

ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ (ΜΕΣΟΛΟΓΓΙ)

**ΠΤΥΧΙΑΚΗ
ΕΡΓΑΣΙΑ**

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ
ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ ΣΕ
ΛΕΙΤΟΥΡΓΙΚΟ ANDROID (QUIZ ΓΝΩΣΕΩΝ)**



**ΕΛΕΝΗ ΓΚΟΡΟΓΙΑ
Α.Ε.Μ.: 14693**

ΜΕΣΟΛΟΓΓΙ, 2014

ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ / ΜΕΣΟΛΟΓΓΙ

Πτυχιακή εργασία

ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ
ΠΑΙΧΝΙΔΙΟΥ ΣΕ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID
(QUIZ ΓΝΩΣΕΩΝ)

Ελένη Γκορόγια
ΑΕΜ: 14693

Επιβλέπων Καθηγητής: Ντόκας Ιωάννης

Μεσολόγγι, 2014

Μεσολόγιο 2014

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Διοίκησης Επιχειρήσεων/Μεσολογίου του ΤΕΙ Δυτικής Ελλάδας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Περιεχόμενα

Εισαγωγή	7
ΚΕΦΑΛΑΙΟ 1	9
Γνωριμία με το Android OS	9
1.1. Τι είναι android;	9
1.2. Χαρακτηριστικά που ενισχύουν τη χρήση του Android.....	9
1.3. Τι είναι το google play;	10
1.4. Πόσο διαδεδομένο είναι το Android OS	11
ΚΕΦΑΛΑΙΟ 2	13
Ιστορική αναδρομή	13
2.1. Gingerbread (Android 2.3.3 APIs, Android 2.3.4 APIs).....	14
2.2. Honeycomb (Android 3.0 APIs, Android 3.1 APIs, Android 3.2 APIs)	14
2.3. Ice Cream Sandwich (Android 4.0 APIs, Android 4.0.3 APIs).....	15
2.4. Jelly Bean (Android 4.1 APIs, Android 4.2 APIs, Android 4.3 APIs).....	15
2.5. Android 4.4 KitKat	16
ΚΕΦΑΛΑΙΟ 3	17
Εγκατάσταση απαιτούμενων προγραμμάτων για την ανάπτυξη μίας Android Εφαρμογής	17
3.1. Τι χρειαζόμαστε για την ανάπτυξη μίας Android Εφαρμογής.....	17
3.2. Εγκατάσταση του Eclipse	17
3.3. Εγκατάσταση του Java SE Development Kit 7 Downloads (για Windows 32bit)	19
3.4. Εγκατάσταση του Android Development Tools plugin	23
3.5. Οδηγός χρήσης Eclipse	28
3.6. Κατεβάζοντας το Eclipse Plugin.....	29
3.7. Δουλεύοντας με τον SDK Manager	34
ΚΕΦΑΛΑΙΟ 4	37
Γνωριμία με τα ξεχωριστά στοιχεία προγραμματισμού σε Android και δημιουργία μιας νέας εφαρμογής.....	37
4.1. Αρχιτεκτονική των εφαρμογών Android	37
4.1.1. Activity (δραστηριότητα).....	37
4.1.2. Service (Υπηρεσία)	37
4.1.3. Content Provider (Πάροχος περιεχομένου).....	37
4.1.4. Broadcast Receiver (Δέκτης εκπομπών).....	38
4.1.5. Intent (Πρόθεση)	38
4.2. Καθορισμός χαρακτηριστικών της Android εφαρμογής.....	38

4.3.	Δημιουργία του Android προγράμματος με τη βοήθεια του Eclipse	39
4.3.1.	Προσθήκη πόρων συμβολοσειράς (string)	44
4.3.1.	Προσθήκη γραφικών πόρων (drawable).....	44
4.3.2.	Δημιουργία των layouts (σχεδιαγράμματα) του προγράμματος	44
4.3.3.	Δημιουργία των απαραίτητων Activities	46
4.4.	Ενημέρωση του αρχείου Android Manifest του Android	49
4.5.	Δημιουργία της εικονικής μηχανής.....	51
4.5.1.	Επεξεργασία των ρυθμίσεων της φόρτωσης της εφαρμογής στην εικονική μηχανή μας.....	54
ΚΕΦΑΛΑΙΟ 5	58
	Σχεδιασμός και υλοποίηση της Οθόνης Εκκίνησης	58
5.1.	Σχεδιασμός Οθόνης Εκκίνησης	58
5.1.1.	Δημιουργία των μηχανισμών ελέγχου Textview	59
5.1.2.	Δημιουργία του colors.xml αρχείου.....	59
5.1.3.	Δημιουργία του dimens.xml αρχείου.....	59
5.1.4.	Συμπλήρωση του αρχείου string.xml.....	60
5.1.5.	Δημιουργία του μηχανισμού ελέγχου ImageView.....	60
5.1.6.	Ο κώδικας της main.xml συνολικά.....	60
5.2.	Προσθήκη κίνησης στο γραφικό της Οθόνης Εκκίνησης	62
5.2.1.	Προσθήκη πόρων κίνησης.....	62
5.3.	Εφαρμογή των Animation στην EuropeActivity	63
5.4.	Σύνδεση της Οθόνης Εκκίνησης με την Κύρια Οθόνη Επιλογών	64
5.5.	Ο κώδικας της Οθόνης Εκκίνησης (EuropeActivity) συνολικά.....	65
ΚΕΦΑΛΑΙΟ 6	67
	Σχεδιασμός και υλοποίηση της Κύριας Οθόνης Επιλογών	67
6.1.	Σχεδιασμός Κύριας Οθόνης Επιλογών	67
6.1.1.	Δημιουργία του menu.xml layout αρχείου	67
6.1.2.	Δημιουργία της ListView στο αρχείο menu.xml.....	68
6.2.	Δημιουργία των απαραίτητων πόρων για την Κύρια Οθόνη Επιλογών	68
6.3.	Δημιουργία νέου Layout με μορφή διάταξης ListView.....	69
6.4.	Δημιουργία της ListView μέσα στην Κύρια Οθόνη Επιλογών.....	70
6.4.1.	Ακρόαση για συμβάντα ListView	70
6.4.2.	Προσθήκη χαρακτηριστικών στο μηχανισμό ελέγχου ListView	71
6.5.	Προσθήκη ενός μενού επιλογών στην Οθόνη του Παιχνιδιού.....	72

6.5.1.	Προσθήκη Πόρων Μενού.....	72
6.5.2.	Προσθήκη ενός Μενού Επιλογών σε μια Δραστηριότητα.....	72
ΚΕΦΑΛΑΙΟ 7		74
Σχεδίαση κι υλοποίηση των Οθονών Βοήθειας και Βαθμολογίας		74
7.1.	Σχεδίαση της Οθόνης Βοήθειας	74
7.1.1.	Δημιουργώντας το αρχείο xml της Οθόνης Βοήθειας	74
7.1.2.	Προσθήκη πόρων Οθόνης Βοήθειας.....	74
7.1.3.	Ενημέρωση της QuizHelpActivity	75
7.2.	Σχεδίαση της Οθόνης Βαθμολογίας.....	76
7.2.1.	Δημιουργώντας το αρχείο xml της Οθόνης των Βαθμολογιών	76
7.2.2.	Προσθήκη πόρων Οθόνης Βαθμολογιών.....	77
7.2.3.	Ενημέρωση της QuizScoresActivity.java	78
ΚΕΦΑΛΑΙΟ 8		81
Σχεδίαση και Υλοποίηση της Οθόνης των Ρυθμίσεων.....		81
8.1.	Σχεδίαση της Οθόνης Ρυθμίσεων	81
8.2.	Δημιουργία του layout settings.xml.....	81
8.2.1.	Προσθήκη των πόρων για την ολοκλήρωση του αρχείου settings.xml.....	84
8.2.2.	Προσθήκη πόρων Πίνακα Συμβολοσειρών.....	84
8.3.	Αποθήκευση δεδομένων της Οθόνης Ρυθμίσεων με τη χρήση του μηχανισμού SharedPreferences	85
8.4.	Ενημέρωση της QuizSettingsActivity.java	85
ΚΕΦΑΛΑΙΟ 9		93
Σχεδίαση κι υλοποίηση της Κύριας Οθόνης του Παιχνιδιού		93
9.1.	Σχεδίαση της Κύριας Οθόνης της εφαρμογής.....	93
9.2.	Δημιουργία του game.xml	93
9.3.	Δημιουργία της κλάσης Question.java.....	96
9.4.	Δημιουργία της κλάσης DBHelper.java	98
9.5.	Δουλεύοντας με την QuizGameActivity.java.....	100
ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....		103
1.	Συμπεράσματα	103
2.	Μελλοντική Εργασία	103
Πίνακας εικόνων		105
Βιβλιογραφία – Δικτυογραφία.....		107

Εισαγωγή

Στις μέρες μας το Android OS κυριαρχεί στις φορητές συσκευές που χρησιμοποιούμε στην καθημερινότητά μας, είτε αυτές είναι κινητά είναι tablets. Προσφέρει πολλά στους χρήστες κι οι εκδόσεις του ποικίλουν ανάλογα με τη συσκευή. Πολλά sites και blogs ασχολούνται με το android και την υλοποίηση εφαρμογών μέσα από αυτό. Όλο και περισσότερες εταιρείες προτιμούν να εντάξουν στις φορητές συσκευές τους το Android.

Όλη αυτή η συζήτηση γύρω από το Android, η περιέργεια να γνωρίσουμε αυτό το λειτουργικό σύστημα, για το οποίο μιλάνε τόσο πολλοί και στο οποίο κάνουν αναφορά πολλά μέσα που ασχολούνται με την τεχνολογία, μας έδωσε την ώθηση να ασχοληθούμε με το συγκεκριμένο λειτουργικό σύστημα.

Σκοπός, λοιπόν, της εργασίας αυτής είναι να γνωρίσουμε το Android, να καταλάβουμε πως λειτουργεί, να κατανοήσουμε γιατί είναι τόσο διαδεδομένο και τέλος να εξοικειωθούμε με το λειτουργικό αυτό σύστημα, δημιουργώντας τη δική μας εφαρμογή.

Η έρευνά μας διήρκεσε αρκετό χρονικό διάστημα και χρειάστηκε αρκετή έρευνα και πολλές ώρες αφιερωμένες στο δέσιμο του κώδικα, που μας οδήγησε στην υλοποίηση της εφαρμογής μας. Στην πορεία αντιμετωπίσαμε αρκετές δυσκολίες, χρειάστηκε να γίνει το πρόγραμμα από την αρχή, αλλά στο τέλος καταφέραμε να επιτύχουμε τον σκοπό μας, να υλοποιήσουμε σε ικανοποιητικό βαθμό την εφαρμογή μας σε Android.

Στην εργασία αυτή έχει γίνει μία σύντομη αναφορά στο λειτουργικό σύστημα Android, τα χαρακτηριστικά του, τις τελευταίες εκδόσεις του, στον τρόπο εγκατάστασης των εργαλείων που χρησιμοποιούνται για τον προγραμματισμό σε Android και τέλος γίνεται ανάλυση βήμα βήμα στο πως μπορεί να υλοποιήσει κάποιος μια απλή εφαρμογή στο λειτουργικό αυτό σύστημα.

Περίληπτικά έχουμε:

Στο 1^ο κεφάλαιο γίνεται αναφορά στο τι είναι το Android, στα χαρακτηριστικά που το κάνουν να διαφέρει, στο πως μπορεί κάποιος να αποκτήσει μια εφαρμογή Android και τέλος πόσο είναι διαδεδομένο ανά τον κόσμο.

Το 2^ο κεφάλαιο κεφάλαιο είναι αφιερωμένο στην εξέλιξη του Android τα τελευταία χρόνια, κάνοντας αναφορά στις πιο πρόσφατες εκδόσεις του.

Στο 3^ο κεφάλαιο αναπτύσσεται ο τρόπος εγκατάστασης των απαραίτητων εργαλείων, που προτείνονται από τους android developers, για όποιον θέλει να προγραμματίσει σε Android. Γίνεται αναφορά ξεχωριστά σε κάθε ένα από αυτά και βήμα βήμα το πώς μπορεί κάποιος να τα εγκαταστήσει στον υπολογιστή του.

Στο 4^ο κεφάλαιο αφού γίνεται η πρώτη γνωριμία με την αρχιτεκτονική των εφαρμογών Android, ξεκινά η υλοποίηση της εφαρμογής, δημιουργώντας την πρώτη οθόνη, τα αρχικά στοιχεία της εφαρμογής και το πρώτο τρέξιμό της με τη χρήση ενός εικονικού κινητού.

Στο 5^ο έως το 9^ο κεφάλαιο συνεχίζεται η υλοποίηση της εφαρμογής, οθόνη – οθόνη, παρουσιάζοντας τον κώδικα της καθεμιάς χωριστά και επεξηγώντας τα διαφορετικά στοιχεία της.

Τέλος, στα συμπεράσματα αναφερόμαστε στο τι αποκομίσαμε από την όλη αυτή προσπάθεια, αλλά και τις μελλοντικές δυνατότητες που έχουμε προκειμένου να βελτιώσουμε και να αναπτύξουμε ακόμη περισσότερο την εφαρμογή μας.

ΚΕΦΑΛΑΙΟ 1

Γνωριμία με το Android OS

1.1. Τι είναι android;

Το android είναι ένα λειτουργικό σύστημα (Operating System – OS) για φορητές συσκευές το οποίο τρέχει τον πυρήνα του λειτουργικού Linux (Kernel). Ο πυρήνας αυτός είναι χαρακτηριστικό παράδειγμα ελεύθερου λογισμικού. Δίνει τη δυνατότητα δημιουργίας καινοτόμων εφαρμογών για φορητές συσκευές σε περιβάλλον προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού αναπτυγμένων από την Google. Επειδή βασίζεται πάνω σε ελεύθερο λογισμικό, δίνει τη δυνατότητα χρήσης ανοικτού κώδικα για την υλοποίηση νέων εφαρμογών, αλλά και τη χρήση όλων των δυνατοτήτων που παρέχονται από μια φορητή συσκευή όπως την πραγματοποίηση κλήσεων, την αποστολή μηνυμάτων κειμένου ή τη χρήση της κάμερας. Προσφέρει, δηλαδή, μία ολοκληρωμένη σειρά λογισμικού για φορητές συσκευές: ένα λειτουργικό σύστημα, ένα ενδιάμεσο λογισμικό και τη χρήση βασικών εφαρμογών για φορητές συσκευές.



Εικόνα 1: Λογότυπο Android

Ο πηγαίος κώδικας του Android προσφέρεται από την Google υπό την άδεια της Apache. Η ελαστικότητα στην χορήγηση αδειών επιτρέπει την ελεύθερη τροποποίηση και διανομή του λογισμικού από τους κατασκευαστές συσκευών, από τις εταιρείες ασύρματης μεταφοράς δεδομένων και από προγραμματιστές που ασχολούνται συστηματικά με το συγκεκριμένο λειτουργικό σύστημα.

1.2. Χαρακτηριστικά που ενισχύουν τη χρήση του Android

➤ **Ανοικτό Λογισμικό**

Το Android δίνει τη δυνατότητα ενός πλούσιου αριθμού πεδίου εφαρμογών για τη δημιουργία καινοτόμων εφαρμογών για κινητές συσκευές σε περιβάλλον προγραμματισμού java. Δημιουργήθηκε με σκοπό να βοηθήσει τους δημιουργούς εφαρμογών για φορητές συσκευές να αξιοποιήσουν στο έπακρο τις δυνατότητες μιας τέτοιας συσκευής. Χτίστηκε για να είναι πραγματικά ένα ελεύθερο λογισμικό. Για παράδειγμα, μία τέτοια εφαρμογή μπορεί να χρησιμοποιήσει οποιεσδήποτε από τις βασικές λειτουργίες ενός τηλεφώνου, όπως την πραγματοποίηση κλήσεων, την αποστολή μηνυμάτων κειμένου ή τη χρήση της κάμερας, επιτρέποντας στους προγραμματιστές να προσφέρουν στους χρήστες πιο πλούσιες και πιο ξεχωριστές εμπειρίες. Επίσης, χρησιμοποιεί μια προσαρμοσμένη εικονική μηχανή που έχει σχεδιαστεί για τη βελτιστοποίηση της μνήμης και των πόρων του υλικού σε ένα περιβάλλον κινητού. (Open Handset Alliance)

➤ **Όλες οι εφαρμογές που δημιουργούνται είναι ισότιμες**

Το Android δεν κάνει διάκριση μεταξύ των βασικών εφαρμογών ενός κινητού και των εφαρμογών από τρίτους. Μπορούν όλοι να κατασκευαστές να έχουν ίση πρόσβαση στις δυνατότητες ενός τηλεφώνου, παρέχοντας στους χρήστες ένα ευρύ φάσμα εφαρμογών και υπηρεσιών. Με συσκευές που κατασκευάστηκαν σε πλατφόρμα Android, οι χρήστες είναι σε θέση να προσαρμόσουν πλήρως το τηλέφωνο τους ανάλογα με τα ενδιαφέροντά τους

και τις ανάγκες τους. Μπορούν να αλλάξουν την αρχική οθόνη του τηλεφώνου, το στυλ του dialer, ή οποιαδήποτε από τις εφαρμογές. Μπορούν ακόμη να εγκαταστήσουν στο τηλέφωνό τους την αγαπημένη τους εφαρμογή προβολής φωτογραφιών και να την χρησιμοποιούν για την προβολή όλων των φωτογραφιών τους.

➤ **Εξάλειψη των ορίων των εφαρμογών**

Το Android εξαλείφει τα όρια στην δημιουργία νέων και καινοτόμων εφαρμογών. Παραδείγματος χάριν, ένας developer μπορεί να συνδυάσει τις πληροφορίες από το web με τα δεδομένα του κινητού τηλέφону ενός χρήστη — όπως οι επαφές του χρήστη, το ημερολόγιο ή η γεωγραφική θέση στην οποία βρίσκεται — παρέχοντας έτσι μια πιο εξατομικευμένη εμπειρία στους χρήστες. Με το android, ένας developer μπορεί να δημιουργήσει μια εφαρμογή που ενημερώνει τους χρήστες για το αν οι φίλοι τους είναι συνδεδεμένοι με την εφαρμογή αυτή και για το όταν υπάρχει δυνατότητα σύνδεσης με αυτούς.

➤ **Γρήγορη κι εύκολη υλοποίηση εφαρμογών**

Το Android παρέχει πρόσβαση σε ένα ευρύ φάσμα χρήσιμων βιβλιοθηκών και εργαλείων που μπορούν να χρησιμοποιηθούν για τη δημιουργία μεγάλου αριθμού εφαρμογών. Για παράδειγμα, το Android δίνει τη δυνατότητα στους προγραμματιστές να βρουν που βρίσκεται η συσκευή και επιτρέπει στις συσκευές να επικοινωνήσουν μεταξύ τους χρησιμοποιώντας πλούσιες peer-to-peer κοινωνικές εφαρμογές. Επιπλέον, το Android περιλαμβάνει ένα πλήρες σύνολο εργαλείων που έχουν δημιουργηθεί μαζί με την πλατφόρμα εξ αρχής, παρέχοντας στους προγραμματιστές υψηλή παραγωγικότητα και βαθιά γνώση όσον αφορά τις εφαρμογές τους.

1.3. Τι είναι το google play;

Το google play, πρώην android market, είναι μία ψηφιακή πλατφόρμα διανομής εφαρμογών για το λειτουργικό Android, ένα online ηλεκτρονικό και ψηφιακό κατάστημα, που υποστηρίζεται από την Google. Η υπηρεσία επιτρέπει στους χρήστες να περιηγηθούν και να κατεβάσουν εφαρμογές που έχουν αναπτυχθεί με το Android SDK¹ και δημοσιεύονται μέσω του Google, καθώς επίσης μουσική, περιοδικά, βιβλία, ταινίες και τηλεοπτικά προγράμματα. Οι χρήστες μπορούν επίσης να αγοράσουν hardware όπως τα Chromebooks, τα νέα laptops της Google, τις Google Nexus κινητές συσκευές, τις Chromecasts, συσκευές με τις οποίες μπορεί κάποιος να απολαύσει μουσική, βίντεο, ταινίες, παιχνίδια απευθείας στην τηλεόρασή του, καθώς επίσης και διάφορα αξεσουάρ μέσω του Google Play.

Η διανομή των εφαρμογών γίνεται είτε δωρεάν είτε επί πληρωμή με μια μικρή χρέωση η οποία μπορεί να κυμανθεί από 0,02€ μέχρι 3€. Αυτό εξαρτάται από την κάθε εφαρμογή και το πόσο ζήτηση έχει. Ο χρήστης μπορεί να κατεβάσει τις εφαρμογές απευθείας σε μια Android συσκευή ή στην Google TV μέσα από την εφαρμογή Play Store του κινητού του, ή αναπτύσσοντας την εφαρμογή σε μια συσκευή χρησιμοποιώντας την ιστοσελίδα της Google Play. Πολλές εφαρμογές μπορούν να απευθύνονται σε συγκεκριμένους χρήστες με βάση ένα συγκεκριμένο χαρακτηριστικό της συσκευής τους, όπως ο αισθητήρας κίνησης (απαραίτητος για τη λειτουργία κάποιων παιχνιδιών) ή η κάμερα στο μπροστινό μέρος (απαραίτητη για την πραγματοποίηση μιας online βιντεοκλήσης).

¹ Android Software Development Kit (Android SDK): Είναι το επίσημο εργαλείο της Google γι' αυτούς που θέλουν να προγραμματίσουν σε Android



Εικόνα 2 Το λογότυπο της Google Play

Στις 6 Μαρτίου 2012 με τη συγχώνευση του Android Market και της Google Music, η υπηρεσία μετονομάστηκε σε Google Play και η μετονομασία συνέπεσε με τον επαναπροσδιορισμό της στρατηγικής ψηφιακής διανομής της Google. Οι υπηρεσίες που προσφέρονται κάτω από το λογότυπο της Google Play είναι: Το Google Play Music (για μουσική), το Google Play Books (για βιβλία), το Google Play Newsstand (για ενημέρωση), το Google Play Movies & TV (για ταινίες και τηλεόραση) και το Google Play Games (για παιχνίδια).

Έως τον Ιούλιο του 2013, το Google Play Store επίσημα είχε φτάσει στις 1 εκατομμύριο δημοσιευμένες εφαρμογές και περίπου στα 50 δισεκατομμύρια downloads. **Η πηγή που καθορίστηκε δεν είναι έγκυρη.**

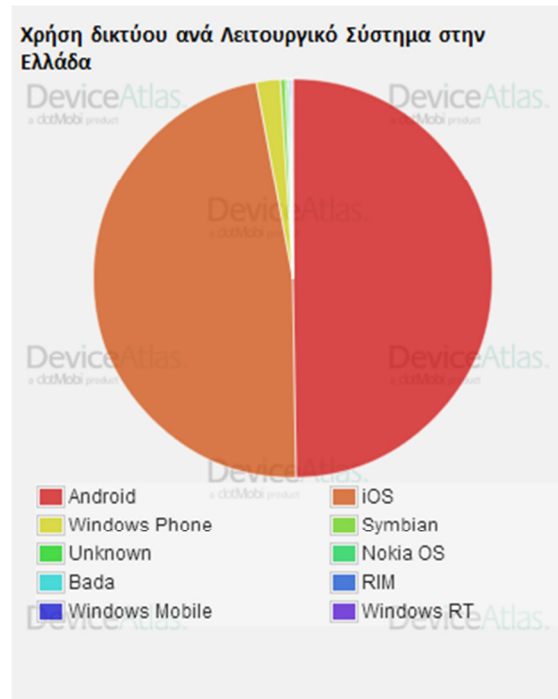
1.4. Πόσο διαδεδομένο είναι το Android OS

Σύμφωνα με τον Παγκόσμιο Χάρτη και κάποια στατιστικά στοιχεία που δημοσίευσε το dotMobi (dotMobi, 2014) φαίνεται ότι το Android λειτουργικό σύστημα υπερέρχει έναντι των άλλων. Από τις 101 χώρες που καλύπτει η ανάλυση, το Android υπερέρχει στις 67 (ως επί το πλείστον αναπτυσσόμενες αγορές) και το iOS σε 34 (δυτικές αγορές), ενώ μόλις δύο αγορές έχουν άλλον πρωτοπόρο (BlackBerry και Nokia). Οι χώρες με το πράσινο χρώμα είναι οι χώρες στις οποίες επικρατεί το Android OS. (Elpidis, 2014)



Εικόνα 3: iOS έναντι Android

Επίσης σύμφωνα με άλλη έρευνα από το dotMobi που παρουσιάζεται από την ιστοσελίδα Device Atlas και έγινε ανά χώρα για τη χρήση του δικτύου ανά λειτουργικό σύστημα, το android λειτουργικό σύστημα επικρατεί και στην Ελλάδα σε σχέση με το υπόλοιπα και κυρίως σε σχέση με το iOS, το οποίο όπως φαίνεται από το διάγραμμα είναι κι ο μεγαλύτερος ανταγωνιστής του.



Εικόνα 4: (Device Data, 2014)

ΚΕΦΑΛΑΙΟ 2

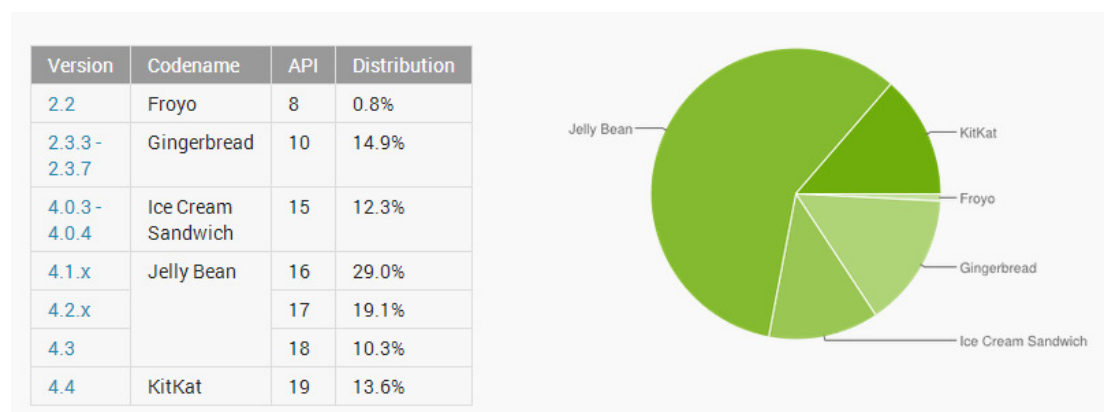
Ιστορική αναδρομή

Αρχικά αναπτύχθηκε από την ομώνυμη εταιρεία Android, Inc., η οποία χρηματοδοτούνταν από την Google κι από την οποία τελικά αγοράστηκε το 2005.

Το λειτουργικό σύστημα Android άρχισε να λειτουργεί ως «beta» έκδοση το Νοέμβριο του 2007. Είναι η εποχή που δημιουργείται κι η Open Handset Alliance (μια κοινοπραξία 47 αρχικά εταιρειών σχετικών με την παραγωγή hardware και software λογισμικού, κατασκευαστών φορητών συσκευών, προγραμματιστικών εφαρμογών, εταιρειών κινητής τηλεφωνίας καθώς και κατασκευαστών chip), η οποία προωθούσε την ανάπτυξη ανοικτών προτύπων για κινητές συσκευές. Η πρώτη εμπορική έκδοση Android 1.0 κυκλοφόρησε το Σεπτέμβριο του 2008. Το λειτουργικό σύστημα Android είναι υπό συνεχόμενη ανάπτυξη από την Google και την Open Handset Alliance και έχουν γίνει μια σειρά από αναβαθμίσεις του βασικού λειτουργικού συστήματος σε σχέση με την αρχική έκδοσή του.

Μέχρι τον Απρίλιο του 2009, όλες οι εκδόσεις Android είχαν γίνει γνωστές με τα κωδικοποιημένα ονόματα που παρέπεμπαν σε όρους ζαχαροπλαστικής. Η ονοματοδοσία είναι με αλφαβητική σειρά: Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), και KitKat (4.4–4.4.3).

Τα παρακάτω στατιστικά δεδομένα αντανακλούν στον αριθμό των συσκευών που τρέχουν τις τελευταίες εφαρμογές που έχουν προστεθεί στο Google Play, οι οποίες είναι συμβατές με συσκευές που υποστηρίζουν Android 2.2 και οποιαδήποτε νεότερη έκδοση. Από το Σεπτέμβριο του 2013 οι συσκευές που είναι παλαιότερες της έκδοσης 2.2 δεν εμφανίζονται στα στατιστικά στοιχεία γιατί δεν υποστηρίζουν τις νέες εφαρμογές του Google Play Store.



Εικόνα 5: Στοιχεία που συλλέχθηκαν μετά από μια εβδομαδιαία έρευνα μέχρι την 4^η Ιουνίου 2014. Οι εκδόσεις με ποσοστό κάτω του 0,1% δεν εμφανίζονται. (Developers, Dashboards/Android Developers, 2014)

2.1. Gingerbread (Android 2.3.3 APIs, Android 2.3.4 APIs)

Η πλατφόρμα Android 2.3 εισήγαγε πολλές νέες και ενδιαφέρουσες δυνατότητες για τους χρήστες και τους προγραμματιστές.

Οι βελτιώσεις που παρατηρούνται όσον αφορά τους χρήστες είναι:

- Βελτιώσεις στην διεπαφή χρήστη με σκοπό την απλότητα και την ταχύτητα.
- Πιο γρήγορη εισαγωγή κειμένου.
- Επιλογή λέξης με ένα άγγιγμα, αντιγραφή κι επικόλληση.
- Βελτιωμένη διαχείριση της εξοικονόμησης ενέργειας της μπαταρίας της φορητής συσκευής.
- Έλεγχος επί των εφαρμογών.
- Νέοι τρόποι επικοινωνίας και οργάνωσης (επικοινωνία μέσω Διαδικτύου, επικοινωνία μεταξύ κοντινών τομέων (near-field communication – NFC), διαχείριση μεταφόρτωσης και χρήση της κάμερας).



Εικόνα 6: Έκδοση Gingerbread

Το Android 2.3 προσφέρει μια ποικιλία δυνατοτήτων και APIs² τα οποία βοηθούν προγραμματιστές να υλοποιήσουν νέες μορφές εφαρμογών σε Android.

- Βελτιώσεις στο gaming
- Νέες μορφές επικοινωνίας
- Μεγαλύτερος αριθμός πολυμέσων

2.2. Honeycomb (Android 3.0 APIs, Android 3.1 APIs, Android 3.2 APIs)

Είναι μία έκδοση αποκλειστικά για tablets, η οποία έφερε αλλαγές στο γραφικό περιβάλλον, πρόσθεσε την υποστήριξη πολλαπλών πυρήνων συνδυασμένη με βελτιωμένα γραφικά.

Νέα χαρακτηριστικά για τους χρήστες:

- Νέα οθόνη διεπαφής σχεδιασμένη από την αρχή για tablets
- Επανασχεδιασμένο πληκτρολόγιο
- Βελτιωμένη δυνατότητα επιλογής, αντιγραφής κι επικόλλησης κειμένου.
- Νέες δυνατότητες επικοινωνίας.
- Αναβαθμίσεις στις βασικές εφαρμογές (browser, κάμερα και συλλογή φωτογραφιών, επαφές, e-mail)



Εικόνα 7: Έκδοση Honeycomb

Η πλατφόρμα Android 3.0 σχεδιάστηκε για να καλύψει τις ανάγκες των εφαρμογών σε συσκευές με μεγαλύτερο μέγεθος οθόνης. Προσέφερε όλα τα εργαλεία εκείνα που είναι απαραίτητα στους προγραμματιστές για να δημιουργήσουν εφαρμογές που θα προσφέρουν απίστευτη οπτική εμπειρία κι αλληλεπίδραση.

Τα νέα χαρακτηριστικά που προσφέρονται είναι:

- Νέο πλαίσιο διεπαφής χρήστη για τη δημιουργία εφαρμογών για tablet
- Ανώτερης ποιότητας 2D και 3D γραφικών

² Application Programming Interface (API): Καλούμε Διεπαφή Προγραμματισμού Εφαρμογών (αγγλ. API, από το Application Programming Interface), γνωστή και ως Διασύνδεση Προγραμματισμού Εφαρμογών (για συντομία διεπαφή ή διασύνδεση), τη διεπαφή των προγραμματιστικών διαδικασιών που ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή παρέχει προκειμένου να επιτρέψει να γίνονται προς αυτό αιτήσεις από άλλα προγράμματα ή / και ανταλλαγή δεδομένων. (ΒΙΚΙΠΑΙΔΕΙΑ)

- Υποστήριξη επεξεργαστών πολλαπλών πυρήνων
- Μεγάλος αριθμός πολυμέσων κι συνδεσιμότητας
- Βελτιώσεις για τις επιχειρήσεις.
- Συμβατότητα με τις υπάρχουσες εφαρμογές.

2.3. Ice Cream Sandwich (Android 4.0 APIs, Android 4.0.3 APIs)

Το Android 4.0 προσφέρει ένα πιο εκλεπτυσμένο και ενοποιημένο User Interface(UI) για κινητά τηλέφωνα και tablets και προσφέρει στους χρήστες και τους προγραμματιστές νέες καινοτόμες δυνατότητες, όπως οι παρακάτω:

Το Android 4.0 βασίζεται πάνω σε χαρακτηριστικά που υπήρξαν πολύ δημοφιλή στις Android συσκευές. Η ευκολία στην εναλλαγή μεταξύ των διάφορων εργασιών, οι πλούσιες ειδοποιήσεις, οι οθόνες με δυνατότητα προσαρμογής στα θέλω του χρήστη, τα προσαρμόσιμα widgets³ και η εμφανής διαδραστικότητα, προσθέτουν νέους ισχυρούς τρόπους επικοινωνίας και διαμοιρασμού των εργασιών του χρήστη.



