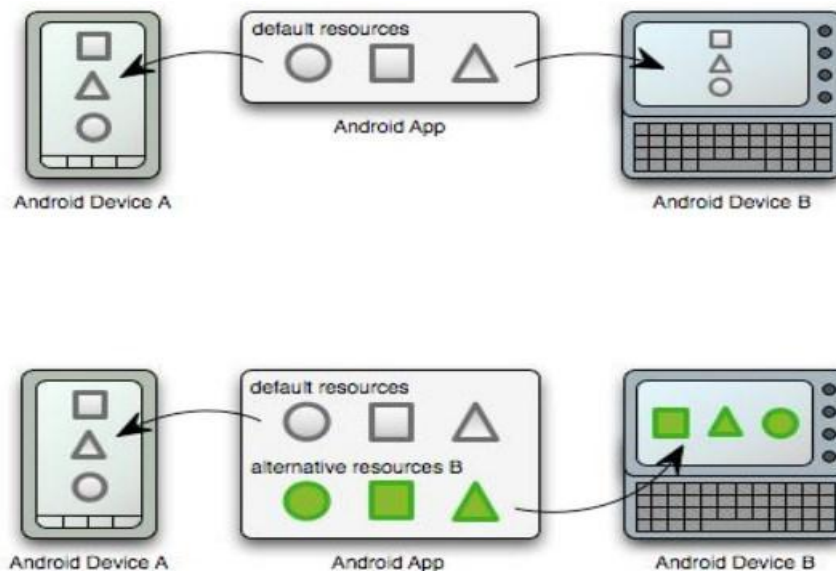


Σχήμα 2.3. Ο κύκλος ζωής μιας δραστηριότητας Android

- **Διαχειριστής παραθύρων (Window Manager):** Η διεπαφή που χρησιμοποιούν οι εφαρμογές για να <<μιλήσουν>> με το διαχειριστή παραθύρων
- **Πάροχος Περιεχομένου (Content Providers):** Οι πάροχοι περιεχομένου αποθηκεύουν και ανακτούν δεδομένα έτσι ώστε να τα κάνουν προσβάσιμα σε όλες τις εφαρμογές. Είναι ο μόνος τρόπος για το διαμοιρασμό δεδομένων μεταξύ των εφαρμογών. Δεν υπάρχει κοινός χώρος αποθήκευσης όπου όλα τα πακέτα Android μπορούν να έχουν πρόσβαση. Το Android περιέχει ένα μεγάλο αριθμό από παρόχους περιεχομένου για διάφορους τύπους δεδομένων, όπως για ήχο, μουσική, εικόνες, προσωπικές πληροφορίες επαφής κ.α. Όλοι οι διαθέσιμοι πάροχοι περιεχομένου βρίσκονται στο `android.provider` πακέτο της Java
- **Προβολή Συστήματος (View System):** Η κλάση αυτή αποτελεί το βασικό δομικό στοιχείο για τη δημιουργία μιας διεπαφής χρήστη. Η δημιουργία μιας κλάσης View καταλαμβάνει μια ορθογώνια περιοχή στην οθόνη και είναι υπεύθυνη για τη κατάρτιση και τη διαχείριση συμβάντων. Η κλάση View είναι βασική κλάση η οποία περιέχει τα εργαλεία διάδρασης (widgets), τα οποία χρησιμοποιούνται για τη δημιουργία διαδραστικών στοιχείων διεπιφάνειας χρήστη (κουμπιά, πεδία επιλογής, πεδία κειμένου κ.α.)
- **Διαχειριστής πακέτων (Package Manager):** Η κλάση αυτή ανακτά πληροφορίες οι οποίες συνδέονται με τα πακέτα εφαρμογών που υπάρχουν εγκατεστημένα στη κινητή συσκευή.



- **Διαχειριστής Τηλεφώνου (Telephony Manager):** Η κλάση αυτή παρέχει πληροφορίες για τις υπηρεσίες τηλεφωνίας στη συσκευή. Οι εφαρμογές μπορούν να χρησιμοποιήσουν τις πληροφορίες αυτές για τον προσδιορισμό των υπηρεσιών και των καταστάσεων του τηλεφώνου καθώς και για τη πρόσβαση σε ορισμένες τύπου πληροφορίες συνδρομητή. Οι εφαρμογές έχουν επίσης τη δυνατότητα να λαμβάνουν ειδοποιήσεις για τις αλλαγές τηλεφωνικής κατάστασης μιας συσκευής. Υπάρχουν περιορισμοί σχετικά με τις πληροφορίες του τηλεφώνου, οι οποίες είναι διαθέσιμες. Κάθε φορά που αναπτύσσουμε μια εφαρμογή, θα πρέπει να καθορίζονται οι επιτρεπόμενες ενέργειες κατά τη δημιουργία της.
- **Διαχειριστής Πόρων (Resource Manager):** Η κλάση αυτή παρέχει πρόσβαση σε πόρους, οι οποίοι είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας π.χ. κωδικοί χρωμάτων, σχεδιαγράμματα, γραφικά, συμβολοσειρές κλπ. Στο πλαίσιο της ανάπτυξης εφαρμογών, οι πόροι (εικόνες, συμβολοσειρές κ.α.) οι οποίοι θα χρησιμοποιούνται θα πρέπει να αποθηκεύονται έξω από την εφαρμογή έτσι ώστε να μπορούν να συντηρηθούν ανεξάρτητα από αυτήν. Εξωτερικεύοντας τους πόρους, δίνεται η δυνατότητα της παροχής διαφορετικών πόρων σε συσκευές υποστηρίζοντας έτσι τις εφαρμογές που λειτουργούν σε συσκευές με διαφορετικές γλώσσες ή διαφορετικά μεγέθη εικόνας. Για κάθε είδος πόρου, μπορούμε να καθορίσουμε τον εξορισμού (default) τύπο ή τον εναλλακτικό (alternative). Ο εξορισμού τύπος χρησιμοποιείται ανεξάρτητα από τη διαμόρφωση της συσκευής ή όταν δεν υπάρχουν εναλλακτικοί πόροι που να ταιριάζουν στη τρέχουσα διαμόρφωση. Ο εναλλακτικός τύπος χρησιμοποιείται όταν υπάρχει συγκεκριμένη διαμόρφωση της συσκευής. Ένα παράδειγμα διαχείρισης πόρων, βλέπουμε στο σχήμα 2.



Σχήμα 2.4. Παράδειγμα διαχείρισης πόρων

- **Διαχειριστής τοποθεσίας (Location Manager):** Η κλάση αυτή παρέχει πρόσβαση στις υπηρεσίες εντοπισμού τοποθεσίας του συστήματος. Οι υπηρεσίες αυτές επιτρέπουν τις εφαρμογές να αποκτούν ανά τακτά χρονικά

διαστήματα ενημερώσεις ,όπως είναι η γεωγραφική θέση της συσκευής ή να προκαλούν την εκκίνηση μιας εφαρμογής όταν η συσκευή μπαίνει σε συγκεκριμένη γεωγραφική τοποθεσία.

- **Διαχειριστής Κοινοποιήσεων (Notification Manager):** Η κλάση αυτή μέσω κοινοποιήσεων ενημερώνει το χρήστη για τα γεγονότα τα οποία συμβαίνουν στο σύστημα.Δηλαδή ο χρήστης αντιλαμβάνεται γεγονότα τα οποία γίνονται στο παρασκήνιο του λειτουργικού συστήματος.Οι κοινοποιήσεις μπορούν να λάβουν διάφορες μορφές ,όπως για παράδειγμα να αναπαριστούνται με ένα εικονίδιο που εμφανίζεται στη γραμμή κατάστασης της κινητής συσκευής και να είναι προσβάσιμες μέσω του μενού.Άλλοι τρόποι εμφάνισης των κοινοποιήσεων είναι είτε ενεργοποιώντας τα LED στοιχεία της συσκευής είτε με την αναπαραγωγή ενός ήχου ή μέσω της δόνησης.

### 2.3.3.Επίπεδο Βιβλιοθηκών (Green layer-Libraries) .

Αυτό στο στρώμα αποτελείται από βιβλιοθήκες android, οι οποίες έχουν γραφτεί σε C,C++ και χρησιμοποιούνται από διάφορα συστήματα. Αυτές οι βιβλιοθήκες λένε στη συσκευή πώς να χειριστεί διαφορετικά είδη δεδομένων, τα οποία εκτίθενται στους developers μέσω του περιβάλλοντος εφαρμογών( application framework). Μερικές από τις βασικές βιβλιοθήκες που παρέχει το εν λόγω λειτουργικό σύστημα είναι οι εξής:

- Βιβλιοθήκη Διαχείρισης Επιφανειών (**Surface Manager** ): είναι βιβλιοθήκη που διαχειρίζεται την εμφάνιση στην οθόνη απεικονίσεων 2D και 3D για τις εφαρμογές.
- Βιβλιοθήκες Μέσων (**Media libraries**): είναι βιβλιοθήκες οι οποίες υποστηρίζουν την αναπαραγωγή και τη καταγραφή πολλών γνωστών πρότυπων εικόνας,βίντεο και ήχου (π.χ. MPEG4,MP3,JPG,PNG κ.α.)
- Βιβλιοθήκη **SQLite** : είναι βιβλιοθήκη ,μέσω της οποίας γίνεται διαθέσιμο ένα σχεσιακό σχήμα βάσης δεδομένων σε όλες τις εφαρμογές που την χρησιμοποιούν.Το σχεσιακό σχήμα βάσης δεδομένων έχει όλες τις λειτουργίες μιας βάσης δεδομένων αλλά είναι πιο ελαφριά στη λειτουργία της.
- Βιβλιοθήκες **3D** : είναι βιβλιοθήκες ,των οποίων η χρήση βασίζεται στο **OpenGL ES 1.0** API και χρησιμοποιούν επιταχυντή υλικού 3D(όπου είναι αυτός διαθέσιμος) ή επιταχυντή λογισμικού 3D.
- Βιβλιοθήκη **Free Type** : είναι βιβλιοθήκη που κάνει διαθέσιμα αρχεία εικονοστοιχείων (bitmap) και διανύσματα απόδοσης γραμματοσειρών (fonts)
- Βιβλιοθήκη **LibWebCore** : είναι βιβλιοθήκη που παρέχει τη δυνατότητα δημιουργίας ενός στοιχείου προβολής Web ή επεκτείνει την εξ'ορισμού εφαρμογή πλοήγησης του Android.
- Βιβλιοθήκη **SGL**: είναι βιβλιοθήκη που παρέχει μηχανή για την αναπαραστάση γραφικών 2D
- Βιβλιοθήκη της C (**libc**) : είναι μια τροποποιημένη βιβλιοθήκη σε σχέση με την στάνταρ βιβλιοθήκη της C η οποία είναι προσαρμοσμένη για να λειτουργεί σε ενσωματωμένα συστήματα τα οποία βασίζονται σε Linux .

### 2.3.4.Επίπεδο Εκτέλεσης (Yellow layer- Runtime Android)

Αυτό το επίπεδο αποτελείται από δυο βασικές συνιστώσες:

- Βασικές βιβλιοθήκες για την αλληλεπίδραση των εφαρμογών Java με το περιβάλλον της συσκευής στην οποία εκτελούνται
- Dalvik virtual machine(DVM) η οποία είναι υπεύθυνη για τη δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να μπορεί να εκκινήσει το λειτουργικό σύστημα

Η Google επέλεξε τη γλώσσα προγραμματισμού Java για την ανάπτυξη Android εφαρμογών όμως δε χρησιμοποίησε τη κλασική εικονική μηχανή της Java(JVM-Java Virtual Machine) αλλά μια εικονική μηχανή κατάλληλη για φορητές υπολογιστικές πλατφόρμες,την Dalvik. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytcodes που προέρχονται από αρχεία .class.Ο πηγαίος κώδικας μεταγλωττίζεται και προκύπτει αρχείο .class και μέσω του εργαλείου <<dx>>,μετατρέπεται σε αρχείο της μορφής .dex και στη συνέχεια σε εκτελέσιμο αρχείο. Σε αντίθεση όμως με ένα αρχείο με κατάληξη .class ,το οποίο περιέχει μια κλάση, ένα αρχείο με κατάληξη .dex περιέχει πολλές κλάσεις και είναι πολύ πιο συμπαγές και αποδοτικό, γεγονός ιδιαίτερα σημαντικό για συσκευές με περιορισμένη μνήμη, υπολογιστικούς πόρους και μπαταρία. Η εικονική μηχανή Dalvik βασίζεται στον πυρήνα Linux παρέχοντας ισχυρή λειτουργικότητα των διεργασιών της, όπως το threading, και της διαχείρισης μνήμης της στο χαμηλό επίπεδο μνήμης.

### 2.3.5.Ο πυρήνας του Linux (Red layer-Kernel)

Το Android βασίζεται στον πυρήνα Linux έκδοση 2,6 για βασικές υπηρεσίες συστήματος ,όπως ασφάλεια,διαχείριση διεργασιών,στοίβα δικτύου,διαχείριση μνήμης και οδηγούς για hardware. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.Το Android αποκτά τεράστια δύναμη και δυνατότητες χρησιμοποιώντας το πυρήνα του Linux.Χρησιμοποιώντας ένα λογισμικό ανοιχτού κώδικα παρέχει την δυνατότητα τόσο σε μεμονωμένα άτομα όσο και σε μεγάλους οργανισμούς να προωθήσουν αυτή τη πλατφόρμα κάτι το οποίο είναι πάρα πολύ σημαντικό σε μια αγορά όπως αυτή των κινητών συσκευών η οποία εξελίσσεται πάρα πολύ γρήγορα.Επίσης πρόκειται για μια δοκιμασμένη πλατφόρμα.Παρέχει αξιοπιστία η οποία είναι πιο σημαντική από τη ποιότητα ειδικά όταν πρόκειται για κινητές συσκευές όπως τα κινητά τηλέφωνα όπου η μετάδοση της φωνής αποτελεί το πρωταρχικό στόχο.Τέλος,δίδεται η δυνατότητα να δημιουργούνται οδηγοί (drivers)στο επίπεδο Linux και να χρησιμοποιούνται όπως και σε οποιαδήποτε άλλη πλατφόρμα Linux.

## 2.4. Συνιστώσες εφαρμογής

Κάθε εφαρμογή Android αποτελείται από 4 βασικά στοιχεία : τις δραστηριότητες(activities),τις υπηρεσίες(services),του παρόχους περιεχομένου (content providers) και τους καθολικούς παραλήπτες μηνυμάτων (broadcast receivers).

### 2.4.1 Δραστηριότητες(activities)

Αποτελούν το βασικότερο στοιχείο κάθε εφαρμογής.Η κάθε δραστηριότητα αποτελεί μια διαφορετική οθόνη(παράθυρο) της εφαρμογής ,με την οποία ο χρήστης αλληλεπιδρά.Σε αυτή φορτώνεται το γραφικό περιβάλλον το οποίο βλέπει ο χρήστης.Το κύκλο ζωής μιας δραστηριότητας το περιγράψαμε στο κεφάλαιο 2.2.

### 2.4.2.Υπηρεσίες(services)

Μια υπηρεσία τρέχει στο παρασκήνιο της εφαρμογής και μπορεί να εκτελέσει αρκετές και χρονοβόρες λειτουργίες.Η διαφορά υπηρεσιών και δραστηριοτήτων είναι ότι η υπηρεσία δε παρέχει user interface.Επίσης η υπηρεσία μπορεί να εκτελείται ακόμη και όταν τρέξουμε κάποια διαφορετική εφαρμογή από αυτή που τη ξεκίνησε.Τέλος,σε αντίθεση με τις δραστηριότητες ,το λειτουργικό σύστημα δε διαχειρίζεται το κύκλο ζωής των υπηρεσιών,επομένως η κάθε υπηρεσία θα πρέπει να δηλώνει το πότε θα σταματήσει με την εντολή stopSelf().

Παράδειγμα υπηρεσίας αποτελεί η εφαρμογή για αναπαραγωγή μουσικής,όταν ακούμε μουσική,και θέλουμε ταυτόχρονα να ανοίξουμε και μια άλλη εφαρμογή.