Εικόνα 8: Έκδοση Ice Cream Sandwich

Φτιαγμένο με βάση τον τρόπο που ζουν οι άνθρωποι το Android 4.0 ενσωματώνει πολλά στοιχεία που βοηθούν στην συνεχή κοινωνικοποίηση του ανθρώπου, προσφέροντάς του δυνατότητα επεξεργασίας και διαμοιρασμού φωτογραφιών, επικοινωνία μέσω voicemail κι άλλων πολλών δυνατοτήτων. Κι όλα αυτά γίνονται με ένα απλό άγγιγμα σε κάθε σημείο του συστήματος καθιστώντας ακόμη πιο εύκολη τη συνομιλία, την αποστολή mail, την συγγραφή μηνύματος και το διαμοιρασμό.

Το Android 4.0 διαθέτει σύνδεση cloud κι αυτή επιτρέπει στους χρήστες να περιηγηθούν στο διαδίκτυο και να συγχρονίσουν φωτογραφίες, εφαρμογές, παιχνίδια, e-mail, τις επαφές τους, όπου κι αν βρίσκονται, σε όλες τις φορητές τους συσκευές. Το Android 4.0 προσθέτει νέες δυνατότητες περιήγησης στο Internet, αλλά και σχετικές με τα e-mail, αφού επιτρέπει στους χρήστες να τα λαμβάνουν όπου κι αν βρίσκονται και να μη χάνουν την επικοινωνία με τις επαφές τους ποτέ.

Για τους developers φέρνει ένα ενοποιημένο UI που τους επιτρέπει να δημιουργούν κομψές και καινοτόμες εφαρμογές για κινητά τηλέφωνα και tablets. Το ενοποιημένο UI σημαίνει νέα εργαλεία στην οθόνη διεπαφής, συνεπείς πρακτικές σχεδιασμού, απλοποιημένους κώδικες και πόρους κι εκσυγχρονισμό όλων των συσκευών που χρησιμοποιούν το λογισμικό Android. (Developers, Ice Cream Sandwich)

2.4. Jelly Bean (Android 4.1 APIs, Android 4.2 APIs, Android 4.3 APIs)

Η Jelly Bean έκδοση Android αποτελείται από 3 υποεκδόσεις, καθεμιά υπάρχει για να βελτιώνει την προηγούμενη και να της προσθέτει πράγματα. Έτσι η 4.1. έκδοση δημιουργήθηκε προκειμένου να βελτιώσει την απόδοση, σχεδιασμένη για να δώσει στο λειτουργικό σύστημα μια ομαλότερη κι ευέλικτη αίσθηση. Οι βελτιώσεις επικεντρώνονταν στο σύστημα κοινοποίησης και την δυνατότητα πια «επέκτασης» των κοινοποιήσεων με action buttons και σε άλλες εσωτερικές αλλαγές.

³ Widgets: Είναι εφαρμογές που βρίσκονται στην αρχική οθόνη (home screen) της φορητής συσκευής με τη μορφή εικονιδίων και παρέχουν στους χρήστες κάποιες σημαντικές πληροφορίες (καιρός, θερμοκρασία, ενεργοποιημένο η όχι bluetooth κ.α.) (Android Developers, Widgets)

Οι δύο επόμενες υποεκδόσεις περιλάμβαναν: η 4.2 περαιτέρω βελτιστοποιήσεις, δυνατότητα υποστήριξης περισσότερων από έναν χρηστών για τα tablets, widgets για το κλείδωμα της οθόνης, γρήγορες ρυθμίσεις και screen savers κι αργότερα η 4.3 περαιτέρω βελτιώσεις κι ενημερώσεις σε σχέση με την συγκεκριμένη πλατφόρμα Android. (wikipedia, 2014)



Εικόνα 9: Έκδοση Jelly Bean

2.5. Android 4.4 KitKat

Το KitKat «σπάει» το φράγμα του περιορισμού μνήμης και έχει τη δυνατότητα πια να "τρέξει" σε συσκευές με RAM από 512MB και πάνω, φέρνοντας τις τελευταίες αναβαθμίσεις του Android ακόμη και σε οικονομικά προσιτά μοντέλα. Σύμφωνα με τη Google χρησιμοποιείται 16 τοις εκατό λιγότερη μνήμη από την προηγούμενη έκδοση Jelly Bean, με αποτέλεσμα ένα συνολικά πιο ελαφρύ λειτουργικό σε όποιο μοντέλο κι αν ενσωματώνεται.



Εικόνα 10: Έκδοση KitKat

Το λειτουργικό σύστημα είναι πιο καθαρό και πιο απλό από ποτέ. Το status και το notification bar της οθόνης έναρξης είναι πλέον διάφανα, ενώ εξαφανίζονται εντελώς όταν ο χρήστης βρίσκεται μέσα σε μία εφαρμογή.

Το Android 4.4. KitKat διαθέτει μία εντελώς ανανεωμένη Phone app, που κάνει τις κλήσεις σας πιο εύκολες από ποτέ, βοηθώντας σας να γίνουν εύκολα και γρήγορα αναζητήσεις στις επαφές του χρήστη, σε κοντινά μέρη, ακόμη και σε Google Apps accounts μέσα από την ίδια την εφαρμογή.

Ενημερωμένη είναι, ωστόσο κι η εφαρμογή των Hangouts, που πλέον διαχειρίζεται όλα τα SMS και MMS σας στο ίδιο σημείο με τις διαδικτυακές επικοινωνίες του χρήστη, ώστε να μη χάνεται κανένα μήνυμα από φίλους σε όποια μορφή και αν έχει σταλεί.

Βελτιώσεις υπάρχουν και στην υπηρεσία φωνητικών αναζητήσεων της Google. Το Voice Search είναι τώρα 25% πιο ακριβές, ενώ μπορεί να ενεργοποιείται και με λέξη κλειδι (keyword) χωρίς καν να ακουμπήσετε τη συσκευή. Και συνολικά, βέβαια, οι αναζητήσεις του χρήστη στο νέο λειτουργικό είναι καλύτερες από ποτέ και πλέον μπορούν να τον οδηγούν ακόμη και σε εφαρμογές και όχι μόνο σε ιστοσελίδες. Το «Google Now» πλέον ενεργοποιείται με ένα swipe από τα αριστερά στα δεξιά στην homescreen σας, έχοντας και αυτό σημαντικές βελτιώσεις -πέρα από τον τομέα της αναζήτησης που αναφέραμε- και στα χαρακτηριστικά των καρτελών του. (Kitsati, 2013)

ΚΕΦΑΛΑΙΟ 3

Εγκατάσταση απαιτούμενων προγραμμάτων για την ανάπτυξη μίας Android Εφαρμογής

3.1. Τι χρειαζόμαστε για την ανάπτυξη μιας Android Εφαρμογής

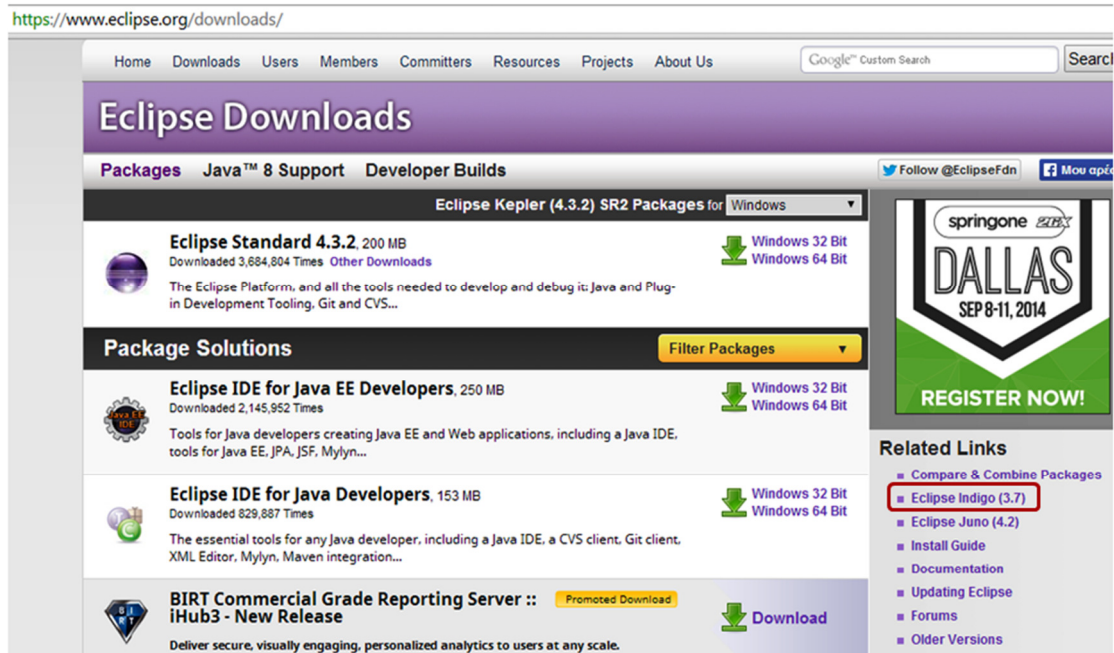
Προκειμένου να προγραμματίσουμε την εφαρμογή Android που θέλουμε, θα επιλέξουμε να δουλέψουμε στο περιβάλλον προγραμματισμού Java του Eclipse. Επιγραμματικά χρειαζόμαστε:

- Να εγκαταστήσουμε το Eclipse 3.7.2. (Indigo) για Windows 32bit (η εφαρμογή υλοποιήθηκε σε υπολογιστή με λειτουργικό Windows 32bit).
- Να έχουμε το Eclipse [JDT](#) (Java Development Tools) plugin (περιέχεται συνήθως στα περισσότερα πακέτα Eclipse IDE)
- Να κατεβάσουμε το [JDK 6](#) (Java Development Kit) ή και νεώτερο (το JRE – Java Runtime Environment, μόνο του δεν λειτουργεί)
- Τέλος να εγκαταστήσουμε τα [Android Development Tools plugin](#)

Παρακάτω παρατίθενται οδηγίες εγκατάστασης των παραπάνω προγραμμάτων.

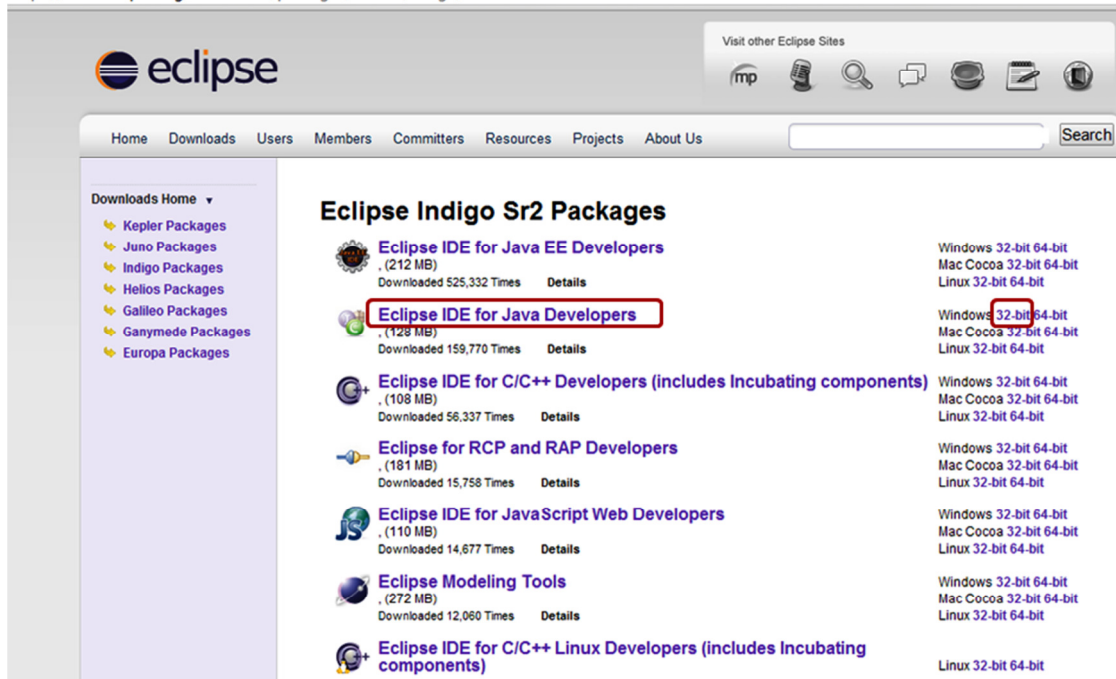
3.2. Εγκατάσταση του Eclipse

Πηγαίνουμε στην επίσημη ιστοσελίδα του Eclipse στις λήψεις (**downloads**) κι από εκεί διαλέγουμε το **Eclipse Indigo (3.7)** για να το κατεβάσουμε στον υπολογιστή μας.



Εικόνα 11: Κεντρική σελίδα των Eclipse Downloads

Στη συνέχεια στη σελίδα του Eclipse Indigo επιλέγουμε στην μπλε γραμμή πάλι downloads κι οδηγούμαστε στον κατάλογο με τα Eclipse Indigo Sr2 Packages.



Εικόνα 12: Η σελίδα για το κατέβασμα του Eclipse Indigo

Σε αυτό επιλέγουμε το **Eclipse IDE for Java Developers** για **Windows 32 bit**. Τέλος στο καινούργιο παράθυρο που ανοίγει διαλέγουμε την προτεινόμενη επιλογή που μας δίνεται για την χώρα στην οποία βρισκόμαστε. Ο συμπιεσμένος φάκελος θα αρχίσει να κατεβαίνει.



Εικόνα 13: Eclipse Download for Windows 32bit

Το αρχείο που κατέβηκε είναι σε μορφή zip. Μετά το τέλος της λήψης του συμπιεσμένου φακέλου του Eclipse, κάνουμε αποσυμπίεση σε έναν προορισμό που εμείς επιλέγουμε.

Το πρόγραμμα του Eclipse δεν χρειάζεται εγκατάσταση. Μόλις αποσυμπιέσουμε το φάκελο, τον ανοίγουμε και πατώντας διπλό κλικ στο αρχείο **eclipse.exe** το πρόγραμμα θα τρέξει. Πριν όμως τρέξουμε το πρόγραμμα θα πρέπει να έχουμε κι άλλα δύο στοιχεία εγκατεστημένα.

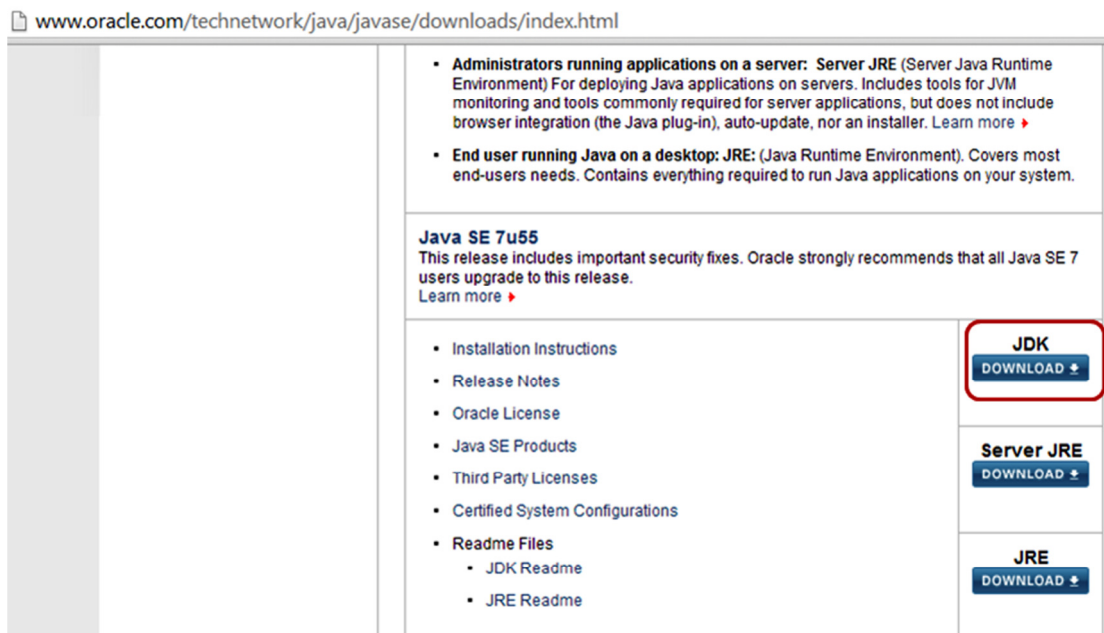
3.3. Εγκατάσταση του Java SE Development Kit 7 Downloads (για Windows 32bit)

Για να προγραμματίσουμε την εφαρμογή μας χρειαζόμαστε επίσης στάνταρντ έκδοση (Java Platform Standard Edition – Java SE) γλώσσας προγραμματισμού Java. Το JDK είναι βασικό εργαλείο της Java για τον προγραμματισμό εφαρμογών, applets, και άλλων τμημάτων που δημιουργούνται σε Java.

Το JDK είναι εργαλείο χρήσιμο για την ανάπτυξη και τον έλεγχο των προγραμμάτων γραμμένων σε Java και τρέχει στην πλατφόρμα Java.

Επομένως είναι απαραίτητο να κατεβεί κάποια έκδοση του JDK που θα χρειαστεί για την σωστή λειτουργία του Eclipse. Το καλύτερο είναι να επιλεγθεί μία έκδοση ανώτερη ή ίση της 6^{ης} έκδοσης. Κατεβάζουμε την έκδοση 7 για Windows 32bit.

Πηγαίνοντας στην επίσημη ιστοσελίδα της Oracle και πατώντας την διεύθυνση: <http://www.oracle.com/technetwork/java/javase/downloads/index.html?ssSourceSiteId=ocomen> ή αλλιώς ακολουθώντας την διαδρομή **Oracle Technology Network > Java > Java SE > Downloads**, οδηγούμαστε στην ιστοσελίδα όπου επιλέγουμε την έκδοση της **Java SE Development Kit 7 update 55**. Οποιαδήποτε έκδοση από την 6^η κι επάνω είναι αποδεκτή για να λειτουργήσει σωστά το πρόγραμμα.



www.oracle.com/technetwork/java/javase/downloads/index.html

- **Administrators running applications on a server: Server JRE** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#) ▶
- **End user running Java on a desktop: JRE**: (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

Java SE 7u55
This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
[Learn more](#) ▶

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK Readme
 - JRE Readme

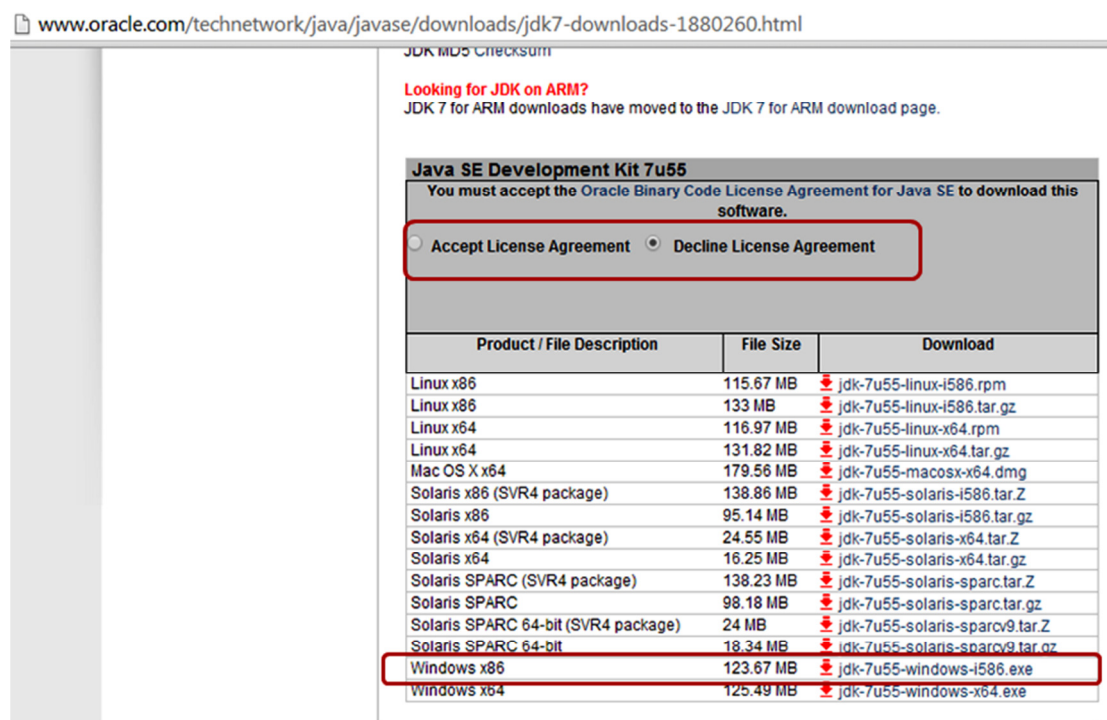
JDK
[DOWNLOAD](#) ▶

Server JRE
[DOWNLOAD](#) ▶

JRE
[DOWNLOAD](#) ▶

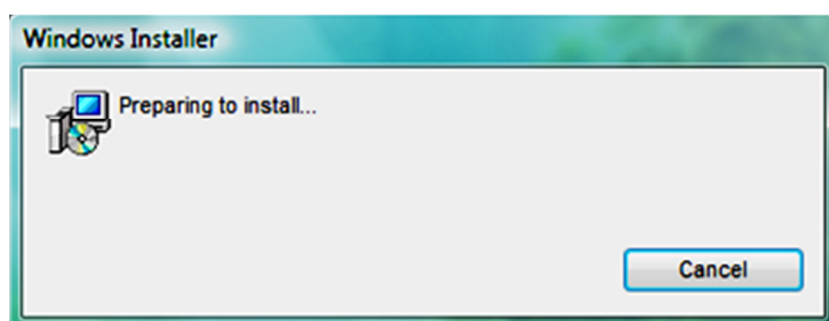
Εικόνα 14: Η σελίδα της Oracle για την εγκατάσταση της Έκδοσης Java SE Development Kit 7 update 55

Έπειτα πατώντας το μπλε κουμπί **JDK Download** όπως φαίνεται στην εικόνα οδηγούμαστε στο αμέσως επόμενο στάδιο της εγκατάστασης, προκειμένου να επιλέξουμε τι θα εγκαταστήσουμε, ανάλογα με το λειτουργικό σύστημα που διαθέτουμε. Για Windows 32bit, άρα χρειάζεται να κατεβάσουμε την έκδοση Windows x86. Αφού πρώτα διαβάσουμε κι αποδεχτούμε τους όρους της συμφωνίας αδειοδότησης της Oracle, κατεβάζουμε το συμπιεσμένο αρχείο [jdk-7u55-windows-i586.exe](#).



Εικόνα 15: Αποδοχή της άδειας και κατέβασμα της έκδοσης της Java για τα Windows 32bit (Windows x86)

Αφού κατεβάσουμε την εφαρμογή (είναι σε μορφή .exe), ξεκινάμε την εγκατάσταση.



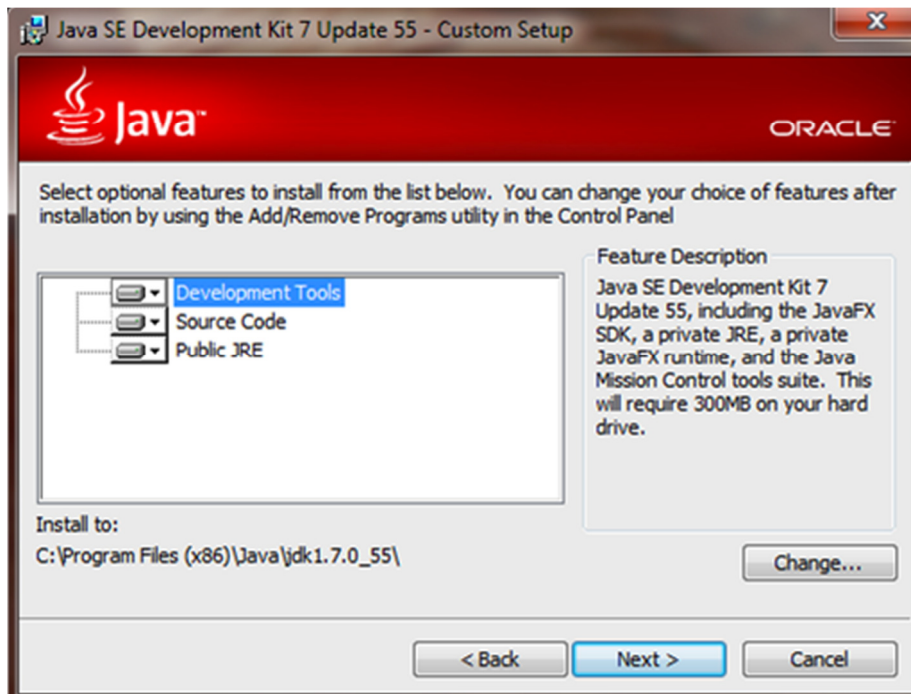
Εικόνα 16: Windows Installer

Στην επόμενη οθόνη πατάμε **Next**.



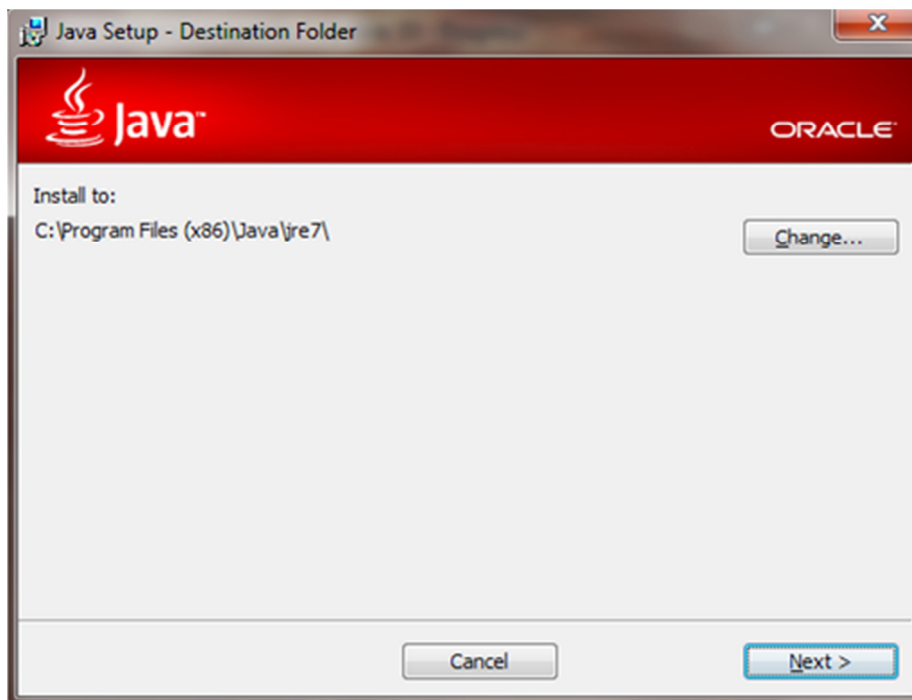
Εικόνα 17: Setup JDK 7u55

Στην επόμενη οθόνη επιλέγουμε τα δευτερεύοντα στοιχεία που θέλουμε να εγκατασταθούν μαζί. Επίσης στην οθόνη αυτή, φαίνεται κι η διαδρομή του φακέλου μέσα στο οποίο θα αποθηκευτούν. Εδώ κρατούμε τα προτεινόμενα, **Development Tools** και τη διαδρομή για το φάκελο **jdk 1.7.0_55** που δημιουργείται μέσα στον φάκελο java και πατάμε **Next**.



Εικόνα 18: Επιλογή των πρόσθετων χαρακτηριστικών (Development Tools) και της επιλογής του φακέλου στον οποίο εγκαθίστανται

Στην επόμενη οθόνη μπορούμε να αλλάξουμε το φάκελο στον οποίο θα εγκατασταθεί το πρόγραμμα. εμείς προτιμούμε να κρατήσουμε τον προτεινόμενο φάκελο(jre7) και πατάμε **Next**.



Εικόνα 19: Προτεινόμενος φάκελος για την εγκατάσταση του προγράμματος

Η επόμενη οθόνη δίνει την εξέλιξη της εγκατάστασης.



Εικόνα 20: Γίνεται η εγκατάσταση...

Φτάνουμε τέλος στην οθόνη που μας ενημερώνει ότι η εγκατάσταση έχει γίνει με επιτυχία. Πατάμε **Close** κι έχουμε τελειώσει με την εγκατάσταση του **Java SE Development Kit 7 Update 55**.



Εικόνα 21: Επιτυχής ολοκλήρωση της εγκατάστασης του Java SE Development Kit 7 Update 55

3.4. Εγκατάσταση του Android Development Tools plugin

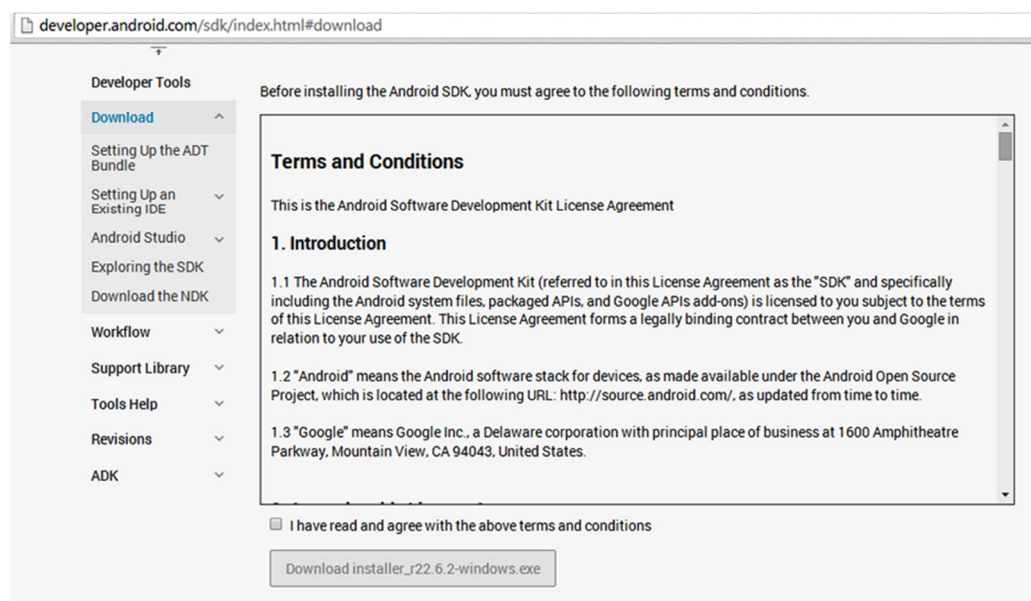
Επισκεπτόμαστε και πάλι την ιστοσελίδα <http://developer.android.com/sdk/index.html> και προχωρώντας προς τα κάτω, πατάμε στο **DOWNLOAD FOR OTHER PLATFORMS** και πηγαίνουμε στον πίνακα με τα SDK Tools Only, όπου επιλέγουμε να εγκαταστήσουμε την εφαρμογή [installer_r22.6.2-windows.exe](#) για Windows 32 & 64-bit.

Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86-20140321.zip	535085536 bytes	b61495a6bf591cc374c31bce4fc46ec0
Windows 64-bit	adt-bundle-windows-x86_64-20140321.zip	535287324 bytes	a6f4699bbdc5a29b371ed60610535651
Mac OS X 64-bit	adt-bundle-mac-x86_64-20140321.zip	501955296 bytes	4a08649cea9b098cdf7349f452294014
Linux 32-bit	adt-bundle-linux-x86-20140321.zip	527971926 bytes	943ae4d28fe7c79108c8bf2aafd5e6d2
Linux 64-bit	adt-bundle-linux-x86_64-20140321.zip	528187678 bytes	f2a2153b5c7dbaeb86b550bf4f770c36

Platform	Package	Size	MD5 Checksum
Windows 32 & 64-bit	android-sdk_r22.6.2-windows.zip	108917505 bytes	6e5bfd7b9c1d231ed6ec78b31551cbf
	installer_r22.6.2-windows.exe (Recommended)	87383126 bytes	2a68b8b22ecd0aba779b1581a914b395
Mac OS X 32 & 64-bit	android-sdk_r22.6.2-macosx.zip	74639394 bytes	2a319c862dd1dcf450bfe2a6b3d9c608
Linux 32 & 64-bit	android-sdk_r22.6.2-linux.tgz	101050024 bytes	ff1541418a44d894bedc5cef10622220

Εικόνα 22: installer_r22.6.2-windows.exe - Ο σύνδεσμος για την εγκατάστασή του στη σελίδα των developer.android.com

Πατώντας επάνω στον σύνδεσμο οδηγούμαστε σε παράθυρο για την ανάγνωση κι αποδοχή των όρων εγκατάστασης του προγράμματος. Τσεκάροντας το κουτάκι που βρίσκεται από κάτω από τους όρους συνεχίζουμε πατώντας το κουμπί [Download installer_r22.6.2-windows.exe](#).



Εικόνα 23: Όροι εγκατάστασης κι αποδοχή τους για την συνέχιση του κατεβάσματος των ADT plugin

Στη συνέχεια εμφανίζεται η οθόνη όπου διαλέγουμε που θέλει να αποθηκευτεί η εφαρμογή και πατάμε Download.

Με την αποθήκευση της εφαρμογής Android SDK, τα εργαλεία των SDK επιτρέπουν να εγκατασταθούν τα επιπρόσθετα packages και τα μελλοντικά updates μέσα από τον SDK Manager.

Το SDK Tools package δεν είναι ένα ολοκληρωμένο SDK πρόγραμμα. Περιέχει μόνο τα βασικά SDK εργαλεία, τα οποία βοηθούν στο να κατεβάσουμε τα υπόλοιπα SDK packages.

Το κατεβασμένο package είναι ένα εκτελέσιμο αρχείο το οποίο ξεκινά έναν installer. Ο installer ελέγχει τον υπολογιστή αν υπάρχουν τα αναγκαία εργαλεία, όπως το απαραίτητο Java SE Development Kit (JDK) και ζητά την εγκατάστασή του αν χρειάζεται. Έπειτα ο installer σώζει τα Android SDK Tools στο φάκελο που θα του ορίσουμε.

Συγκεκριμένα:

1. Κάνουμε διπλό κλικ στο αρχείο του installer που είναι σε μορφή .exe, για να ξεκινήσει η εγκατάσταση.
2. Σημειώνουμε το όνομα και την τοποθεσία στην οποία ο installer εγκαθιστά το SDK στον υπολογιστή. Όταν θα εγκαταστήσουμε αργότερα το ADT plugin, θα χρειαστεί να αναφέρουμε το φάκελο του SDK.
3. Όταν η εγκατάσταση ολοκληρωθεί ο installer μας ζητά να ανοίξει τον Android SDK Manager. Επειδή δουλεύουμε με Eclipse, δεν δεχόμαστε να τον ανοίξει τώρα, αλλά θα τον ανοίξουμε εμείς στη συνέχεια μέσω του Eclipse (Android Developers: Setting Up an Existing IDE).