Δυο βασικές λειτουργίες συνδέονται με τις υπηρεσίες :

- startService() ,που καλείται από κάποια activity για να ξεκινήσει η service και θα συνεχίσει να εκτελείται ακόμη και όταν η activity που τη κάλεσε,τερματιστεί.Έτσι η service δεν επιστρέφει κάποιο αποτέλεσμα στην activity που τη κάλεσε αλλά εκτελεί κάποια λειτουργία,όπως το κατέβασμα κάποιου αρχείου από το διαδίκτυο.
- buildService() ,που δίνει τη service με κάποιο άλλο στοιχείο της εφαρμογής π.χ. με μια activity,έτσι ώστε να υπάρχει αλληλεπίδραση μεταξύ των στοιχείων αυτών ,όπως π.χ. αποστολή και λήψη αποτελεσμάτων.Αν τερματιστεί το στοιχείο που έχει δεθεί με τη service ,τότε τερματίζει και η service.Η service δε παρέχει διεπαφή με το χρήστη,οπότε για να ενημερώνεται ο χρήστης για τα αποτελέσματα της υπηρεσίας,χρησιμοποιούμε την ειδοποίηση Toast ή την ειδοποίηση Status Bar.Οι ειδοποιήσεις αυτές αναλύονται στη παράγραφο ,αφού αφορούν στο user interface.

### 2.4.3.Πάροχοι περιεχομένου (content providers)

Οι πάροχοι περιεχομένου διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα,οι οποίοι είναι προσπελάσιμοι από οποιαδήποτε εφαρμογή.Αποτελούν το μοναδικό τρόπο για αποθήκευση δεδομένων,τα οποία θέλουμε να είναι προσβάσιμα από αρκετές εφαρμογές.Παράδειγμα παρόχου περιεχομένων αποτελεί μια εφαρμογή τηλεφωνικού καταλόγου ,η οποία αποθηκεύει σε συγκεκριμένο χώρο τις επαφές μας,ώστε αργότερα αν θελήσουμε να χρησιμοποιήσουμε διαφορετική εφαρμογή ,να μπορεί αυτή να εντοπίσει και να επεξεργαστεί τα ήδη υπάρχοντα δεδομένα.Άλλα παραδείγματα είναι τα αρχεία ήχου,εικόνας,βίντεο κ.α. τα οποία αποθηκεύονται σε προεπιλεγμένους χώρους,ώστε να είναι προσβάσιμα από πολλές εφαρμογές.

#### **2.4.4. Προθέσεις (Intents)**

Είναι ο συνδετικός κρίκος μεταξύ των δραστηριοτήτων. Είναι υπεύθυνες για την επικοινωνία και την μετάβαση από μια δραστηριότητα στην άλλη. Επίσης, εξασφαλίζουν την ανταλλαγή δεδομένων είτε μεταξύ των δραστηριοτήτων είτε μεταξύ των εφαρμογών. Δεν μπορούν να προβάλουν μια διεπαφή χρήστη, όμως μπορούν να ενημερώσουν τον χρήστη με την προβολή Ειδοποίησης για κάποιο γεγονός.

#### **2.4.5. Καθολικοί παραλήπτες μηνυμάτων (broadcast receivers).**

Οι broadcast receivers ενημερώνονται για κάποιο συγκεκριμένο γεγονός από το λειτουργικό σύστημα και τότε ενεργοποιούνται. Παραδείγματα τέτοιων γεγονότων είναι το σβήσιμο της οθόνης, η χαμηλή μπαταρία κ.α. Αρκετές εφαρμογές λήψης φωτογραφιών απενεργοποιούν το flash σε περίπτωση χαμηλής μπαταρίας για εξοικονόμηση ενέργειας. Ενημερώνονται δηλαδή από το broadcast receiver για το γεγονός αυτό και πράττουν ανάλογα. Οι broadcast receivers δεν έχουν γραφικό περιβάλλον (user interface) αλλά μπορούν να ενημερώσουν το χρήστη για κάποιο γεγονός μέσω μπάρας ειδοποιήσεων (status bar notifications). Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής

## 2.5.Διεπαφή χρήστη (user interface)

Τον χρήστη δε τον ενδιαφέρει τι συμβαίνει πίσω από την εφαρμογή που τρέχει αλλά το γραφικό περιβάλλον και η ευχρηστία της εφαρμογής. Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δε μπορεί να υπάρξει το άλλο. Πολλές φορές μάλιστα είναι δυσκολότερος ο σχεδιασμός ενός όμορφου και εύχρηστου περιβάλλοντος εργασίας, παρά η ίδια η λειτουργικότητα της εφαρμογής. Καθίσταται σαφές, λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μία συγκεκριμένη εφαρμογή έναντι μίας άλλης, με τις ίδιες λειτουργίες.

Η διεπιφάνεια χρήστη κατασκευάζεται χρησιμοποιώντας αντικείμενα View και ViewGroup.Υπάρχουν πολλοί τύποι αντικειμένων View και ViewGroup,όπου ο κάθε τύπος κληρονομεί από τη κλάση View.Τα αντικείμενα View είναι η βασική μονάδα έκφρασης μιας διεπιφάνειας χρήστη στη πλατφόρμα Android.Η κλάση View αποτελεί βασική κλάση για υποκλάσεις που ονομάζονται <<widgets>>,οι οποίες προσφέρουν αντικείμενα διεπαφής χρήστη,όπως πεδία κειμένου(text fields) και κουμπιά(buttons).Η κλάση ViewGroup αποτελεί βασική κλάση για υποκλάσεις που ονομάζονται <<layouts>>,οι οποίες προσφέρουν διαφορετικούς τρόπους διάταξης του περιεχομένου μιας διεπαφής χρήστη.Ένα αντικείμενο View είναι μια δομή δεδομένων,όπου οι ιδιότητες της αποθηκεύουν τις παραμέτρους και το περιεχόμενο της διάταξης για συγκεκριμένη ορθογώνια περιοχή της οθόνης.Δηλαδή ,το αντικείμενο View είναι το σημείο αλληλεπίδρασης μεταξύ του χρήστη και του δέκτη των γεγονότων της αλληλεπίδρασης.

Σε κάθε οθόνη της εφαρμογής πρωταρχικό στοιχείο του γραφικού περιβάλλοντος αποτελεί η διάταξη των γραφικών στοιχείων της οθόνης (layout).

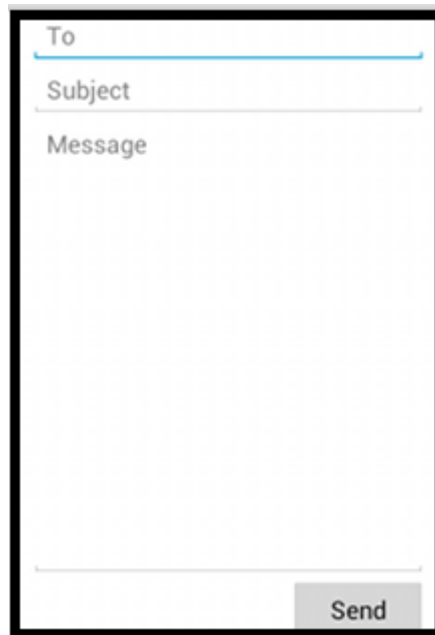
### 2.5.1.Διάταξη γραφικών στοιχείων οθόνης (layout)

Το layout περιλαμβάνει όλα τα γραφικά στοιχεία της οθόνης τα οποία μπορεί να είναι διατεταγμένα σε επιμέρους layouts. Υπάρχουν 4 είδη layout, τα LinearLayout (γραμμική διάταξη), RelativeLayout (σχετική διάταξη), FrameLayout (διάταξη πλαισίου) και TableLayout (διάταξη πίνακα). (Υπάρχει και το AbsoluteLayout, το οποίο όμως έχει προταθεί να μην χρησιμοποιείται πλέον, διότι ορίζει τις απόλυτες θέσεις κάθε στοιχείου, οι οποίες όμως διαφέρουν ανάλογα με το κινητό τηλέφωνο και την οθόνη που χρησιμοποιείται η εφαρμογή).

- **η γραμμική διάταξη (Linear Layout)** : η διάταξη των στοιχείων γίνεται σε οριζόντια ή κατακόρυφη διάταξη.Όταν δηλώσουμε τρία στοιχεία σε ένα οριζόντιο LinearLayout τότε αυτά θα εμφανίζονται στην οθόνη σε οριζόντια θέση το ένα δίπλα στο άλλο με τη σειρά που δηλώθηκαν.Αντίθετα θα διαταχθούν σε μια κατακόρυφη θέση.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
```

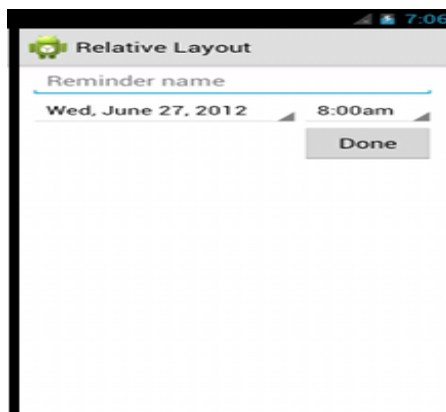
```
android:orientation="vertical" >
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/to" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/subject" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="top"
    android:hint="@string/message" />
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="@string/send" />
</LinearLayout>
```



Εικόνα 2.14. Παράδειγμα LinearLayout

- η **σχετική διάταξη (Relative Layout)** : η διάταξη που δίνει στον προγραμματιστή μια μεγαλύτερη ευχέρεια και ελευθερία στη δήλωση των γραφικών στοιχείων, με την έννοια ότι μπορούμε να εμφανίσουμε οποιοδήποτε αντικείμενο σε όποια θέση θέλουμε, αρχή, μέση και τέλος της οθόνης ακόμα και να ορίσουμε την θέση σε σχέση με κάποιο άλλο αντικείμενο.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```



Εικόνα 2.15. Παράδειγμα Relative Layout



- **η διάταξη πλαισίου (Frame Layout)** : η πιο απλή διάταξη στοιχείων, καθώς δημιουργεί ένα κενό χώρο όπου μπορεί ο προγραμματιστής να βάλει ένα αντικείμενο, για παράδειγμα μια εικόνα, και αυτό το στοιχείο θα έχει τον ρόλο της ρίζας (root) στο δέντρο των γραφικών στοιχείων της οθόνης ,όπου όλα τα υπόλοιπα θα διατάσσονται σύμφωνα με αυτό.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <ImageView
    android:id="@+id/ImageView01"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:src="@drawable/Lake"
    android:scaleType="matrix"></ImageView>
  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="#000"
    android:textSize="40dp"
    android:text="@string/top_text" />
  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/bottom_text"
    android:layout_gravity="bottom"
    android:gravity="right"
    android:textColor="#fff"
    android:textSize="50dp" />
</FrameLayout>
```



Εικόνα 2.16. Παράδειγμα Frame Layout

- **η διάταξη πίνακα (Table Layout)** : η διάταξη πίνακα, μπορεί δηλαδή να διατάσσει τα παιδιά του (children) σε γραμμές και στήλες. Στις γραμμές, στις στήλες καθώς και στα κελιά δεν εμφανίζονται γραμμές (borders). Ο πίνακας έχει τόσες στήλες όσες έχει η γραμμή με τα περισσότερα κελιά. Ένας πίνακας μπορεί να έχει άδεια κάποια κελιά αλλά τα κελιά δε μπορούν να εκτείνονται σε στήλες όπως στην HTML. Τα αντικείμενα TableRow είναι παιδιά του Table Layout (κάθε αντικείμενο του TableRow απεικονίζει μια γραμμή στο πίνακα). Κάθε γραμμή του πίνακα έχει 0 ή περισσότερα κελιά καθένα από τα οποία περιέχουν αντικείμενα View (π.χ. button).

```
<?xml version="1.0" encoding="utf-8"?>

<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent" android:layout_width="fill_parent"
android:background="#000044" >

<TableRow>
<TextView android:text="User Name:" android:width="120px"/>
<EditText android:id="@+id/txtUserName" android:width="200px" />
</TableRow>
<TableRow>
<TextView android:text="Password:" />
<EditText android:id="@+id/txtPassword" android:password="true" />
</TableRow>
<TableRow>
<TextView />
<CheckBox android:id="@+id/chkRememberPassword"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Remember Password" />
</TableRow>

<TableRow>
<Button android:id="@+id/buttonSignIn" android:text="Log In" />
</TableRow>

</TableLayout>
```



Εικόνα 2.17. Παράδειγμα Table Layout

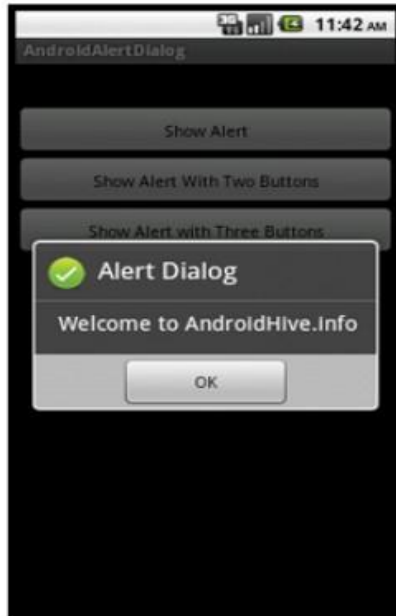
## 2.5.2. Παράθυρα διαλόγου(dialogs)

Ο διάλογος είναι ένα μικρό παράθυρο που εμφανίζεται στην οθόνη μπροστά από την activity μέσα από την οποία ο χρήστης το κάλεσε με κάποια ενέργεια του. Η activity αυτή χάνει την εστίαση που είχε και το παράθυρο διαλόγου είναι το μοναδικό με το οποίο μπορεί να αλληλεπιδράσει ο χρήστης. Χρησιμοποιείται είτε για ενημέρωση του χρήστη για κάποιο γεγονός είτε για να ορίσει ο χρήστης την επόμενη ενέργεια του. Τα κυριότερα είδη διαλόγων είναι ο διάλογος ειδοποίησης(Alert Dialog) και ο διάλογος προόδου(ProgressDialog).

### 2.5.2.1. Alert Dialog

Ο Alert Dialog είναι ο πιο συνηθισμένος διάλογος. Αποτελείται από ένα τίτλο, ένα μήνυμα, κάποια κουμπιά ή μια λίστα από επιλογές. Για κάθε κουμπί του διαλόγου, ορίζουμε μέσα στην activity τις ενέργειες που θα ακολουθήσουν όταν το πατήσει ο χρήστης.

```
AlertDialog alertDialog = new AlertDialog.Builder(
AlertDialogActivity.this).create();
alertDialog.setTitle("Alert Dialog");
alertDialog.setMessage("Welcome to AndroidHive.info");
alertDialog.setIcon(R.drawable.tick);
    alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
Toast.makeText(getApplicationContext(), "You clicked on OK",
Toast.LENGTH_SHORT).show();
}
});
alertDialog.show();
```

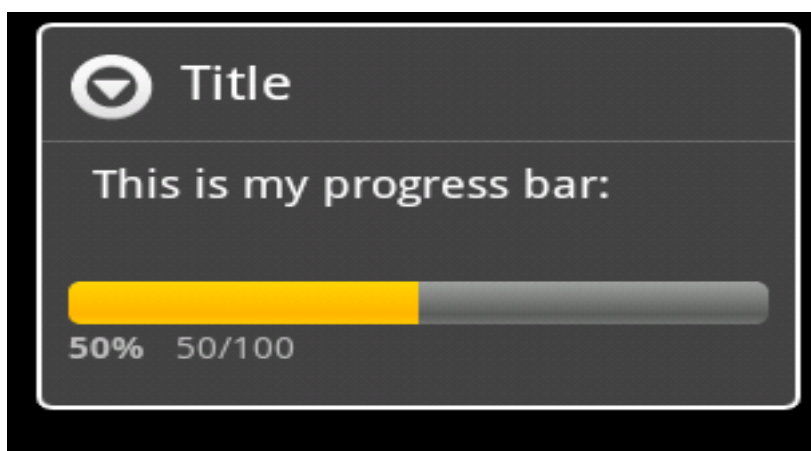


Εικόνα 2.18. Παράδειγμα Alert Dialog

#### 2.5.2.2. ProgressDialog

Ο ProgressDialog αποτελεί επέκταση του Alert Dialog και χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στο χρήστη τη πρόοδο για κάποια ενέργεια. Για να μη βλέπει ο χρήστης μια μαύρη οθόνη, όπου δε μπορεί να κάνει τίποτα και μπορεί να νομίζει ότι η εφαρμογή κάνει κάποιο bug, εμφανίζουμε ένα ,σαν το επόμενο:

```
ProgressDialog dialog = ProgressDialog.show(MyActivity.this, "",  
"Loading. Please wait...", true);
```



Εικόνα 2.19. Παράδειγμα ProgressDialog

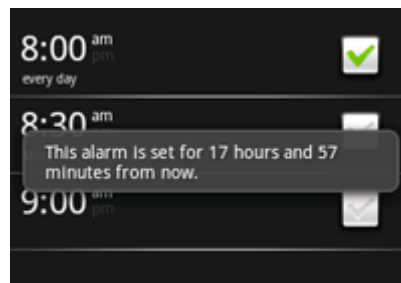
Για να ακυρώσουμε ένα διάλογο, υπάρχουν δυο τρόποι. Συνήθως θέλουμε να δώσουμε την ελευθερία στο χρήστη για να ακυρώσει ένα διάλογο με το πιο φυσικό τρόπο του Android, δηλαδή το back button. Το γεγονός αυτό το επιτυγχάνουμε ορίζοντας το διάλογό μας ακυρώσιμο (cancellable). Στις ενέργειες που ακολουθούν όταν ο χρήστης πατήσει κάποιο κουμπί, τις περισσότερες φορές πρέπει να συμπεριλάβουμε και την εντολή `myDialog.dismiss()` (αν έχουμε ονομάσει το διάλογό μας `myDialog`) ώστε ο διάλογός μας να εξαφανιστεί και έπειτα να συνεχίσει η ροή της εφαρμογής ανάλογα με την επιλογή του χρήστη.