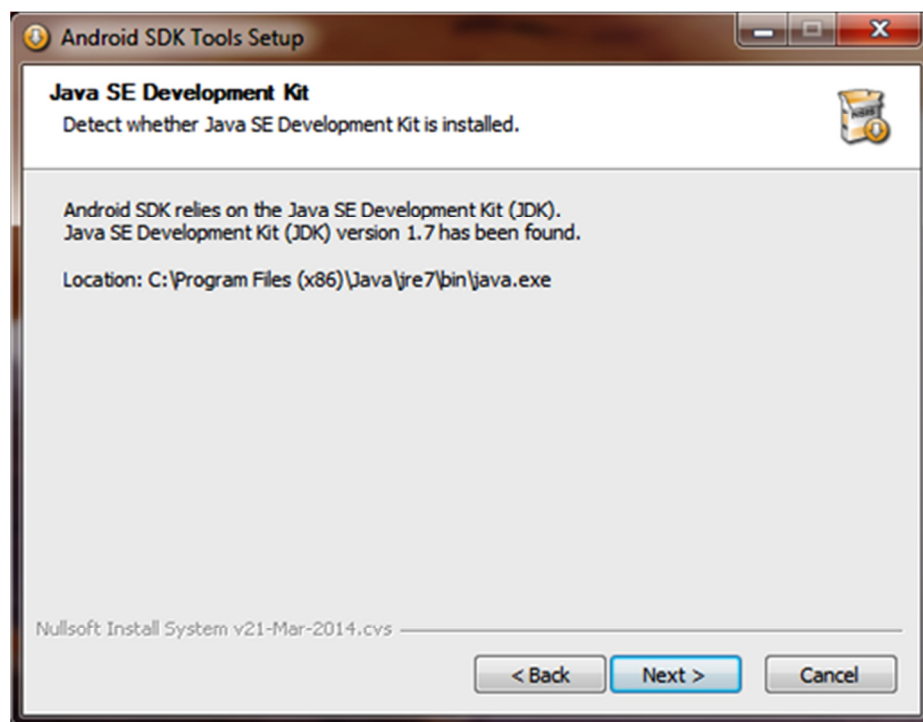
Αρχικά όταν κάνουμε διπλό κλικ στην εφαρμογή, βγαίνει μήνυμα προειδοποίησης επαλήθευσης του εκδότη. Εκεί πατάμε εκτέλεση.

Στη συνέχεια ξεκινά η εγκατάσταση των Android SDK Tools.



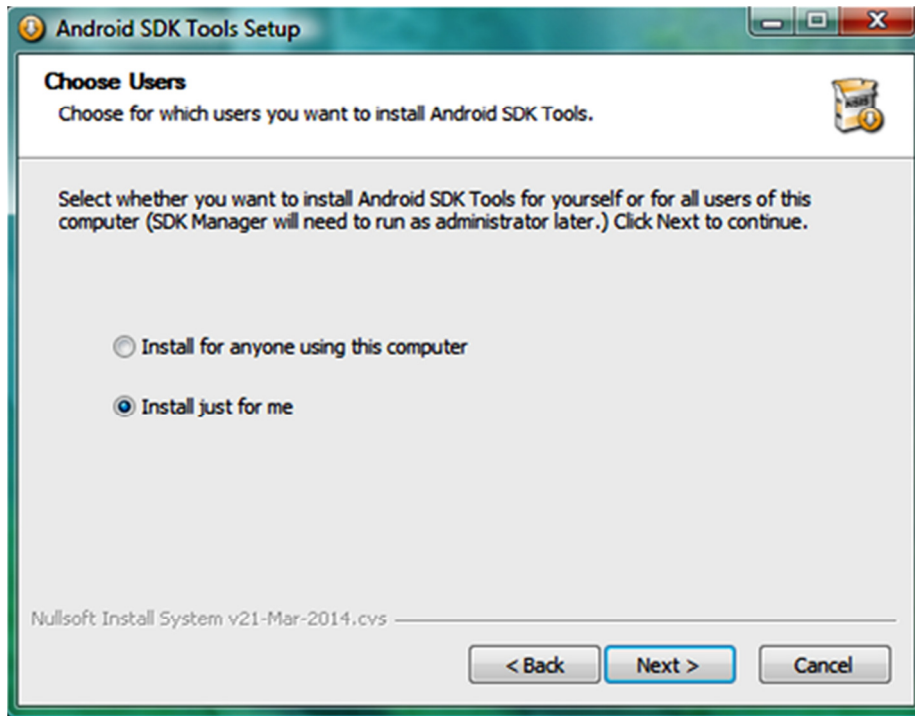
Εικόνα 24: Android SDK Tools Setup Wizard

Γίνεται έλεγχος αν υπάρχει εγκαταστημένο Java SE Development Kit (JDK). Ο έλεγχος βρίσκει ότι έχουμε εγκατεστημένη την έκδοση 1.7. Πατάμε **Next**.

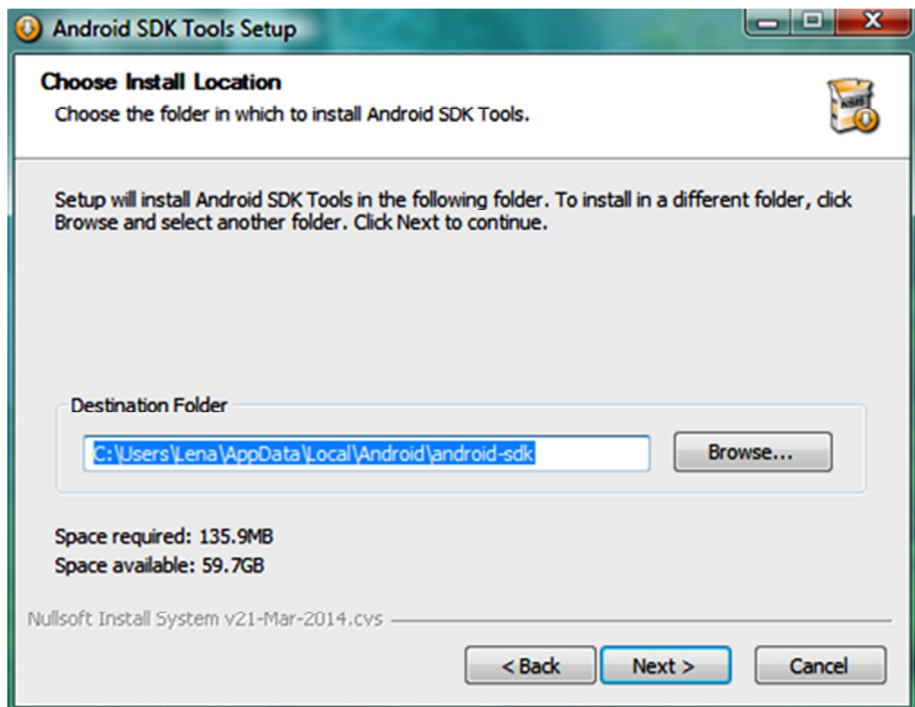


Εικόνα 25: Επιτυχής εύρεση εγκατεστημένης έκδοσης Java

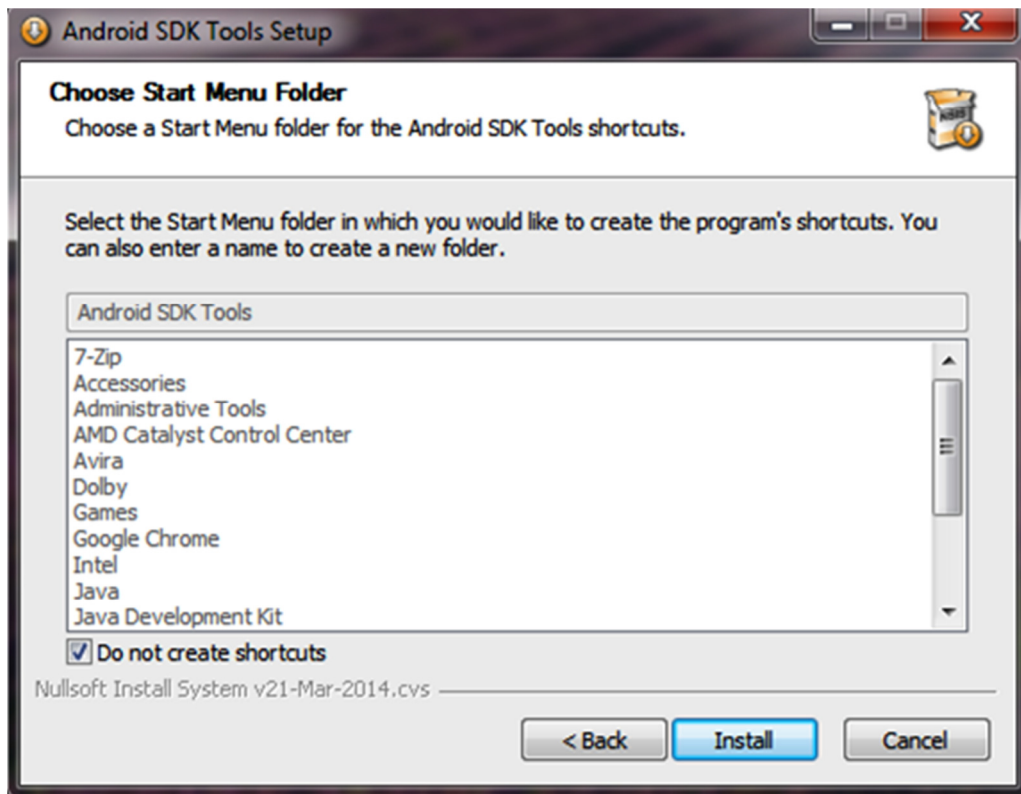
Διαλέγουμε την επιλογή μόνο για μας και πατάμε Next.



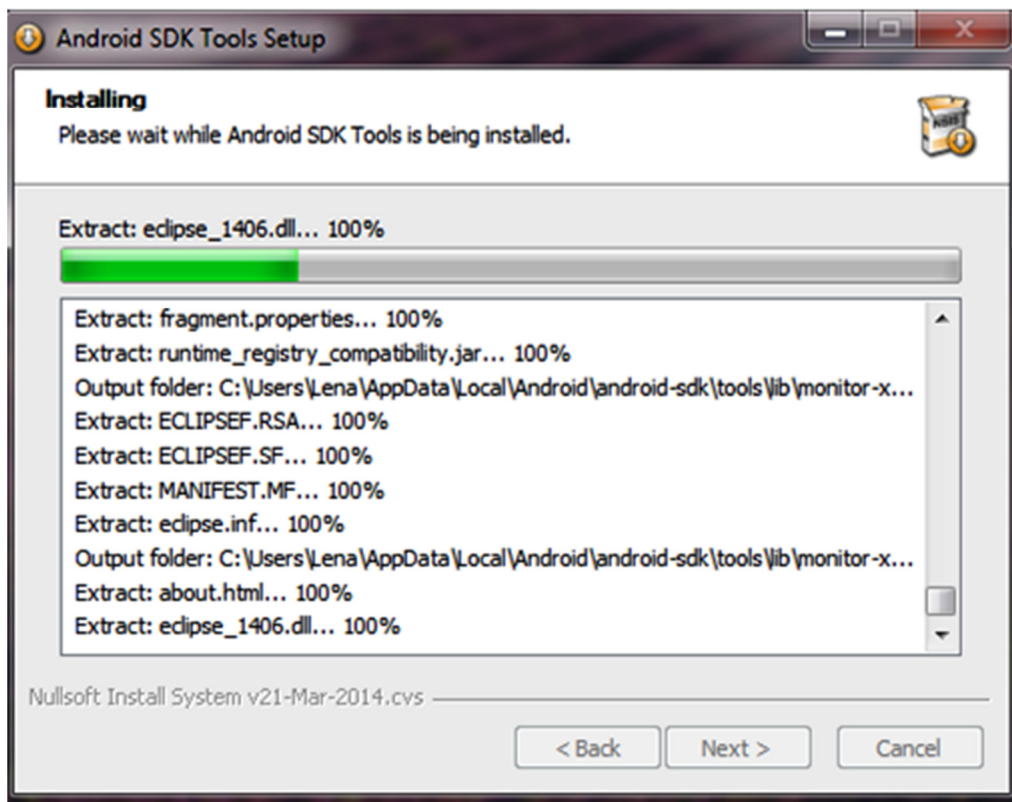
Εικόνα 26: Επιλογή χρήστη



Εικόνα 27: Επιλογή φακέλου προορισμού των αρχείων



Εικόνα 28: Επιλογή Start Menu Folder



Εικόνα 29: Εξαγωγή αρχείων...

Όταν τελειώσει η εγκατάσταση πατάμε **Next**. Στην επόμενη οθόνη το σύστημα μας ειδοποιεί ότι η εγκατάσταση έχει ολοκληρωθεί και δίνει τη δυνατότητα να ξεκινήσουμε το SDK Manager άμεσα. Όμως, επειδή εμείς θα δουλέψουμε με το Eclipse δεν επιλέγουμε το κουτάκι, προκειμένου να κάνουμε την εγκατάσταση των εργαλείων μέσα από το Eclipse. Τέλος πατάμε **Finish**.



Εικόνα 30: Ολοκλήρωση της εγκατάστασης των SDK Tools

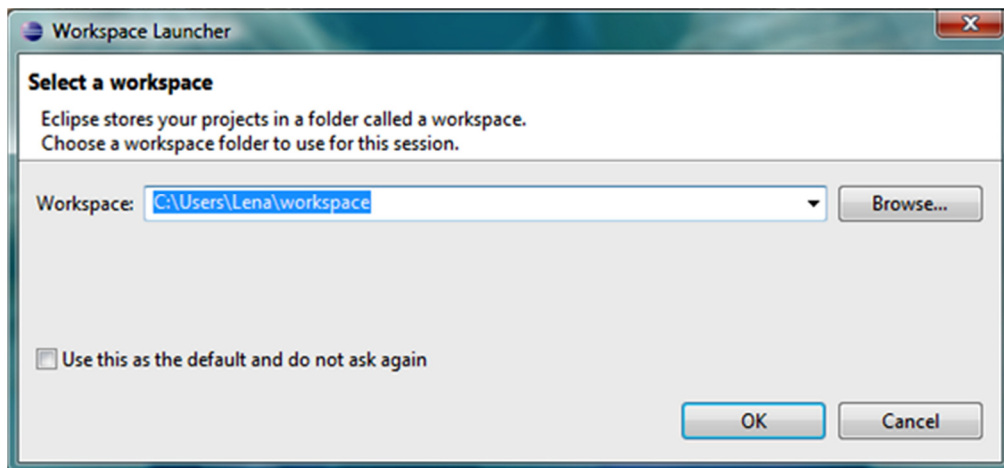
3.5. Οδηγός χρήσης Eclipse

Μέσα από το φάκελο του Eclipse που έχει δημιουργηθεί και πατάμε στο **eclipse.exe**.



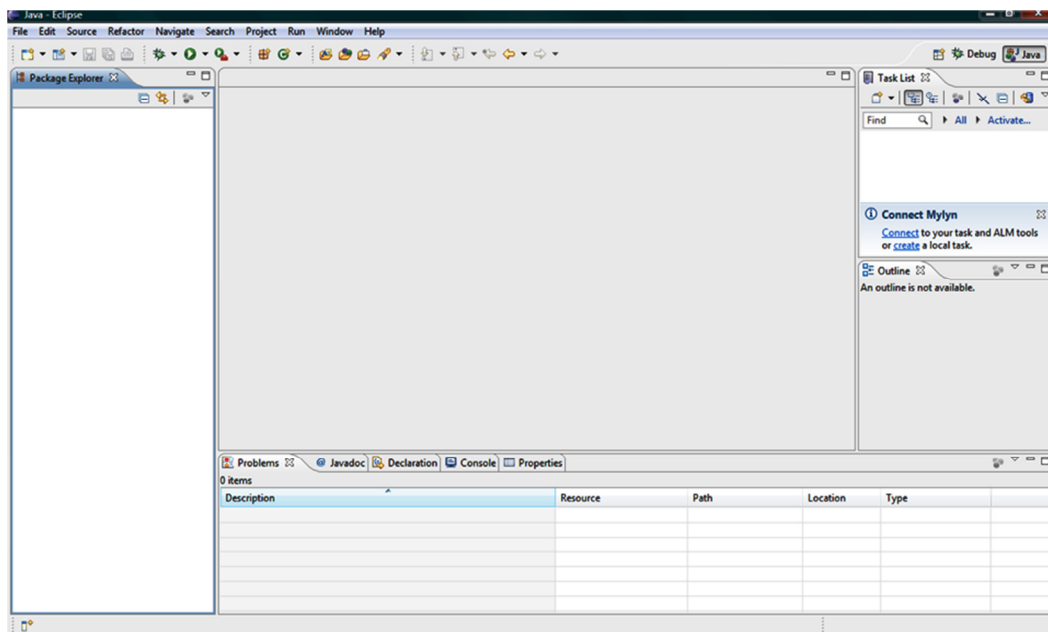
Εικόνα 31: Έναρξη του Eclipse

Με το που θα ξεκινήσουμε το Eclipse, το πρόγραμμα θα μας ζητήσει να καθορίσουμε σε ποιο φάκελο θα αποθηκεύονται τα project μας.



Εικόνα 32: Καθορισμός της διαδρομής του φακέλου workspace

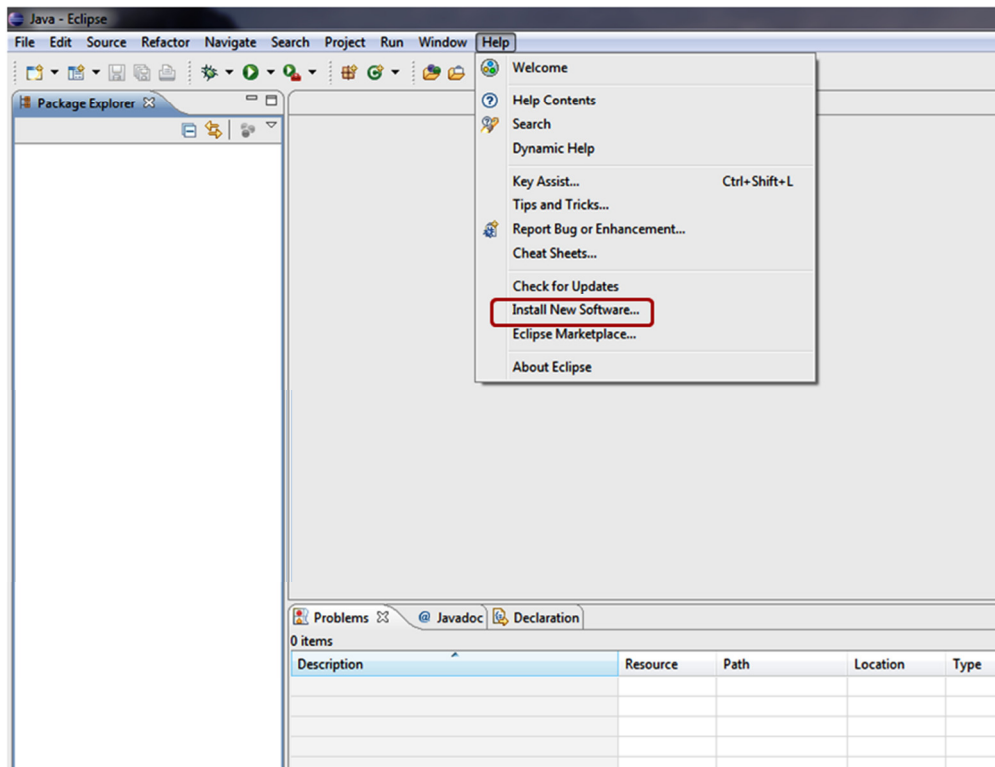
Έπειτα μπαίνουμε στο περιβάλλον του Eclipse.



Εικόνα 33: Επιφάνεια εργασίας του προγράμματος Eclipse

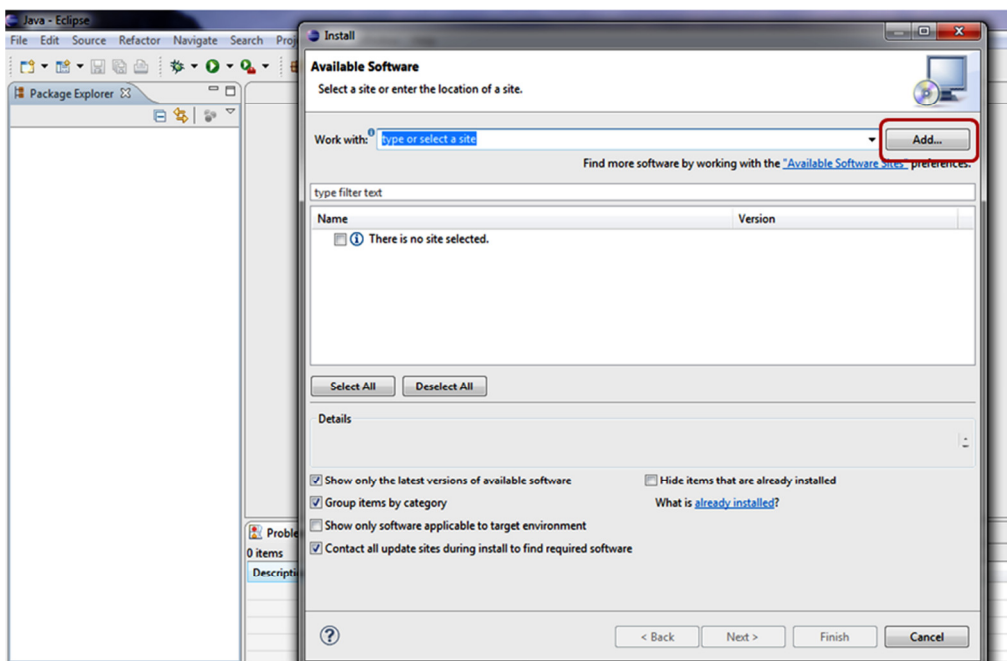
3.6. Κατεβάζοντας το Eclipse Plugin

Από τη γραμμή εργαλείων επιλέγουμε **Help** → **Install New Software**.



Εικόνα 34: Help → Install New Software

Επιλέγουμε **Add** πάνω δεξιά.



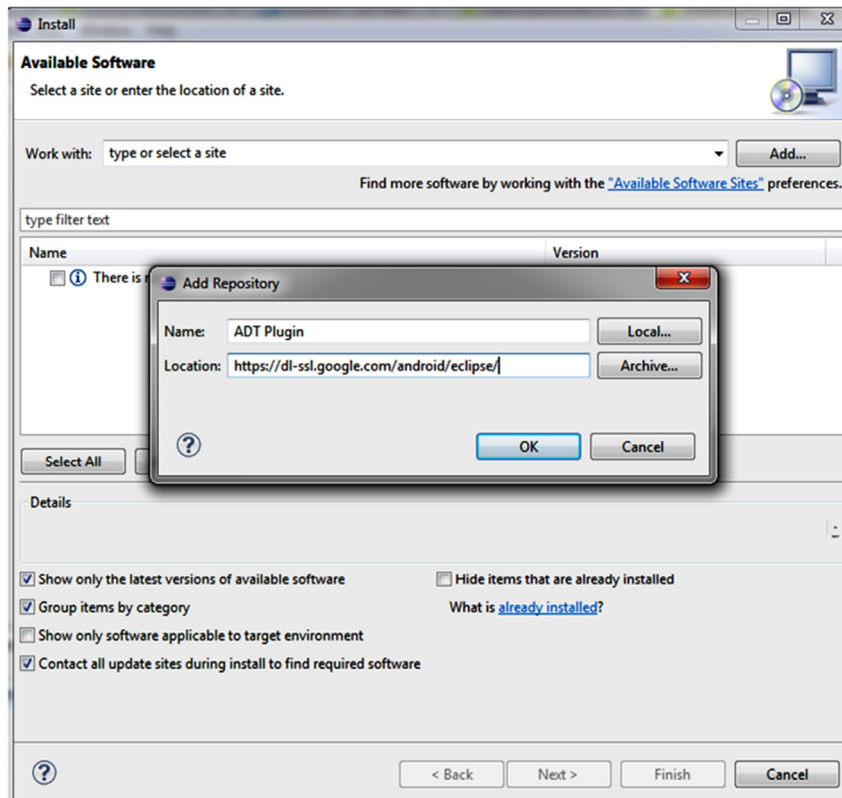
Εικόνα 35: Add Software

Στο διάλογο που ανοίγει (Add Repository⁴) γράφουμε:

Name: *ADT Plugin*

⁴ Repository: Τοποθεσία αποθήκευσης απ' όπου τα πακέτα λογισμικού (software packages) ανακτώνται και εγκαθιστούνται στον υπολογιστή. (Wikipedeia, 2014)

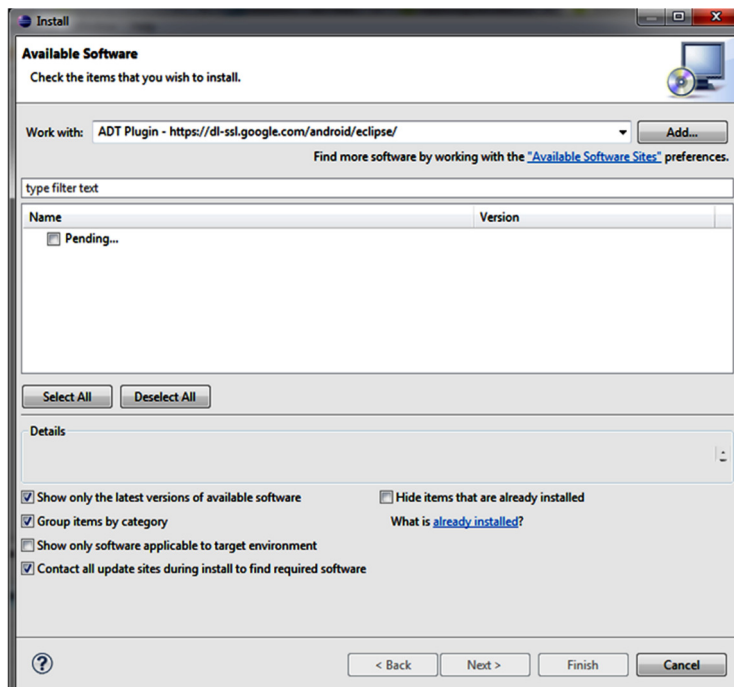
Location: <https://dl-ssl.google.com/android/eclipse/>



Εικόνα 36: Εισαγωγή Name και Location του ιστότοπου

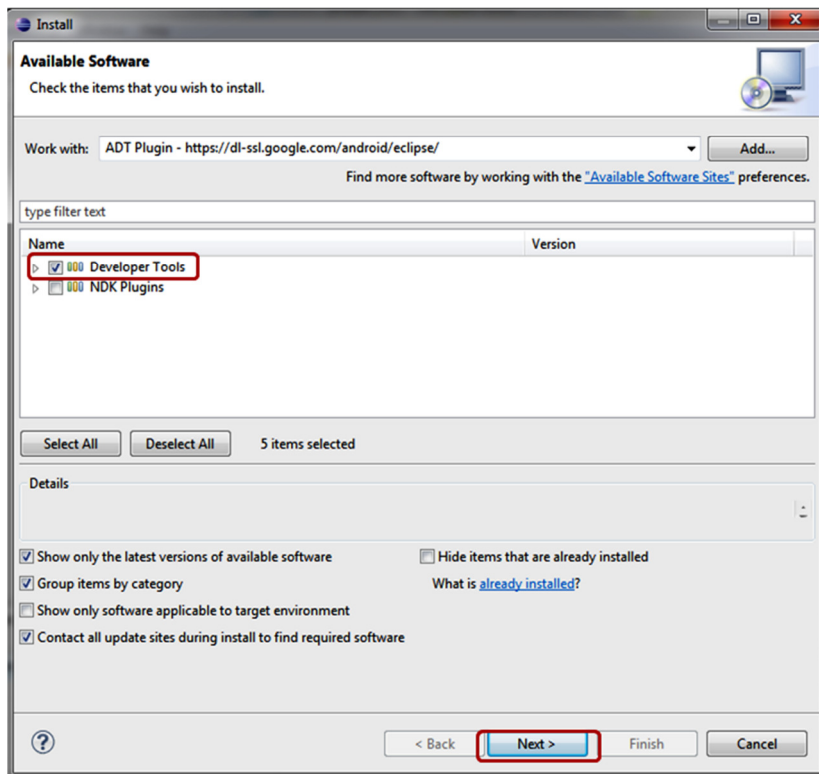
Από αυτή τη διεύθυνση το Eclipse θα μπορέσει να κατεβάσει και να ενσωματώσει το Android Plug-in.

Πατάμε **OK**.

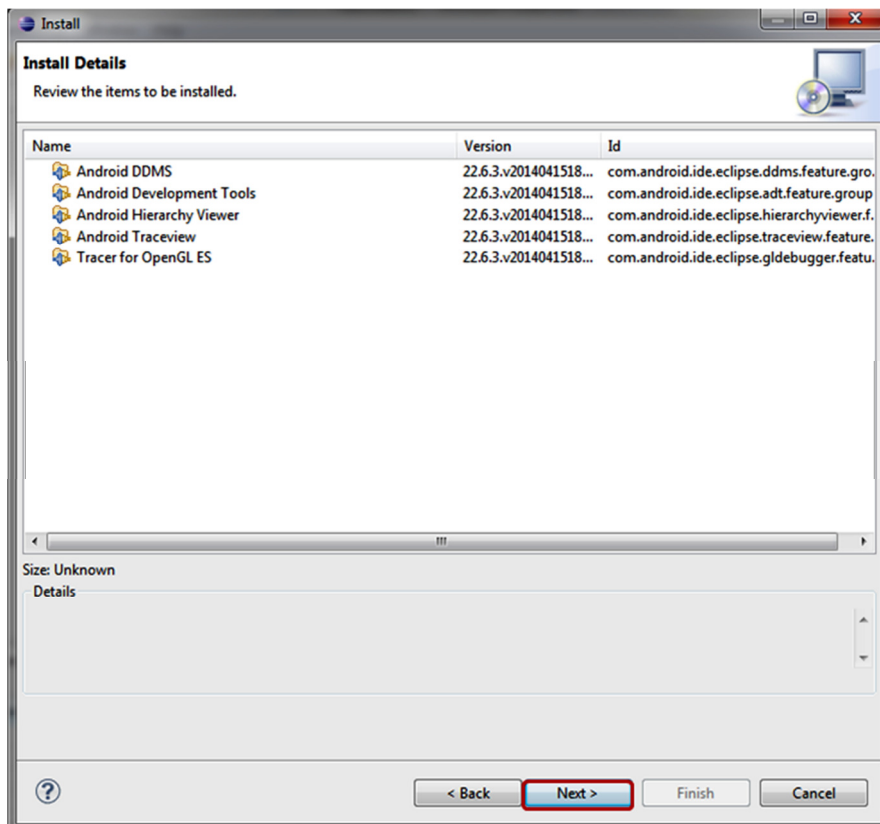


Εικόνα 37: Pending

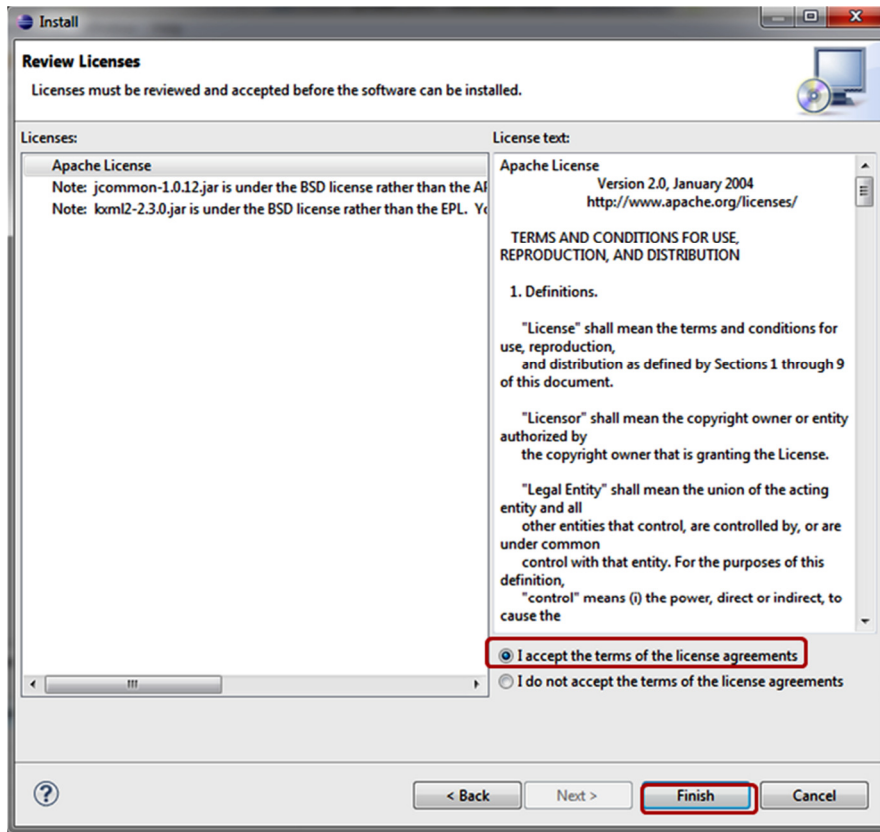
Στο παράθυρο που ανοίγει κι αφού περιμένουμε λιγάκι να «φορτώσει» τα δεδομένα, τσεκάρουμε το **Developer Tools** και πατάμε **Next**.