### 2.5.3. Ειδοποιήσεις χρήστη (notifications)

Συνήθως θέλουμε να ειδοποιήσουμε τον χρήστη για κάποιο γεγονός ή αποτέλεσμα σχετικά με την εφαρμογή μας. Σε κάποια από αυτά ο χρήστης πρέπει να δώσει μια απάντηση σε άλλα όχι. Για παράδειγμα όταν ο χρήστης αποθηκεύει ένα αρχείο είναι σωστό να τον ενημερώσουμε το πότε αυτό έγινε. Ή όταν η εφαρμογή μας κατεβάζει μια εικόνα στο παρασκήνιο θα πρέπει να ειδοποιήσει τον χρήστη όταν κατέβηκε για το αν θέλει να ανοίξει την εικόνα ή όχι. Στο πρώτο παράδειγμα, όπου δεν είναι αναγκαία η παρέμβαση του χρήστη θα χρησιμοποιήσουμε `Toast notification`, ενώ στην δεύτερη `status bar notification`.

#### 2.5.3.1. Toast notification

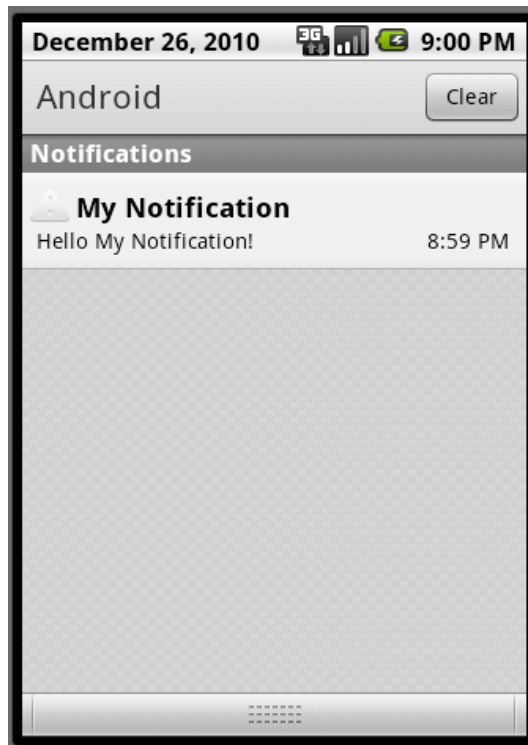
Πρώτα θα εξετάσουμε τα `Toast notifications` (Εικόνα 20). Ένα `toast notification` είναι ένα μήνυμα που εμφανίζεται για λίγα δευτερόλεπτα στο παράθυρο που βρίσκεται ο χρήστης σε οποιαδήποτε εφαρμογή και αν βρίσκεται. Ο χώρος που καταλαμβάνει είναι ο ελάχιστος απαιτούμενος ώστε αυτό να είναι εμφανές, ενώ ο χρήστης είναι ελεύθερος να αλληλεπιδρά με την `activity` στην οποία βρίσκεται. Δεν υπάρχει κάποια αλληλεπίδραση του χρήστη με αυτό. Χρησιμοποιείται συνήθως για μικρά μηνύματα που απώτερου σκοπού έχουν μόνο την ενημέρωση του χρήστη και τίποτα άλλο, για παράδειγμα «Το αρχείο κατέβηκε επιτυχώς», «Το ζυπνητήρι ορίστηκε σε ...». Συγκεκριμένα:



Εικόνα 2.20. Παράδειγμα toastnotification

### 2.5.3.2. Status bar notification

Στην συνέχεια έχουμε τα status bar notifications που όπως φανερώνει και το όνομα του είναι μια ειδοποίηση που εμφανίζεται στη status bar της συσκευής μας και την οποία μπορούμε να ανοίξουμε είτε βρισκόμαστε στην αρχική οθόνη (home screen), είτε μέσα σε οποιαδήποτε εφαρμογή. Αντίθετα με το προηγούμενο, η status bar notification μπορεί να επιλεχθεί και να ξεκινήσει κάποια λειτουργία ανάλογα με τις ενέργειες που έχουμε ορίσει στο κώδικα της εφαρμογής. Για παράδειγμα ,όταν κατεβάζουμε ένα αρχείο από το διαδίκτυο,όταν η λήψη ολοκληρωθεί ,θα θέλαμε να επιλέξουμε την ειδοποίηση αυτή και με τον τρόπο αυτό ,είτε να ανοίξουμε το φάκελο στον οποίο βρίσκεται το αρχείο ,είτε να το τρέξουμε.Τις περισσότερες φορές,οι Toast notification ενεργοποιούνται από activities,ενώ οι status bar notifications από services.



Εικόνα 2.21. Παράδειγμα status bar notification

## ΚΕΦΑΛΑΙΟ 3: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ANDROID
















### 3.1 Εργαλεία προγραμματισμού

Η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη εφαρμογών Android είναι η Java, ενώ τους τελευταίους μήνες γίνονται προσπάθειες για να συμπεριληφθούν και άλλες γλώσσες όπως η C και η C++.

Οπότε βασική προϋπόθεση είναι να διαθέτουμε τα αντίστοιχα εργαλεία της γλώσσας προγραμματισμού και συγκεκριμένα το Java Development Kit (JDK). Ακόμη χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment ή IDE) για να μεταγλωττίσουμε και να τρέξουμε τα προγράμματα μας, όπως είναι για παράδειγμα το Eclipse (εικόνα ). Το βασικότερο εργαλείο αποτελεί το Android SDK ( software development kit ) το οποίο ουσιαστικά μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα για την κατασκευή μιας εφαρμογής Android. Η σύνδεση του Android SDK με το γραφικό μας περιβάλλον (Eclipse) γίνεται μέσω μιας επέκτασης ( Android Development Tools ή ADT Plug-in ) που κάνουμε εγκατάσταση στο Eclipse, ώστε να μπορέσουμε να μεταγλωττίσουμε την εφαρμογή μας και έπειτα να την τρέξουμε.

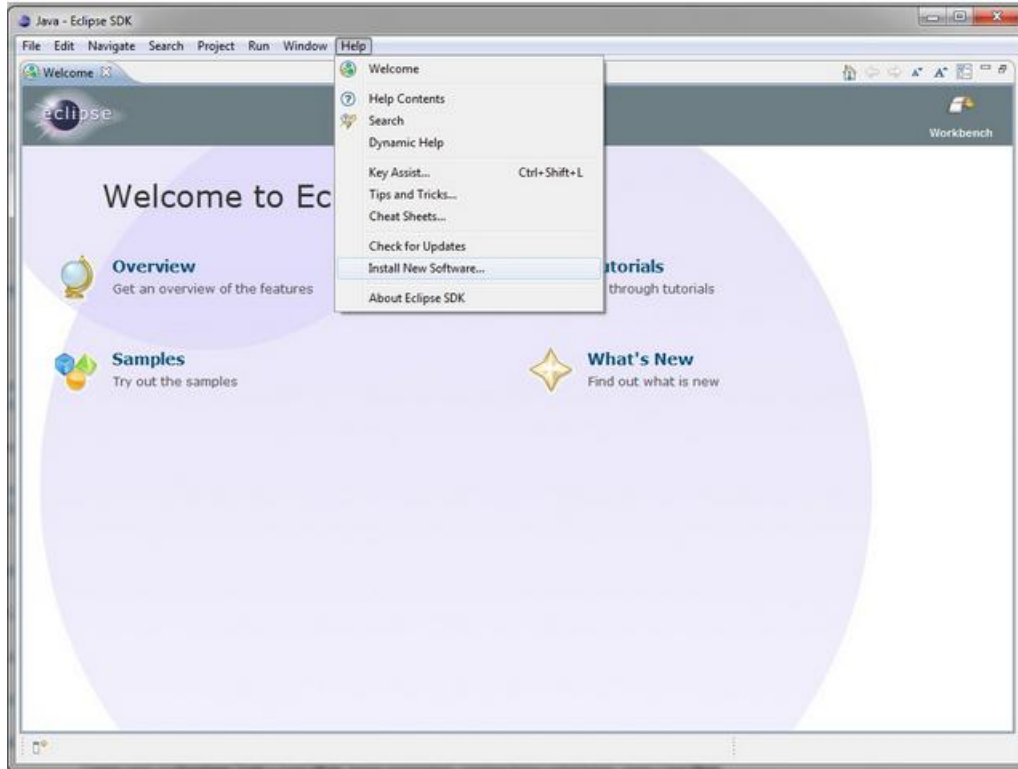
#### 3.1.1. Δημιουργία ενός Android Development Environment

Κατεβάζουμε το “**Java Development Kit**” (JDK) v.7 (οποιαδήποτε έκδοση >= v.6 είναι κατάλληλη) από την επίσημη ιστοσελίδα επιλέγοντας την έκδοση του λειτουργικού συστήματός μας, όπως φαίνεται στην εικόνα . Εκτελούμε το αρχείο που κατεβάσαμε. Στη διαδικασία της εγκατάστασης, αποδεχόμαστε τις προεπιλεγμένες ρυθμίσεις. Με την επιτυχή περάτωση της διαδικασίας, το JDK θα έχει εγκατασταθεί στη διαδρομή “C:\Program Files\Java\jdk1.7.0\_17”.

Java SE Development Kit 7u25		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	80.38 MB	 jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	 jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	 jdk-7u25-linux-x64.rpm
Linux x64	91.85 MB	 jdk-7u25-linux-x64.tar.gz
Mac OS X x64	144.43 MB	 jdk-7u25-macosx-x64.dmg
Solaris x86 (SVR4 package)	136.02 MB	 jdk-7u25-solaris-i586.tar.Z
Solaris x86	92.22 MB	 jdk-7u25-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.77 MB	 jdk-7u25-solaris-x64.tar.Z
Solaris x64	15.09 MB	 jdk-7u25-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	16 MB	 jdk-7u25-solaris-sparc.tar.Z
Solaris SPARC	9 MB	 jdk-7u25-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	 jdk-7u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.67 MB	 jdk-7u25-solaris-sparcv9.tar.gz
Windows x86	89.09 MB	 jdk-7u25-windows-i586.exe
Windows x64	90.66 MB	 jdk-7u25-windows-x64.exe

Εικόνα 3.1. JDK

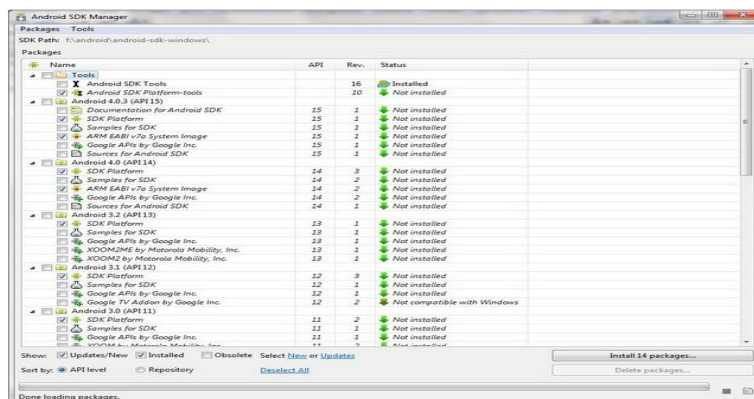
Έχοντας ολοκληρώσει την εγκατάσταση του JDK, συνεχίζουμε με την εγκατάσταση και παραμετροποίηση του περιβάλλοντος ανάπτυξης Android. κάνουμε χρήση του ολοκληρωμένου περιβάλλοντος ανάπτυξης Android (ουσιαστικά περιλαμβάνει ειδική έκδοση **Eclipse IDE** με ενσωματωμένο Android SDK), του λεγόμενου ADT (Android Development Tools) bundle. Επιλέγουμε και κατεβάζουμε την ανάλογη έκδοση που αντιστοιχεί στο ΛΣ μας (αφότου συμβουλευτούμε τις επίσημες απαιτήσεις του λογισμικού, στην ενότητα “SYSTEM REQUIREMENTS”), όπως φαίνεται στην ακόλουθη εικόνα.



Εικόνα 3.2.Eclipse IDE

Αποσυμπιέζουμε το αρχείο “zip” και μετονομάζουμε τον παραγόμενο φάκελο, δίνοντάς του ένα όνομα της μορφής “adt-bundle-windows-x86\_32”.

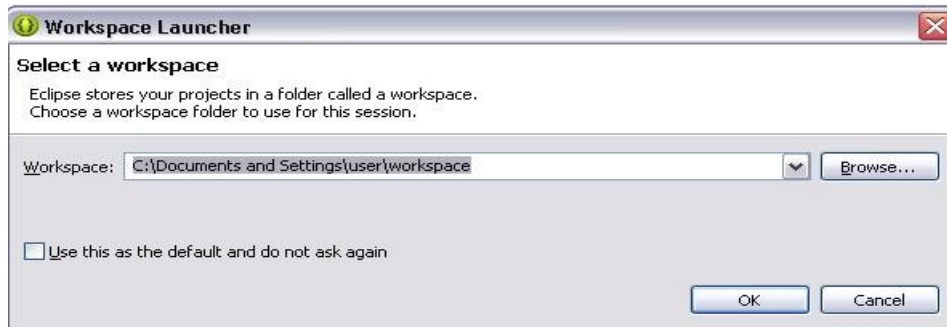
Ανοίγουμε τον φάκελο “adt-bundle-windows-x86\_32” και εκτελούμε το πρόγραμμα “SDK Manager”, όπως αυτό φαίνεται στην ακόλουθη εικόνα.



Εικόνα 3.3.SDK Manager



Μόλις ανοίξει η εφαρμογή, θα μας ζητήσει να επιλέξουμε «χώρο εργασίας» (workspace) . Αφήνουμε την προεπιλεγμένη τιμή (στην συγκεκριμένη περίπτωση είναι "C:\Documents and Settings\Users\C\workspace"), όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 3.4.Επιλογή χώρου εργασίας

### 3.1.2.Δημιουργία εικονικής συσκευής Android (Android Virtual Devices – AVD)

Ο developer πρέπει πριν να εκδώσει την εφαρμογή του να την δοκιμάσει σε ένα αριθμό συσκευών για να εξασφαλίσει την ομαλή λειτουργία της σε όλες τις συνθήκες. Φυσικά το κόστος των συσκευών είναι αρκετά μεγάλο για να αποθαρρύνει τον προγραμματιστή να έχει στην κατοχή του 10-20 συσκευές για να ελέγξει σε όλες τις λειτουργίες και την εμφάνιση της εφαρμογής του. Το πρόβλημα αυτό έρχεται να λύσει η ύπαρξη των εικονικών συσκευών του Android.

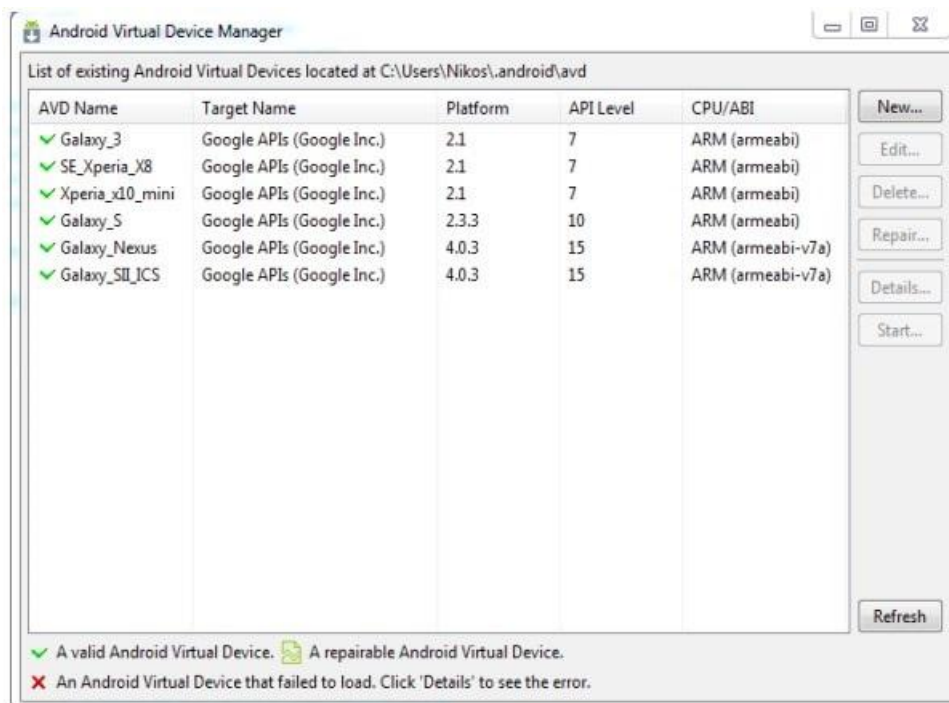
Πρόκειται για μια συσκευή εξομοίωσης η οποία μας επιτρέπει να εξομοιώσουμε την λειτουργία και συμπεριφορά μιας κανονικής συσκευής, ορίζοντας τις επιλογές υλικού και λογισμικού που θέλουμε στον εξομοιωτή του Android. Με αυτό τον τρόπο ο developer μπορεί να ελέγξει την εφαρμογή του σε μια σειρά από πραγματικά σενάρια λειτουργίας και να πάρει γρήγορα και άμεσα feedback για τη λειτουργία της εφαρμογής του.

Μια εικονική συσκευή αποτελείται από:

- Το προφίλ του υλικού: Σε αυτό προσδιορίζονται οι ιδιότητες και τα χαρακτηριστικά της εικονικής συσκευής. Μπορούμε παραδείγματος χάρη να ορίσουμε την ανάλυση της οθόνης και την πυκνότητα σε pixel (dpi), το μέγεθος της μνήμης RAM, αν η συσκευή θα έχει κάμερα, υποστήριξη GPS, κλπ.
- Την έκδοση του Android: Επιλογή της έκδοσης της πλατφόρμας του Android που θέλουμε να εξομοιώσει η εικονική συσκευή. Μπορούμε επίσης να επιλέξουμε και μεταξύ ειδικών εκδόσεων της πλατφόρμας, μεταξύ των οποίων τις Google TV, και άλλων.
- Έξτρα χώρος αποθήκευσης: Εδώ αποθηκεύονται όλα τα δεδομένα της εφαρμογής, και επίσης μπορούμε να ορίσουμε μια εικονική κάρτα μνήμης ώστε να επεκτείνουμε τον αποθηκευτικό χώρο, όπως θα κάναμε και σε μια πραγματική συσκευή.

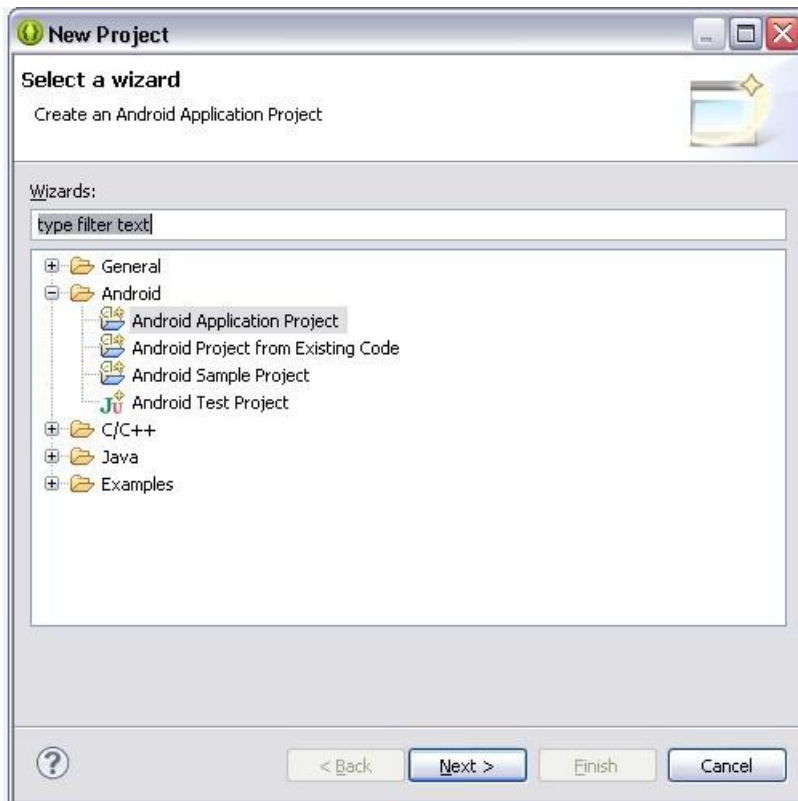
Δημιουργούμε έναν εξομοιωτή Android στο σύστημα μας. Για να τρέξει ο Java κώδικας της εφαρμογής σας, τα Android εργαλεία στηρίζονται σε έναν emulator (εξομοιωτής) ο οποίος προσποιείται ότι είναι μια Android κινητή συσκευή. Από την πλευρά του προγραμματιστή, αυτό είναι ένα πολύ καλό και θετικό στοιχείο γιατί μπορούμε να αναπτύσουμε Android εφαρμογές χωρίς να έχουμε Android συσκευή και να δοκιμάζουμε τις εφαρμογές μας αν τρέχουν σωστά σε Android περιβάλλον.

Μέσα από το Android SDK και το AVD Manager που κατεβάσαμε και εγκαταστήσαμε μπορούμε να δημιουργήσουμε ένα AVD. Πατάμε το εικονίδιο που έχει ήδη τοποθετηθεί στο κύριο μενού του Eclipse ή Επιλέγουμε την επιλογή “Android Virtual Device Manager” από το μενού “Window”, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 3.5.AVD Manager

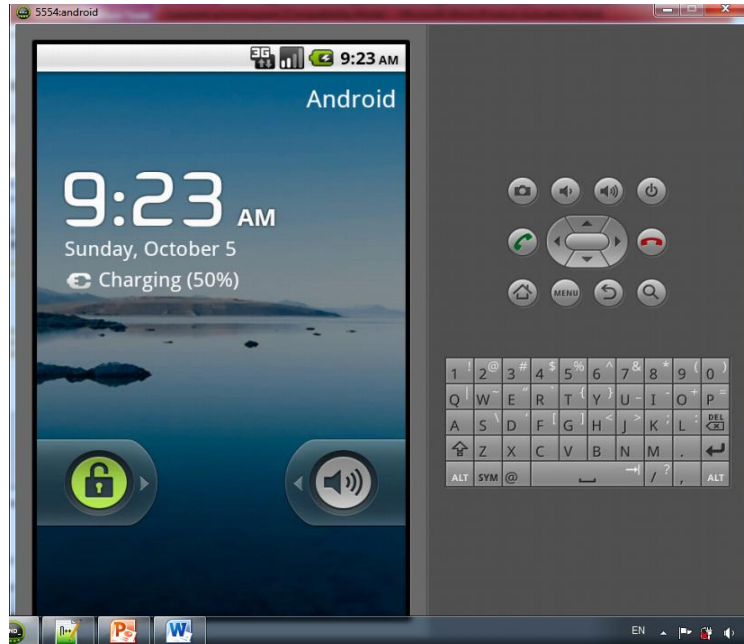
Πατώντας το κουμπί New εμφανίζεται ένα νέο παράθυρο δημιουργίας εικονικής συσκευής (Εικόνα 3.6), στο οποίο μπορούμε να συμπληρώσουμε τα χαρακτηριστικά του hardware που επιθυμούμε. Στο *Create new Android Virtual Device (AVD)* παράθυρο συμπληρώνουμε τα εξής στοι



Εικόνα 3.6. Παράθυρο δημιουργίας νέας Εικονικής μηχανής

- Name: Το όνομα της συσκευής που δημιουργούμε.
- Target: Εδώ επιλέγουμε μια από τις API εκδόσεις που κατεβάσαμε ήδη όταν χρησιμοποιήσαμε τον SDK Manager. Για το παράδειγμά μας, επιλέγουμε το Android 2.1 – API Level 7.
- SD Card (Size): Εδώ καθορίζουμε το μέγεθος της SD card. Κάθε σημερινή κινητή συσκευή μπορεί να έχει SD χώρο που φτάνει σε αρκετά GB. Αλλά για το δικό μας παράδειγμα δεν χρειάζεται να δηλώσουμε τόσο μεγάλο αποθηκευτικό χώρο. Μια SD card του μεγέθους των 512 MB είναι αρκετή.
- Skin: Εδώ ορίζουμε την ανάλυση που θέλουμε να έχει το virtual device που θα δημιουργήσουμε. Η προκαθορισμένη επιλογή (Default WVGA800) που έχει ανάλυση οθόνης 800×480 είναι ικανοποιητική για το παράδειγμά μας.

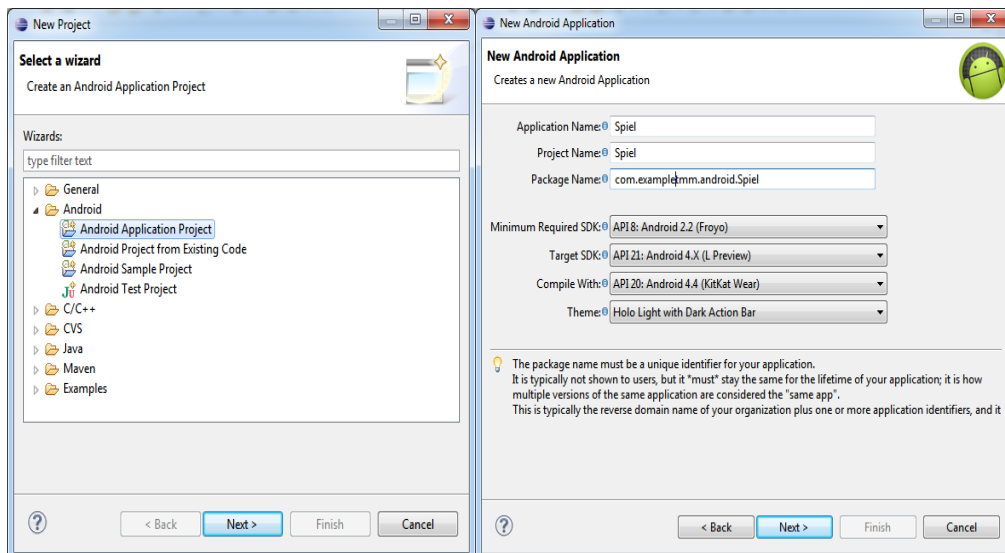
Πατάμε το *Create AVD* κουμπί για να δημιουργηθεί η συσκευή μας:



Εικόνα 3.7.Εικονική συσκευή

### 3.1.3. Δημιουργία Android εφαρμογής

Σαν πρώτο βήμα, ξεκινήστε το Eclipse. Ύστερα, πηγαίετε στο *File* από το κύριο μενού και επιλέξτε *New* και έπειτα *Project*.



Εικόνα 3.8.Τα στάδια δημιουργίας ενός νέου project

Όταν εμφανιστεί το *New Project* παράθυρο, κάνετε κλικ επάνω στον Android φάκελο και από τα περιεχόμενα του επιλέξετε *Android Project*. Πατήστε το κουμπί *NEXT* για να συνεχίσετε στο επόμενη βήμα:

Φτάνουμε στο πιο σημαντικό παράθυρο (*Application Info*) της διαδικασίας δημιουργίας ενός Android Project το οποίο καθορίζει το όνομα της εφαρμογής, όπως και το όνομα της java κλάσης η οποία θα εκτελεστεί. -Στο πεδίο *Application Name* γράφουμε το όνομα που θέλουμε να δώσουμε στην εφαρμογή μας. Είναι το ίδιο όνομα που θα εμφανίζεται και στην Android συσκευή όταν εγκαταστήσουμε την εφαρμογή μας. Οπότε προσέξτε με την ονομασία. -Κάθε κλάση στον προγραμματισμό και ειδικότερα στην java περιέχεται μέσα σε ένα πακέτο.

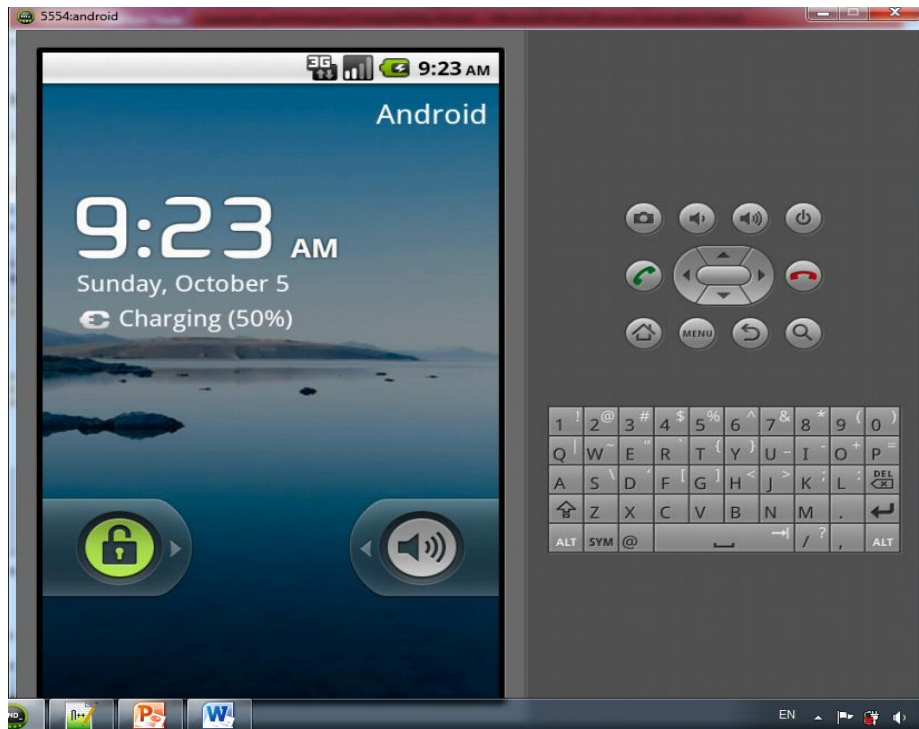
Τα packages, όπως είναι η ονομασία τους, ταξινομούν και ομαδοποιούν .java αρχεία για ευκολότερη αντιμετώπιση προβλημάτων σε επίπεδο κώδικα αλλά και για να καθορίζουν δικαιώματα πρόσβασης. Σε εφαρμογές Android είναι υποχρεωτικό να δημιουργήσουμε ένα πακέτο μέσα στο οποίο θα περιέχεται η java κλάση της Android εφαρμογής.

-Το όνομα του Activity είναι το όνομα της java κλάσης μέσα από την οποία ξεκινάει η εκτέλεση της Android εφαρμογής. Θα αναλύσουμε αυτή την κλάση εκτενέστερα σε μελλοντικά μαθήματα.

Ο αριθμός στο πεδίο *Minimum SDK* βασίζεται στην επιλογή του API που κάναμε στο προηγούμενο βήμα. Δεν χρειάζεται να αλλάξουμε αυτό τον αριθμό. Όταν συμπληρώσετε τα αναγκαία πεδία, θα ενεργοποιηθεί το κουμπί *FINISH*:

Μετά το τέλος της διαδικασίας, θα δείτε στον Package Explorer του Eclipse τα αρχεία από τα οποία αποτελείται το project σας. Για να τρέξουμε την εφαρμογή μας, κάνουμε δεξί κλικ επάνω στο όνομα του project και επιλέγουμε *Run As* και μετά *Android Application*. Αφού το περιβάλλον του Eclipse επιβεβαιώσει τον κώδικα του project (η διαδικασία του compilation), μετά θα ξεκινήσει τον emulator που είχαμε ορίσει έτσι ώστε να εμφανιστεί ένα virtual smartphone στην οθόνη σας.

Για να δοκιμάσουμε την εφαρμογή μας και να δούμε τα αποτελέσματα της θα χρειαστούμε μια φυσική συσκευή, όπως ένα κινητό τηλέφωνο ή ένα tablet που χρησιμοποιούν λειτουργικό σύστημα Android, συνδέοντας το απλά σε μια θύρα USB του υπολογιστή μας ή μια εικονική συσκευή AVD.