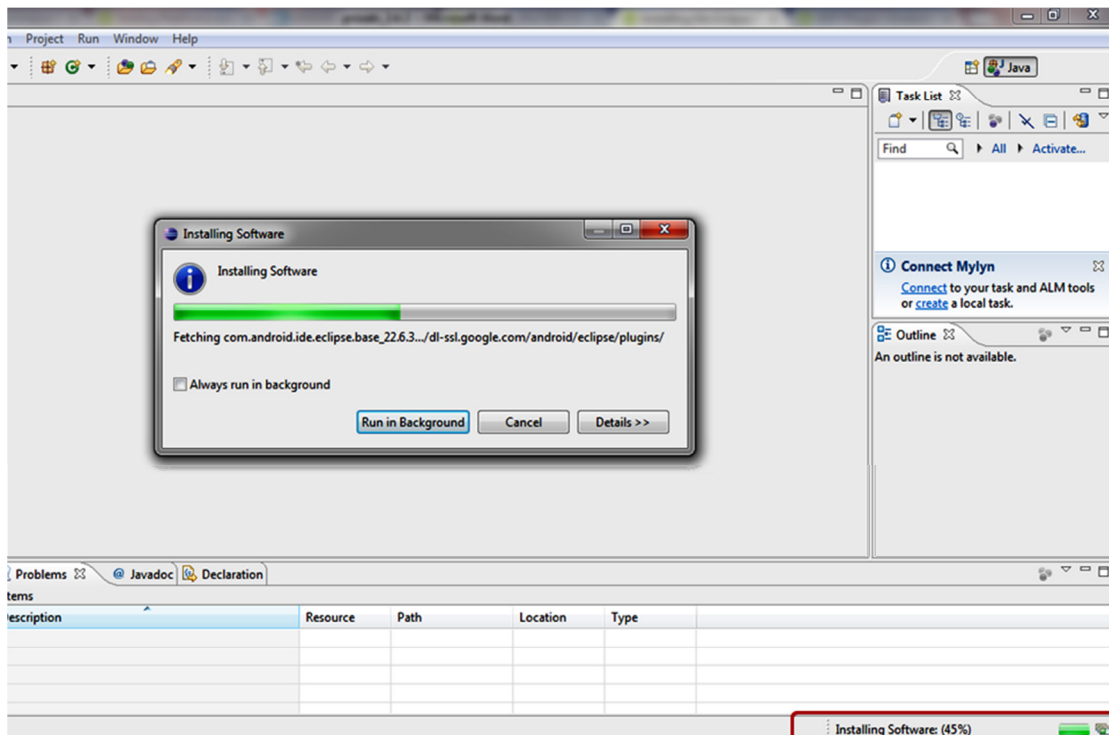
Εικόνα 38: Διαθέσιμα Developer Tools



Εικόνα 39: Ανασκόπηση των εργαλείων που θα κατεβούν

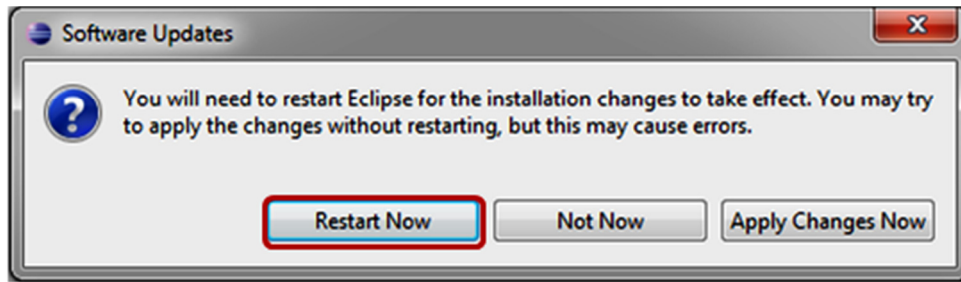


Εικόνα 40: Αποδοχή των όρων εγκατάστασης



Εικόνα 41: Installing Software...

Στην προειδοποίηση (**Security Warning**) που ακολουθεί ότι το software που εγκαθιστούμε περιέχει άγνωστα στοιχεία πατάμε **OK**.



Εικόνα 42: Ειδοποίηση για επανεκκίνηση του Eclipse

Στο τέλος κάνουμε επανεκκίνηση του Eclipse για να δεχτεί όλες τις αναβαθμίσεις και αλλαγές που κάναμε.

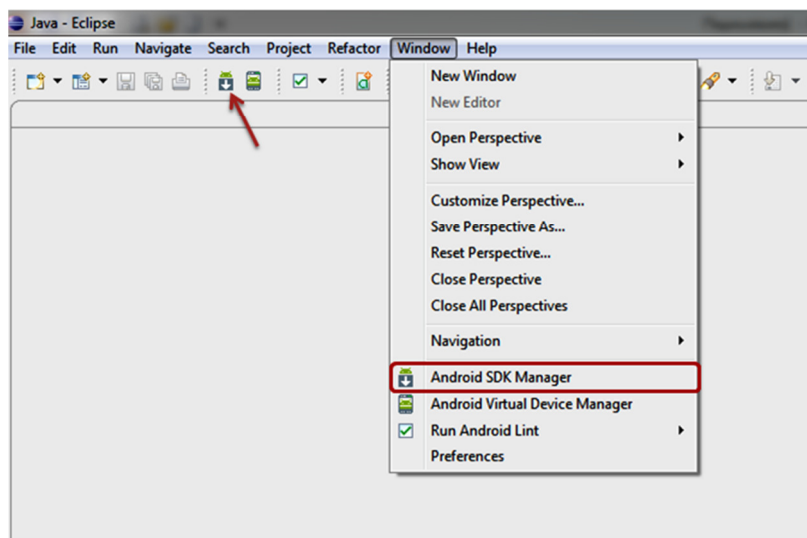
3.7. Δουλεύοντας με τον SDK Manager

Όταν ξανανοίγουμε την εφαρμογή του Eclipse μας προειδοποιεί ότι πρέπει να εγκαταστήσουμε τα SDK tools packages μέσα από τον SDK Manager και μας δίνει την δυνατότητα να τον ανοίξουμε άμεσα. Το Android SDK χωρίζει τα εργαλεία (tools), τις πλατφόρμες (platforms) και διάφορα άλλα στοιχεία (components) σε πακέτα (packages) τα οποία μπορούμε να κατεβάσουμε χρησιμοποιώντας τον Android SDK Manager. Το αρχικό SDK package που έχουμε ήδη κατεβάσει περιέχει μόνο τα SDK tools. Για να αναπτύξουμε μία Android εφαρμογή, χρειάζεται να κατεβάσουμε τουλάχιστον μία Android platform και τα τελευταία SDK Platform – Tools.

Το SDK Manager δείχνει όλα τα SDK packages που είναι διαθέσιμα στον χρήστη για κατέβαση στο Android SDK. Ως ελάχιστες ρυθμίσεις για το SDK προτείνονται να υπάρχουν:

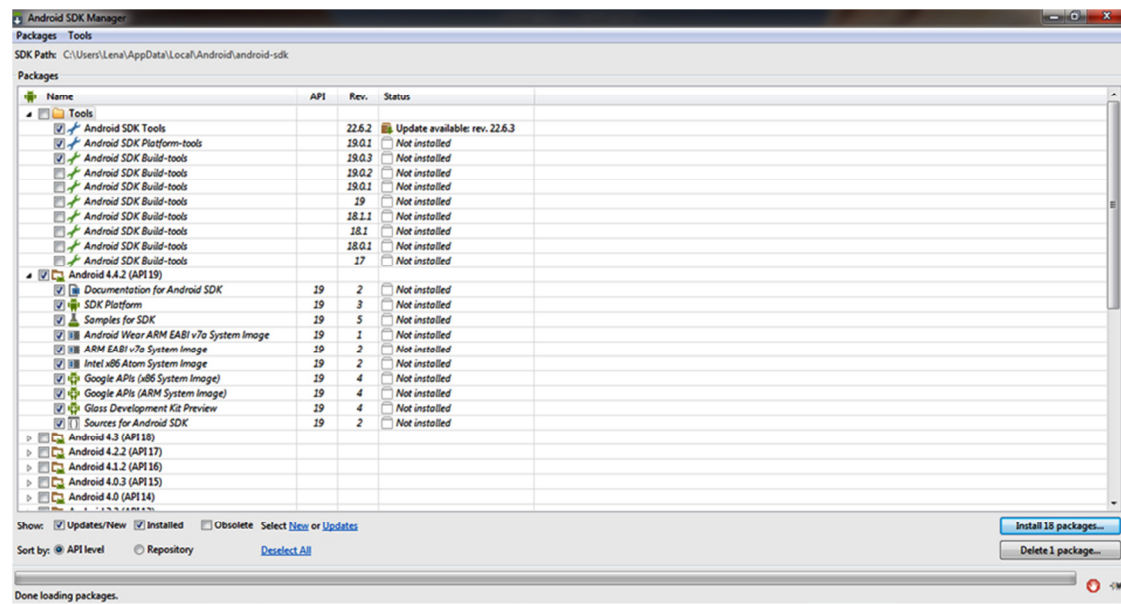
- Τα τελευταία Tools packages (βρίσκονται στο φάκελο Tools)
- Η τελευταία έκδοση Android (είναι ο πρώτος φάκελος Android)
- Η Android Support Library (βρίσκουμε το φάκελο Extras και τσεκάρουμε την επιλογή Android Support Library). (Android Developers: Adding platforms and packages)

Έτσι πάμε στον SDK Manager, τον οποίο ανοίγουμε είτε από την γραμμή εργαλείων, είναι από το μενού **Window** → **Android SDK Manager** (όπως φαίνεται στην εικόνα) και τσεκάρουμε τα παραπάνω στοιχεία.



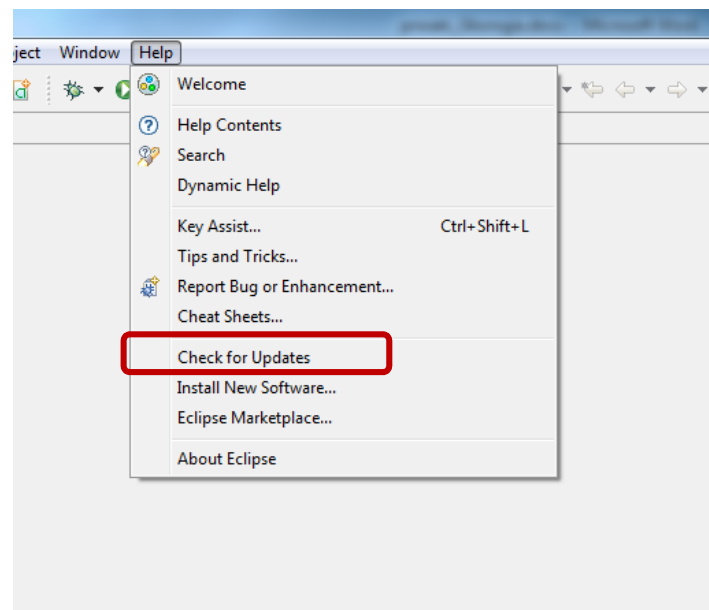
Εικόνα 43: Εγκατάσταση του Android SDK Manager (Windows → Android SDK Manager)

Έπειτα πατάμε **Install**.

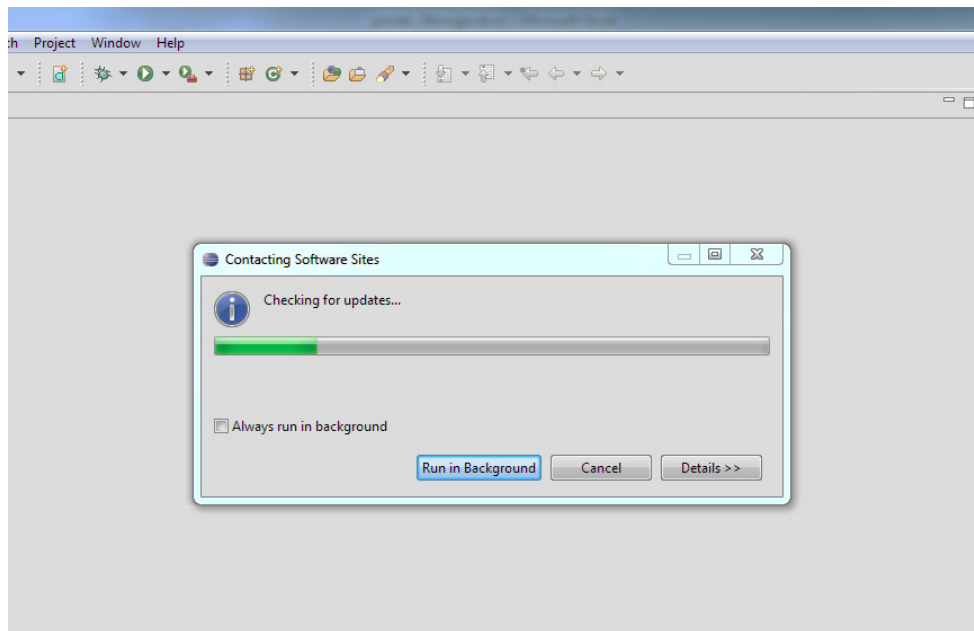


Εικόνα 44: SDK Packages

Μόλις ο SDK Manager κατεβάσει όσα packages έχουμε τσεκάρει, μπορεί να μας ζητηθεί να κάνουμε έλεγχο για updates στο πρόγραμμα του Eclipse. Τότε πάμε στο μενού **Help** → **Check for Updates**.



Εικόνα 45: Έλεγχος για ενημερώσεις στο Eclipse



Εικόνα 46: Έλεγχος για updates

ΚΕΦΑΛΑΙΟ 4

Γνωριμία με τα ξεχωριστά στοιχεία προγραμματισμού σε Android και δημιουργία μιας νέας εφαρμογής

Ο προγραμματισμός σε Android χρησιμοποιεί κάποια στοιχεία που τον κάνουν να διαφέρει. Τα στοιχεία αυτά της αρχιτεκτονικής του αποτελούν κύρια συστατικά στη δημιουργία μιας νέας εφαρμογής, γι' αυτό και είναι καλό να τα γνωρίζουμε πριν προχωρήσουμε στην εφαρμογή μας.

4.1. Αρχιτεκτονική των εφαρμογών Android

Ο προγραμματισμός σε Android περιέχει κάποια στοιχεία που τον ξεχωρίζουν από τον απλό προγραμματισμό σε Java. Αυτά τα στοιχεία είναι απαραίτητα για την δημιουργία μιας ολοκληρωμένης εφαρμογής σε Android.

4.1.1. Activity (δραστηριότητα)

Η Activity παριστά μία εργασία που μπορεί να κάνει η εφαρμογή, είναι το γραφικό περιβάλλον της εφαρμογής που βλέπει ο χρήστης. Κάθε Activity αντιστοιχεί και σε διαφορετική οθόνη. Το User Interface του χρήστη είναι μία Activity. Η αρχική οθόνη που βλέπει ο χρήστης είναι συνήθως κι η κύρια δραστηριότητα (main activity) και δημιουργείται μαζί με τη δημιουργία της εφαρμογής.

Μία εφαρμογή αποτελείται, συνήθως, από διάφορες activities. Μεταξύ των activities υπάρχει δυνατότητα εναλλαγής με τη χρήση της εντολής Intent (βλέπε παρακάτω). Δηλαδή, ο χρήστης μπορεί να περάσει από μία activity σε μία άλλη, χωρίς αυτό να σημαίνει ότι η προηγούμενη activity, εξαφανίζεται. Η προηγούμενη activity συνεχίζει να λειτουργεί στο παρασκήνιο. Έτσι αν ο χρήστης αποφασίσει να γυρίσει πίσω στην προηγούμενη οθόνη, η οθόνη ξαναγυρνά στο προσκήνιο και γίνεται και πάλι το user interface του χρήστη.

4.1.2. Service (Υπηρεσία)

Η Service (Υπηρεσία) είναι ένα κομμάτι της εφαρμογής που τρέχει στο παρασκήνιο, δεν έχει δική του οθόνη διεπαφής (User Interface – UI). Αντιπροσωπεύει είτε την ευχέρεια της εφαρμογής να ζητήσει από το σύστημα να εκτελέσει μια λειτουργία στο παρασκήνιο (ακόμη κι όταν ο χρήστης δεν αλληλεπιδρά άμεσα με την εφαρμογή), είτε την ευχέρεια της εφαρμογής να εκθέσει μερικές από τις λειτουργίες της σε άλλες εφαρμογές. (Android Developers, Service)

Για παράδειγμα μία υπηρεσία μπορεί να παίζει μουσική στο παρασκήνιο ενώ ο χρήστης βρίσκεται σε κάποια άλλη εφαρμογή ή να κατεβάζει δεδομένα από το Internet χωρίς να επηρεάζει την αλληλεπίδραση του χρήστη με μια Activity. (Tutorials Point)

4.1.3. Content Provider (Πάροχος περιεχομένου)

Ο πάροχος περιεχομένου διαχειρίζεται την πρόσβαση σε ένα διαρθρωμένο σύνολο δεδομένων. Ενσωματώνει τα δεδομένα, και παρέχει μηχανισμούς για τον καθορισμό της ασφάλειας των δεδομένων. Ο πάροχος περιεχομένου είναι η τυπική διασύνδεση που συνδέει τα δεδομένα μιας διαδικασίας με τον κώδικα που τρέχει σε μια άλλη διαδικασία.

Ο πάροχος περιεχομένου διαμοιράζει δεδομένα από μία εφαρμογή σε άλλες, ύστερα από αίτημα. Μπορεί να χρησιμοποιεί διαφορετικούς τρόπους για να αποθηκεύσει τα δεδομένα του. Τα δεδομένα μπορούν να αποθηκευτούν σε μια βάση δεδομένων, σε αρχεία ή ακόμη και μέσω ενός δικτύου, είτε αυτό είναι το Internet, είτε ένα τοπικό δίκτυο.

Κάθε Android εφαρμογή τρέχει τις δικές της διεργασίες, με τα δικά της δικαιώματα, τα οποία κρατούν τα δεδομένα της εφαρμογής κρυφά από τις άλλες εφαρμογές. Όμως, κάποιες φορές είναι υποχρεωμένη να μοιράζεται τα δεδομένα αυτά με τις άλλες εφαρμογές. Για το σκοπό αυτό, του διαμοιρασμού, είναι χρήσιμος ένας πάροχος περιεχομένου. (Tutorialspoint, Android Content Providers)

4.1.4. Broadcast Receiver (Δέκτης εκπομπών)

Ένας δέκτης εκπομπών απλά ανταποκρίνεται στα εκπεμπόμενα μηνύματα από άλλες εφαρμογές ή από το ίδιο το σύστημα. Τα μηνύματα αυτά ονομάζονται συχνά γεγονότα (events) ή προθέσεις (Intents). Για παράδειγμα, οι εφαρμογές μπορούν να ξεκινήσουν μία εκπομπή προκειμένου να γνωστοποιήσουν στις άλλες εφαρμογές ότι κάποια δεδομένα έχουν ληφθεί στη συσκευή κι είναι διαθέσιμα σε αυτές να τα χρησιμοποιήσουν. Αυτό είναι ένας δέκτης εκπομπών ο οποίος θα παρακολουθήσει αυτήν την επικοινωνία και θα κινήσει τις απαραίτητες διαδικασίες. (Tutorialspoint, Android Broadcast Receivers)

4.1.5. Intent (Πρόθεση)

Μια android πρόθεση είναι ένα αντικείμενο που μεταφέρει μία πρόθεση, δηλαδή ένα μήνυμα από ένα στοιχείο σε ένα άλλο, δια μέσου της εφαρμογής ή έξω από την εφαρμογή. Οι προθέσεις μπορούν να μεταφέρουν μηνύματα ανάμεσα στις τρεις βασικές συνιστώσες της εφαρμογής: Activities, services, broadcast receivers.

Η ίδια η πρόθεση ως έννοια, ένα αντικείμενο πρόθεσης, είναι μία παθητική δομή δεδομένων που περιέχουν μία αφηρημένη περιγραφή μιας λειτουργίας, η οποία πρέπει να εκτελεστεί. (Tutorialspoint, Android Intents and Filters)

4.2. Καθορισμός χαρακτηριστικών της Android εφαρμογής

Αφού κάναμε μία σύντομη αναφορά στα συστατικά της αρχιτεκτονικής του Android που το κάνουν ξεχωριστό, προχωρούμε στην υλοποίηση της εφαρμογής μας.

Η εφαρμογή που έχουμε σκοπό να αναπτύξουμε είναι ένα παιχνίδι γνώσεων, ένα παιχνίδι ερωτήσεων πολλαπλής επιλογής. Θα έχει σχέση με τις γνώσεις του παίκτη σχετικά με την Ευρώπη.

Για παράδειγμα:

Σε ποια χώρα βρίσκεται το Διεθνές Δικαστήριο της Χάγης;

- i. Στην Ελβετία
- ii. Στην Γαλλία
- iii. Στην Ολλανδία**
- iv. Στην Δανία

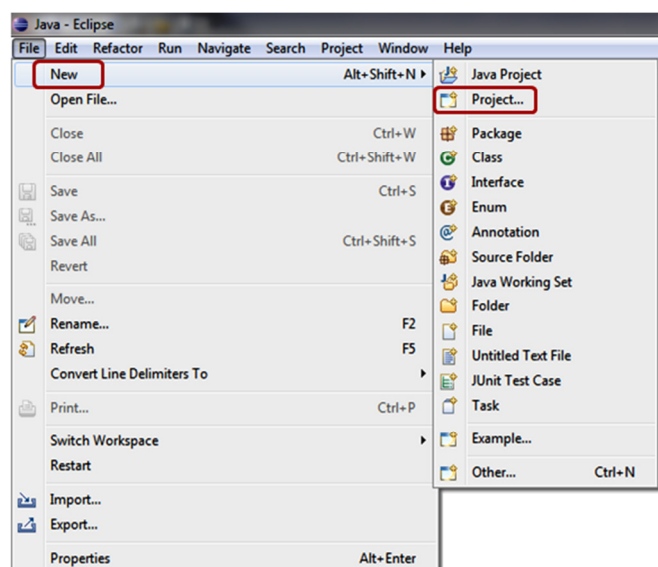
Μόνο μία είναι η σωστή απάντηση. Υπάρχουν πιο δύσκολες και πιο εύκολες ερωτήσεις. Ο κάθε παίκτης έχει ένα συγκεκριμένο αριθμό ερωτήσεων να απαντήσει κι ανάλογα τον αριθμό των σωστών απαντήσεων μαζεύει και κάποιο αριθμό πόντων. Ανάλογα με τους πόντους που μαζεύει κατατάσσεται και στον πίνακα των σκόρερ.

Για την σωστή λειτουργία του παιχνιδιού χρειαζόμαστε:

- Μία οθόνη έναρξης (EuropeActivity)
- Μία οθόνη επιλογών (QuizMenuActivity).
- Μία οθόνη των σκορ(QuizScoresActivity).
- Μια οθόνη με οδηγίες για το παιχνίδι προς τον χρήστη (QuizHelpActivity).
- Μια οθόνη ρυθμίσεων (QuizSettingsActivity)
- Την κύρια οθόνη, όπου θα εμφανίζονται οι ερωτήσεις του παιχνιδιού (QuizGameActivity).

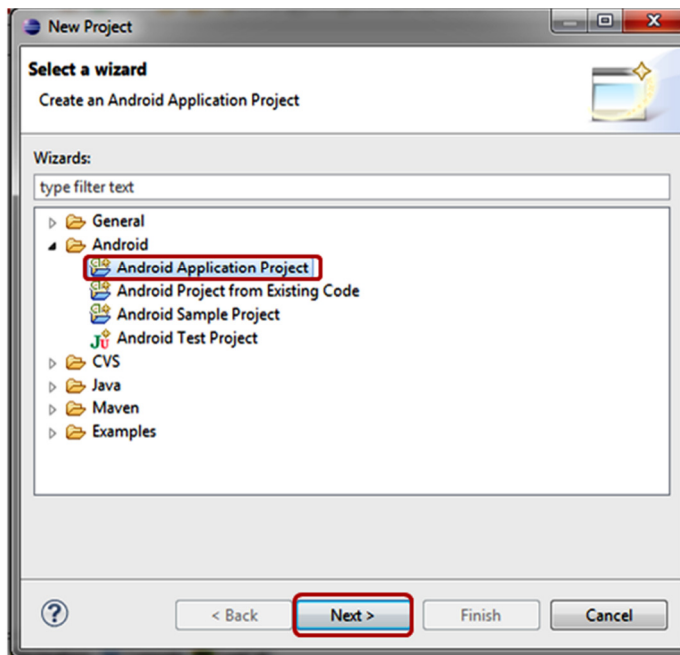
4.3. Δημιουργία του Android προγράμματος με τη βοήθεια του Eclipse

Ανοίγουμε το Eclipse. Πηγαίνουμε στη γραμμή εργαλείων και πατάμε **File →New →Project**



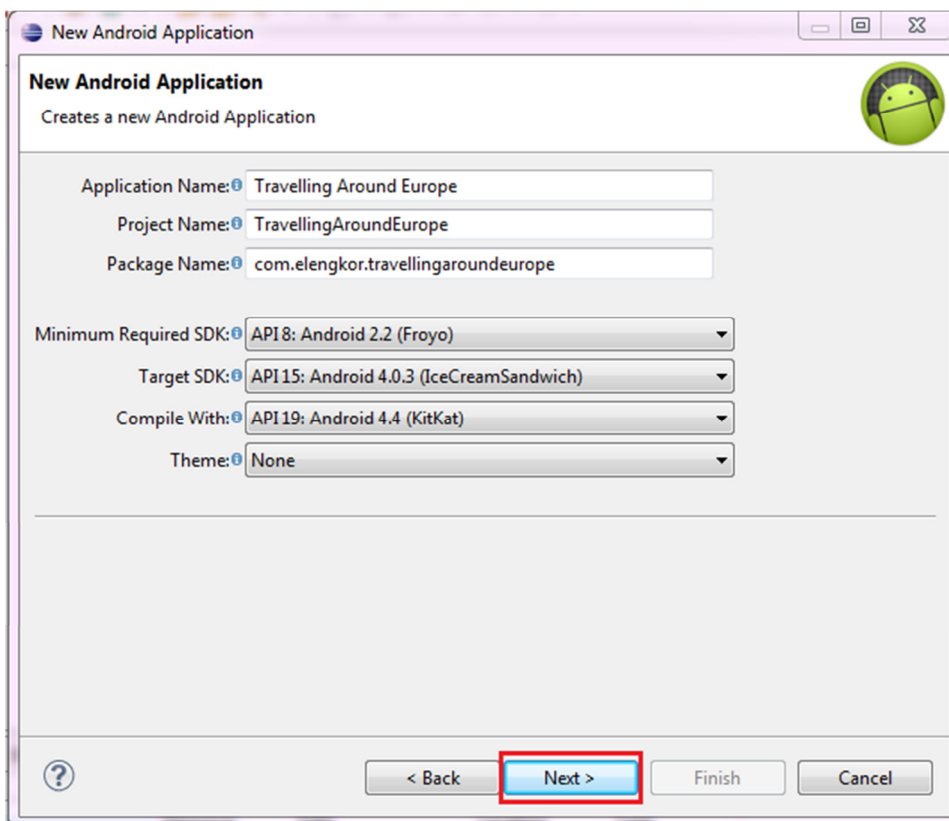
Εικόνα 47: Δημιουργία νέου project

Έπειτα πηγαίνουμε στον φάκελο **Android** και διαλέγουμε **Android Application Project** και **Next**.



Εικόνα 48: Επιλογή δημιουργίας Android Application Project

Στην επόμενη οθόνη δηλώνουμε τα δεδομένα της εφαρμογής που θα εμφανίζεται στην android συσκευή.



Εικόνα 49: Ονοματοδότηση project κι επιλογή SDK

Στο **Application Name** δίνουμε το όνομα της εφαρμογής που θα δημιουργήσουμε. Στο **Project Name** σχηματίζεται το όνομα του project που θα δουλέψουμε, είναι το όνομα της εφαρμογής μας με μία λέξη.

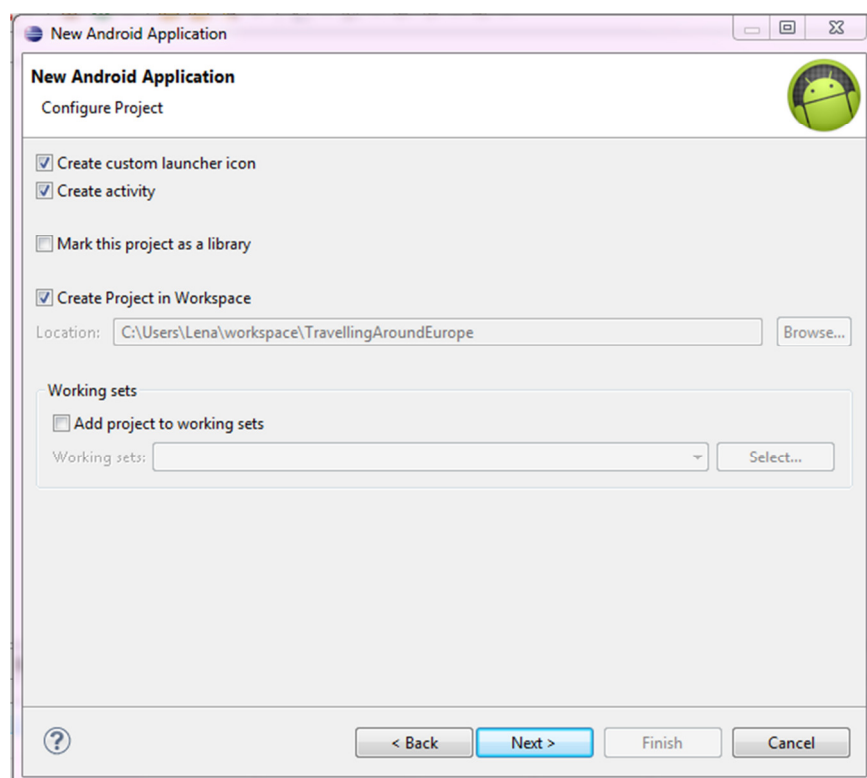
Στο **Package Name** έχουμε το όνομα του package μας, το οποίο είναι μοναδικό για κάθε εφαρμογή που υπάρχει μέσα στο Android σύστημα. Γι' αυτό το λόγο πολλοί χρησιμοποιούν αντίστροφα το όνομα της εταιρείας ή του οργανισμού τους. Χρησιμοποιούμε στα δύο πρώτα μέρη ότι εμείς θέλουμε και το τρίτο μέρος είναι το όνομα του project μας.

Στην επιλογή **Minimum Required SDK** αναφερόμαστε στην παλαιότερη έκδοση με την οποία η εφαρμογή μας θα είναι συμβατή.

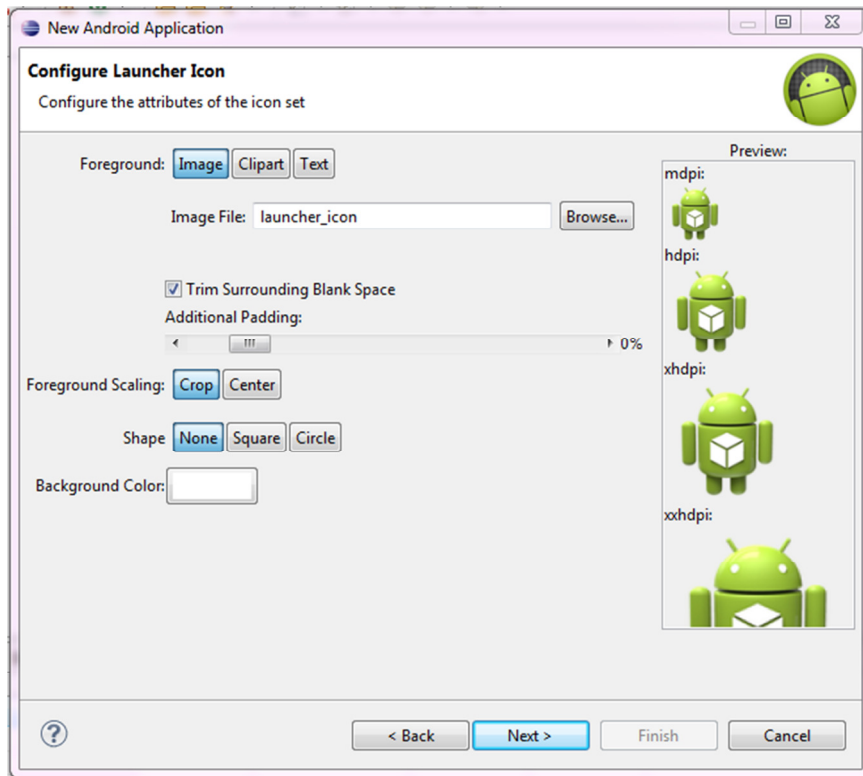
Στο **Target SDK** επιλέγουμε το API στο οποίο θα τρέξει η εφαρμογή μας.

Τέλος στο **Compile With** δηλώνουμε συνήθως την τελευταία έκδοση του Android με την οποία θα είναι συμβατή η εφαρμογή.

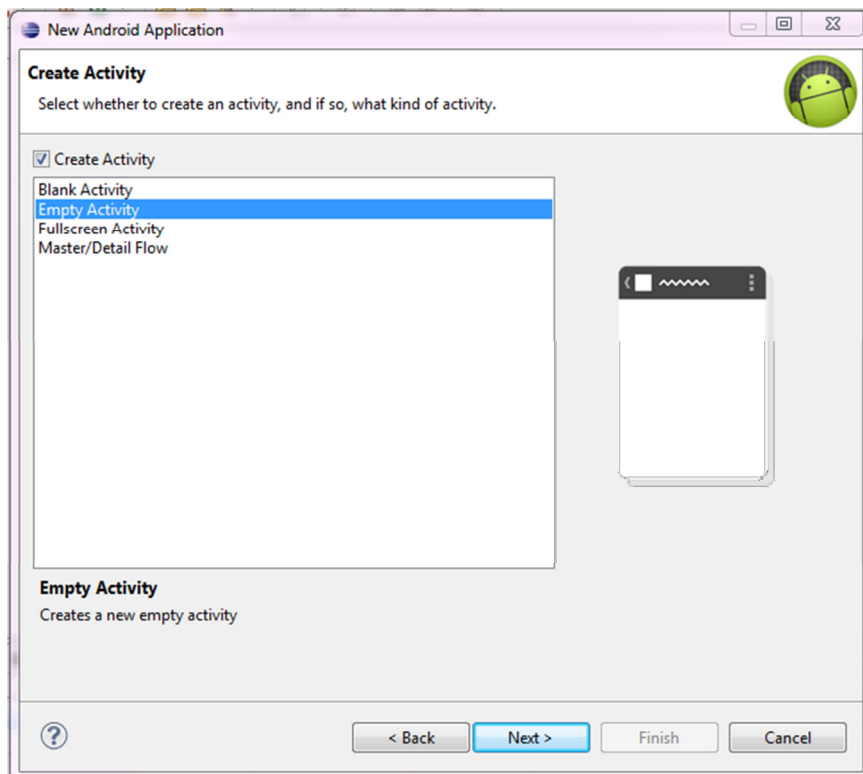
Σε κάθε βήμα το ίδιο το πρόγραμμα δίνει τις κατάλληλες συμβουλές για την επιλογή που θα κάνει ο κάθε χρήστης – δημιουργός εφαρμογής Android σε Eclipse.



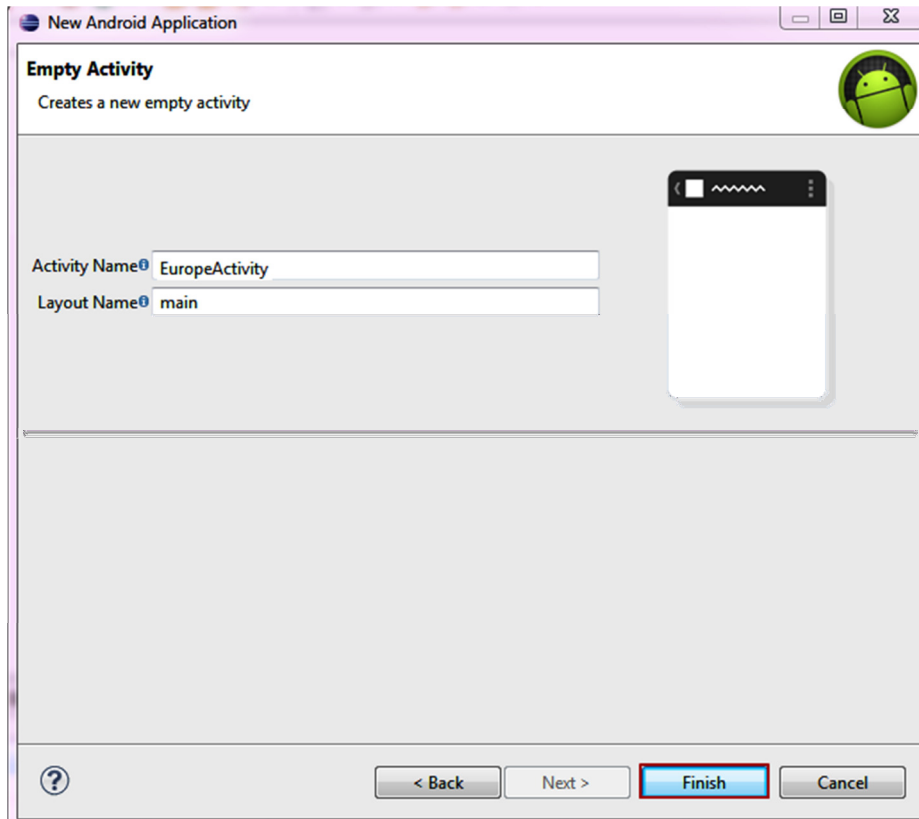
Εικόνα 50: Επιλογή της διαδρομής του φακέλου workspace, της δημιουργίας ή μη της αρχικής δραστηριότητας και επιλογή δημιουργίας εικονιδίου



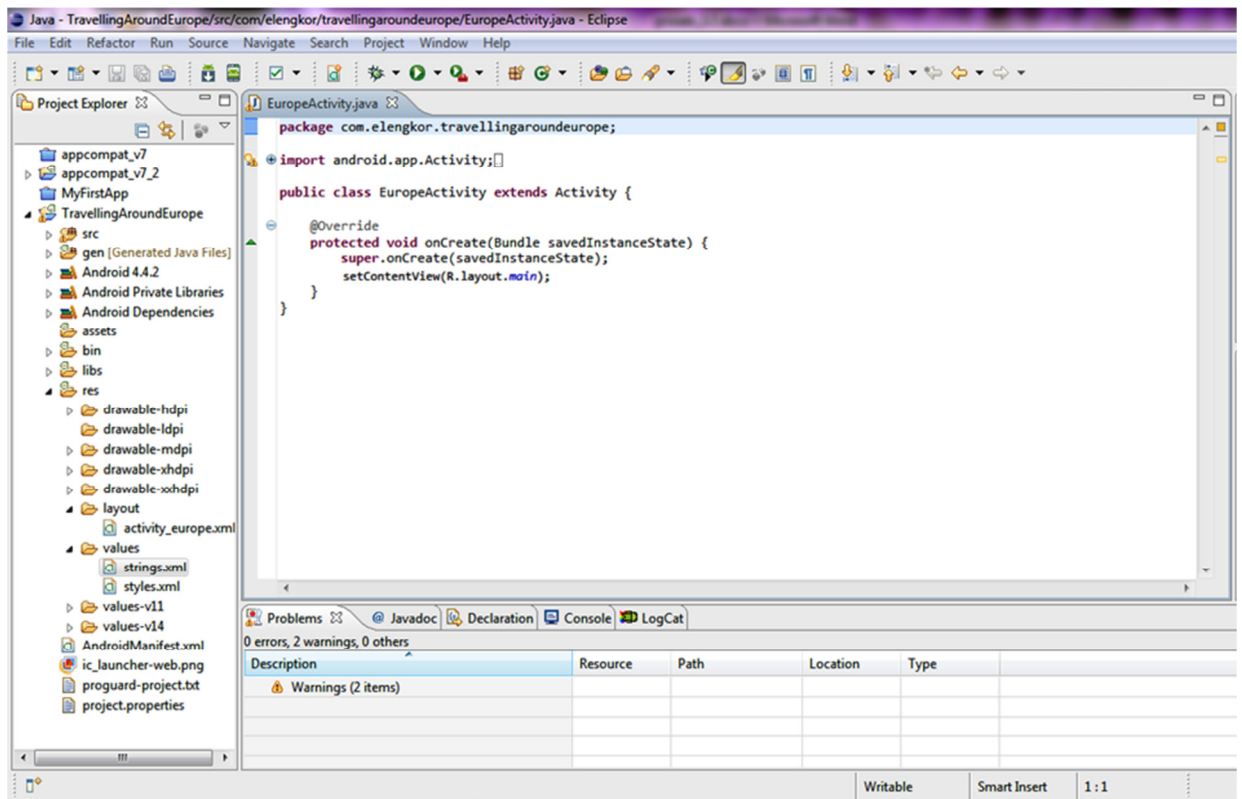
Εικόνα 51: Configure Launcher Icon (μπορούμε να διαλέξουμε το λογότυπο της εφαρμογής μας)



Εικόνα 52: Create Activity (επιλογή του είδους της Activity)



Εικόνα 53: Δήλωση του ονόματος της αρχικής Activity και του Layout που τη συνοδεύει



Εικόνα 54: Η activity που δημιουργείται στην αρχική της μορφή

4.3.1. Προσθήκη πόρων συμβολοσειράς (string) Στο φάκελο TravellingAroundEurope/res/values έχει δημιουργηθεί φάκελος με το όνομα strings.xml, όπου θα αποθηκεύονται όλες οι συμβολοσειρές που θα χρειαστούμε για την εφαρμογή μας.

Για αρχή υπάρχουν μόνο 2 strings με τα ονόματα "app_name" και το "hello". Το "app_name" δηλώνει το όνομα της εφαρμογής και το "hello" είναι κείμενο που εμφανίζεται στην αρχική activity που δημιουργήθηκε με το project, πριν την τροποποίησή της.

Σε κάθε activity θα αντιστοιχεί κι ένα string ως τίτλος οθόνης. Το string αυτό θα δηλώνεται μέσα στο layout που συνοδεύει κάθε activity (βλέπε παρακάτω).

Αλλάζουμε το όνομα του string "app_name" σε «Ταξιδεύοντας στην Ευρώπη» ενώ το "hello" το αφήνουμε κενό (χρησιμοποιείται μετέπειτα στην Οθόνη Έναρξης).

Στη συνέχεια δημιουργούμε άλλα 5 strings, ένα για κάθε οθόνη που θέλουμε να δημιουργήσουμε. Έτσι έχουμε τα παρακάτω:

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

    <string name="app_name">Ταξιδεύοντας στην Ευρώπη</string>
    <string name="helLo"> </string> //Οθόνη Έναρξης
    <string name="helP">Βοήθεια</string> //Οθόνη βοήθειας
    <string name="settings">Ρυθμίσεις</string> //Οθόνη Ρυθμίσεων
    <string name="game"> Έναρξη </string> //Κύρια Οθόνη παιχνιδιού
    <string name="scores">Βαθμολογία</string> //Οθόνη Βαθμολογιών
    <string name="menu">Μενού</string> //Κύρια οθόνη επιλογών

</resources>
```

4.3.1. Προσθήκη γραφικών πόρων (drawable)

Δεξί κλικ στο φάκελο res και ακολουθώντας τα βήματα New → Folder δημιουργούμε ένα νέο φάκελο με το όνομα drawable, όπου εκεί μέσα θα επικολλήσουμε το εικονίδιο που έχουμε δημιουργήσει για την εφαρμογή μας, με το όνομα europe_1.png.



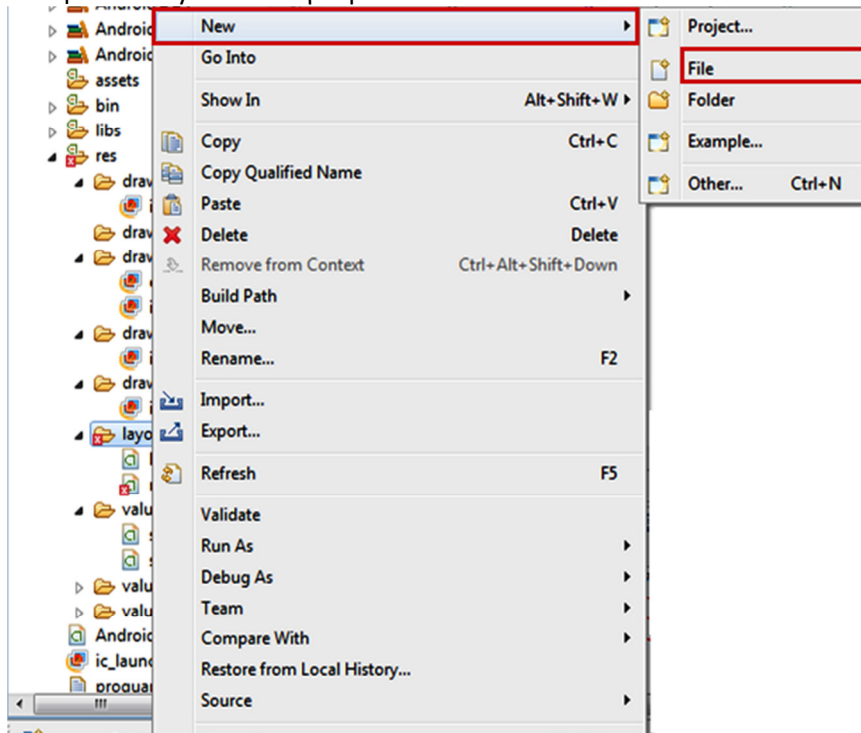
Εικόνα 55: Το λογότυπο της εφαρμογής μας

4.3.2. Δημιουργία των layouts (σχεδιαγράμματα) του προγράμματος

Ένα layout καθορίζει την αρχιτεκτονική της διεπαφής χρήστη (User Interface – UI), είναι η γραφική δομή της κάθε οθόνης. Δείχνει πως τα διάφορα αντικείμενα που υπάρχουν στην οθόνη, τοποθετούνται στο χώρο του UI και τι μορφή (Κείμενο, Εικόνα κλπ.) έχουν.

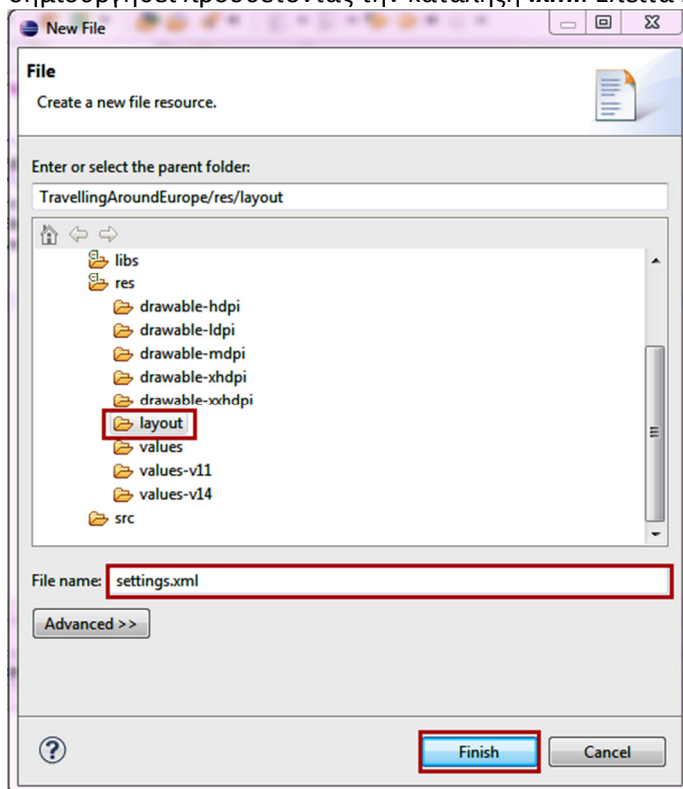
Για το πρόγραμμά μας χρειαζόμαστε 6 layouts που θα καθορίζουν το γραφικό περιβάλλον των activities που θα δημιουργήσουμε. Τα layout έχουν κατάληξη της μορφής .xml και δημιουργούμε ένα για κάθε activity χωριστά.

Για να δημιουργήσουμε ένα νέο layout, πηγαίνουμε στο φάκελο **TravellingAroundEurope/res/layout** που έχει δημιουργηθεί και πατώντας δεξί κλικ επάνω στο φάκελο layout διαλέγουμε **New → File**.



Εικόνα 56: Δημιουργία νέου αρχείου

Στο παράθυρο που ανοίγει δίνουμε το όνομα που θέλουμε στο καινούργιο layout που θα δημιουργηθεί προσθέτοντας την κατάληξη **.xml**. Έπειτα πατάμε **Finish**.



Εικόνα 57: Δήλωση του ονόματος, της κατάληξης και του φακέλου που θα αποθηκευτεί

Έτσι δημιουργούμε τα παρακάτω αρχεία της μορφής .xml: help.xml, settings.xml, scores.xml, menu.xml και το game.xml.

Στην αρχή όταν δημιουργείται το καθένα από αυτά, είναι ένα κενό αρχείο. Πάμε και κάνουμε αντιγραφή του κώδικα του πρώτου layout (main.xml) που δημιουργήθηκε αυτόματα μαζί με την πρώτη κλάση που δημιουργήσαμε (QuizActivity). Πάμε έπειτα στο TextView κι εκεί που δηλώνουμε το string, το αλλάζουμε ανάλογα με το αρχείο .xml που θα συνδέεται. Εδώ παραθέτουμε τον κώδικα του αρχείου settings.xml, ενός από τα .xml προκειμένου να γίνει αντιληπτός ο κώδικας.

```
<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"
    tools:context="{packageName}.{activityClass}" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/settings" />

</RelativeLayout>
```

Για την τοποθέτηση των αντικειμένων στο χώρο της οθόνης, χρειαζόμαστε ένα μηχανισμό ελέγχου (Layout). Στον παραπάνω κώδικα χρησιμοποιείται ο RelativeLayout⁵ ο οποίος τοποθετεί τα αντικείμενα σε μία σχετική θέση με τα άλλα αντικείμενα μέσα σε διάταξη ή σε σχέση με την οθόνη της συσκευής.

Επίσης δίνουμε ύψος και πλάτος στη στήλη μας έτσι ώστε να ταιριάζει ακριβώς στις διαστάσεις της οθόνης της συσκευής μας. Με τη χρησιμοποίηση της εντολής match_parent(fill_parent) που δίνεται στις διατάξεις layout_height και layout_width καθορίζεται το ύψος και το πλάτος αντίστοιχα, έτσι ώστε το ύψος και το πλάτος του layout, να ταιριάζει ακριβώς στις διαστάσεις της οθόνης της συσκευής μας.

Εκτός από το fill_parent υπάρχει και το wrap_content που βρίσκουμε παρακάτω στον κώδικα του TextView (δημιουργία κειμένου), το οποίο και θα χρησιμοποιήσουμε σε πολλά σημεία του κώδικά μας. Η διαφορά του είναι ότι ο μηχανισμός ελέγχου View σε αυτή την περίπτωση κρατά τις διαστάσεις του ή μεταβάλλεται πάντα σε συνάρτηση με το αρχικό Layout, προκειμένου να περιβάλλεται από το αρχικό.

Στη θέση text δηλώνουμε τον τίτλο του layout μέσα από ένα string(έχουν δημιουργηθεί παραπάνω). Έτσι για κάθε layout στη θέση android:text ακολουθούμε τον κώδικα android:text="@string/τίτλο_layout" όπως έχει δηλωθεί στο αρχείο strings.xml.

4.3.3. Δημιουργία των απαραίτητων Activities

Δημιουργώντας το project μας δημιουργήθηκε η αρχική μας κλάση, η EuropeActivity. Στη συνέχεια δημιουργούμε την βασική μας κλάση την QuizActivity, αντιγράφοντας την πρώτη

⁵ Αντίθετα με το RelativeLayout υπάρχει κι άλλες μορφές Layout, όπως είναι το LinearLayout με το οποίο κάθε αντικείμενο τοποθετείται το ένα κάτω από το άλλο ή το ένα δίπλα στο άλλο ανάλογα με τον προσανατολισμό που θα χρησιμοποιηθεί.

μας κλάση. Την κλάση αυτή θα κληρονομούν όλες οι υπόλοιπες κλάσεις που θα δημιουργήσουμε. Στην αρχή δηλώνονται σε αυτή μόνο οι σταθερές μεταβλητές που έχουν σχέση με τις επιλογές του παιχνιδιού.

```
public static final String GAME_PREFERENCES = "GamePrefs";
```

Έτσι έχουμε τον αρχικό κώδικα της QuizActivity:

```
package com.elengkor.travellingaroundeurope;

import android.app.Activity;

public class QuizActivity extends Activity {

    public static final String GAME_PREFERENCES = "GamePrefs";
}
```

Στην πορεία της υλοποίησης της εφαρμογής θα προσθέτουμε σταδιακά κι άλλα στοιχεία.

Τροποποιούμε την EuropeActivity έτσι ώστε να κληρονομεί την QuizActivity αντί της Activity.

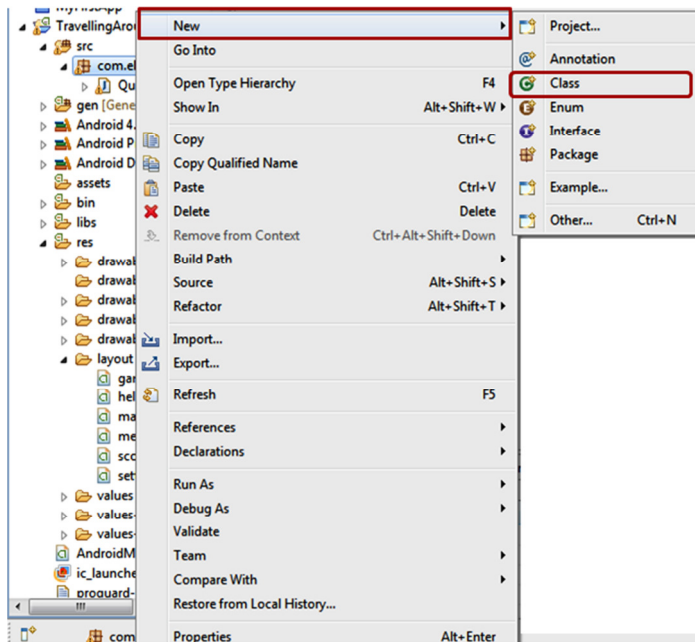
```
import android.os.Bundle;

public class EuropeActivity extends QuizActivity{

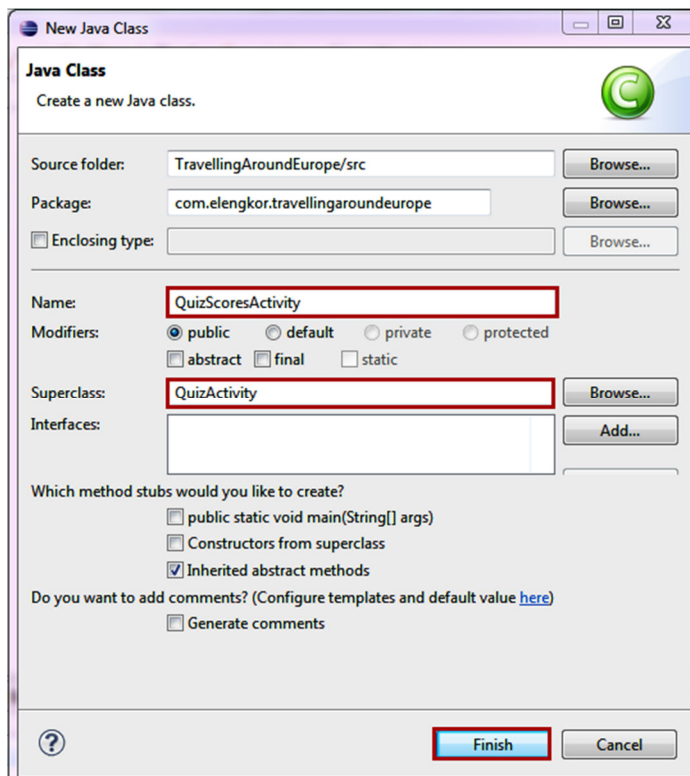
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
} (ΔΟΥΚΑΣ, 2011)
```

Δημιουργήσουμε και τις υπόλοιπες activities που είναι απαραίτητες για την υλοποίηση του παιχνιδιού μας; QuizMenuActivity, QuizSettingsActivity, QuizHelpActivity, QuizScoresActivity και την QuizGameActivity.

Κάνοντας δεξί κλικ πάνω στο package TravellingAroundEurope που έχει σχηματιστεί μέσα στο φάκελο src (source) κι ο οποίος ήδη περιέχει τις δύο πρώτες activities, πατάμε New → Class.



Εικόνα 58: Δημιουργία νέας κλάσης



Εικόνα 59: Δήλωση ονόματος της νέας activity και της activity που θα κληρονομεί

Ο αρχικός κώδικας της Scores activity

```

package com.elengkor.travellingarouneurope;

import android.os.Bundle;

public class QuizScoresActivity extends QuizActivity {

    @Override
  
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.scores);
        }
    }
}

```

Στην μέθοδο `setContentView()` δηλώνουμε το layout που αντιστοιχεί στην κάθε οθόνη που δημιουργούμε. Έτσι για παράδειγμα ενώ στην `EuropeActivity` η μέθοδος `setContentView()` παραπέμπει στο layout με το όνομα `main` (`R.layout.main`), στη μέθοδο που σχετίζεται με την οθόνη της βαθμολογίας καλούμε το layout με το όνομα `scores` (`R.layout.scores`). Η ίδια αντικατάσταση θα γίνει σε όλες τις activities που σχετίζονται με οθόνη προκειμένου η κάθε οθόνη να έχει το αντίστοιχο layout της (στο οποίο δηλώνεται ο τίτλος της).

4.4. Ενημέρωση του αρχείου Android Manifest του Android

Το αρχικό αρχείο του Android Manifest δημιουργήθηκε μαζί με το package μας, έχοντας κάποια χαρακτηριστικά από το σύστημα δοσμένα, όπως είναι το εικονίδιο της εφαρμογής (`ic_launcher.png`). Αυτό το εικονίδιο είναι αποθηκευμένο μέσα στο φάκελο `res/drawable` του package μας.

Για να αλλάξουμε το εικονίδιο της εφαρμογής που δημιουργήσαμε και να το αντικαταστήσουμε με το εικονίδιο που είχαμε προσθέσει σε προηγούμενη φάση μέσα στον ίδιο φάκελο (`europa_1.png`). Πηγαίνουμε στην γραμμή του κώδικα του αρχείου `AndroidManifest.xml` που αναφέρεται στο όνομα και κάνουμε την αντικατάσταση:

Δηλαδή αντί του

```
android:icon="@drawable/ic_launcher"
```

γράφουμε

```
android:icon="@drawable/europa_1".
```

Στη συνέχεια αντιγράφουμε την αρχική activity που εμφανίζεται (τη σχετική με τη `EuropeActivity`)

```

<activity
    android:name="com.elengkor.travellingaroundeuropa.EuropaActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

```

τόσες φορές όσες κι οι activities που δημιουργήσαμε. Σε κάθε ένα αντιγραμμένο κομμάτι, πηγαίνουμε και αντικαθιστούμε το όνομα της activity στο σημείο που δηλώνεται ("`com.elengkor.travellingaroundeuropa.όνομα_Activity`"). Έπειτα στο `intent-filter`, στη δήλωση `action`, αντικαθιστούμε το αντίστοιχο με την Activity όνομα του layout ("`android.intent.action.ONOMA_LAYOUT`"). Τέλος στην δήλωση `category` αντικαθιστούμε το `LAUNCHER` με το `DEFAULT`.

```

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

```



```

<application
    android:allowBackup="true"
    android:icon="@drawable/europe_1"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity

android:name="com.eLengkor.travellingaroundeurope.EuropeActivity"
    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="com.eLengkor.travellingaroundeurope.QuizActivity"
        android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    <activity
android:name="com.eLengkor.travellingaroundeurope.QuizGameActivity"
        android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.GAME" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    <activity
android:name="com.eLengkor.travellingaroundeurope.QuizSettingsActivity"
        android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.SETTINGS" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    <activity
android:name="com.eLengkor.travellingaroundeurope.QuizScoresActivity"
        android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.SCORES" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    <activity
android:name="com.eLengkor.travellingaroundeurope.QuizHelpActivity"
        android:label="@string/app_name" >

```