Εικόνα 3.9 .Android Virtual Device ή AVD

Όλα τα παραπάνω προγραμματιστικά εργαλεία μπορεί όποιος επιθυμεί να τα αναζητήσει εύκολα και δωρεάν στο διαδίκτυο από τους κατασκευές στις επίσημες ιστοσελίδες τους.



Εικόνα 3.10. Λογότυπο Eclipse

## ΚΕΦΑΛΑΙΟ 4: Η ΕΦΑΡΜΟΓΗ

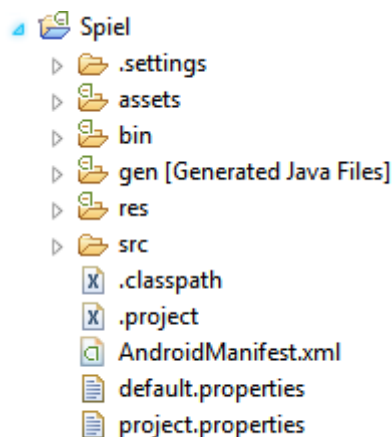
### 4.1 Περιγραφή

Ο κύριος στόχος της εφαρμογής είναι η κατασκευή ενός παιχνιδιού, το οποίο δίνει τη δυνατότητα στο χρήστη να

### 4.2.Δομή

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Θα παρουσιάσουμε πρώτα την δομή της εφαρμογής, πώς δηλαδή

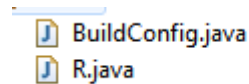
Εclipse στο οποίο αναπτύχθηκε η εφαρμογή, το project εμφανίζεται με τη μορφή :



Εικόνα 4.1.Δομή της εφαρμογής Spiel

#### 4.2.1.Φάκελος gen /

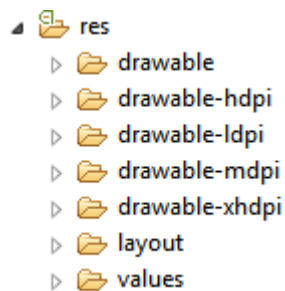
Στο φάκελο gen (generated java files) βρίσκονται διάφορα αρχεία τα οποία κρατάει η κλάση R η οποία κρατάει τα αρχεία που εισάγονται. Κάθε φορά που εισάγεται ένα αρχείο οπουδήποτε στον κατάλογο /res το plugin παρατηρεί την αλλαγή και προσθέτει τα κατάλληλα αρχεία. Το αρχείο R.java μένει πάντα συγχρονισμένο.



Εικόνα 4.2.Περιεχόμενα φακέλου gen/

#### 4.2.2.Φάκελος res /

Ο φάκελος res (resources) περιέχει όλα τα application resources τα οποία χρησιμοποιούμε ,δηλαδή αρχεία εικόνας, κειμένου, xml,layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο src.Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους.



Εικόνα 4.3.Περιεχόμενα φακέλου res /

Οι υποφάκελοι του κύριου φακέλου res, είναι :

- ο φάκελος **drawable** ο οποίος περιέχει τα αρχεία εικόνας (.png,.jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή μας.Εδώ τοποθετούμε το εικονίδιο της εφαρμογής μας. Το Android ζητάει η κάθε ανάλυση να έχει το δικό της εικονίδιο, αλλιώς στην θέση του εμφανίζει το προεπιλεγμένο εικονίδιο με το πράσινο ρομποτάκι. Επειδή δεν το θέλουμε αυτό θα κάνουμε resize το εικονίδιο της εφαρμογής μας για να τοποθετήσουμε την σωστή διάσταση εικονιδίου εφαρμογής στον σωστό φάκελο. Για τις διαστάσεις ισχύουν οι τιμές του παρακάτω πίνακα.Έτσι εξασφαλίζουμε ότι σε καμία περίπτωση οθόνης δεν θα έχουμε φαινόμενα cropping των εικονιδίων μας. Το λειτουργικό σύστημα θα εντοπίσει αυτόματα την οθόνη της συσκευής του χρήστη και θα επιλέξει τα κατάλληλα αρχεία. (Χαμηλή- ldpi,μέτρια-mdpi, υψηλή-hdpi και πολύ υψηλή-xhdpi )



	ldpi (120 dpi) (Low density screen)	mdpi (160 dpi) (Medium density screen)	hdpi (240 dpi) (High density screen)	xhdpi (320 dpi) (Extra-high density screen)
Launcher Icon Size	36 x 36 px	48 x 48 px	72 x 72 px	96 x 96 px

Πίνακας 4.1.Icon size

## *background.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"
    android:src="@drawable/Logosmall"
    android:tileMode="repeat" />
```



Εικόνα 4.4.background.xml

## *button-selector.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_pressed="true"
        android:drawable="@drawable/green_button" />

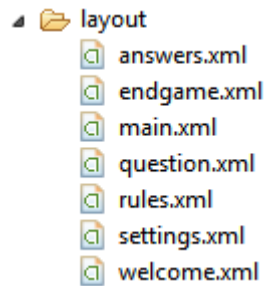
    <!-- <item android:state_focused="true"
        android:drawable="@drawable/button_focused" /> -->

    <item android:drawable="@drawable/black_button" />
</selector>
```



Εικόνα 4.5.button-selector.xml

- ο φάκελος **layout** ο οποίος περιέχει όλα τα αρχεία xml ,στα οποία δηλώνουμε το user interface της κάθε οθόνης.



Εικόνα 4.6.Περιεχόμενα φακέλου layout

## *main.xml*

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

## welcome.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    android:background="@drawable/background">

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Play" android:id="@+id/playBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"

            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:textColor="#ffffff" />

    </LinearLayout>

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Settings"
            android:id="@+id/settingsBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:textColor="#ffffff" />

    </LinearLayout>

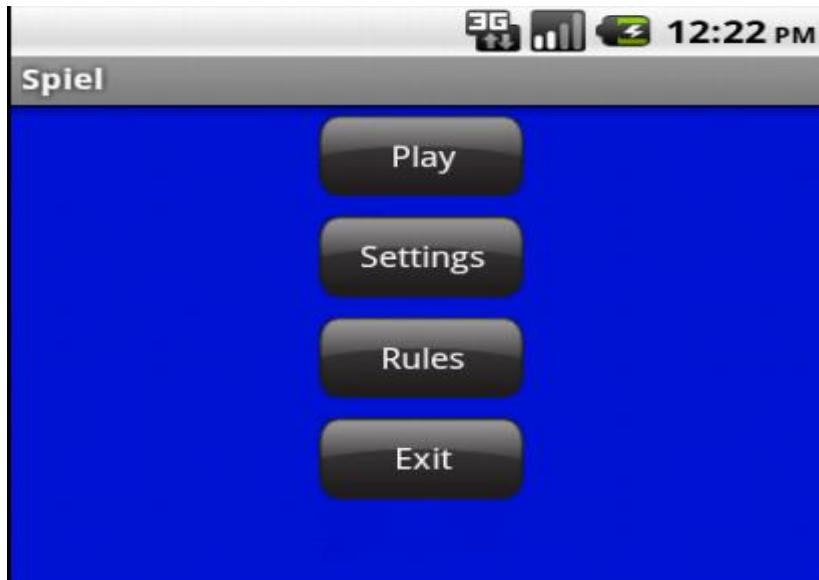
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Rules" android:id="@+id/rulesBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:textColor="#ffffff" /></LinearLayout>
```

```
        <LinearLayout android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:gravity="center_horizontal">

            <Button android:text="Exit" android:id="@+id/exitBtn"
                android:layout_width="80dip"
                android:layout_alignParentRight="true"
                android:layout_height="wrap_content"
                android:background="@drawable/button_selector"
                android:paddingTop="5dip" android:paddingBottom="5dip"
                android:textColor="#ffffff" />

        </LinearLayout>
</LinearLayout>
```



Εικόνα 4.7.Περιεχόμενα υποφακέλου welcome.xml φακέλου layout

## *question.xml*

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    android:background="@drawable/background">
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <RadioGroup android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:id="@+id/group1">

            <TextView android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:id="@+id/question"/>

            <RadioButton android:checked="false"
                android:id="@+id/answer1" />

            <RadioButton android:checked="false"
                android:id="@+id/answer2" />

            <RadioButton android:checked="false"
                android:id="@+id/answer3" />

            <RadioButton android:checked="false"
                android:id="@+id/answer4" />

        </RadioGroup>

    </LinearLayout>

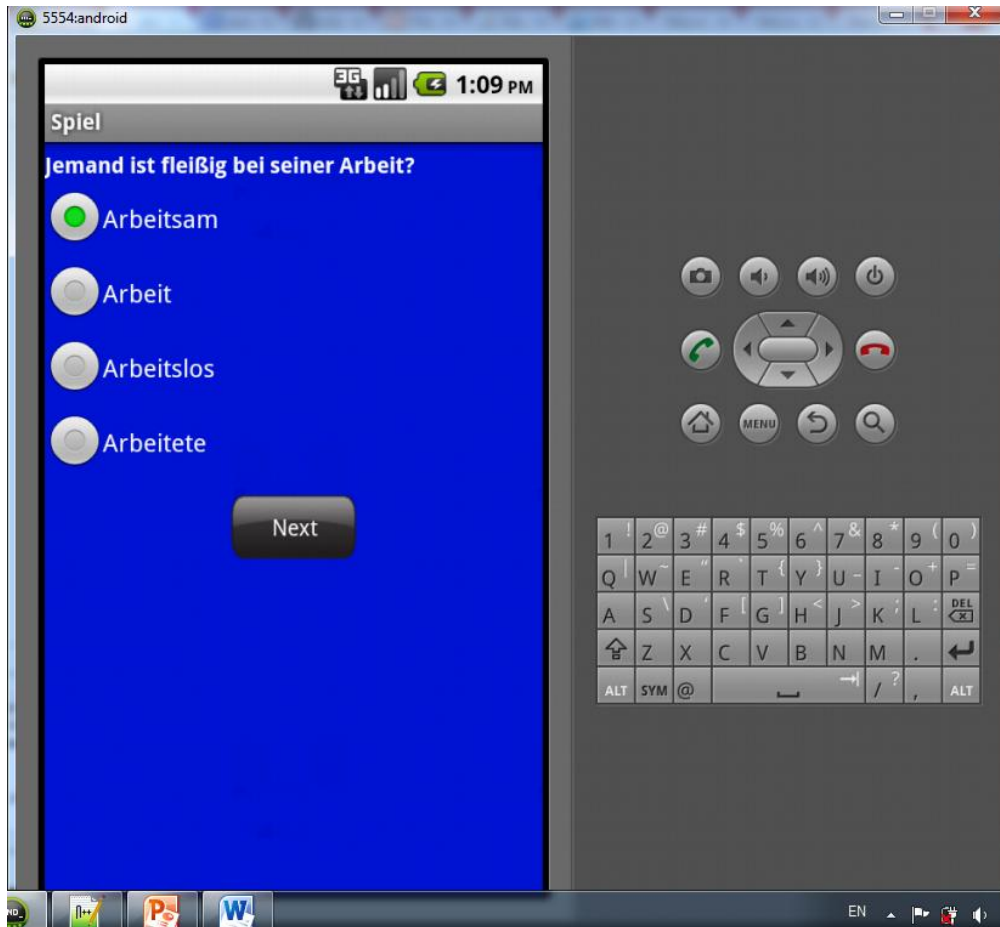
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Next" android:id="@+id/nextBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:textColor="#ffffff" />

```

```
</LinearLayout>
```

```
</LinearLayout>
```



Εικόνα 4.8. Περιεχόμενα υποφακέλου question.xml φακέλου layout

## *answers.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/ScrollView01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center_horizontal"
        android:background="@drawable/background">

        <LinearLayout android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:paddingTop="5dip" android:paddingBottom="5dip">
```

```

        android:gravity="center_horizontal">

        <TextView android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="#ffffff"
            android:textStyle="bold" android:id="@+id/answers"
            android:paddingLeft="15dip"
            android:paddingRight="15dip"
            android:gravity="center_horizontal"/>

    </LinearLayout>

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Back" android:id="@+id/finishBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"

            android:paddingTop="5dip"
            android:paddingBottom="5dip"
            android:textColor="#ffffff" />

    </LinearLayout>

</LinearLayout>
</ScrollView>

```

## *endgame.xml*

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/ScrollView01" android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center_horizontal"
        android:background="@drawable/background">

        <LinearLayout android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:gravity="center_horizontal">

            <TextView android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:id="@+id/endgameResult"

```

```
                android:paddingLeft="15dip"
android:paddingRight="15dip"
                android:gravity="center_horizontal" />

</LinearLayout>

<LinearLayout android:orientation="horizontal"
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
    android:paddingTop="1dip" android:paddingBottom="1dip"
    android:gravity="center_horizontal">

    <ImageView android:id="@+id/resultPage"
        android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:scaleType="fitXY" />

</LinearLayout>

<LinearLayout android:orientation="horizontal"
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
    android:paddingTop="5dip" android:paddingBottom="5dip"
    android:gravity="center_horizontal">

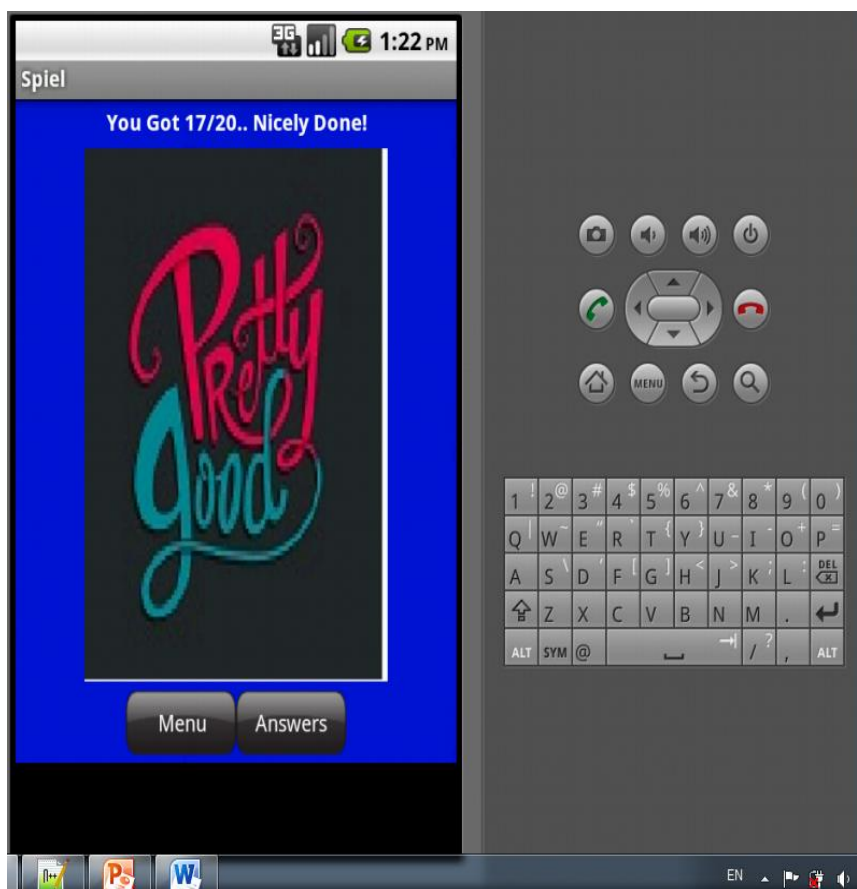
    <Button android:text="Menu" android:id="@+id/finishBtn"
        android:layout_width="80dip"
android:layout_alignParentRight="true"
        android:layout_height="wrap_content"
android:background="@drawable/button_selector"
        android:paddingTop="5dip"
android:paddingBottom="5dip"
        android:textColor="#ffffff" />

    <Button android:text="Answers"
android:id="@+id/answerBtn"
        android:layout_width="80dip"
android:layout_alignParentRight="true"
        android:layout_height="wrap_content"
android:background="@drawable/button_selector"
        android:paddingTop="5dip"
android:paddingBottom="5dip"
        android:textColor="#ffffff" />

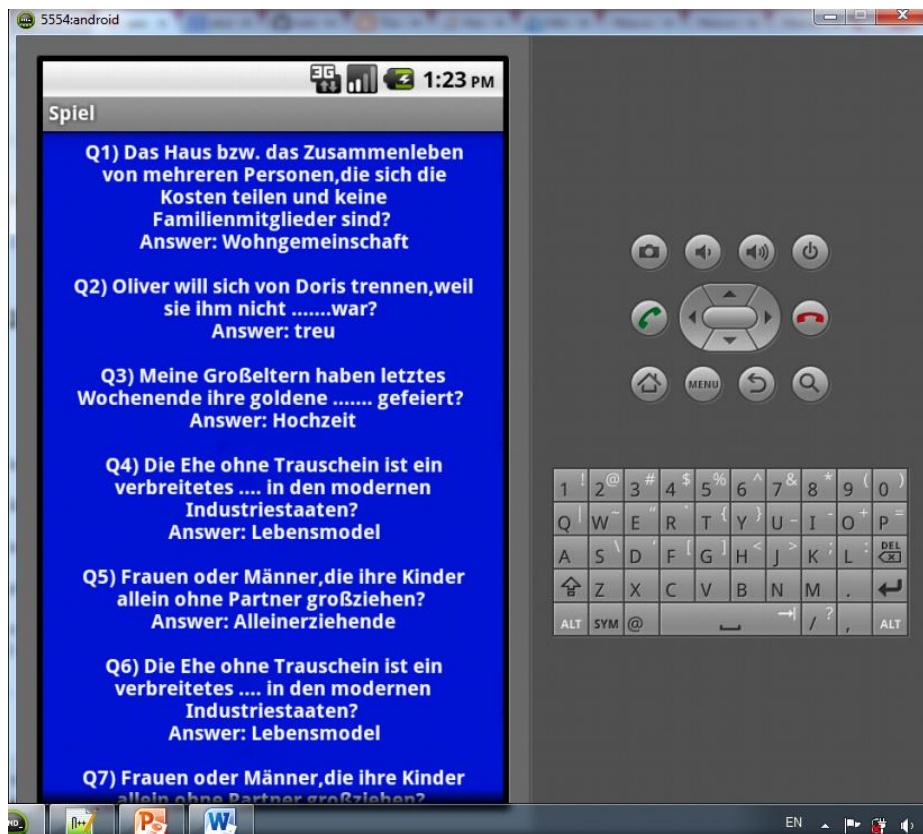
</LinearLayout>

</LinearLayout>
</ScrollView>
```