```

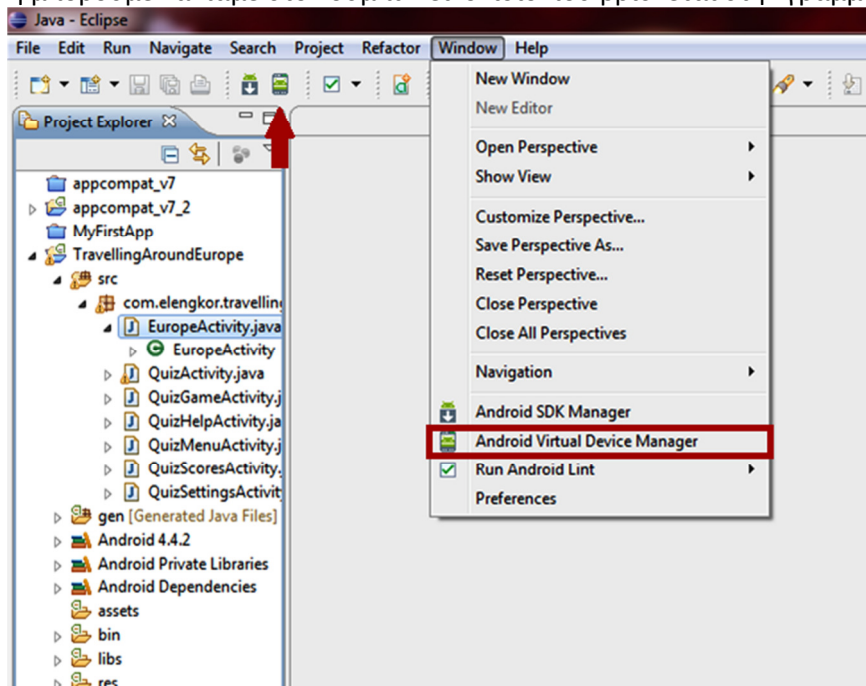
<intent-filter>
    <action android:name="android.intent.action.HELP" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>
</application>
<activity
    android:name="com.elengkor.travellingaroundeurope.QuizMenuActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MENU" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</manifest>

```

4.5. Δημιουργία της εικονικής μηχανής

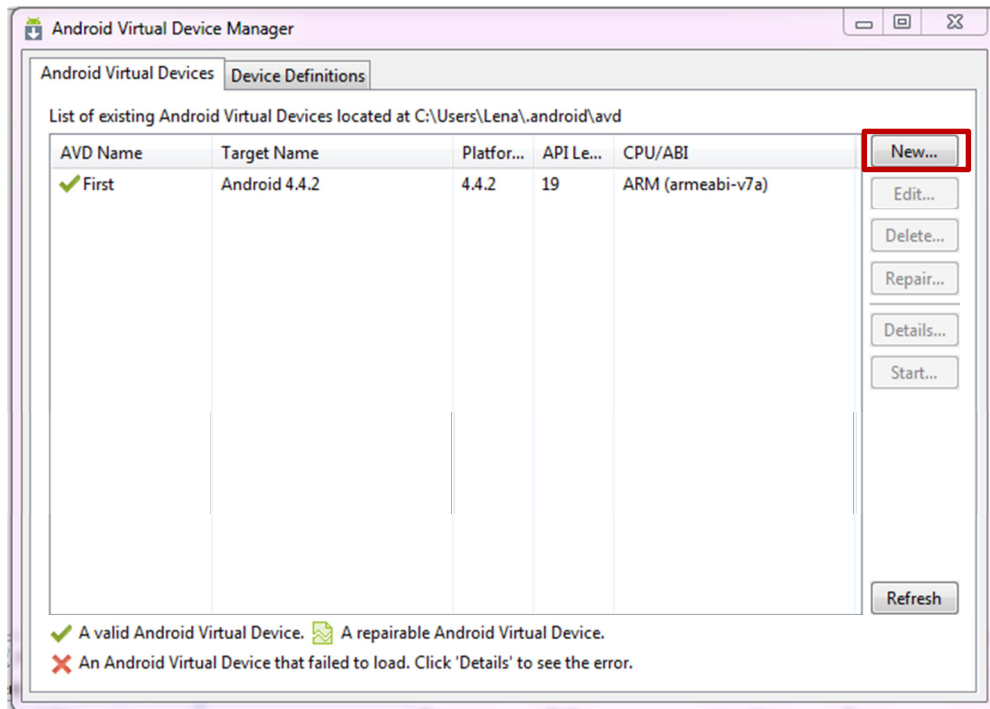
Χρησιμοποιήσουμε μία εικονική μηχανή(λειτουργεί όπως ένα κανονικό κινητό, όσον αφορά τις λειτουργίες και τη χρήση των κουμπιών του), η οποία θα μας βοηθήσει να τρέξουμε την εφαρμογή μας και να διαπιστώσουμε κατά πόσο μπορεί να λειτουργεί σωστά.

Την εικονική μηχανή (Android Virtual Device – AVD) θα την δημιουργήσουμε με τη βοήθεια του Android Virtual Device Manager, τον οποίο μπορούμε να βρούμε στην επιλογή Window → Android Virtual Device Manager που βρίσκεται στην γραμμή εργασιών του προγράμματος ή μπορούμε να πάμε στο κουμπί – εικονίδιο που βρίσκεται στην γραμμή εργαλείων.



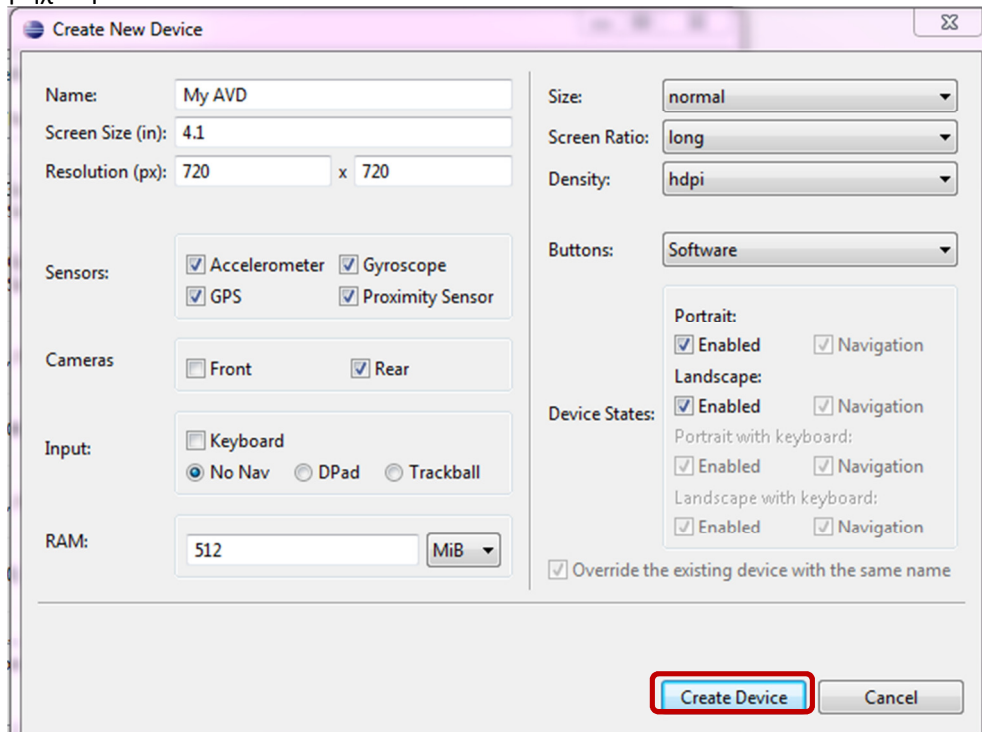
Εικόνα 60: Δημιουργία εικονικής μηχανής

Το πρόγραμμα δίνει κάποιες έτοιμες εικονικές μηχανές στην καρτέλα Device Definitions. Εμείς θα σχεδιάσουμε τη δική μας εικονική μηχανή. Έτσι μόλις εμφανιστεί η καρτέλα **Android Virtual Devices** πατάμε **New**.



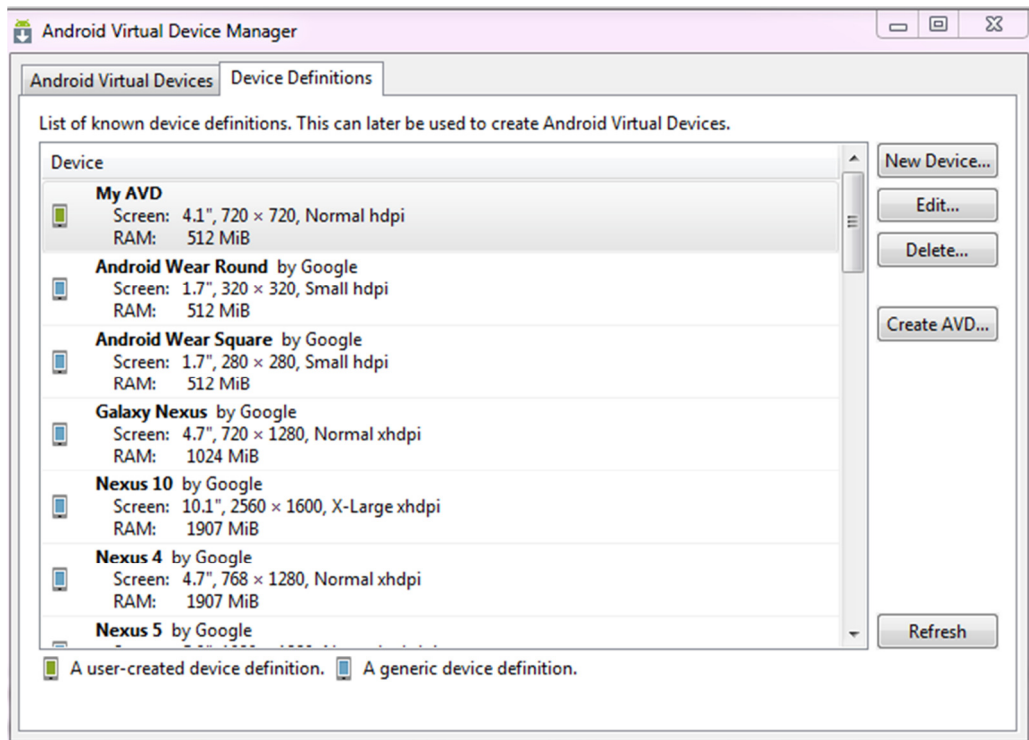
Εικόνα 61: Δημιουργία νέας εικονικής μηχανής

Στην καρτέλα **Create New Device** που εμφανίζεται διαλέγουμε ένα όνομα για την μηχανή μας, το μέγεθος οθόνης θα έχει και την ανάλυση, ότι εμείς πιστεύουμε καλύτερο προκειμένου να έχουμε ένα καλύτερο αποτέλεσμα στην δοκιμαστική εφαρμογή του παιχνιδιού μας. Τέλος πατάμε **Create Device** κι έτσι δημιουργήθηκε η νέα μας εικονική μηχανή.

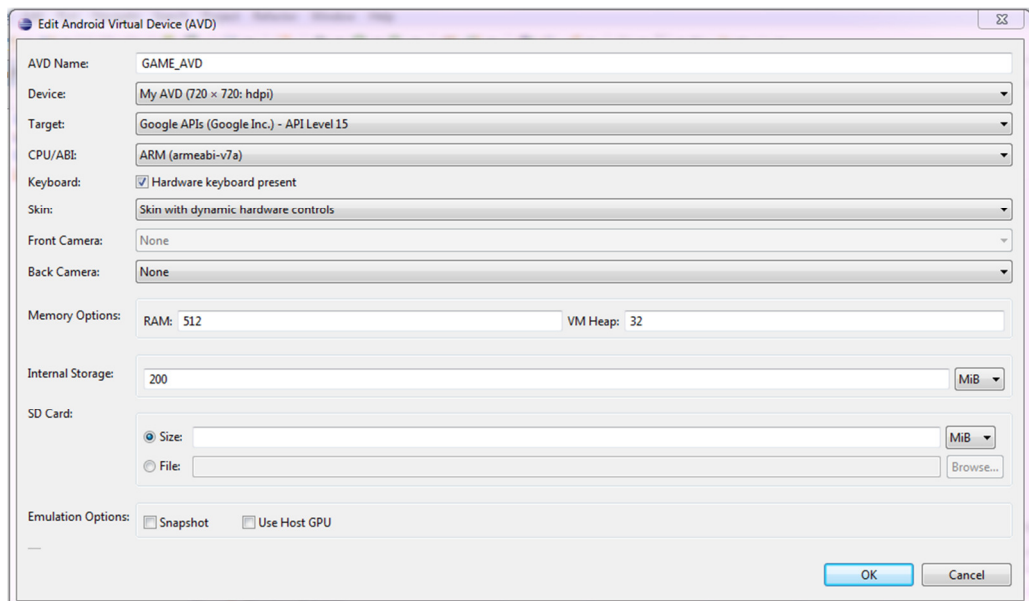


Εικόνα 62: Επιλογή ονόματος και χαρακτηριστικών της εικονικής μηχανής

Η δημιουργία της φαίνεται στην καρτέλα με τις μηχανές. Από εκεί μπορούμε και να την επεξεργαστούμε πατώντας το κουμπί **Edit**.

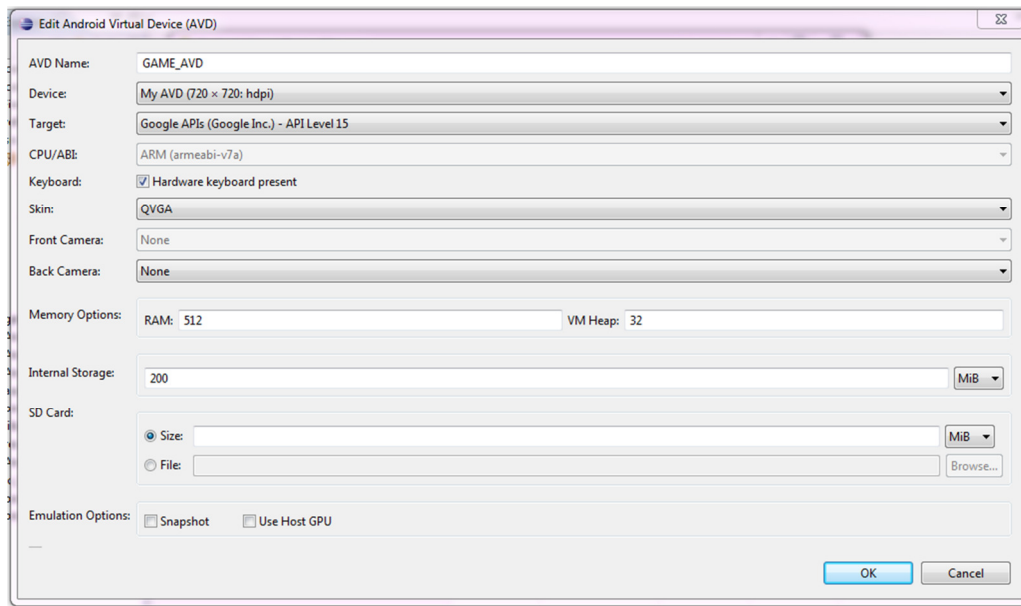


Εικόνα 63: Το σύνολο των διαθέσιμων εικονικών μηχανών και η δυνατότητα επεξεργασίας τους



Εικόνα 64: Επεξεργασία εικονικής μηχανής

Αλλάζουμε το χαρακτηριστικό skin και δοκιμάζουμε το QVGA.

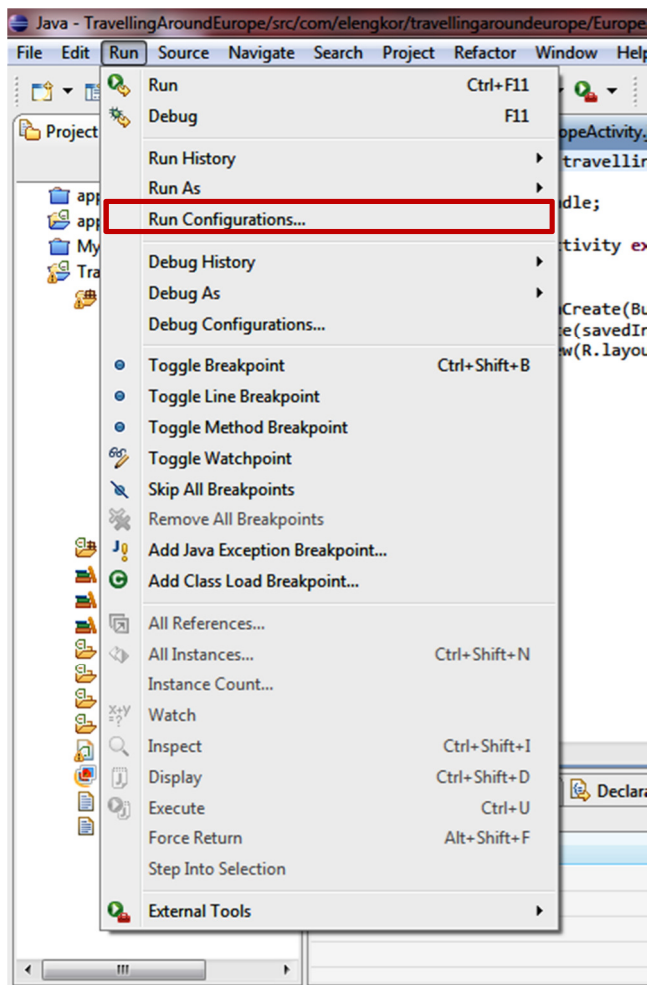


Εικόνα 65: Τα χαρακτηριστικά της εικονικής μηχανής που δημιουργούμε

4.5.1. Επεξεργασία των ρυθμίσεων της φόρτωσης της εφαρμογής στην εικονική μηχανή μας.

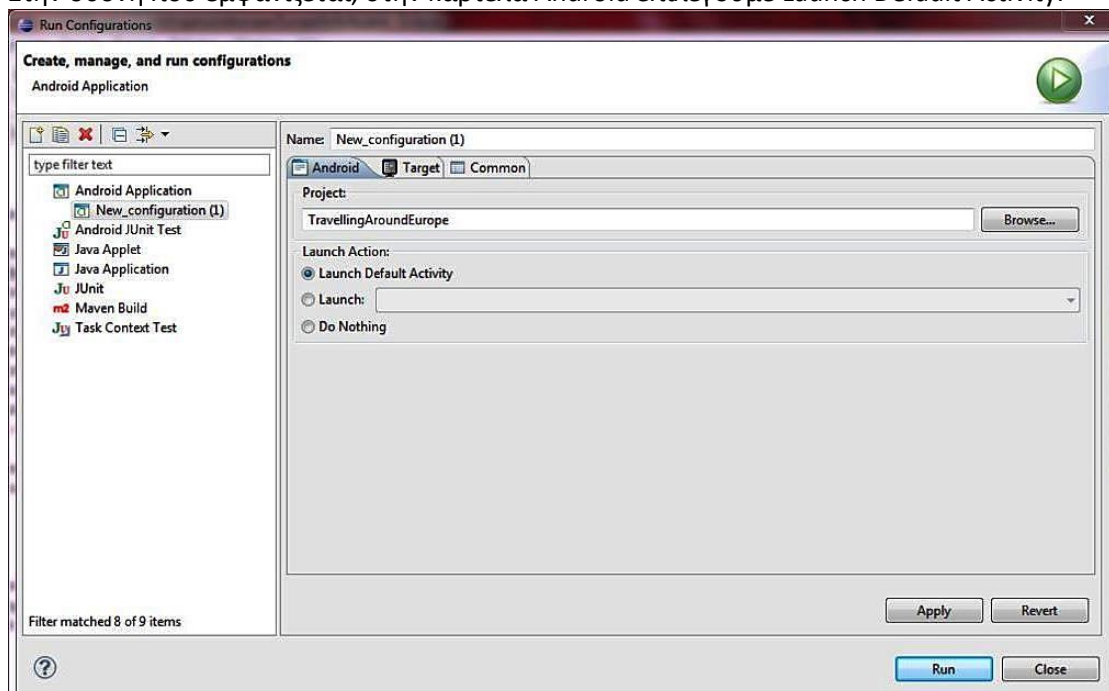
Τώρα θα πρέπει να εξειδικεύσουμε τις ρυθμίσεις της εικονικής μηχανής προκειμένου να τρέξει η εφαρμογή που δημιουργούμε.

Για να γίνει αυτό πηγαίνουμε στη επιλογή **Run → Run Configurations...**



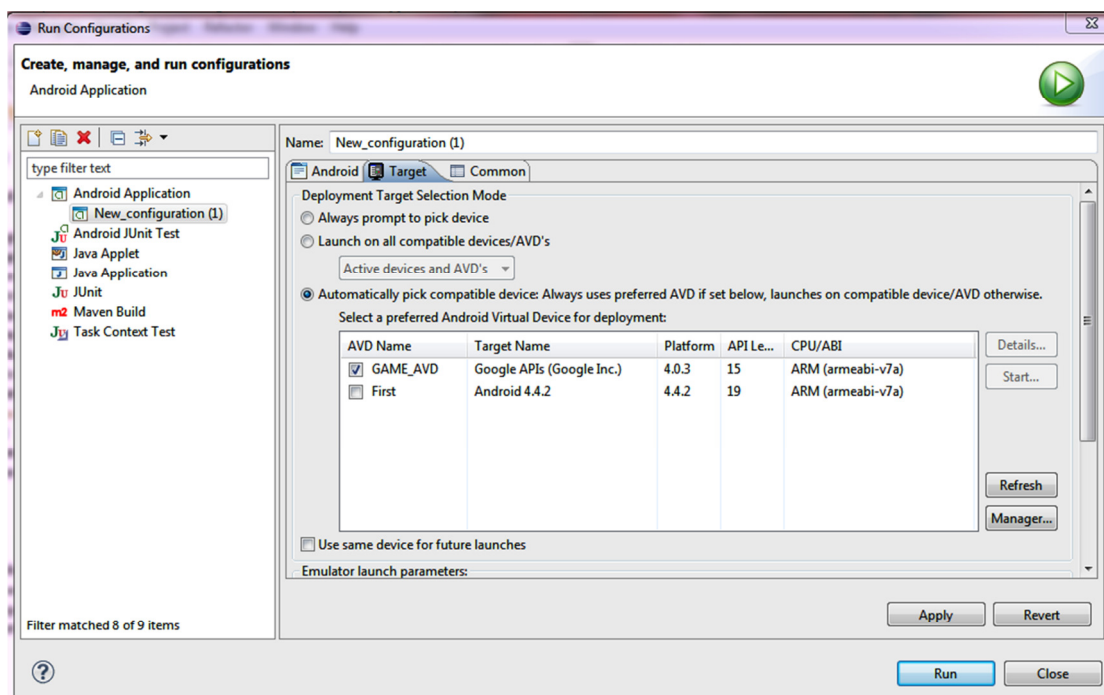
Εικόνα 66: Run → Run Configurations

Στην οθόνη που εμφανίζεται, στην καρτέλα Android επιλέγουμε Launch Default Activity.



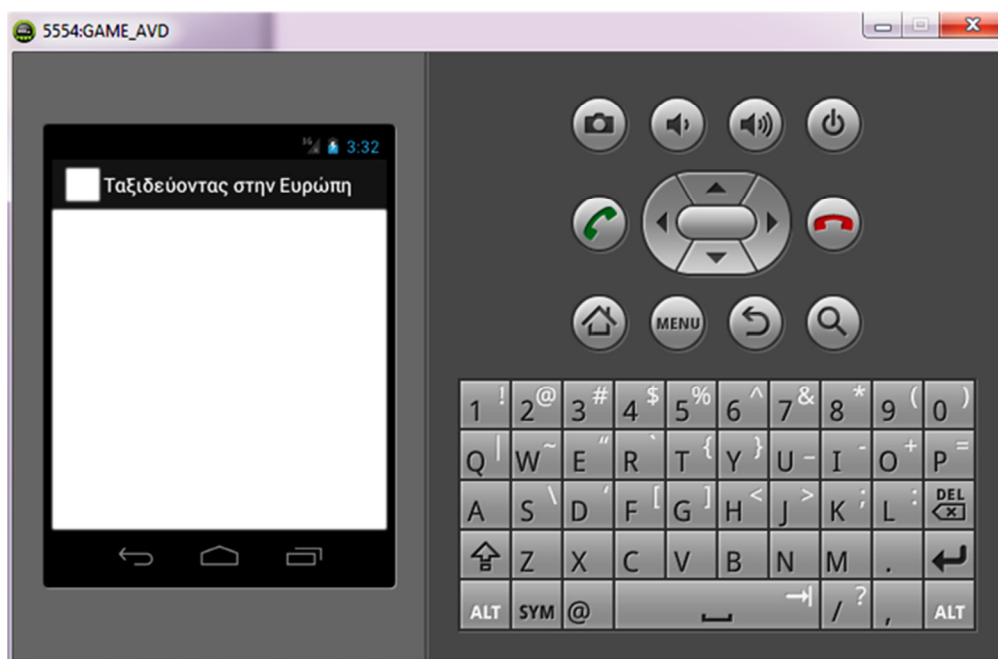
Εικόνα 67: Create, manage and run configurations

Στην καρτέλα **Target** τσεκάρουμε την AVD που δημιουργήσαμε (**GAME_AVD**) προκειμένου η εφαρμογή μας να τρέχει μέσα από αυτή. Έπειτα πατάμε **Run**.



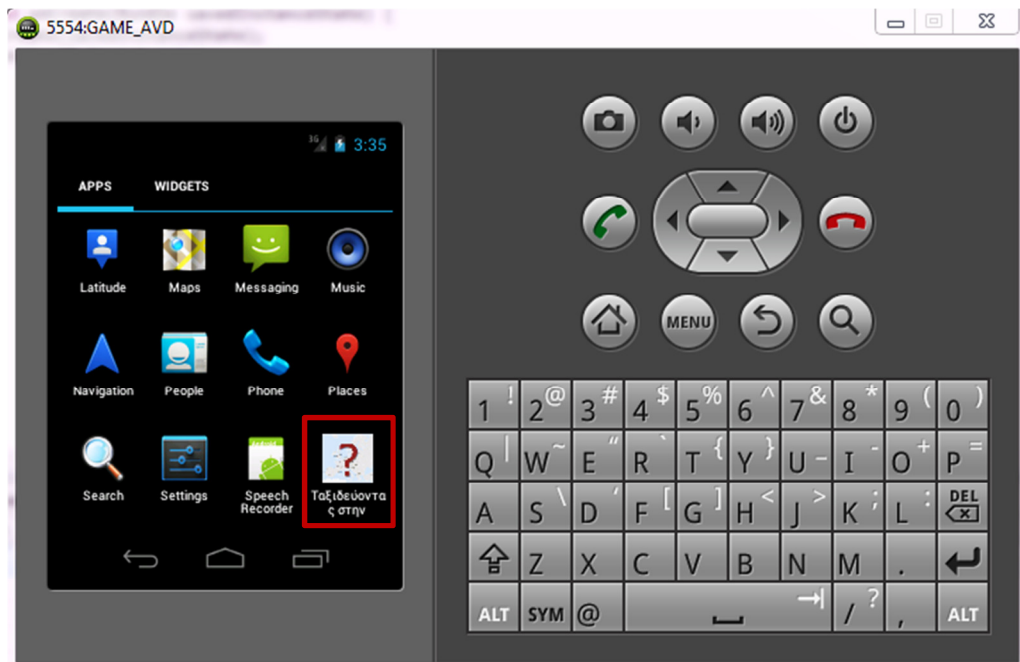
Εικόνα 68: Επιλογή target για το πρόγραμμά μας

Έπειτα από αναμονή κάποιων λεπτών, η εφαρμογή αρχίζει και τρέχει εμφανίζοντας την πρώτη activity που είχε δημιουργηθεί μαζί με το project, τη `EuropeActivity`. Επειδή δεν έχουμε δώσει κάποια στοιχεία ακόμη στην συγκεκριμένη activity, εδώ εμφανίζεται για την ώρα κενή.



Εικόνα 69: Η εφαρμογή μας μετά την πρώτη φόρτωση

Πηγαίνοντας στο μενού τώρα του εικονικού κινητού και διαλέγοντας να εμφανιστούν οι εφαρμογές που υπάρχουν μέσα σε αυτό, εμφανίζεται μαζί και το εικονίδιο της εφαρμογής που δημιουργήσαμε.



Εικόνα 70: Το λογότυπο της εφαρμογής στον κατάλογο των εφαρμογών του εικονικού κινητού μας

Αυτό ήταν και το επιθυμητό μας αποτέλεσμα.

ΚΕΦΑΛΑΙΟ 5

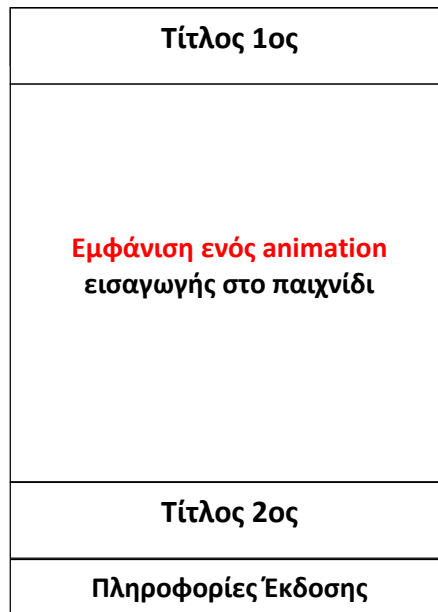
Σχεδιασμός και υλοποίηση της Οθόνης Εκκίνησης

5.1. Σχεδιασμός Οθόνης Εκκίνησης

Επεξεργαζόμαστε τις οθόνες – activities της εφαρμογής μας χωριστά την κάθε μία. Ξεκινάμε με την πρώτη που βλέπει ο χρήστης, την οθόνη εκκίνησης, την EuropeActivity.

Η οθόνη αυτή θα περιέχει έναν τίτλο στην κορυφή μία εικόνα στη μέση, με ένα animation να τρέχει, ένα δεύτερο τίτλο κάτω από την εικόνα και στο κάτω μέρος κάποιες πληροφορίες για το παιχνίδι.

Η διάταξη της οθόνης αυτής θα έχει τη μορφή:



Για να δημιουργήσουμε την οθόνη αυτή πρέπει να επεξεργαστούμε το main.xml, το οποίο είχαμε δημιουργήσει προηγουμένως και το οποίο σχετίζεται άμεσα με την EuropeActivity.

Για αρχή, χρειαζόμαστε ένα μηχανισμό ελέγχου (Layout) που θα τοποθετεί τα αντικείμενα της οθόνης σε μία κάθετη στήλη, το ένα κάτω από το άλλο. Για το λόγο αυτό θα χρησιμοποιούμε τη LinearLayout και δίνουμε εντολή για κατακόρυφο προσανατολισμό

```
android:orientation="vertical".
```

Επίσης δίνουμε ύψος και πλάτος στη στήλη μας έτσι ώστε να ταιριάζει ακριβώς στις διαστάσεις της οθόνης της συσκευής μας. Αυτό επιτυγχάνεται με την εντολή fill_parent που δίνεται στις διατάξεις layout_height και layout_width που καθορίζουν το ύψος και το πλάτος αντίστοιχα.

```
android:layout_width="fill_parent"  
android:layout_height="fill_parent"
```

Τέλος θα δηλώσουμε και τη φωτογραφία που θα φαίνεται ως φόντο στην οθόνη εκκίνησης. Αφού περάσουμε τη φωτογραφία στους όλους τους φακέλους res/drawable κάνοντας αντιγραφή της φωτογραφίας κι επικόλλησης σε αυτούς, τη δηλώνουμε μέσα στο Layout με

το όνομα της χωρίς την κατάληξη που υποδηλώνει σε τι μορφή είναι (boudapesti). Έτσι έχουμε:

```
android:background="@drawable/boudapesti"
```

5.1.1. Δημιουργία των μηχανισμών ελέγχου TextView

Έπειτα θα δημιουργήσουμε τρεις μηχανισμούς ελέγχου TextView που θα εμφανίζουν κείμενο (τον πρώτο και δεύτερο τίτλο και τις πληροφορίες έκδοσης του παιχνιδιού).

Τα TextView θα τα δημιουργήσουμε μέσα στο LinearLayout. Για το καθένα θα δηλωθεί και πάλι το πλάτος, το ύψος του Layout, ανάλογα με τις προτιμήσεις μας. Επίσης εδώ μπορούμε να χρησιμοποιήσουμε κι άλλες ιδιότητες προκειμένου να δώσουμε κάποια συγκεκριμένη μορφοποίηση στα TextView μας. Έτσι μπορούμε να επιλέξουμε στοίχιση για τη συμβολοσειρά μας (layout_gravity), να δηλώσουμε το χρώμα (textColor), το στυλ (textStyle) και το μέγεθος (font_size) που θα έχει η γραμματοσειρά μας. Επίσης μπορεί να δηλωθεί κι απόσταση μεταξύ των γραμμών με τη χρήση του lineSpacingExtra. Τέλος σε κάθε TextView θα πρέπει να δοθεί μία μοναδική ταυτότητα, ένα id, καθώς και να δηλωθεί ως string τι θα περιέχει το κάθε TextView.

Όλες αυτές οι δηλώσεις όμως για τη σωστή τους λειτουργία και την αναγνώρισή του από το πρόγραμμα προϋποθέτουν τη δημιουργία 2 νέων αρχείων xml, το **colors.xml** όπου θα δηλώνονται τα χρώματα που χρησιμοποιούμε και το **dimens.xml** όπου θα δηλώνονται οι μορφοποιήσεις.

5.1.2. Δημιουργία του colors.xml αρχείου

Αρχικά δημιουργούμε το αρχείο /res/values/colors.xml χρησιμοποιώντας τα ίδια βασικά στοιχεία με το αρχείο string.xml (την κεφαλίδα και τα resources ως τίτλους). Δίνουμε ένα όνομα σε κάθε χρώμα προκειμένου το πρόγραμμα όταν το καλούμε με το όνομα αυτό να το αναγνωρίζει κι έπειτα δηλώνουμε τον κωδικό του χρώματος. Η χρήση των κωδικών χρωμάτων που θεωρούνται ασφαλή για την html, είναι ο καλός οδηγός για την σωστή κωδικοποίηση αυτών που θέλουμε για την εφαρμογή μας.

Έτσι το αρχείο αυτό περιέχει 2 χρώματα, το χρώμα του κειμένου και το χρώμα του φόντου πίσω από το κείμενο, του περιγράμματος.

colors.xml

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

    <color name="purple">#CC0066</color>
    <color name="blue_purple">#330066</color>

</resources>
```

5.1.3. Δημιουργία του dimens.xml αρχείου

Έπειτα δημιουργούμε το αρχείο /res/values/dimens.xml όπου δηλώνουμε διαστάσεις. Για την ώρα δηλώνουμε το μέγεθος γραμματοσειράς, την απόσταση των γραμμών του τίτλου και της έκδοσης μεταξύ τους (διάστιχο) και το μέγεθος της γραμματοσειράς της έκδοσης.

dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="font_size">12pt</dimen>
```

```
<dimen name="title_spacing">3pt</dimen>
<dimen name="version_size">6pt</dimen>
<dimen name="version_spacing">3pt</dimen>
```

```
</resources>
```

5.1.4. Συμπλήρωση του αρχείου string.xml

Προσθέτουμε και στο string.xml τις συμβολοσειρές που είναι απαραίτητες για την οθόνη εκκίνησης, τους τίτλους και τις πληροφορίες της έκδοσης της εφαρμογής. Εδώ χρησιμοποιώντας τις γνώσεις μας από την html δηλώνουμε τους τίτλους να εμφανίζονται με πλάγια γραφή.

```
<string name="hello"><i>Ταξιδεύοντας στην Ευρώπη! </i></string>
<string name="hello1"><i>Γνωρίστε τη! </i></string>
<string name="app_version_info">Version 1.0 </string>
```

5.1.5. Δημιουργία του μηχανισμού ελέγχου ImageView

Για να ολοκληρωθεί ο κώδικας της πρώτης οθόνης χρειαζόμαστε κι έναν μηχανισμό ο οποίο θα εμφανίζει το γραφικό – εικόνα, στο κέντρο της οθόνης. Για να το πετύχουμε αυτό χρησιμοποιούμε το μηχανισμό ελέγχου ImageView. Δίνουμε μία ταυτότητα στον μηχανισμό μας (id), καθορίζουμε το ύψος και το πλάτος της εικόνας – γραφικού που θα χρησιμοποιήσουμε, την στοίχιση στο πλαίσιο της και δηλώνουμε και το όνομα της εικόνας(europe3), όπως θα εμφανίζεται στο φάκελο res/drawable, αφού την έχουμε προσθέσει πρώτα μέσα στους φακέλους. Το ImageView το τοποθετούμε στον κώδικα μετά το πρώτο Textview έτσι ώστε να εμφανίζεται ανάμεσα στους δύο τίτλους.

5.1.6. Ο κώδικας της main.xml συνολικά

Συνολικά ο κώδικας που διαμορφώνεται για τη main.xml είναι:

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

  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:id="@+id/TextViewTopTitle"
    android:text="@string/hello"
    android:textColor="@color/purple"
    android:textStyle="bold"
    android:lineSpacingExtra="@dimen/title_spacing"
    android:textSize="@dimen/font_size"
    android:gravity="center" />

  <ImageView
    android:id="@+id/ImageView"
    android:src="@drawable/europe3"
```

```

        android:layout_height="170dp"
        android:layout_width="150dp"
        android:layout_gravity="center"
        android:contentDescription="@string/desc">
</ImageView>

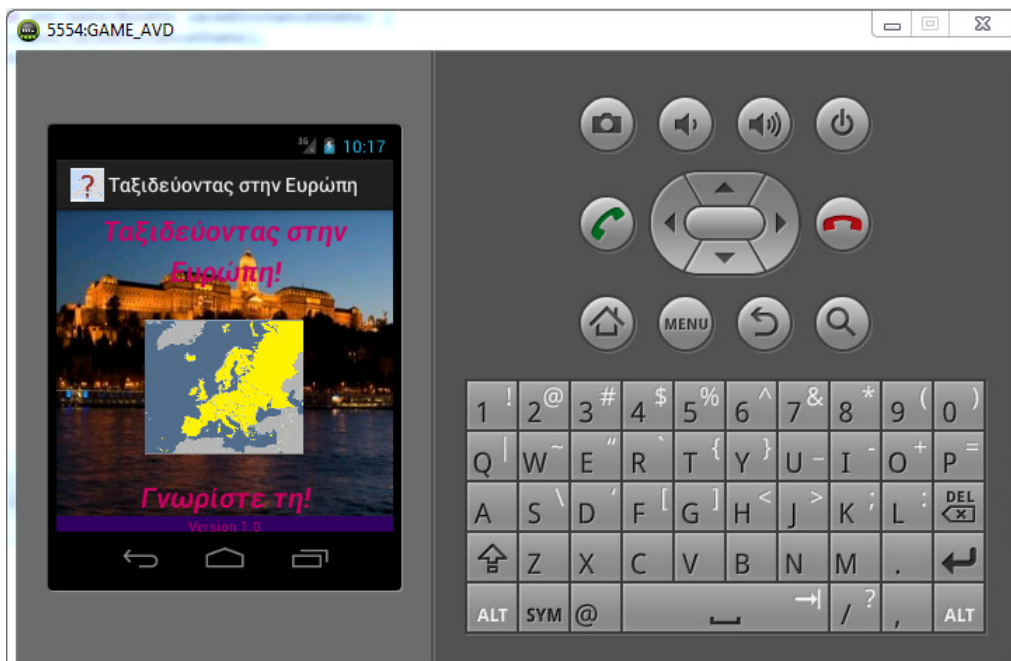
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:id="@+id/TextViewBottomTitle"
    android:text="@string/hello1"
    android:textColor="@color/purple"
    android:textStyle="bold"
    android:textSize="@dimen/font_size"
    android:gravity="center" />

<TextView
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:id="@+id/TextViewBottomVersion"
    android:background="@color/blue_purple"
    android:text="@string/app_version_info"
    android:textColor="@color/purple"
    android:textSize="@dimen/version_size"
    android:lineSpacingExtra="@dimen/version_spacing"
    android:gravity="center">
</TextView>

</LinearLayout>

```

Τρέχοντας την εφαρμογή παράγεται η εικόνα του παρακάτω σχήματος στο οποίο απεικονίζεται τι έχουμε δημιουργήσει μέχρι τώρα.



Εικόνα 71: Η εφαρμογή με την τροποποίηση της main.xml

5.2. Προσθήκη κίνησης στο γραφικό της Οθόνης Εκκίνησης

Θέλουμε τώρα να χρησιμοποιήσουμε τις δυνατότητες που έχει το android για τη δημιουργία κινούμενων γραφικών. Έτσι επειδή θέλουμε να δώσουμε μία κίνηση στο γραφικό μας, να εμφανίζεται σταδιακά, να περιστρέφεται, να αλλάζει μέγεθος και τέλος να μετακινείται, θα χρησιμοποιήσουμε έναν από τους τύπους κίνησης γραφικών Android, την κίνηση με ενδιάμεσα καρέ (tweened Animation).

5.2.1. Προσθήκη πόρων κίνησης

Δημιουργούμε ένα νέο φάκελο /res/anim μέσα στον οποίο θα τοποθετήσουμε τρία αρχεία xml, το καθένα θα αναφέρεται σε κάποια προσαρμοσμένη κίνηση χωριστά. Τα αρχεία αυτά θα τα ονομάσουμε fade_in.xml, fade_out.xml και custom_anim.xml.

Η πρώτη κίνηση fade_in.xml εμφανίζει σταδιακά την εικόνα, από μία τιμή alpha 0 (διαφανές) σε μια τιμή alpha ίση με 1(αδιαφανές), σε μια χρονική περίοδο 2,5 δευτερόλεπτα(2500 χιλιοστά του δευτερολέπτου), ξεκινώντας από το 1,5 δευτερόλεπτο. Η συνολική διάρκεια, δηλαδή φτάνει στα 4 δευτερόλεπτα. Επίσης, επειδή ένας παρεμβολέας (Interpolator) ορίζει το ρυθμό μεταβολής μιας κινούμενης εικόνας κι επιτρέπει τα βασικά εφέ κίνησης (όπως είναι η alpha διαφάνεια), η κλίμακα, η μετακίνηση, η περιστροφή) να επιταχύνουν, να επιβραδύνουν, να επαναλαμβάνονται, κλπ., θα πρέπει να δώσουμε τιμή και σε αυτόν. Επειδή όμως θέλουμε να έχουμε διαφορετικούς παρεμβολείς σε κάθε μας κινητικό σετ, θέτουμε την επιλογή `shareInterpolator="false"`

Έτσι ο κώδικας της fade_in.xml είναι:

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <alpha
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="2500"
        android:startOffset="1500">
    </alpha>
</set>
```

Την κίνηση αυτή θα την εφαρμόσουμε στο πρώτο τίτλο που εμφανίζεται στην Οθόνη εκκίνησης.

Ο κώδικας της fade_in2.xml είναι παρόμοιος με τον κώδικα του προηγούμενου .xml. Το μόνο που διαφέρει είναι το startOffset, το οποίο θέτουμε να ξεκινά στα 2,5 δευτερόλεπτα (1 δευτερόλεπτο μετά το πρώτο animation).

```
android:startOffset="2500"
```

Συνολικός χρόνος του δευτέρου animation είναι τα 5 δευτερόλεπτα.

Και τα δύο αυτά animation αφορούν τα δύο πρώτα TextView που έχουν δημιουργηθεί στη main.xml

Στο 3^ο αρχείο custom_anim.xml, θα αφορά το ImageView και θα επηρεάσει την κίνηση της εικόνας. Ουσιαστικά με τον παρακάτω κώδικα δίνουμε εντολή στην εικόνα να περιστραφεί κατά 360° γύρω από το κέντρο του, στο 50% του άξονα X και Y και η διάρκεια της περιστροφής να είναι 3 δευτερόλεπτα. Επίσης, δίνουμε εντολή να εμφανίζεται σταδιακά από μια τιμή alpha ίση με 0 (διαφανές) μέχρι μια τιμή ίση με 1 (αδιαφανές). Η διάρκεια της

ενέργειας αυτής είναι και πάλι 3 δευτερόλεπτα. Η τελευταία εντολή που δίνουμε, είναι αυτή της κλιμάκωσης. Η εικόνα του ImageView θα κλιμακώνεται από 10% (fromScale=".1") σε 100% (toScale="1.0"). Συνολικά το animation του ImageView θα διαρκέσει 3 δευτερόλεπτα.

```
<?xml version="1.0" encoding="utf-8" ?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="3000"
    />
    <alpha
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="3000"
    >
    </alpha>
    <scale
        android:pivotX="50%"
        android:pivotY="50%"
        android:fromXScale=".1"
        android:fromYScale=".1"
        android:toXScale="1.0"
        android:toYScale="1.0"
        android:duration="3000"/>
</set> (Conder, 2012)
```

5.3. Εφαρμογή των Animation στην EuropeActivity

Πηγαίνοντας τώρα στην EuropeActivity θα πρέπει να δώσουμε τις κατάλληλες εντολές προκειμένου να φορτωθούν τα κινούμενα γραφικά που δημιουργήσαμε στο xml αρχείο. Γενικά επειδή ένα animation χρειάζεται πολύ χρόνο για να φορτωθεί στην εφαρμογή καλό θα είναι να έχουμε προνοήσει ώστε αν για κάποιο λόγο η εφαρμογή διακοπεί ξαφνικά, με την επαναφορά της εφαρμογής, το animation να συνεχίσει από εκεί που σταμάτησε.

Ξεκινώντας από τον πρώτο τίτλο με το όνομα `TextViewTopTitle` και θέλοντας να εφαρμόσουμε την κίνηση `fade_in`, χρησιμοποιούμε την μέθοδο `onCreate` της κλάσης `EuropeActivity` για να ανακτήσουμε ένα στιγμιοτύπο του μηχανισμού `TextView`. Δημιουργούμε το αντικείμενο `Animation` κι έπειτα μέσα σ' αυτό φορτώνουμε τον πόρο κίνησης. Τέλος καλούμε τη μέθοδο `startAnimation` για την έναρξη της κίνησης του πρώτου τίτλου.

```
// Επάνω Τίτλος
TextView logo1 = (TextView) findViewById(R.id.TextViewTopTitle);
Animation fade1 = AnimationUtils.LoadAnimation(this, R.anim.fade_in);
logo1.startAnimation(fade1);
```

Ακολουθούμε την ίδια διαδικασία και στα άλλα δύο μέρη της οθόνης, δηλαδή την εικόνα (χρησιμοποιώντας τον μηχανισμό ImageView) και τον τελευταίο τίτλο χρησιμοποιώντας και πάλι το μηχανισμό TextView).

```
// Εικόνα
ImageView logo2 = (ImageView) findViewById(R.id.ImageView);
Animation fade2 = AnimationUtils.LoadAnimation(this, R.anim.custom_anim);
logo2.startAnimation(fade2);
// Κάτω Τίτλος
TextView logo3 = (TextView) findViewById(R.id.TextViewBottomTitle);
Animation fade3 = AnimationUtils.LoadAnimation(this, R.anim.fade_in2);
logo3.startAnimation(fade3);
```

Όταν τα animations πρέπει να σταματήσουν, καλούμε τη μέθοδο clearAnimation() και για τα τρία κινητά μέρη της οθόνης μέσα στην μέθοδο onPause() της δραστηριότητάς μας.

```
protected void onPause()
{
    super.onPause();
    // Διακοπή animation
    TextView logo1 = (TextView) findViewById(R.id.TextViewTopTitle);
    logo1.clearAnimation();
    ImageView logo2 = (ImageView) findViewById(R.id.ImageView);
    logo2.clearAnimation();
    TextView logo3 = (TextView) findViewById(R.id.TextViewBottomTitle);
    logo3.clearAnimation();
}
```

5.4. Σύνδεση της Οθόνης Εκκίνησης με την Κύρια Οθόνη Επιλογών

Αφού δημιουργήσουμε τα animation της Οθόνης Εκκίνησης, θα πρέπει να δημιουργήσουμε τον τρόπο με τον οποίο με το που τελειώνει η κίνηση του τελευταίου animation (του τελευταίου τίτλου), να γίνεται μια ομαλή μετάβαση από την Οθόνη Εκκίνησης (EuropeActivity), στην Κύρια Οθόνη Επιλογών (QuizMenuActivity).

Για να γίνει αυτό δημιουργούμε ένα νέο μηχανισμό ελέγχου Intent που θα καλεί την QuizMenuActivity μέσα στη μέθοδο startActivity(). Καλούμε τη μέθοδο finish() της EuropeActivity, προκειμένου να την τερματίσουμε και να μην μείνει στην μνήμη.

```
startActivity(new Intent(EuropeActivity.this, QuizMenuActivity.class));
EuropeActivity.this.finish();
```

Το animation του τελευταίου τίτλου (fade_in2) είναι αυτό που θα μας οδηγήσει στη μετάβαση στην επόμενη οθόνη (είναι αυτό που τελειώνει τελευταίο από τα 3 animation). Δημιουργούμε ένα αντικείμενο της κλάσης AnimationListener, που περιέχει τα τρία στοιχεία του κύκλου της ζωής της κίνησης, την έναρξη (start), τη λήξη (end) και την επανάληψη (repeat). Στην περίπτωση μας χρειαζόμαστε τη λήξη της κίνησης, τη μέθοδο onAnimationEnd(), προκειμένου να εισάγουμε μέσα σε αυτή την έναρξη της επόμενης δραστηριότητας, όπως αναφέρθηκε παραπάνω. Οι άλλες δύο (onAnimationStart, onAnimationRepeat) δεν θα περιέχουν κάτι.

```
fade3.setAnimationListener(new AnimationListener()
{
    public void onAnimationStart(Animation arg0)
```

```

{
    // Den periexei kati
}

public void onAnimationRepeat(Animation arg0)
{
    // Den periexei kati
}

public void onAnimationEnd(Animation animation)
{
    startActivity(new Intent(EuropeActivity.this,
QuizMenuActivity.class));
    EuropeActivity.this.finish();
}
});

```

5.5. Ο κώδικας της Οθόνης Εκκίνησης (EuropeActivity) συνολικά

```

package com.elengkor.travellingaroundeurope;

import android.content.Intent;
import android.os.Bundle;
import android.view.animation.Animation;
import android.view.animation.Animation.AnimationListener;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

public class EuropeActivity extends QuizActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Επάνω Τίτλος
        TextView logo1 = (TextView) findViewById(R.id.TextViewTopTitle);
        Animation fade1 = AnimationUtils.LoadAnimation(this,
R.anim.fade_in);
        logo1.startAnimation(fade1);
        // Εικόνα
        ImageView logo2 = (ImageView) findViewById(R.id.ImageView);
        Animation fade2 = AnimationUtils.LoadAnimation(this,
R.anim.custom_anim);
        logo2.startAnimation(fade2);
        // Κάτω Τίτλος
        TextView logo3 = (TextView) findViewById(R.id.TextViewBottomTitle);
        Animation fade3 = AnimationUtils.LoadAnimation(this,
R.anim.fade_in2);
        logo3.startAnimation(fade3);

        fade3.setAnimationListener(new AnimationListener()
        {

```

```

        public void onAnimationStart(Animation arg0)
        {
            // Den periexei kati
        }

        public void onAnimationRepeat(Animation arg0)
        {
            // Den periexei kati
        }
        public void onAnimationEnd(Animation animation)
        {
            startActivity(new
Intent(EuropeActivity.this, QuizMenuActivity.class));
            EuropeActivity.this.finish();
        }
    });
}
@Override
protected void onPause()
{
    super.onPause();
    // Διακοπή animation
    TextView logo1 = (TextView)
findViewById(R.id.TextViewTopTitle);
    logo1.clearAnimation();
    ImageView logo2 = (ImageView) findViewById(R.id.ImageView);
    logo2.clearAnimation();
    TextView logo3 = (TextView)
findViewById(R.id.TextViewBottomTitle);
    logo3.clearAnimation();
}
}

```


ΚΕΦΑΛΑΙΟ 6

Σχεδιασμός και υλοποίηση της Κύριας Οθόνης Επιλογών

6.1. Σχεδιασμός Κύριας Οθόνης Επιλογών

Η κύρια Οθόνη επιλογών θα έχει περίπου την παρακάτω μορφή

logo	Τίτλος Οθόνης	logo
	Έναρξη	
	Βαθμολογία	
	Ρυθμίσεις	
	Βοήθεια	

6.1.1. Δημιουργία του menu.xml layout αρχείου

Αρχικά χρησιμοποιούμε την διάταξη ενός τίτλου (TextView) στη μέση και δύο εικόνων (ImageView) δεξιά κι αριστερά του, στο επάνω μέρος της οθόνης. Για την διάταξη αυτή τοποθετούμε τα τρία στοιχεία αυτά μέσα σε ένα Relative Layout ώστε να έχουν μία σχετική διάταξη μεταξύ τους. Καθορίζουμε το Layout_width και το Layout_height έτσι ώστε να είναι wrap_content, δίνουμε ένα διαφορετικό id για το καθένα και καθορίζουμε τα χαρακτηριστικά που θέλουμε να δώσουμε στο καθένα.

```
<RelativeLayout
    android:id="@+id/RelativeLayout01"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content">
    <ImageView
        android:layout_alignParentTop="true"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:id="@+id/ImageView_MenuHeader"
        android:layout_alignParentLeft="true"
        android:src="@drawable/europe4"
        android:contentDescription="@string/desc"></ImageView>

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
```

```

android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:shadowColor="@color/menu_glow"
android:shadowDx="0"
android:shadowDy="0"
android:shadowRadius="5"
android:text="@string/menu"
android:textColor="@color/purple"
android:textSize="@dimen/font_size_menu"
android:textStyle="bold" />

```

```

<ImageView
    android:layout_alignParentTop="true"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:id="@+id/ImageView_MenuHeader2"
    android:layout_alignParentRight="true"
    android:src="@drawable/europe4"
    android:contentDescription="@string/desc"></ImageView>

```

6.1.2. Δημιουργία της ListView στο αρχείο menu.xml

Επειδή θέλουμε τα στοιχεία μας να είναι σε λίστα χρησιμοποιούμε τη διάταξη List View στο αρχείο μας menu.xml μέσα στο αρχικό Linear Layout που είχαμε, αλλά σε δικό του Relative Layout σε σχέση με τα προηγούμενα στοιχεία.

```
</RelativeLayout>
```

```

<RelativeLayout
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:id="@+id/RelativeLayout02">

    <ListView
        android:id="@+id/ListView_Menu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:divider="@drawable/divider"
        android:listSelector="@drawable/textured" >

        </ListView>
    </RelativeLayout>

```

6.2. Δημιουργία των απαραίτητων πόρων για την Κύρια Οθόνη Επιλογών

Προσθέτουμε στο αρχείο /res/value/strings.xml τα παρακάτω χαρακτηριστικά που είναι απαραίτητα για τη λίστα μας:

```

<string name="help">Βοήθεια</string>
<string name="settings">Ρυθμίσεις</string>
<string name="game">Επιλογές </string>
<string name="scores">Βαθμολογία</string>

```

```
<string name="menu">Μενού</string>
```

Έπειτα πηγαίνουμε στο φάκελο `/res/value/colors.xml` και προσθέτουμε το χρώμα της λάμπης των γραμμάτων. Επιλέγουμε να είναι ανοιχτό μωβ. Χρησιμοποιούμε τον κωδικό του προκειμένου να το περάσουμε στο χαρακτηριστικό `color`.

```
<color name="menu_glow">#F27BE8</color>
```

Έπειτα πηγαίνουμε στο φάκελο των διαστάσεων `/res/value/dimens.xml` και προσθέτουμε μέγεθος γραμματοσειράς για τα γράμματα του μενού.

```
<dimen name="font_size_menu">14pt</dimen>
```

Τέλος προσθέτουμε τη φωτογραφία που είναι απαραίτητη για να συμπληρωθούν τα στοιχεία του `ImageView` στο φάκελο `res/drawable`.

6.3. Δημιουργία νέου `Layout` με μορφή διάταξης `ListView`

Δημιουργούμε το αρχείο διάταξης `menu_item.xml`, το οποίο περιέχει ένα μηχανισμό ελέγχου `TextView` για να εμφανίζει το όνομα του στοιχείου μενού (Εναρξη, Βοήθεια, κλπ.). Στον μηχανισμό ελέγχου `TextView` εφαρμόζουμε όλα τα τυπικά χαρακτηριστικά εκτός ενός: του ίδιου του κειμένου (δεν υπάρχει κάποιο `string` που να μας παραπέμπει σε κείμενο). Από εδώ μπορούμε να πειραματιστούμε με τα χρώματα του κειμένου, το οποίο έχουμε δημιουργήσει προηγουμένως και το μέγεθος του κειμένου.

Έτσι έχουμε:

menu_item.xml

```
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:textSize="@dimen/menu_item_size"
    android:layout_gravity="center_horizontal"
    android:layout_height="50dp"
    android:shadowRadius="3"
    android:gravity="center"
    android:textColor="@color/menu_color"
    android:textStyle="bold"
    android:shadowColor="@color/menu_glow"
    android:shadowDy="1"
    android:shadowDx="1"
    android:id="@+id/TextView" />
```

colors.xml

```
<color name="menu_color">#CC0066</color>
```

dimens.xml

```
<dimen name="menu_item_size">12pt</dimen>
```

6.4. Δημιουργία της ListView μέσα στην Κύρια Οθόνη Επιλογών

Πηγαίνουμε στον κώδικα της Κύριας Οθόνης Επιλογών (QuizMenuActivity.java) και προσαρμόζουμε τον κώδικα, δημιουργώντας ένα ListView ελεγκτή, μέσα από τον οποίο ο χρήστης θα παραπέμπεται στην κατάλληλη οθόνη μετά από πάτημα μιας από τις επιλογές που εμφανίζονται στην λίστα. Η λίστα αυτή περιέχει τέσσερα στοιχεία. Επομένως χρειαζόμαστε έναν απλό πίνακα μορφής String μίας στήλης με τέσσερις γραμμές. Ανακτούμε τον πίνακα καλώντας τη μέθοδο setContentView() της onCreate()

```
ListView menuList = (ListView) findViewById(R.id.ListView_Menu);
```

Έπειτα καθορίζουμε τις τιμές String που θα χρησιμοποιήσουμε για να συμπληρώσουμε τα στοιχεία του TextView μέσα στον ListView. Εδώ θα συμπληρώσουμε 4 τέσσερις συμβολοσειρές πόρων, που αντιστοιχούν στις επιλογές που θέλουμε να έχουμε στη λίστα μας.

```
String[] items = { getResources().getString(R.string.menu_item_play),  
                  getResources().getString(R.string.menu_item_scores),  
                  getResources().getString(R.string.menu_item_settings),  
                  getResources().getString(R.string.menu_item_help) };
```

Έπειτα πηγαίνουμε στο xml με το όνομα strings.xml και δημιουργούμε τα strings στοιχεία που αναφέραμε παραπάνω. Δηλαδή:

```
<string name="menu_item_play">Εναρξη</string>  
<string name="menu_item_scores">Βαθμολογία</string>  
<string name="menu_item_settings">Ρυθμίσεις</string>  
<string name="menu_item_help">Βοήθεια</string>
```

Χρησιμοποιούμε έναν προσαρμογέα δεδομένων προκειμένου να αντιστοιχίσουμε τα δεδομένα με το πρότυπο διάταξης, το οποίο δημιουργήσαμε προηγουμένως (menu_item.xml). Επειδή έχουμε πίνακα, χρησιμοποιούμε έναν ArrayAdapter:

```
ArrayAdapter<String> adapt =  
    new ArrayAdapter<String>(this,R.layout.menu_item, items);
```

Έπειτα δίνουμε την εντολή στο μηχανισμό ListView να χρησιμοποιήσει τον ArrayAdapter, χρησιμοποιώντας τη μέθοδο setAdapter().

```
menuList.setAdapter(adapt);
```

6.4.1. Ακρόαση για συμβάντα ListView

Η σωστή λειτουργία του μενού προϋποθέτει την «ακρόαση» και απόκριση σε συγκεκριμένα γεγονότα του ελεγκτή ListView. Εδώ θέλουμε να δείξουμε τι θα συμβεί όταν ο χρήστης επιλέξει ένα από τα στοιχεία της λίστας μας. Αυτό το πετυχαίνουμε χρησιμοποιώντας την μέθοδο setOnItemClickListener() του ListView και υλοποιούμε τη μέθοδο onItemClick της κλάσης AdapterView.OnItemClickListener. η μέθοδος αυτή ελέγχει σε ποιο στοιχείο έγινε κλικ κι εκκινεί την κατάλληλη δραστηριότητα της εφαρμογής. Επομένως, έχουμε τον παρακάτω κώδικα:

```
menuList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```

        public void onItemClick(AdapterView<?> parent, View
itemClicked, int
        position, long id) {
            TextView textView = (TextView) itemClicked;
            String strText = textView.getText().toString();
            if
(strText.equalsIgnoreCase(getResources().getString(R.string.menu_item_play)
)) {
                // Εκκίνηση της Game Activity
                startActivity(new Intent(QuizMenuActivity.this,
QuizGameActivity.class));
            } else if

(strText.equalsIgnoreCase(getResources().getString(R.string.menu_ite
m_help))) {
                // Εκκίνηση της Help Activity
                startActivity(new Intent(QuizMenuActivity.this,
QuizHelpActivity.class));
            } else if

(strText.equalsIgnoreCase(getResources().getString(R.string.menu_ite
m_settings))) {
                // Εκκίνηση της Settings Activity
                startActivity(new Intent(QuizMenuActivity.this,
QuizSettingsActivity.class));
            } else if

(strText.equalsIgnoreCase(getResources().getString(R.string.menu_ite
m_scores))) {
                // Εκκίνηση της Scores Activity
                startActivity(new Intent(QuizMenuActivity.this,
QuizScoresActivity.class));
            }
        }
    });

```

6.4.2. Προσθήκη χαρακτηριστικών στο μηχανισμό ελέγχου ListView

Για να ολοκληρωθεί η μορφοποίηση που θέλουμε να δώσουμε στην κύρια Οθόνη επιλογών, θα προσθέσουμε ένα διαιρέτη, μία γραμμή που θα εμφανίζεται ανάμεσα στις τέσσερις επιλογές που έχουμε στη λίστα μας. Ουσιαστικά θα εισάγουμε έναν διαιρέτη ως εικόνα (@drawable/divider.png).

Έτσι μέσα στη ListView μας στο αρχείο menu.xml δηλώνουμε:

```
android:divider="@drawable/divider"
```

Επίσης προσθέτουμε κι έναν επιλογέα έτσι ώστε όταν ο χρήστης διαλέγει μία από τις επιλογές μας αυτή να αλλάζει συμπεριφορά σε σχέση με τις υπόλοιπες. Εδώ θα αλλάξει χρώμα το φόντο της επιλογής μας. Αυτό το δημιουργούμε με τον ίδιο ακριβώς τρόπο όπως προσθέσαμε και το διαιρέτη. Τον εισάγουμε πρώτα ως εικόνα (@drawable/textured.png), κι έπειτα προσθέτουμε τον κώδικά του στην ListView μας:

```
android:listSelector="@drawable/textured"
```



Εικόνα 72: Η Κύρια Οθόνη Επιλογών μετά από τις παραπάνω ρυθμίσεις

6.5. Προσθήκη ενός μενού επιλογών στην Οθόνη του Παιχνιδιού

6.5.1. Προσθήκη Πόρων Μενού

Δημιουργούμε τον φάκελο menu μέσα στον κατάλογο res κι έπειτα το αρχείο gameoptions.xml μέσα στο φάκελο menu.

```
<menu
  xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/scores_menu_item"
    android:icon="@drawable/scores"
    android:title="@string/menu_item_scores"
    android:showAsAction="ifRoom"></item>
  <item
    android:id="@+id/settings_menu_item"
    android:icon="@drawable/prefs"
    android:title="@string/menu_item_settings"
    android:showAsAction="ifRoom"></item>
  <item
    android:id="@+id/help_menu_item"
    android:icon="@drawable/help"
    android:title="@string/menu_item_help"
    android:showAsAction="ifRoom"></item>
</menu>
```

6.5.2. Προσθήκη ενός Μενού Επιλογών σε μια Δραστηριότητα

Δουλεύουμε στην QuizGameActivity

```
public boolean onCreateOptionsMenu(Menu menu)
{
```

```

        super.onCreateOptionsMenu(menu);
        getMenuInflater().inflate(R.menu.gameoptions, menu);

        menu.findItem(R.id.scores_menu_item).setIntent(new Intent(this,
QuizScoresActivity.class));
        menu.findItem(R.id.settings_menu_item).setIntent(new Intent(this,
QuizSettingsActivity.class));
        menu.findItem(R.id.help_menu_item).setIntent(new Intent(this,
QuizHelpActivity.class));

        return true;
    }

```

Για να αντιλαμβάνεται η εφαρμογή ότι ο χρήστης επιλέγει κάποια από τις επιλογές υλοποιούμε την μέθοδο `onOptionsItemSelected()` της δραστηριότητας.

```

@Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        super.onOptionsItemSelected(item);
        startActivity(item.getIntent());
        return true;
    }

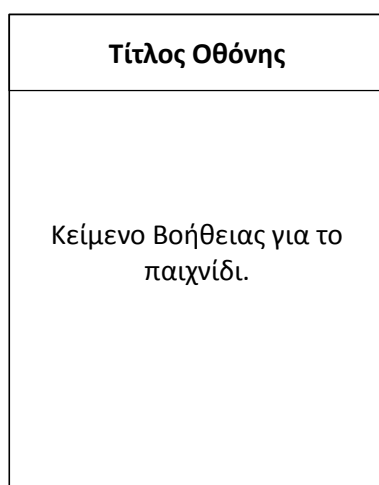
```

ΚΕΦΑΛΑΙΟ 7

Σχεδίαση κι υλοποίηση των Οθονών Βοήθειας και Βαθμολογίας

7.1. Σχεδίαση της Οθόνης Βοήθειας

Η Οθόνη Βοήθειας είναι η οθόνη στην οποία ο χρήστης θα αναζητά πληροφορίες για το τι είναι το παιχνίδι, με τι έχει σχέση και το πώς παίζεται. Επίσης, στην οθόνη αυτή μπορούν να δίνονται κι επιπλέον πληροφορίες για το συγκεκριμένο παιχνίδι (πχ. συχνά προβλήματα που μπορούν να αντιμετωπιστούν άμεσα από τους χρήστες, χωρίς την παρέμβαση του δημιουργού ή του διαχειριστή του παιχνιδιού).



7.1.1. Δημιουργώντας το αρχείο xml της Οθόνης Βοήθειας

Η help.xml περιλαμβάνει ένα μηχανισμό ελέγχου TextView, εκτός από τα αρχικά στοιχεία (τις δύο εικόνες και τον τίτλο της οθόνης), ο οποίος θα δηλώνει τα χαρακτηριστικά που θα έχει το κείμενο της οθόνης.

help.xml

```
<TextView
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:id="@+id/TextView_HelpText"
    android:linksClickable="true"
    android:text="@+id/TextView"
    android:autoLink="all"
    android:textSize="@dimen/help_text_size"
    android:textColor="@color/purple"></TextView>
```

7.1.2. Προσθήκη πόρων Οθόνης Βοήθειας

Προσθέτουμε ένα αρχείο raw στο φάκελο res, δηλαδή ένα αυθαίρετο αρχείο που αποθηκεύεται σε ακατέρραστη μορφή. Στο φάκελο αυτό θα αποθηκεύσουμε ένα αρχείο κειμένου res/raw/quizhelp.txt. το αρχείο κειμένου αυτό θα εντάσσει τον χρήστη στη λογική του παιχνιδιού, για το πώς παίζεται, δηλαδή, το παιχνίδι, το πώς λειτουργεί η

βαθμολόγηση, πώς θα μετριοούνται οι σωστές και λάθος απαντήσεις κλπ. και άλλες γενικές πληροφορίες για την κατασκευή, την έκδοση, τον δημιουργό του παιχνιδιού κ.α. γενικά χαρακτηριστικά του παιχνιδιού.

7.1.3. Ενημέρωση της QuizHelpActivity

```
public class QuizHelpActivity extends QuizActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.help);

        // Read raw file into string and populate TextView
        InputStream iFile = getResources().openRawResource(R.raw.quizhelp);
        try
        {
            TextView helpText = (TextView)
findViewById(R.id.TextView_HelpText);
            String strFile = inputStreamToString(iFile);
            helpText.setText(strFile);
        }
        catch (Exception e)
        {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    @SuppressWarnings("deprecation")
    public String inputStreamToString(InputStream is) throws IOException
    {
        StringBuffer sBuffer = new StringBuffer();
        DataInputStream dataIO = new DataInputStream(is);
        String strLine = null;
        while((strLine = dataIO.readLine()) !=null)
        {
            sBuffer.append(strLine + "\n");
        }
        dataIO.close();
        is.close();

        return sBuffer.toString();
    }
}
```

7.2. Σχεδίαση της Οθόνης Βαθμολογίας

Δημιουργούμε μία οθόνη η οποία θα περιέχει πάνω πάνω τα τρία βασικά χαρακτηριστικά που έχουν κι οι προηγούμενες οθόνες (2 εικόνες και τον τίτλο της οθόνης) κι έπειτα δημιουργούμε έναν πίνακα με Κεφαλίδα και τρεις στήλες (Username, score και rank)

ΕΙΚΟΝΑ	ΒΑΘΜΟΛΟΓΙΑ	ΕΙΚΟΝΑ
<p>Τίτλος</p> <p>Λίστα των scorers</p>		

7.2.1. Δημιουργώντας το αρχείο xml της Οθόνης των Βαθμολογιών

Αρχικά στο αρχείο scores.xml χρησιμοποιούμε τα στοιχεία που χρησιμοποιήσαμε και στα προηγούμενα xml, δηλαδή τον τίτλο και τις εικόνες με την αντίστοιχη διάταξη. Έπειτα χρησιμοποιούμε ένα TextView για τον τίτλο του πίνακα των σκόρερ. Στη συνέχεια χρησιμοποιούμε τη διάταξη ScrollView προκειμένου αν ο πίνακας είναι μεγάλος, να έχουμε τη δυνατότητα να περιηγηθούμε κατεβαίνοντας προς τα κάτω και στις σειρές που δεν φαίνονται στην οθόνη αρχικά. Μέσα στη ScrollView δημιουργούμε έναν πίνακα με τη διάταξη TableLayout κι επίσης καθορίζουμε και τις στήλες με την TableRow και τα επιμέρους χαρακτηριστικά για την κάθε διάταξη.

```
</RelativeLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:id="@+id/LinearLayout">

    <RelativeLayout
        android:id="@+id/RelativeLayout02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/TextView02"
            android:layout_height="wrap_content"
            android:text="@string/top20"
            android:textSize="@dimen/top20_header"
            android:textStyle="bold"
            android:shadowDx="0"
```

```

        android:shadowDy="0"
        android:shadowRadius="10"
        android:layout_width="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textColor="@color/Logo_color"></TextView>

<ScrollView
    android:id="@+id/ScrollViewALLScores"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:scrollbars="vertical"
    android:layout_below="@+id/TextView02">

<TableLayout
    android:id="@+id/TableLayout_AllScores"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*">

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/TableRow"
    android:layout_gravity="center"
    android:gravity="center">
    <TextView
        android:id="@+id/TextView_Nickname"
        android:paddingLeft="3dip"/>
    <TextView
        android:id="@+id/TextView_Score"
        android:paddingRight="3dip"/>
    <TextView
        android:id="@+id/TextView_Rank"
        android:paddingLeft="3dip"/>
</TableRow>
</TableLayout>
</ScrollView>
</RelativeLayout>
</LinearLayout>

```

7.2.2. Προσθήκη πόρων Οθόνης Βαθμολογιών

Στη συνέχεια δημιουργούμε τα strings, το χρώμα που είναι απαραίτητα για την οθόνη βαθμολογιών.

```

<string name="top20"><u>Τα 20 καλύτερα σκορ</u></string>
<string name="username"><i>Username</i></string>
<string name="score"><i>Score</i></string>
<string name="rank"><i>Ranking</i></string>
<string name="no_scores">No scores</string>

<color name="Logo_color">#CC0066</color>

```

Έπειτα δημιουργούμε ένα αρχείο xml το οποίο θα περιέχει κάποιες ψεύτικες βαθμολογίες που θα αντικατοπτρίζουν τον τρόπο με τον οποίο θα δομούνται οι πραγματικές. Έτσι δημιουργούμε το αρχείο /res/xml/allscores.xml όπου καταχωρούμε τρεις χρήστες με την παρακάτω δομή:

```
<?xml version="1.0" encoding="utf-8"?>
<scores>
  <score
    username="Φανή"
    score="100"
    rank="1"/>
  <score
    username="Κώστας"
    score="90"
    rank="2"/>
  <score
    username="Mag"
    score="80"
    rank="3"/>
</scores>
```

Ένα στοιχείο <scores> έχει έναν αριθμό επιμέρους <score> τα οποία περιέχουν τρία χαρακτηριστικά: Username, score και rank.

7.2.3. Ενημέρωση της QuizScoresActivity.java

Πηγαίνουμε τώρα στην QuizScoresActivity για να ρυθμίσουμε κι από κει τον πίνακα των βαθμολογιών. Ξεκινάμε με τη χρήση ενός ελεγκτή `TableLayout` ο οποίος αντικατοπτρίζει τη διάταξη που έχουμε χρησιμοποιήσει στην οθόνη. Δίνουμε την εντολή στο πρόγραμμα να αναγνωρίζει τον πίνακα από το id του.

```
TableLayout allScoresTable =
(TableLayout)findViewById(R.id.TableLayout\_ALLScores);

initializeHeaderRow(allScoresTable);
XmlResourceParser mockAllScores = getResources().getXml(R.xml.allscores);
try {
    processScores(allScoresTable, mockAllScores);
}
catch (Exception e)
{
    //δεν περιέχει κάτι
}
}
// print the headers like "name, score, ranking value"
private void initializeHeaderRow(TableLayout scoreTable)
{
    TableRow headerRow = new TableRow(this);
    int textColor = getResources().getColor(R.color.Logo\_color);
    float textSize =
getResources().getDimension(R.dimen.score\_text\_size);
    addTextToRowWithValues(headerRow,
getResources().getString(R.string.username), textColor, textSize);
    addTextToRowWithValues(headerRow,
getResources().getString(R.string.score), textColor, textSize);
```

```

        addTextToRowWithValues(headerRow,
            getResources().getString(R.string.rank), textColor, textSize);
        scoreTable.addView(headerRow);
    }

    // reading values as score, name, ranking from xml/allscores.xml
    private void processScores(TableLayout scoreTable, XmlResourceParser
scores) throws XmlPullParserException, IOException
    {
        int eventType = -1;
        boolean bFoundScores = false;
        // Find Score records from XML
        while (eventType != XmlResourceParser.END_DOCUMENT)
        {
            if (eventType == XmlResourceParser.START_TAG)
            {
                // Get the name of the tag (eg scores or score)
                String strName = scores.getName();
                if (strName.equals("score"))
                {
                    bFoundScores = true;
                    String scoreValue = scores.getAttributeValue(null,
"score");
                    String scoreRank = scores.getAttributeValue(null,
"rank");
                    String scoreUserName = scores.getAttributeValue(null,
"username");
                    insertScoreRow(scoreTable, scoreValue, scoreRank,
scoreUserName);
                }
            }
            eventType = scores.next();
        }
        // Handle no scores available
        if (bFoundScores == false)
        {
            final TableRow newRow = new TableRow(this);
            TextView noResults = new TextView(this);
            noResults.setText(getResources().
                getString(R.string.no_scores));
            newRow.addView(noResults);
            scoreTable.addView(newRow);
        }
    }

    // creating a row includes attributes as score, name, rank
    private void insertScoreRow(TableLayout scoreTable, String scoreValue,
String scoreRank, String scoreUserName)
    {
        final TableRow newRow = new TableRow(this);
        int textColor = getResources().getColor(R.color.score_color);
        float textSize = getResources().getDimension(R.dimen.score_text_size);
        addTextToRowWithValues(newRow, scoreUserName, textColor, textSize);
        addTextToRowWithValues(newRow, scoreValue, textColor, textSize);
        addTextToRowWithValues(newRow, scoreRank, textColor, textSize);
        scoreTable.addView(newRow);
    }
}

```

```
// printing the score row
private void addTextToRowWithValues(final TableRow tableRow, String text,
int textColor, float textSize)
{
    TextView textView = new TextView(this);
    textView.setTextSize(textSize);
    textView.setTextColor(textColor);
    textView.setText(text);
    tableRow.addView(textView);
} (softy-warely, 2011)
```

ΚΕΦΑΛΑΙΟ 8

Σχεδίαση και Υλοποίηση της Οθόνης των Ρυθμίσεων

8.1. Σχεδίαση της Οθόνης Ρυθμίσεων

Στην οθόνη αυτή ο χρήστης θα μπορεί να διαχειρίζεται τα στοιχεία του. Θα προσθέτει ψευδώνυμο, e-mail, κωδικό ο οποίος θα πρέπει να επαληθεύεται κάθε φορά που χρησιμοποιείται, ημερομηνία γέννησης, με επιλογή από έναν μηχανισμό και τέλος το γένος, με επιλογή επίσης από ένα πτυσσόμενο μηχανισμό στον οποίο ο χρήστης θα έχει τη δυνατότητα να μη δηλώσει κανένα από τα δύο γένη. Το σχεδιάγραμμα της οθόνης είναι το παρακάτω:

logo	Τίτλος οθόνης	logo
<hr/>		
Nickname		
<hr/>		
E-mail		

Κωδικός	Μήνυμα Σ ή Λ Κωδικού	
Ημερομηνία Γέννησης	Μήνυμα με την ημερομηνία που επιλέχθηκε	
Γένος		

8.2. Δημιουργία του layout settings.xml

Στο αρχείο /res/layout/settings.xml, στην αρχή δημιουργούμε τα 3 βασικά στοιχεία που υπάρχουν και στις προηγούμενες οθόνες (εικόνες και τίτλο οθόνης). Έπειτα δημιουργούμε μια δομή ScrollView προκειμένου αν η οθόνη είναι μικρή ο χρήστης να μπορεί να κατεβάζει την μπάρα κύλισης μέχρι το κάτω μέρος της οθόνης.