Εικόνα 4.9. Περιεχόμενα υποφακέλου endGame.xml φακέλου layout



Εικόνα 4.10. Σωστές απαντήσεις

## *settings.xml*

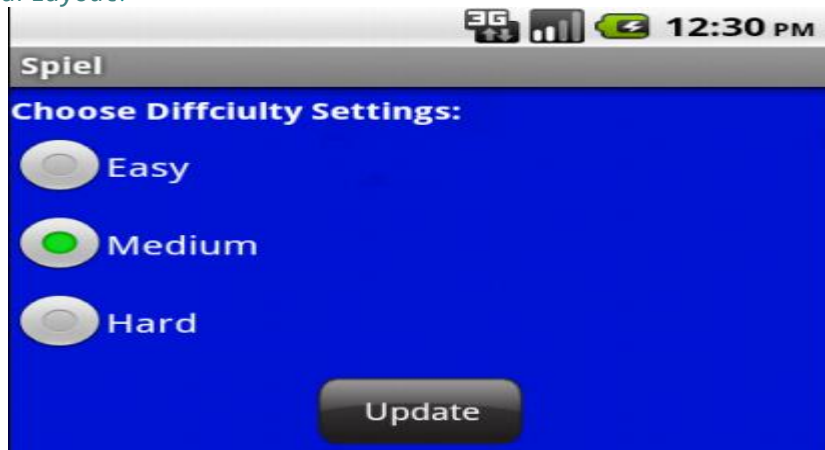
```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    android:background="@drawable/background">
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">
        <RadioGroup android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:id="@+id/group1">
            <TextView android:layout_width="fill_parent" android:text="Choose
            Difficulty Settings:"
                android:layout_height="wrap_content"
                android:textColor="#ffffff">
```

```

                android:textStyle="bold"
android:id="@+id/question"/>
    <RadioButton android:checked="false" android:id="@+id/easySetting"
android:text="Easy"/>
    <RadioButton android:checked="false" android:id="@+id/mediumSetting"
android:text="Medium" />
    <RadioButton android:checked="false" android:id="@+id/hardSetting"
android:text="Hard" />
</RadioGroup>
</LinearLayout>
<LinearLayout android:orientation="horizontal"
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
    android:paddingTop="5dip" android:paddingBottom="5dip"
    android:gravity="center_horizontal">
    <Button android:text="Update" android:id="@+id/nextBtn"
        android:layout_width="80dip"
android:layout_alignParentRight="true"
        android:layout_height="wrap_content"
android:background="@drawable/button_selector"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:textColor="#ffffff" />
</LinearLayout>
</LinearLayout>

```



Εικόνα 4.11. Περιεχόμενα υποφακέλου settings.xml φακέλου layout

## *rules.xml*

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    android:background="@drawable/background">

    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <TextView android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="#ffffff"
            android:textStyle="bold" android:id="@+id/rulesText"
            android:paddingLeft="15dip" android:paddingRight="15dip"
            android:text="@string/rules"/>

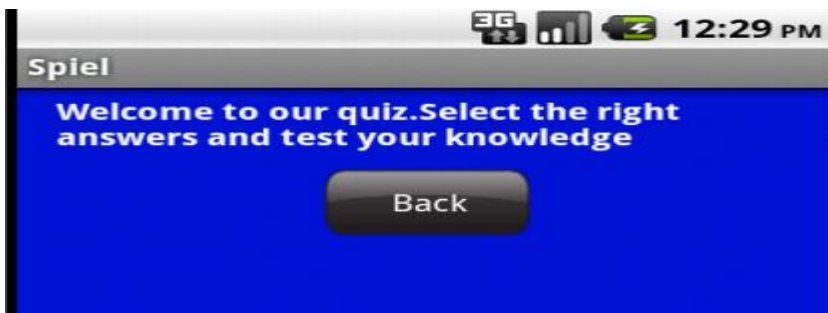
    </LinearLayout>
    <LinearLayout android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dip" android:paddingBottom="5dip"
        android:gravity="center_horizontal">

        <Button android:text="Back" android:id="@+id/backBtn"
            android:layout_width="80dip"
            android:layout_alignParentRight="true"
            android:layout_height="wrap_content"
            android:background="@drawable/button_selector"
            android:paddingTop="5dip" android:paddingBottom="5dip"
            android:textColor="#ffffff" />

    </LinearLayout>

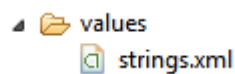
</LinearLayout>

```



Εικόνα 4.12. Περιεχόμενα υποφακέλου rules.xml φακέλου layout

- ο φάκελος **values** στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου (String Resources) που χρησιμοποιούνται στην εφαρμογή. Οι πόροι αυτοί μπορεί να αποτελούνται από αλφαριθμητικά (Strings), πίνακες αλφαριθμητικών, ακέραιους, πίνακες ακεραίων, λίστες με κωδικούς χρωμάτων για εύκολη αναφορά κλπ. Στην περίπτωση μας χρησιμοποιήσαμε το αρχείο strings.xml με όλα τα string της εφαρμογής δηλωμένα εκεί, το αρχείο colors.xml όπου γίνεται η διαβάθμιση των διαφορετικών χρωμάτων που χρησιμοποιεί η εφαρμογή και το αρχείο styles.xml στο οποίο δηλώνονται τα διαφορετικά styles που χρησιμοποιεί κάθε γραφικό στοιχείο ή ολόκληρη η οθόνη.



Εικόνα 4.13. Περιεχόμενα φακέλου values

## *strings.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World</string>
  <string name="app_name">Spiel</string>
  <string name="rules">Welcome to our quiz.Select the right answers and
test your knowledge
  </string>
</resources>
```

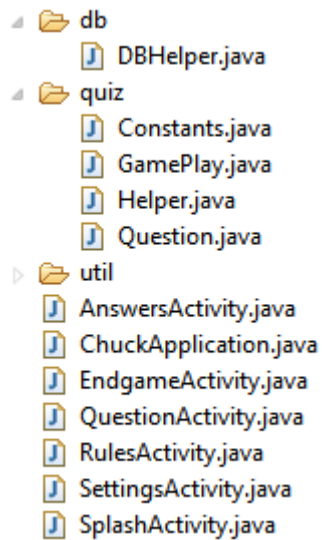
Εικόνα 4.14. Περιεχόμενα υποφακέλου strings.xml φακέλου values

### 4.2.3. Φάκελος bin /

Στον φάκελο bin/ βρίσκονται τα αρχεία που παράγονται μετά τη μεταγλώττιση της εφαρμογής. Εκεί βρίσκεται για παράδειγμα το πακέτο ark της εφαρμογής το οποίο μπορεί να χρησιμοποιηθεί και να διανεμηθεί το παραπάνω πακέτο μέσω της υπηρεσίας Google Play, θα πρέπει πρώτα να υπογραφεί ψηφιακά. Το Android SDK παρέχει ένα εργαλείο που βοηθά τον προγραμματιστή στη , ένα πακέτο μπορεί να υπογραφεί ψηφιακά μέσω , νατας την εξαγωγή της εφαρμογής Eclipse IDE.

#### 4.2.4. Φάκελος src /

Στον φάκελο src (source) περιέχονται τα αρχεία κλάσης τις Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία, κλπ. Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας.



Εικόνα 4.15. Περιεχόμενα φακέλου src /

## 4.2.4.1.Κλάσεις Java

*DBHelper.java*

Το SQLite είναι μια open-source σχεσιακή βάση δεδομένων που χρησιμοποιείται για την εκτέλεση λειτουργιών της βάσης δεδομένων για Android συσκευές, όπως αποθήκευση, χειρισμό ή ανάκτηση από τη βάση δεδομένων. Είναι ενσωματωμένη στο android από προεπιλογή. Έτσι, δεν υπάρχει καμία ανάγκη για να εκτελέσει οποιαδήποτε ρύθμιση της βάσης δεδομένων. Η κλάση android.database.sqlite.SQLiteOpenHelper χρησιμοποιείται για τη δημιουργία της βάσης δεδομένων και τη διαχείριση της έκδοσης.

Υπάρχουν πολλές μέθοδοι στην SQLiteOpenHelper τάξη. Μερικές από αυτές είναι οι ακόλουθες:

Method	Description
<b>public abstract void onCreate(SQLiteDatabase db)</b>	called only once when database is created for the first time.
<b>public abstract void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)</b>	called when database needs to be upgraded.
<b>public synchronized void close ()</b>	closes the database object.
<b>public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)</b>	called when database needs to be downgraded.

```

package com.tmm.android.Spiel.db;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;

import com.tmm.android.Spiel.quiz.Question;

public class DBHelper extends SQLiteOpenHelper{

```

```
//The Android's default system path of your application database.
private static String DB_PATH =
"/data/data/com.tmm.android.chuck/databases/";
private static String DB_NAME = "questionsDb";
private SQLiteDatabase myDataBase;
private final Context myContext;

/**
 * Constructor
 * Takes and keeps a reference of the passed context in order to
access to the application assets and resources.
 * @param context
 */
public DBHelper(Context context) {
    super(context, DB_NAME, null, 1);
    this.myContext = context;
}

/**
 * Creates a empty database on the system and rewrites it with your
own database.
 * */
public void createDataBase() throws IOException{

    boolean dbExist = checkDataBase();
    if(!dbExist)
    {
        //By calling this method and empty database will be
created into the default system path
//of your application so we are gonna be able to
overwrite that database with our database.
        this.getReadableDatabase();

        try {
            copyDataBase();
        } catch (IOException e) {
            throw new Error("Error copying database");
        }
    }
}

/**
 * Check if the database already exist to avoid re-copying the file
each time you open the application.
 * @return true if it exists, false if it doesn't
 */
private boolean checkDataBase(){
    SQLiteDatabase checkDB = null;
    try{
        String myPath = DB_PATH + DB_NAME;
        checkDB = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READONLY);
    }catch(SQLiteException e){
        //database doesn't exist yet.
    }
    if(checkDB != null){
        checkDB.close();
    }
}
```



```

        return checkDB != null ? true : false;
    }

    /**
     * Copies your database from your local assets-folder to the just
    created empty database in the
     * system folder, from where it can be accessed and handled.
     * This is done by transferring byteStream.
     */
    private void copyDataBase() throws IOException{

        //Open your local db as the input stream
        InputStream myInput = myContext.getAssets().open(DB_NAME);

        // Path to the just created empty db
        String outFileName = DB_PATH + DB_NAME;

        //Open the empty db as the output stream
        OutputStream myOutput = new FileOutputStream(outFileName);

        //transfer bytes from the inputfile to the outputfile
        byte[] buffer = new byte[1024];
        int length;
        while ((length = myInput.read(buffer))>0){
            myOutput.write(buffer, 0, length);
        }

        //Close the streams
        myOutput.flush();
        myOutput.close();
        myInput.close();
    }

    public void openDataBase() throws SQLException{
        //Open the database
        String myPath = DB_PATH + DB_NAME;
        myDataBase = SQLiteDatabase.openDatabase(myPath, null,
        SQLiteDatabase.OPEN_READONLY);
    }

    @Override
    public synchronized void close() {
        if(myDataBase != null)
            myDataBase.close();
        super.close();
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
    }

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

    // Add your public helper methods to access and get content from the
    database.

```

```
// You could return cursors by doing "return myDataBase.query(...)"
so it'd be easy
// to you to create adapters for your views.
```

```
public List<Question> getQuestionSet(int difficulty, int numQ){
    List<Question> questionSet = new ArrayList<Question>();
    Cursor c = myDataBase.rawQuery("SELECT * FROM QUESTIONS WHERE
DIFFICULTY=" + difficulty +
        " ORDER BY RANDOM() LIMIT " + numQ, null);
    while (c.moveToNext()){
        //Log.d("QUESTION", "Question Found in DB: " +
c.getString(1));
        Question q = new Question();
        q.setQuestion(c.getString(1));
        q.setAnswer(c.getString(2));
        q.setOption1(c.getString(3));
        q.setOption2(c.getString(4));
        q.setOption3(c.getString(5));
        q.setRating(difficulty);
        questionSet.add(q);
    }
    return questionSet;
}
}
```

## Question.java

```
package com.tmm.android.Spiel.quiz;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Question {

    private String question;
    private String answer;
    private String option1;
    private String option2;
    private String option3;
    private int rating;

    /**
     * @return the question
     */
    public String getQuestion() {
        return question;
    }
    /**
     * @param question the question to set
     */
    public void setQuestion(String question) {
        this.question = question;
    }
    /**
     * @return the answer
     */
    public String getAnswer() {
        return answer;
    }
    /**
     * @param answer the answer to set
     */
    public void setAnswer(String answer) {
        this.answer = answer;
    }
    /**
     * @return the rating
     */
    public int getRating() {
        return rating;
    }
    /**
     * @param rating the rating to set
     */
    public void setRating(int rating) {
        this.rating = rating;
    }
}
```

```
/**
 * @return the option1
 */
public String getOption1() {
    return option1;
}
/**
 * @param option1 the option1 to set
 */
public void setOption1(String option1) {
    this.option1 = option1;
}
/**
 * @return the option2
 */
public String getOption2() {
    return option2;
}
/**
 * @param option2 the option2 to set
 */
public void setOption2(String option2) {
    this.option2 = option2;
}
/**
 * @return the option3
 */
public String getOption3() {
    return option3;
}
/**
 * @param option3 the option3 to set
 */
public void setOption3(String option3) {
    this.option3 = option3;
}

public List<String> getQuestionOptions(){
    List<String> shuffle = new ArrayList<String>();
    shuffle.add(answer);
    shuffle.add(option1);
    shuffle.add(option2);
    shuffle.add(option3);
    Collections.shuffle(shuffle);
    return shuffle;
}
}
```

## Helper.java

```
package com.tmm.android.Spiel.quiz;

import com.tmm.android.Spiel.R;

public class Helper {

    /**
     * This method selects a end game response based on the players score
     * and current difficulty level
     *
     * @param numCorrect - num correct answers
     * @param numRounds - number of questions
     * @param diff - the difficulty level
     * @return String comment
     */
    public static String getResultComment(int numCorrect, int numRounds,
int diff)
    {
        String comm="";
        int percentage = calculatePercentage(numCorrect, numRounds);
        switch (diff)
        {
            case Constants.EXTREME :
                if (percentage > 90){
                    comm = "Bravo ...";
                }else if (percentage >= 80){
                    comm="Nicely Done..";
                }else if (percentage >= 60){
                    comm="Not bad..";
                }else if (percentage >= 40){
                    comm="try again..";
                }else{
                    comm="you lost..";
                }
                break;

            default:
                if (percentage > 90){
                    comm = "Awesome!";
                }else if (percentage >= 80){
                    comm="Nicely Done!";
                }else if (percentage >= 60){
                    comm="Not bad..";
                }else if (percentage >= 40){
                    comm="try again..";
                }else{
                    comm="you lost..";
                }
        }
    }
}
```

```
        return comm;
    }

/**
 * Method to return an image to use for the end of game screen
 *
 * @param numCorrect - number of correct answers
 * @param numRounds - number of rounds
 * @param diff - difficulty level
 * @return int Image ID
 */
public static int getResultImage(int numCorrect, int numRounds, int
diff){
    //calculate percentage
    int percentage = calculatePercentage(numCorrect, numRounds);

    //work out which image
    if (percentage > 90){
        if (diff == Constants.EXTREME){
            return R.drawable.hard_winner;
        }else{
            return R.drawable.easy_winner;
        }
    }else if (percentage >= 80){
        return R.drawable.prettygood;
    }else if (percentage >= 60){
        return R.drawable.notbad;
    }else if (percentage >= 40){
        return R.drawable.tryagain;
    }else{
        return R.drawable.Loser;
    }
}

/**
 * Calculate the percentage result based on the number correct and
number of questions
 *
 * @param numCorrect - number of questions right
 * @param numRounds - total number of questions
 * @return int percentage correct
 */
private static int calculatePercentage(int numCorrect, int numRounds)
{
    double frac = (double)numCorrect/((double)numRounds);
    int percentage = (int) (frac*100);
    return percentage;
}
}
```

## GamePlay.java

```
package com.tmm.android.Spiel.quiz;

import java.util.ArrayList;
import java.util.List;

public class GamePlay {

    private int numRounds;
    private int difficulty;
    private String playerName;
    private int right;
    private int wrong;
    private int round;

    private List<Question> questions = new ArrayList<Question>();

    /**
     * @return the playerName
     */
    public String getPlayerName() {
        return playerName;
    }

    /**
     * @param playerName the playerName to set
     */
    public void setPlayerName(String playerName) {
        this.playerName = playerName;
    }

    /**
     * @return the right
     */
    public int getRight() {
        return right;
    }

    /**
     * @param right the right to set
     */
    public void setRight(int right) {
        this.right = right;
    }

    /**
     * @return the wrong
     */
    public int getWrong() {
        return wrong;
    }

    /**
     * @param wrong the wrong to set
     */
    public void setWrong(int wrong) {
        this.wrong = wrong;
    }
}
```

```
    * @return the round
    */
    public int getRound() {
        return round;
    }
    /**
     * @param round the round to set
     */
    public void setRound(int round) {
        this.round = round;
    }
    /**
     * @param difficulty the difficulty to set
     */
    public void setDifficulty(int difficulty) {
        this.difficulty = difficulty;
    }
    /**
     * @return the difficulty
     */
    public int getDifficulty() {
        return difficulty;
    }
    /**
     * @param questions the questions to set
     */
    public void setQuestions(List<Question> questions) {
        this.questions = questions;
    }

    /**
     * @param q the question to add
     */
    public void addQuestions(Question q) {
        this.questions.add(q);
    }

    /**
     * @return the questions
     */
    public List<Question> getQuestions() {
        return questions;
    }

    public Question getNextQuestion(){

        //get the question
        Question next = questions.get(this.getRound());
        //update the round number to the next round
        this.setRound(this.getRound()+1);
        return next;
    }

    /**
     * method to increment the number of correct answers this game
     */
    public void incrementRightAnswers(){
        right ++;
    }
}
```



```
    }

    /**
     * method to increment the number of incorrect answers this game
     */
    public void incrementWrongAnswers(){
        wrong ++;
    }
    /**
     * @param numRounds the numRounds to set
     */
    public void setNumRounds(int numRounds) {
        this.numRounds = numRounds;
    }
    /**
     * @return the numRounds
     */
    public int getNumRounds() {
        return numRounds;
    }

    /**
     * method that checks if the game is over
     * @return boolean
     */
    public boolean isGameOver(){
        return (getRound() >= getNumRounds());
    }
}

}
```

## *Constants.java*

```
package com.tmm.android.Spiel.quiz;

import android.view.Menu;

public class Constants {

    public static final int EASY = 1;
    public static final int MEDIUM = 2;
    public static final int HARD = 3;
    public static final int EXTREME = 3;

    public static final int PLAYBUTTON = Menu.FIRST;
    public static final int RULESBUTTON = Menu.FIRST + 1;
    public static final int SETTINGSBUTTON = Menu.FIRST + 2;

    public static final String SETTINGS = "SETTINGS";
    public static final String DIFFICULTY = "DIFFICULTY";
    public static final String NUM_ROUNDS = "NUM_ROUNDS";

}
```

## Utility.java

```
package com.tmm.android.Spiel.util;

import java.util.List;

import com.tmm.android.Spiel.quiz.Question;

public class Utility {

    /**
     * Method to capitalise the first letter of a given string
     *
     * @param s
     * @return
     */
    public static String capitalise(String s){
        if (s==null || s.length()==0) return s;

        String s1 = s.substring(0, 1).toUpperCase() + s.substring(1);
        return s1;
    }

    /**
     * Method to get set of answers for a list of questions
     * @param questions
     * @return
     */
    public static String getAnswers(List<Question> questions) {
        int question = 1;
        StringBuffer sb = new StringBuffer();

        for (Question q : questions){
            sb.append("Q").append(question).append("
").append(q.getQuestion()).append("? \n");
            sb.append("Answer:
").append(q.getAnswer()).append("\n\n");
            question ++;
        }

        return sb.toString();
    }
}

package com.tmm.android.Spiel;

import java.io.IOException;
import java.util.List;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.SQLException;
import android.os.Bundle;
```

```

import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

import com.tmm.android.Spiel.db.DBHelper;
import com.tmm.android.Spiel.quiz.Constants;
import com.tmm.android.Spiel.quiz.GamePlay;
import com.tmm.android.Spiel.quiz.Question;

public class SplashActivity extends Activity implements OnClickListener{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.welcome);

        ////////////////////////////////////////////////////
        ////////////// GAME MENU
        ////////////////////////////////////////////////////
        Button playBtn = (Button) findViewById(R.id.playBtn);
        playBtn.setOnClickListener(this);
        Button settingsBtn = (Button) findViewById(R.id.settingsBtn);
        settingsBtn.setOnClickListener(this);
        Button rulesBtn = (Button) findViewById(R.id.rulesBtn);
        rulesBtn.setOnClickListener(this);
        Button exitBtn = (Button) findViewById(R.id.exitBtn);
        exitBtn.setOnClickListener(this);
    }

    /**
     * Listener for game menu
     */
    @Override
    public void onClick(View v) {
        Intent i;

        switch (v.getId()){
            case R.id.playBtn :
                //once logged in, load the main page
                //Log.d("LOGIN", "User has started the game");

                //Get Question set //
                List<Question> questions = getQuestionSetFromDb();

                //Initialise Game with retrieved question set ///
                GamePlay c = new GamePlay();
                c.setQuestions(questions);
                c.setNumRounds(getNumQuestions());
                ((ChuckApplication)getApplication()).setCurrentGame(c);

                //Start Game Now.. //
                i = new Intent(this, QuestionActivity.class);
                startActivityForResult(i, Constants.PLAYBUTTON);
                break;

            case R.id.rulesBtn :

```

```

        i = new Intent(this, RulesActivity.class);
        startActivityForResult(i, Constants.RULESBUTTON);
        break;

    case R.id.settingsBtn :
        i = new Intent(this, SettingsActivity.class);
        startActivityForResult(i, Constants.SETTINGSBUTTON);
        break;

    case R.id.exitBtn :
        finish();
        break;
    }
}

/**
 * Method that retrieves a random set of questions from
 * the database for the given difficulty
 * @return
 * @throws Error
 */
private List<Question> getQuestionSetFromDb() throws Error {
    int diff = getDifficultySettings();
    int numQuestions = getNumQuestions();
    DBHelper myDbHelper = new DBHelper(this);
    try {
        myDbHelper.createDataBase();
    } catch (IOException ioe) {
        throw new Error("Unable to create database");
    }
    try {
        myDbHelper.openDataBase();
    } catch (SQLException sqle){
        throw sqle;
    }
    List<Question> questions = myDbHelper.getQuestionSet(diff,
numQuestions);
    myDbHelper.close();
    return questions;
}

/**
 * Method to return the difficulty settings
 * @return
 */
private int getDifficultySettings() {
    SharedPreferences settings =
getSharedPreferences(Constants.SETTINGS, 0);
    int diff = settings.getInt(Constants.DIFFICULTY,
Constants.MEDIUM);
    return diff;
}

/**
 * Method to return the number of questions for the game
 * @return

```

```
        */
        private int getNumQuestions() {
            SharedPreferences settings =
getSharedPreferences(Constants.SETTINGS, 0);
            int numRounds = settings.getInt(Constants.NUM_ROUNDS, 20);
            return numRounds;
        }
    }
}
```

## *ChuckApplication.java*

```
package com.tmm.android.Spiel;

import com.tmm.android.Spiel.quiz.GamePlay;
import android.app.Application;

public class ChuckApplication extends Application{
    private GamePlay currentGame;

    /**
     * @param currentGame the currentGame to set
     */
    public void setCurrentGame(GamePlay currentGame) {
        this.currentGame = currentGame;
    }

    /**
     * @return the currentGame
     */
    public GamePlay getCurrentGame() {
        return currentGame;
    }
}
```

## *QuestionActivity.java*

```
package com.tmm.android.Spiel;

import java.util.List;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
```

```
import android.widget.TextView;

import com.tmm.android.Spiel.quiz.GamePlay;
import com.tmm.android.Spiel.quiz.Question;
import com.tmm.android.Spiel.util.Utility;

public class QuestionActivity extends Activity implements OnClickListener{

    private Question currentQ;
    private GamePlay currentGame;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.question);
        /**
         * Configure current game and get question
         */
        currentGame = ((ChuckApplication)getApplication()).getCurrentGame();
        currentQ = currentGame.getNextQuestion();
        Button nextBtn = (Button) findViewById(R.id.nextBtn);
        nextBtn.setOnClickListener(this);

        /**
         * Update the question and answer options..
         */
        setQuestions();
    }

    /**
     * Method to set the text for the question and answers from the
     current games
     * current question
     */
    private void setQuestions() {
        //set the question text from current question
        String question = Utility.capitalize(currentQ.getQuestion()) +
"?";

        TextView qText = (TextView) findViewById(R.id.question);
        qText.setText(question);

        //set the available options
        List<String> answers = currentQ.getQuestionOptions();
        TextView option1 = (TextView) findViewById(R.id.answer1);
        option1.setText(Utility.capitalize(answers.get(0)));

        TextView option2 = (TextView) findViewById(R.id.answer2);
        option2.setText(Utility.capitalize(answers.get(1)));

        TextView option3 = (TextView) findViewById(R.id.answer3);
        option3.setText(Utility.capitalize(answers.get(2)));

        TextView option4 = (TextView) findViewById(R.id.answer4);
        option4.setText(Utility.capitalize(answers.get(3)));
    }
}
```

```

@Override
public void onClick(View arg0) {
    //Log.d("Questions", "Moving to next question");

    /**
     * validate a checkbox has been selected
     */
    if (!checkAnswer()) return;

    /**
     * check if end of game
     */
    if (currentGame.isGameOver()){
        //Log.d("Questions", "End of game! lets add up the
scores..");
        //Log.d("Questions", "Questions Correct: " +
currentGame.getRight());
        //Log.d("Questions", "Questions Wrong: " +
currentGame.getWrong());
        Intent i = new Intent(this, EndgameActivity.class);
        startActivity(i);
        finish();
    }
    else{
        Intent i = new Intent(this, QuestionActivity.class);
        startActivity(i);
        finish();
    }
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    switch (keyCode)
    {
        case KeyEvent.KEYCODE_BACK :
            return true;
    }

    return super.onKeyDown(keyCode, event);
}

/**
 * Check if a checkbox has been selected, and if it
 * has then check if its correct and update gamescore
 */
private boolean checkAnswer() {
    String answer = getSelectedAnswer();
    if (answer==null){
        //Log.d("Questions", "No Checkbox selection made -
returning");
        return false;
    }
    else {

```

```

        //Log.d("Questions", "Valid Checkbox selection made -
check if correct");
        if (currentQ.getAnswer().equalsIgnoreCase(answer))
        {
            //Log.d("Questions", "Correct Answer!");
            currentGame.incrementRightAnswers();
        }
        else{
            //Log.d("Questions", "Incorrect Answer!");
            currentGame.incrementWrongAnswers();
        }
        return true;
    }
}

/**
 *
 */
private String getSelectedAnswer() {
    RadioButton c1 = (RadioButton)findViewById(R.id.answer1);
    RadioButton c2 = (RadioButton)findViewById(R.id.answer2);
    RadioButton c3 = (RadioButton)findViewById(R.id.answer3);
    RadioButton c4 = (RadioButton)findViewById(R.id.answer4);
    if (c1.isChecked())
    {
        return c1.getText().toString();
    }
    if (c2.isChecked())
    {
        return c2.getText().toString();
    }
    if (c3.isChecked())
    {
        return c3.getText().toString();
    }
    if (c4.isChecked())
    {
        return c4.getText().toString();
    }

    return null;
}
}

```

## *AnswersActivty.java*

```

package com.tmm.android.Spiel;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

```



```
import android.widget.TextView;

import com.tmm.android.Spiel.quiz.GamePlay;
import com.tmm.android.Spiel.util.Utility;

public class AnswersActivity extends Activity implements OnClickListener {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.answers);
        GamePlay currentGame =
        ((ChuckApplication)getApplication()).getCurrentGame();

        TextView results = (TextView)findViewById(R.id.answers);
        String answers = Utility.getAnswers(currentGame.getQuestions());
        results.setText(answers);

        //handle button actions
        Button finishBtn = (Button) findViewById(R.id.finishBtn);
        finishBtn.setOnClickListener(this);
    }

    /** (non-Javadoc)
     * @see android.app.Activity#onKeyDown(int, android.view.KeyEvent)
     *
     * This method is to override the back button on the phone
     * to prevent users from navigating back in to the quiz
     */
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        switch (keyCode)
        {
            case KeyEvent.KEYCODE_BACK :
                return true;
        }

        return super.onKeyDown(keyCode, event);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.finishBtn :
                finish();
        }
    }
}
```

## EndgameActivity.java

```
package com.tmm.android.Spiel;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.tmm.android.Spiel.quiz.Constants;
import com.tmm.android.Spiel.quiz.GamePlay;
import com.tmm.android.Spiel.quiz.Helper;

public class EndgameActivity extends Activity implements OnClickListener {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.endgame);
        GamePlay currentGame =
        ((ChuckApplication)getApplication()).getCurrentGame();
        String result = "You Got " + currentGame.getRight() + "/" +
        currentGame.getNumRounds() + ".. ";
        String comment = Helper.getResultComment(currentGame.getRight(),
        currentGame.getNumRounds(), getDifficultySettings());

        TextView results = (TextView)findViewById(R.id.endgameResult);
        results.setText(result + comment);

        int image = Helper.getResultImage(currentGame.getRight(),
        currentGame.getNumRounds(), getDifficultySettings());
        ImageView resultImage =
        (ImageView)findViewById(R.id.resultPage);
        resultImage.setImageResource(image);

        //handle button actions
        Button finishBtn = (Button) findViewById(R.id.finishBtn);
        finishBtn.setOnClickListener(this);
        Button answerBtn = (Button) findViewById(R.id.answerBtn);
        answerBtn.setOnClickListener(this);
    }

    /**
     * Method to return the difficulty settings
     */
}
```



## *RulesActivity.java*

```
package com.tmm.android.Spiel;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class RulesActivity extends Activity implements OnClickListener{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.rules);

        //finish button
        Button backBtn = (Button) findViewById(R.id.backBtn);
        backBtn.setOnClickListener(this);
    }

    @Override
    public void onClick(View arg0) {
        /**
         * if the back button is clicked then go back to the main menu
         */
        finish();
    }
}
```

## SettingsActivity.java

```
package com.tmm.android.Spiel;

import android.app.Activity;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;

import com.tmm.android.Spiel.quiz.Constants;

public class SettingsActivity extends Activity implements OnClickListener{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.settings);

        /**
         * set listener on update button
         */
        Button updateBtn = (Button) findViewById(R.id.nextBtn);
        updateBtn.setOnClickListener(this);

        /**
         * Set selected button if saved
         */
        updateButtonWithPreferences();
    }

    /**
     * Method to update default check box
     */
    private void updateButtonWithPreferences() {
        RadioButton c1 = (RadioButton)findViewById(R.id.easySetting);
        RadioButton c2 = (RadioButton)findViewById(R.id.mediumSetting);
        RadioButton c3 = (RadioButton)findViewById(R.id.hardSetting);

        SharedPreferences settings =
        getSharedPreferences(Constants.SETTINGS, 0);
        int diff = settings.getInt(Constants.DIFFICULTY,
        Constants.MEDIUM);

        switch (diff)
        {
            case Constants.EASY :
                c1.toggle();
                break;

            case Constants.MEDIUM :
```

```
                c2.toggle();
                break;

            case Constants.EXTREME :
                c3.toggle();
                break;
        }
    }

@Override
public void onClick(View arg0) {
    /**
     * check which settings set and return to menu
     */
    if (!checkSelected())
    {
        return;
    }
    else
    {
        SharedPreferences settings =
getSharedPreferences(Constants.SETTINGS, 0);
        Editor e = settings.edit();
        e.putInt(Constants.DIFFICULTY, getSelectedSetting());
        e.commit();
        finish();
    }
}

/**
 * Method to check that a checkbox is selected
 *
 * @return boolean
 */
private boolean checkSelected() {
    RadioButton c1 = (RadioButton)findViewById(R.id.easySetting);
    RadioButton c2 = (RadioButton)findViewById(R.id.mediumSetting);
    RadioButton c3 = (RadioButton)findViewById(R.id.hardSetting);
    return (c1.isChecked() || c2.isChecked() || c3.isChecked());
}

/**
 * Get the selected setting
 */
private int getSelectedSetting() {
    RadioButton c1 = (RadioButton)findViewById(R.id.easySetting);
    RadioButton c2 = (RadioButton)findViewById(R.id.mediumSetting);
    if (c1.isChecked())
    {
        return Constants.EASY;
    }
    if (c2.isChecked())
```

```
        {  
            return Constants.MEDIUM;  
        }  
        return Constants.EXTREME;  
    }  
}
```

**Πριν ακόμα χρειαστεί να γράψουμε εμείς κάποιο κώδικα, το eclipse έχει δημιουργήσει το απαραίτητο hello world! Για να ρίξουμε μια ματιά στον κώδικα και στο layout xml.**

Όπως βλέπουμε το eclipse έχει κάνει import τις απαραίτητες κλάσεις (και κάποιες που δεν θα τις χρειαστούμε) και έχει δημιουργήσει μια νέα activity. Η μέθοδος onCreate βρίσκεται σε κάθε κλάση που κάνει extend την κλάση Activity και είναι το πρώτο πράγμα που εκτελείται μόλις τρέξει η εφαρμογή. Στον συγκεκριμένο κώδικα, η μέθοδος αυτή δεν κάνει τίποτα άλλο από το να φορτώσει το layout αρχείο που θα μελετήσουμε παρακάτω.

```
package app.android.greektuts;  
import android.os.Bundle;  
import android.app.Activity;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.support.v4.app.NavUtils;  
public class MainActivity extends Activity {  
    //Δημιουργία μιας νέας activity  
    @Override public  
    void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

**Το αρχείο activity\_main.xml (και όλα τα άλλα αρχεία στον φάκελο /res/layout) είναι υπεύθυνο για τον σχεδιασμό της εφαρμογής μας. Για να ρίξουμε μια ματιά στα περιεχόμενα του.**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent" >

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:padding="@dimen/padding_medium"
android:text="@string/hello_world"
tools:context=".MainActivity" />

</RelativeLayout>
```

Όπως βλέπουμε, το eclipse έχει δημιουργήσει ένα Relative Layout με ένα text view.



#### 4.2.5. Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομα του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Δημιουργείται αυτόματα όταν ξεκινάμε ένα καινούργιο project για μια εφαρμογή Android. Μεταξύ άλλων, το AndroidManifest κάνει τα ακόλουθα :

- Ονομάζει το Java πακέτο της εφαρμογής. Το όνομα αυτό λειτουργεί ως μοναδικό αναγνωριστικό ID για την εφαρμογή
- Ονομάζει τις κλάσεις που εφαρμόζουν καθεμία από τις συνιστώσες και δημοσιεύει τις ικανότητές τους
- Δηλώνει ποια δικαιώματα η εφαρμογή πρέπει να έχει προκειμένου να γίνει δυνατή η πρόσβαση σε προστατευμένες περιοχές του API και να μπορεί να αλληλεπιδρά με άλλες εφαρμογές
- Ο αριθμός έκδοσης της εφαρμογής Δηλώνει τα δικαιώματα που οι άλλοι οφείλουν να έχουν προκειμένου να αλληλεπιδρούν με στοιχεία της εφαρμογής
- Δηλώνει το ελάχιστο επίπεδο του Android API που απαιτεί η εφαρμογή
- Περιγράφει τις συνιστώσες της εφαρμογής , όπως οι δραστηριότητες (activities), πάροχοι περιεχομένου (content providers), υπηρεσίες (services) κλπ,

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

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tmm.android.chuck"
    android:versionCode="2"
    android:versionName="2.0">

    <application android:icon="@drawable/chuck_icon_bg"
        android:label="@string/app_name"
        android:name=".ChuckApplication" >

        <activity android:name=".SplashActivity"
            android:label="@string/app_name">

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".QuestionActivity" />
        <activity android:name=".RulesActivity" />
        <activity android:name=".EndgameActivity" />
        <activity android:name=".SettingsActivity" />
        <activity android:name=".AnswersActivity" />
    </application>
    <uses-sdk android:minSdkVersion="2" />
</manifest>
```

Εικόνα 4.16. .Περιεχόμενα φακέλου AndroidManifest.xml

Στοιχεία του αρχείου AndroidManifest.xml :

Τα στοιχεία που χρησιμοποιούνται στον ανωτέρω αρχείο XML περιγράφονται παρακάτω.

**<manifest>** είναι το root στοιχείο του αρχείου AndroidManifest.xml

**<application>** είναι το υποστοιχείο του manifest. Περιλαμβάνει τη δήλωση namespace. Το στοιχείο αυτό περιέχει αρκετά υποστοιχεία που δηλώνουν τα στοιχεία της εφαρμογής, όπως η δραστηριότητα κ.λπ. Τα χαρακτηριστικά αυτού του στοιχείου που χρησιμοποιούνται συνήθως είναι το εικονίδιο, η ετικέτα, το θέμα κλπ

android:icon , αντιπροσωπεύει το εικονίδιο για όλα τα στοιχεία της android εφαρμογής.

android:label , λειτουργεί ως προεπιλεγμένη ετικέτα για όλα τα στοιχεία της εφαρμογής.

**<activity>** είναι η υποστοιχείο της αίτησης και αποτελεί μια δραστηριότητα που πρέπει να οριστεί στο αρχείο AndroidManifest.xml. Έχει πολλά χαρακτηριστικά, όπως η ετικέτα, το όνομα, το θέμα, launchMode κ.λπ.

android:label , αντιπροσωπεύει δηλαδή ετικέτα που εμφανίζεται στην οθόνη.

android:name , αντιπροσωπεύει ένα όνομα για την κατηγορία δραστηριότητας. Είναι απαραίτητο χαρακτηριστικό.

**<intent-filter>** είναι το υπο-στοιχείο της δραστηριότητας που περιγράφει τον τύπο των intent στην οποία δραστηριότητα, υπηρεσία ή μετάδοση μπορεί να απαντήσει.

**<action>** Προσθέτει μια δράση για το intent-filter ,το οποίο πρέπει να έχει τουλάχιστον ένα στοιχείο δράσης.

**<category>** Προσθέτει ένα όνομα κατηγορίας στο intent-filter.

**4.2.6. Το αρχείο project.properties**, το οποίο δημιουργείται αυτόματα όταν ξεκινάμε ένα καινούργιο project για μια εφαρμογή Android, αφορά το σύστημα και δεν πρέπει να το επεξεργαζόμαστε

#### 4.2.7. Φάκελος libs /

Στον κατάλογο libs/ τοποθετούνται οι εξωτερικές βιβλιοθήκες που μπορεί να

επιπλέον χαρακτηριστικά σε μια εφαρμογή. Αυτές οι βιβλιοθήκες πρέπει να δηλωθούν ως "linked libraries" στο Eclipse IDE ώστε να γίνουν link

Android χρησιμοποιεί τη γλώσσα προγραμματισμού Java, έτσι και οι εγγενείς βιβλιοθήκες που υποστηρίζει είναι οι βιβλιοθήκες της Java .jar

χρησιμοποιεί. Στην εφαρμογή μας χρησιμοποιήθηκε η Android Support Library για τη χρήση χαρακτηριστικών

## ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ

### 5.1. Συμπεράσματα

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με τη δημιουργία μιας εφαρμογής-παιχνίδι για το λειτουργικό σύστημα Android. Το Android είναι ακόμη σε αρχικό στάδιο αλλά φαίνεται πως θα έχει μεγάλη εξέλιξη στα επόμενα χρόνια. Οι εφαρμογές που υπάρχουν στο play store ,στο χώρο από τον οποίο δίνεται η δυνατότητα να κατεβάσει,όποιος ενδιαφέρεται ,εφαρμογές,αφορούν τη διασκέδαση αλλά και τα παιχνίδια γνώσεων. Το θετικό είναι ότι όλο και περισσότεροι χρήστες κατεβάζουν εφαρμογές. Όμως,επειδή αφορά πλατφόρμα ανοιχτού κώδικα,οι περισσότεροι χρήστες δεν αγοράζουν εφαρμογές που έχουν κόστος.

στην ανάπτυξη εφαρμογών. Δημιουργήσαμε αρχεία Java και αντίστοιχα xml για να πάρουμε το συγκεκριμένο αποτέλεσμα.

### 5.2. Μελλοντικές εξελίξεις

Με τη προσθήκη κάποιων επιπλέον χαρακτηριστικών στην εφαρμογή,θα μπορούσε εύκολα να εξελιχθεί σε ένα πιο ολοκληρωμένο παιχνίδι γνώσεων,όπως :

- ✓ Δημιουργία καλύτερου interface παιχνιδιού, προσθέτοντας 3D γραφικά ή κανοντας χρήση της τεχνολογίας του accelerometer των κινητών συσκευών
- ✓ Κατηγοριοποίηση των ερωτήσεων ανάλογα με το είδος (πολλαπλής επιλογής,συμπλήρωσης κενών,σταυρόλεξο,συνώνυμα-αντώνυμα)
- ✓ Μετάφραση του περιβάλλοντος διεπαφής της εφαρμογής σε πολλές γλώσσες ώστε να αποκτήσουν περισσότεροι χρήστες το κίνητρο να τη χρησιμοποιήσουν

**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [ 1 ] Android SDK Guide-Google
- [ 2 ] J.Friesen,2010, “Learn Java for Android Development” ,Apress
- [ 3 ] M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition”, CommonsWare
- [ 4 ] J. Friesen, 2010, “Learn Java for Android Development”, Apress
- [ 5 ] Nearchos Paspallis, Middleware-based development of context-aware applications with reusable components, Ph.D. Thesis, University of Cyprus, September 2009
- [ 6 ] J. Morris, 2011, “Android User Interface Development Beginner's Guide”, Pakt Publishing
- [ 7 ] A. Bar-Noy and I. Kessler, “Tracking Mobile Users in Wireless Networks”, IEEE Trans. Info.Theory, vol. 39, 1993, pp. 1877--1886
- [ 8 ] Nisarg Gandhewar,Rahila Sheikh,Google Android : An Emerging Software Platform For Mobile Devices,International Journal on Computer Science and Engineering (IJCSSE),2011
- [ 9 ] Darren Carlson, Andreas Schrader, A wide-area context-awareness approach for Android, Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, pp. 383-386
- [ 10 ] W.M. Lee, 2011, “Beginning Android Application Development”, Wrox
- [ 11 ] Mark Murphy,Beginning Android 3,Apress
- [ 12 ] J. Steele, 2010, “The Android Developer's Cookbook”, Addison & Wesley
- [ 13 ] Shafik G.Punja & Richard P.Mislan,Mobile Device Analysis,June 2008
- [ 14 ] R. Meier, 2010, “Professional Android 2 Application Development”, Wrox
- [ 15 ] Ed Burnette. “Hello Android Introducing Google’s Mobile Development Platform
- [ 16 ] C. Hasenan, 2008, “Android Essentials”, Firstpress
- [ 17 ] Laurent Darcey and Shane Conder “Ανάπτυξη εφαρμογών σε Android” (Second Edition)
- [ 18 ] Steven J. Vaughan-Nichols, August 18, 2011, “Linus Torvalds on Android, the Linux fork”, Zdnet.com
- [ 19 ] Balakrishnan, V. K. (1997-02-01). Graph Theory (1st ed.). McGraw-Hill. ISBN 0-07-005489-4
- [ 20 ] J. Steele, 2010, “The Android Developer's Cookbook”, Addison & Wesley
- [ 21 ] Hello, Android, Introducing Google’s Mobile Development Platform 3rd Edition – Ed Burnette, The pragmatic programmers
- [ 22 ] Professional Android 2 Application Development – Reto Meier, Wrox
- [ 23 ] Beginning Android 2 – Mark L. Murphy, APress
- [ 24 ] Μελέτη πλατφόρμας Android και υλοποίηση ενός multiplayer παινιδιού – Τσώτας Σωτήριος
- [ 25 ] Το λειτουργικό σύστημα Android και η υλοποίηση εφαρμογής με το Google SDK –Στυλιανός Βουκάτας

## ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- [ 1 ] Android OS.  
<http://developer.android.com/index.html>
- [ 2 ] Ανάλυση της αρχιτεκτονικής του Android  
<http://developer.android.com/guide/basics/what-is-android.html>
- [ 3 ] Οδηγίες εγκατάστασης του Android SDK  
<http://developer.android.com/sdk/installing.html>
- [ 4 ] Οδηγίες εγκατάστασης του ADT plugin  
<http://developer.android.com/sdk/eclipse-adt.html#installing>
- [ 5 ] Η επίσημη ιστοσελίδα του Eclipse IDE, το οποίο χρησιμοποιήθηκε κατά την υλοποίηση του project και είναι και προτεινόμενο από την Google.  
<http://www.eclipse.org/>
- [ 6 ] Ο καταμερισμός των εκδόσεων του Android όπως αυτά καταγράφονται από την πρόσβαση των συσκευών στο Google Play Store.  
<http://developer.android.com/resources/dashboard/platform-versions.html>
- [ 7 ] Ανάλυση των βασικών συστατικών μιας εφαρμογής του Android, και ο συσχετισμός τους με την αρχιτεκτονική του συστήματος  
<http://developer.android.com/guide/basics/what-is-android.html>
- [ 8 ] Πληροφορίες σχετικά με τη δομή και τη χρήση του του αρχείου AndroidManifest.xml  
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [ 9 ] Πληροφορίες για την Google  
<http://www.google.com/intl/en/corporate/>
- [ 10 ] Πληροφορίες για την Open Handset Alliance  
<http://www.openhandsetalliance.com/>
- [ 11 ] Το logo της Android  
[http://en.wikipedia.org/wiki/File:Android\\_logo.svg](http://en.wikipedia.org/wiki/File:Android_logo.svg)
- [ 12 ] Πληροφορίες για την αρχιτεκτονική και τα χαρακτηριστικά του Android  
<http://developer.android.com/guide/basics/what-is-android.html>
- [ 13 ] Η ανάπτυξη του παραδείγματος της Google για την Android  
<http://developer.android.com/resources/tutorials/hello-world.html>
- [ 14 ] Περισσότερες πληροφορίες για το AndroidManifest.xml και τα άλλα αρχεία xml  
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [ 15 ] Αναλυτική περιγραφή των τύπων και των χαρακτηριστικών των Layouts  
<http://developer.android.com/guide/topics/ui/layout-objects.html>
- [ 16 ] Αναλυτική περιγραφή των μεθόδων και του κύκλου ζωής του Activity  
<http://developer.android.com/reference/android/app/Activity.html>
- [ 17 ] Android Tutorial  
<http://www.javacodegeeks.com/page/2/?s=Android+Tutorial>
- [ 18 ] Wikipedia - Android software development,  
[http://en.wikipedia.org/wiki/Android\\_software\\_development](http://en.wikipedia.org/wiki/Android_software_development)
- [ 19 ] Wikipedia-JavaDevelopmentKit(JDK)  
[http://en.wikipedia.org/wiki/Java\\_Development\\_Kit](http://en.wikipedia.org/wiki/Java_Development_Kit)
- [ 20 ] Wikipedia - Google Play,  
[http://en.wikipedia.org/wiki/Google\\_Play](http://en.wikipedia.org/wiki/Google_Play)