```
<ScrollView
    android:id="@+id/ScrollViewSettings"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:scrollbars="vertical"
    android:scrollbarAlwaysDrawVerticalTrack="true">
```


Στη συνέχεια δημιουργούμε μέσα σε ένα LinearLayout, ένα μηχανισμό TextView και EditText για το Nickname και το e-mail, ξεχωριστά για το καθένα, έτσι ώστε ο χρήστης να μπορεί να εισάγει το όνομα που θέλει και το mail του και να μπορεί να το επεξεργαστεί.

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/LinearLayout">
    <TextView
        android:id="@+id/TextView_Nickname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/settings_nickname"
        android:textSize="@dimen/settings_text_size"
        android:textColor="@color/purple"
        android:textStyle="bold" />
    <EditText android:id="@+id/EditText_Nickname"
        android:layout_height="wrap_content"
        android:maxLength="20"
        android:layout_width="fill_parent"
        android:maxLines="1"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/TextView_Email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/settings_email"
        android:textSize="@dimen/settings_text_size"
        android:textColor="@color/purple"
        android:textStyle="bold" />
        <EditText
            android:id="@+id/EditText_Email"
            android:layout_height="wrap_content"
            android:layout_width="fill_parent"
            android:maxLines="1"
            android:inputType="textEmailAddress"/>
```

Έπειτα, δημιουργούμε σε ξεχωριστό LinearLayout (μέσα όμως στο προηγούμενο LinearLayout) ένα μηχανισμό για την εισαγωγή και την επαλήθευση του κωδικού του χρήστη μετά από το πάτημα σε ένα κουμπί.

```
<TextView
    android:id="@+id/TextView_Password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/settings_password"
    android:textSize="@dimen/settings_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold"/>
<LinearLayout
    android:id="@+id/LinearLayoutPassword"
    android:orientation="horizontal"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent">
    <Button
        android:id="@+id/Button_Password"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="@string/settings_button_pwd"
        android:onClick="onSetPasswordButtonClick" />
<TextView
    android:id="@+id/TextView_Password_Info"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textSize="@dimen/settings_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold"
    android:gravity="center" />
</LinearLayout>

```

Επίσης σε ξεχωριστό LinearLayout (επίσης μέσα στο αρχικό LinearLayout)δίνουμε τα χαρακτηριστικά του μηχανισμού της επιλογής ημερομηνίας από ένα ημερολόγιο.

```

<TextView
    android:id="@+id/TextView_DOB"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/settings_dob"
    android:textSize="@dimen/settings_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold" />
<LinearLayout
    android:id="@+id/LinearLayout01"
    android:orientation="horizontal"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent">
    <Button
        android:id="@+id/Button_DOB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/settings_button_dob"
        android:onClick="onPickDateButtonClick" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:textSize="@dimen/settings_text_size"
        android:textColor="@color/purple"
        android:textStyle="bold"
        android:gravity="center"
        android:id="@+id/TextView_DOB_Info" />
</LinearLayout>

```

Τέλος έχουμε τα χαρακτηριστικά για την εισαγωγή ενός πτυσσόμενου μενού επιλογής με τη χρήση Spinner για την επιλογή του γένους.

```

<TextView
    android:id="@+id/TextView_Gender"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/settings_gender"
    android:textSize="@dimen/settings_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold" />
<Spinner
    android:id="@+id/Spinner_Gender"
    android:layout_height="wrap_content"

```

```

        android:layout_width="fill_parent"
        android:entries="@array/genders" />
    </LinearLayout>
</ScrollView>

```

8.2.1. Προσθήκη των πόρων για την ολοκλήρωση του αρχείου settings.xml

Ξεκινάμε με το χρώμα που χρησιμοποιούμε σε περίπτωση που οι κωδικοί δεν συμφωνούν (colors.xml).

```
<color name="error_color">#F00</color>
```

Συνεχίζουμε με τα strings που πρέπει να προστεθούν έτσι ώστε να λειτουργήσει σωστά η οθόνη (strings.xml):

```

<string name="settings_nickname">Όνομα</string>
<string name="settings_email">Email</string>
<string name="settings_password">Κωδικός:</string>
<string name="set_password_string">Βάλε κωδικό:</string>
<string name="settings_password2">Επιβεβαίωση κωδικού:</string>
<string name="settings_pwd_set">Κωδικός Σωστός</string>
<string name="settings_pwd_not_set">Λανθασμένος Κωδικός</string>
<string name="settings_pwd_not_equal">Οι κωδικοί δεν ταιριάζουν, προσπάθησε
    Ξανά.</string>
<string name="settings_pwd_equal">Οι κωδικοί ταιρίαξαν.</string>
<string name="settings_button_pwd">Βάλε κωδικό:</string>
<string name="settings_dob">Ημερομηνία Γέννησης:</string>
<string name="settings_dob_not_set">Δεν έχει μπει ημερομηνία</string>
<string name="settings_button_dob">Διάλεξε Ημερομηνία</string>
<string name="settings_gender">Γένος:</string>
<string name="gender_male">Ανδρας</string> //Για το spinner
<string name="gender_female">Γυναίκα</string> //Για το spinner
<string name="gender_neutral">Προτιμώ να μην πω</string> //Για το spinner

```

Ακολουθεί η ενημέρωση του αρχείου dimens.xml για το μέγεθος του κάθε κειμένου που θα χρησιμοποιούμε μέσα στην οθόνη αυτή.

```
<dimen name="settings_text_size">6pt</dimen>
```

8.2.2. Προσθήκη πόρων Πίνακα Συμβολοσειρών

Για τη λειτουργία του μηχανισμού ελέγχου Spinner θα δημιουργήσουμε έναν αρχείο xml το οποίο θα περιέχει έναν πίνακα με τα στοιχεία που θα εμφανίζονται σε λίστα επιλογής όταν κάποιος πατάει το κουτί με την επιλογή του γένους. Οι επιλογές θα είναι τρεις επομένως κι ο πίνακας θα έχει τρεις σειρές. Έτσι το νέο αρχείο /res/values/arrays.xml θα έχει την παρακάτω μορφή:

```

<resources>
    <string-array name="genders">
        <item>@string/gender_male</item>
        <item>@string/gender_female</item>
        <item>@string/gender_neutral</item>
    </string-array>
</resources>

```

8.3. Αποθήκευση δεδομένων της Οθόνης Ρυθμίσεων με τη χρήση του μηχανισμού SharedPreferences

Η χρήση του μηχανισμού SharedPreferences μας βοηθά να αποθηκεύσουμε τα δεδομένα που έχει πληκτρολογήσει κι επιλέξει ο χρήστης στη φόρμα – οθόνη των ρυθμίσεων.

Ενημερώνουμε την QuizActivity με τα χαρακτηριστικά που μας είναι απαραίτητα για την ενημέρωση της QuizSettingsActivity, προσθέτοντας μεταβλητές sting για κάθε μία από τις ρυθμίσεις.

```
public static final String GAME_PREFERENCES_NICKNAME = "Nickname";
// String
public static final String GAME_PREFERENCES_EMAIL = "Email"; // String
public static final String GAME_PREFERENCES_PASSWORD = "Password";
// String
public static final String GAME_PREFERENCES_DOB = "DOB"; // Long
//DOB= Date of Birth
public static final String GAME_PREFERENCES_GENDER = "Gender"; // Integer,
στο συγκεκριμένο πίνακα (in array order): Άνδρας (1), Γυναίκα (2), και
Προτιμώ να μην πω (0)
```

8.4. Ενημέρωση της QuizSettingsActivity.java

Ο κώδικας για την QuizSettingsActivity είναι ο παρακάτω:

```
public class QuizSettingsActivity extends QuizActivity
{
    SharedPreferences mGameSettings;

    //Αρχικοποίηση των id' s
    static final int DATE_DIALOG_ID = 0;
    static final int PASSWORD_DIALOG_ID = 1;

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

        // Ανάκτηση των κοινών προτιμήσεων
        mGameSettings = getSharedPreferences(GAME_PREFERENCES,
            Context.MODE_PRIVATE);

        // Αρχικοποιεί την εισαγωγή του nickname
        initNicknameEntry();
        // Αρχικοποιεί την εισαγωγή του email
        initEmailEntry();
        // Αρχικοποιεί τον επιλογέα Password
        initPasswordChooser();
        // Αρχικοποιεί τον Date picker
        initDatePicker();
        // Αρχικοποιεί τον spinner
        initGenderSpinner();
    }
}
```

```

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

    //Αποθηκεύει το κείμενο που δίνεται από το χρήστη ως Nickname
    EditText nicknameText = (EditText)
    findViewById(R.id.EditText_Nickname);

    //Αποθηκεύει το mail του χρήστη
    EditText emailText = (EditText)
    findViewById(R.id.EditText_Email);

    //Εξάγει το Nickname του χρήστη
    String strNickname = nicknameText.getText().toString();
    //Εξάγει το mail του χρήστη
    String strEmail = emailText.getText().toString();

    Editor editor = mGameSettings.edit();
    editor.putString(GAME_PREFERENCES_NICKNAME, strNickname);
    editor.putString(GAME_PREFERENCES_EMAIL, strEmail);

    editor.commit();
}

@Override
protected void onDestroy()
{
    Log.d(DEBUG_TAG, "SHARED PREFERENCES");
    Log.d(DEBUG_TAG, "Nickname is: " + m
    GameSettings.getString(GAME_PREFERENCES_NICKNAME, "Not set"));
    Log.d(DEBUG_TAG, "Email is: " +
    mGameSettings.getString(GAME_PREFERENCES_EMAIL, "Not set"));
    Log.d(DEBUG_TAG, "Gender (M=1, F=2, U=0) is: " +
    mGameSettings.getInt(GAME_PREFERENCES_GENDER, 0));
    // We are not saving the password yet
    Log.d(DEBUG_TAG, "Password is: " +
    mGameSettings.getString(GAME_PREFERENCES_PASSWORD, "Not
    set"));
    // We are not saving the date of birth yet
    Log.d(DEBUG_TAG, "DOB is: " + DateFormat.format("MMMM dd,
    yyyy", mGameSettings.getLong(GAME_PREFERENCES_DOB, 0)));
    super.onDestroy();
}

/**
 * Initialize the nickname entry
 */
private void initNicknameEntry()
{
    EditText nicknameText = (EditText)
    findViewById(R.id.EditText_Nickname);
    if (mGameSettings.contains(GAME_PREFERENCES_NICKNAME))
    {
        nicknameText.setText(mGameSettings.getString
        (GAME_PREFERENCES_NICKNAME, ""));
    }
}

```

```

}

/**
 * Initialize the email entry
 */
private void initEmailEntry()
{
    EditText emailText = (EditText)
    findViewById(R.id.EditText_Email);
    if (mGameSettings.contains(GAME_PREFERENCES_EMAIL))
    {
        emailText.setText(mGameSettings.getString
            (GAME_PREFERENCES_EMAIL, ""));
    }
}

/**
 * Initialize the Password chooser
 */
private void initPasswordChooser()
{
    // Set password info
    TextView passwordInfo = (TextView)
    findViewById(R.id.TextView_Password_Info);
    if (mGameSettings.contains(GAME_PREFERENCES_PASSWORD))
    {
        passwordInfo.setText(R.string.settings_pwd_set);
    }
    else
    {
        passwordInfo.setText(R.string.settings_pwd_not_set);
    }
}

/**
 * Called when the user presses the Set Password button
 *
 * @param view
 *         the button
 */
@SuppressWarnings("deprecation")
public void onSetPasswordButtonClick(View view)
{
    showDialog(PASSWORD_DIALOG_ID);
}

/**
 * Initialize the Date picker
 */
private void initDatePicker()
{
    // Set password info
    TextView dobInfo = (TextView)
    findViewById(R.id.TextView_DOB_Info);
    if (mGameSettings.contains(GAME_PREFERENCES_DOB))
    {
        dobInfo.setText(DateFormat.format("MMMM dd, yyyy",
            mGameSettings.getLong(GAME_PREFERENCES_DOB, 0)));
    }
}

```

```

        else
        {
            dobInfo.setText(R.string.settings_dob_not_set);
        }
    }

/**
 * Called when the user presses the Pick Date button
 *
 * @param view
 *         The button
 */

@SuppressWarnings("deprecation")

public void onPickDateButtonClick(View view)
{
    showDialog(DATE_DIALOG_ID);
}

/**
 * Initialize the spinner
 */
private void initGenderSpinner()
{
    // Populate Spinner control with genders
    final Spinner spinner = (Spinner)
        findViewById(R.id.Spinner_Gender);

    ArrayAdapter<?> adapter =
        ArrayAdapter.createFromResource(this, R.array.genders,
            android.R.layout.simple_spinner_item);

    adapter.setDropDownViewResource
        (android.R.layout.simple_spinner_dropdown_item);

    spinner.setAdapter(adapter);
    if (mGameSettings.contains(GAME_PREFERENCES_GENDER))
    {
        spinner.setSelection(mGameSettings.getInt
            (GAME_PREFERENCES_GENDER, 0));
    }
    // Handle spinner selections
    spinner.setOnItemClickListener(new
        AdapterView.OnItemClickListener()
        {
            public void onItemClick(AdapterView<?> parent, View
                itemSelected, int selectedItemPosition, long
                selectedId)
            {
                Editor editor = mGameSettings.edit();
                editor.putInt(GAME_PREFERENCES_GENDER,
                    selectedItemPosition);
                editor.commit();
            }

            public void onNothingSelected(AdapterView<?> parent)
            {
            }
        }
    )
}

```

```

    });
}

@Override
protected Dialog onCreateDialog(int id)
{
    switch (id)
    {
        case DATE_DIALOG_ID:
            final TextView dob = (TextView)
                findViewById(R.id.TextView_DOB_Info);
            Calendar now = Calendar.getInstance();

            DatePickerDialog dateDialog = new DatePickerDialog(this, new
                DatePickerDialog.OnDateSetListener()
            {
                public void onDateSet(DatePicker view, int year, int
                    monthOfYear, int dayOfMonth)
                {
                    Time dateOfBirth = new Time();
                    dateOfBirth.set(dayOfMonth, monthOfYear, year);
                    long dtDob = dateOfBirth.toMillis(true);
                    dob.setText(DateFormat.format("MMMM dd, yyyy",
                        dtDob));

                    Editor editor = mGameSettings.edit();
                    editor.putLong(GAME_PREFERENCES_DOB, dtDob);
                    editor.commit();
                }
            },

                now.get(Calendar.YEAR), now.get(Calendar.MONTH),
                now.get(Calendar.DAY_OF_MONTH));
            return dateDialog;

        case PASSWORD_DIALOG_ID:
            LayoutInflater inflater = (LayoutInflater)
                getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            final View layout = inflater.inflate(R.layout.password_dialog,
                (ViewGroup) findViewById(R.id.root));
            final EditText p1 = (EditText)
                layout.findViewById(R.id.EditText_Pwd1);
            final EditText p2 = (EditText)
                layout.findViewById(R.id.EditText_Pwd2);
            final TextView error = (TextView)
                layout.findViewById(R.id.TextView_PwdProblem);
            p2.addTextChangedListener(new TextWatcher()
            {
                public void afterTextChanged(Editable s)
                {
                    String strPass1 = p1.getText().toString();
                    String strPass2 = p2.getText().toString();
                    if (strPass1.equals(strPass2))
                    {
                        error.setText(R.string.settings_pwd_equal);
                    }
                    else
                    {

```



```

        error.setText(R.string.settings_pwd_not_equal);
    }
}

// ... other required overrides need not be implemented

public void beforeTextChanged(CharSequence s, int
start, int count, int after)
{

}

public void onTextChanged(CharSequence s, int start,
int before, int count)
{

}
});
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setView(layout);
// Now configure the AlertDialog
builder.setTitle(R.string.settings_button_pwd);
builder.setNegativeButton(android.R.string.cancel, new
DialogInterface.OnClickListener()
{
    @SuppressWarnings("deprecation")
    public void onClick(DialogInterface dialog, int
whichButton)
    {
        // We forcefully dismiss and remove the Dialog, so it
        // cannot be used again (no cached info)
        QuizSettingsActivity.this.removeDialog
(PASSWORD_DIALOG_ID);
    }
});
builder.setPositiveButton(android.R.string.ok, new
DialogInterface.OnClickListener()
{
    @SuppressWarnings("deprecation")
    public void onClick(DialogInterface dialog, int which)
    {
        TextView passwordInfo = (TextView)
findViewById(R.id.TextView_Password_Info);
String strPassword1 = p1.getText().toString();
String strPassword2 = p2.getText().toString();
if (strPassword1.equals(strPassword2))
        {
            Editor editor = mGameSettings.edit();
            editor.putString(GAME_PREFERENCES_PASSWORD,
strPassword1);
            editor.commit();
            passwordInfo.setText(R.string.settings_pwd_set);
        }
        else
        {
            Log.d(DEBUG_TAG, "Passwords do not match. Not
saving. Keeping old password (if set).");
        }
        // We forcefully dismiss and remove the Dialog, so it
        // cannot be used again

```

```

        QuizSettingsActivity.this.removeDialog(PASSWORD_DIALOG_
        ID);
    }
});
// Create the AlertDialog and return it
AlertDialog passwordDialog = builder.create();
return passwordDialog;
}
return null;
}

@SuppressWarnings("deprecation")
@Override
protected void onPrepareDialog(int id, Dialog dialog)
{
    super.onPrepareDialog(id, dialog);
    switch (id) {
        case DATE_DIALOG_ID:
            // Handle any DatePickerDialog initialization here
            DatePickerDialog dateDialog = (DatePickerDialog)
            dialog;
            int iDay,
            iMonth,
            iYear;
            // Check for date of birth preference
            if (mGameSettings.contains(GAME_PREFERENCES_DOB))
            {
                // Retrieve Birth date setting from preferences
                long msBirthDate =
                mGameSettings.getLong(GAME_PREFERENCES_DOB, 0);
                Time dateOfBirth = new Time();
                dateOfBirth.set(msBirthDate);
                iDay = dateOfBirth.monthDay;
                iMonth = dateOfBirth.month;
                iYear = dateOfBirth.year;
            }
            else
            {
                Calendar cal = Calendar.getInstance();
                // Today's date fields
                iDay = cal.get(Calendar.DAY_OF_MONTH);
                iMonth = cal.get(Calendar.MONTH);
                iYear = cal.get(Calendar.YEAR);
            }
            // Set the date in the DatePicker to the date of birth OR to
            // the current date
            dateDialog.updateDate(iYear, iMonth, iDay);
            return;
        case PASSWORD_DIALOG_ID:
            // Handle any Password Dialog initialization here
            // Since we don't want to show old password dialogs, just set
            // ones, we need not do anything here
            // Because we are not "reusing" password dialogs once they have
            // finished, but removing them from
            // the Activity Dialog pool explicitly with removeDialog() and
            // recreating them as needed.
            return;
    }
}

```

}
}
} (Conder, 2012)

ΚΕΦΑΛΑΙΟ 9

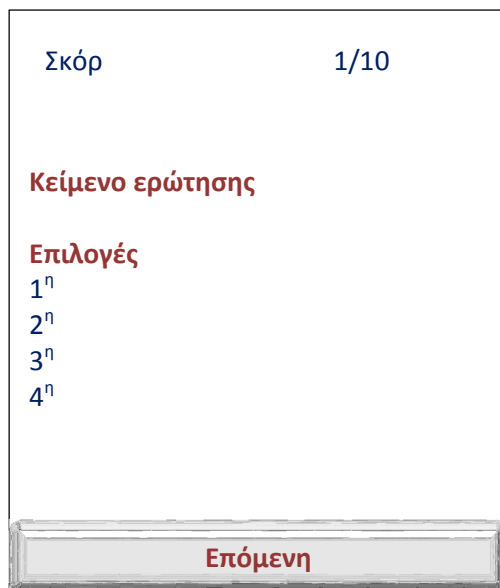
Σχεδίαση κι υλοποίηση της Κύριας Οθόνης του Παιχνιδιού

Ένα από τα τελευταία μέρη της εργασίας αλλά και το σπουδαιότερο είναι η σχεδίαση κι υλοποίηση της οθόνης του παιχνιδιού. Η οθόνη που θα διεξάγεται το παιχνίδι κι η οποία παίζει το σημαντικότερο ρόλο στην επιτυχία ενός τέτοιου παιχνιδιού. Θα πρέπει να προσελκύει το χρήστη με τη διαδραστικότητα της και να τον κάνει να θέλει να ξαναπαίξει.

9.1. Σχεδίαση της Κύριας Οθόνης της εφαρμογής

Μέχρι τώρα στην κύρια Οθόνη είχαμε να εμφανίζονται μόνο τα βοηθητικά μενού στην μπάρα τίτλου του παιχνιδιού (βαθμολογία, ρυθμίσεις, βοήθεια). Στην οθόνη αυτή ο χρήστης έχει πρόσβαση πατώντας στην επιλογή έναρξη. Μπαίνοντας στην οθόνη, το παιχνίδι αρχίζει και τρέχει. Αριστερά στο επάνω μέρος θα φαίνεται το σκορ του παίκτη, ενώ αριστερά θα φαίνεται η αρίθμηση των ερωτήσεων, δηλαδή, σε ποια ερώτηση είμαστε την κάθε στιγμή. Στη συνέχεια θα εμφανίζεται η ερώτηση με τέσσερις επιλογές για την σωστή απάντηση. Τέλος θα υπάρχει ένα κουμπί στο κάτω μέρος της οθόνης, το οποίο θα μας οδηγεί στην επόμενη ερώτηση.

Έτσι έχουμε περίπου το παρακάτω σχεδιάγραμμα:



9.2. Δημιουργία του game.xml

Για να δημιουργήσουμε το /res/layout/game.xml χρειαζόμαστε ένα RelativeLayout με δύο TextView στοιχισμένα δεξιά και αριστερά (στις νεότερες εκδόσεις δηλώνονται ως τέλος κι αρχή αντίστοιχα), για την ερώτηση και το σκορ αντίστοιχα.

```
<RelativeLayout
    android:layout_height="wrap_content"
    android:id="@+id/RelativeLayout02"
```

```

        android:layout_alignParentTop="true"
        android:layout_width="fill_parent"
        android:padding="10dp"
        android:gravity="center">
<TextView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textSize="@dimen/game_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold"
    android:text="@string/game_score"
    android:gravity="start"
    android:id="@+id/TextView_CurrentScore" />
<TextView
    android:id="@+id/questionNumberTextView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textSize="@dimen/game_text_size"
    android:text="@string/question"
    android:textColor="@color/purple"
    android:textStyle="bold"
    android:gravity="end"/>

</RelativeLayout>

```

Έπειτα δημιουργούμε το LinearLayout το οποίο θα περιέχει τις ερωτήσεις μας.

Θέτουμε όλα τα `android:layout_alignParent= "True"` εκτός από το `android:layout_alignParentTop="false"` που το θέτουμε σε κατάσταση false προκειμένου το Layout να τοποθετείται κάτω από το προηγούμενο Layout.⁶ Θέτουμε το `orientation = "vertical"` ώστε τα στοιχεία μας να στοιχισθούν το ένα κάτω από το άλλο.

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentTop="false"
    android:gravity="center"
    android:orientation="vertical" >

```

Στη συνέχεια, δημιουργούμε την ερώτηση (TextView) και τις πιθανές μας απαντήσεις με μορφή Radiobuttons⁷. Για το παιχνίδι μας χρειαζόμαστε τέσσερα radiobuttons. Όλα αυτά τοποθετούνται μέσα σε μια διάταξη RadioGroup. Χρησιμοποιούμε, επίσης, έναν μηχανισμό ScrollView για τη δημιουργία μπάρας κύλισης, έτσι ώστε στην πιθανότητα μιας μικρής

⁶ Το `android:layout_alignParentTop="True"` δίνει την εντολή στο Layout να έχει ως αρχή του την πάνω πλευρά του θυγατρικού Layout που ήδη έχουμε.

⁷ Radio Buttons: Κυκλικά κουμπιά επιλογής που επιτρέπουν στον χρήστη μία επιλογή από ένα σύνολο. Όταν ο χρήστης διαλέξει ένα από αυτά, αυτό γεμίζει (τσεκάρεται), ενώ όλα τα υπόλοιπα αδειάζουν (δεν φαίνονται τσεκαρισμένα).

οθόνης να φαίνονται όλες οι πιθανές απαντήσεις κυλώντας την μπάρα κύλισης προς τα κάτω.

```
<ScrollView
    android:id="@+id/ScrollViewGame"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:scrollbars="vertical"
    android:scrollbarAlwaysDrawVerticalTrack="true">
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/LinearLayout">
<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="fill_parent"
    android:layout_height="0dip"
    android:layout_weight="0.04"
    android:gravity="top"
    android:orientation="vertical"
    android:paddingBottom="@dimen/padding_bottom1" >

<RadioButton
    android:id="@+id/radio0"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.02"
    android:textSize="@dimen/game_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold"
/>

<RadioButton
    android:id="@+id/radio1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.02"
    android:textColor="@color/purple"
    android:textSize="@dimen/game_text_size"
    android:textStyle="bold"
/>

<RadioButton
    android:id="@+id/radio2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.02"
    android:textSize="@dimen/game_text_size"
    android:textColor="@color/purple"
    android:textStyle="bold"
/>

<RadioButton
    android:id="@+id/radio3"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.02"
    android:textColor="@color/purple"
```

```
        android:textSize="@dimen/game_text_size"
        android:textStyle="bold"
    />
```

```
</RadioGroup>
```

Τέλος προσθέτουμε το κουμπί που θα μας οδηγήει στην επόμενη ερώτηση.

```
        <Button
            android:id="@+id/button1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="@color/purple"
            android:text="@string/str_next" />
    </LinearLayout>
</ScrollView>
```

```
</LinearLayout>
```

9.3. Δημιουργία της κλάσης Question.java

Δημιουργούμε την κλάση αυτή για την καλύτερη εξυπηρέτηση του προγράμματός μας. Είναι καλύτερα να αναφέρεσαι σε ένα αντικείμενο παρά σε μια ομάδα αλφαριθμητικών, όταν καλείς μια ερώτηση.

Έτσι δημιουργούμε την βοηθητική κλάση Question όπου δηλώνουμε την ερώτηση, τις τέσσερις πιθανές απαντήσεις και την σωστή απάντηση. Επίσης δηλώνουμε και το id που θα έχει η κάθε ερώτηση. Επειδή έχουμε σκοπό να δημιουργήσουμε μία βάση με τις ερωτήσεις, η κάθε ερώτηση θα έχει το δικό της μοναδικό id. Έπειτα με ξεχωριστές μεθόδους καλούμε ξεχωριστά τα παραπάνω στοιχεία. Έπειτα θέτουμε αυτά τα στοιχεία και πάλι σε ξεχωριστές μεθόδους το καθένα.

```
public class Question {
    private int ID;
    private String QUESTION;
    private String OPTA;
    private String OPTB;
    private String OPTC;
    private String OPTD;
    private String ANSWER;
    public Question()
    {
        ID=0;
        QUESTION="";
        OPTA="";
        OPTB="";
        OPTC="";
        OPTD="";
        ANSWER="";
    }
    public Question(String qQUESTION, String oPTA, String oPTB, String oPTC, String oPTD,
        String aANSWER) {
```

```

        QUESTION = qQUESTION;
        OPTA = oPTA;
        OPTB = oPTB;
        OPTC = oPTC;
        OPTD = oPTD;
        ANSWER = aANSWER;
    }
    public int getID()
    {
        return ID;
    }
    public String getQUESTION() {
        return QUESTION;
    }
    public String getOPTA() {
        return OPTA;
    }
    public String getOPTB() {
        return OPTB;
    }
    public String getOPTC() {
        return OPTC;
    }
    public String getOPTD() {
        return OPTD;
    }

    public String getANSWER() {
        return ANSWER;
    }
    public void setID(int id)
    {
        ID=id;
    }
    public void setQUESTION(String qQUESTION) {
        QUESTION = qQUESTION;
    }
    public void setOPTA(String oPTA) {
        OPTA = oPTA;
    }
    public void setOPTB(String oPTB) {
        OPTB = oPTB;
    }
    public void setOPTC(String oPTC) {
        OPTC = oPTC;
    }
    public void setOPTD(String oPTD) {
        OPTD = oPTD;
    }
    public void setANSWER(String aANSWER) {
        ANSWER = aANSWER;
    }

```



```
}  
} (A Simple Quiz Game in Android, 2013)
```

9.4. Δημιουργία της κλάσης DBHelper.java

Αφού δημιουργήσαμε την κλάση της ερώτησης τώρα είναι ώρα να δημιουργήσουμε τη βάση. Χρησιμοποιούμε μία κλάση που θα μας βοηθήσει στην εκτέλεση των λειτουργιών της βάσης δεδομένων. Η κλάση αυτή, η DBHelper, θα κληρονομεί την SQLiteOpenHelper. Επίσης προσθέτουμε δύο μεθόδους, την onCreate() και την onUpgrade().

Η onCreate() καλείται μόνο στην αρχή προκειμένου να αποκτήσουμε πρόσβαση στην βάση. Περιέχει όλα τα στοιχεία που απαρτίζουν τη βάση.

Η onUpgrade() καλείται σε κάθε περίπτωση που αλλάζει κάτι στη βάση. (Building a TODO List app in Android using SQLite, 2013)

Έτσι έχουμε την παρακάτω κλάση χρησιμοποιώντας και τη λογική που λειτουργεί μία βάση δεδομένων, δηλαδή, το μοναδικό κλειδί για τη βάση, δήλωση του κάθε στοιχείου και το τι είναι το καθένα (Integer, Text κλπ.):

```
public class DBHelper extends SQLiteOpenHelper  
{  
    private static final int DATABASE_VERSION = 1;  
    // Το όνομα της βάσης  
    private static final String DATABASE_NAME = "triviaQuiz";  
    // Το όνομα των εργασιών του πίνακα  
    private static final String TABLE_QUEST = "quest";  
    // Τα ονόματα των στηλών του πίνακα  
    private static final String KEY_ID = "id";  
    private static final String KEY_QUES = "question";  
    private static final String KEY_ANSWER = "answer"; //σωστή απάντηση  
    private static final String KEY_OPTA= "opta"; //option a  
    private static final String KEY_OPTB= "optb"; //option b  
    private static final String KEY_OPTC= "optc"; //option c  
    private static final String KEY_OPTD= "optd"; //option d  
    private SQLiteDatabase dbase;  
  
    public DBHelper(Context context)  
    {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    }  
    //Καλούμε την βάση  
    @Override  
    public void onCreate(SQLiteDatabase db)  
    {  
        dbase=db;  
        String sql = "CREATE TABLE IF NOT EXISTS " + TABLE_QUEST + " ( "+  
KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + KEY_QUES + " TEXT,  
" + KEY_ANSWER+ " TEXT, "+KEY_OPTA + " TEXT, " +KEY_OPTB + " TEXT,  
"+KEY_OPTC + " TEXT, "+KEY_OPTD + " TEXT)";  
        db.execSQL(sql);  
        addQuestions();  
        //db.close();  
    }  
  
    //Προσθήκη των ερωτήσεων στη βάση
```

```

private void addQuestions()
{
    Question q1=new Question("Ποια χώρα δηλώνει διεθνώς " + "τα
λατινικά γράμματα SE;", "Σλοβενία", "Σερβία", "Σουηδία",
"Σλοβακία", "Σουηδία");
    this.addQuestion(q1);
    Question q2=new Question("Σε ποια χώρα βρίσκονται οι περιοχές"
+ " της Λομβαρδίας και της Τοσκάνης;", "Ιταλία", "Γαλλία",
"Ελβετία", "Γερμανία", "Ιταλία");
    this.addQuestion(q2);

//Εδώ προσθέτουμε όσες ερωτήσεις είναι απαραίτητες για το παιχνίδι
μας ακολουθώντας την ίδια λογική με τις παραπάνω ερωτήσεις.
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {
    // Αν υπάρχει προηγούμενος πίνακας τον διαγράφει
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_QUEST);
    // Δημιουργεί τους πίνακες ξανά
    onCreate(db);
}
// Προσθήκη νέων ερωτήσεων
public void addQuestion(Question quest) {
    //SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_QUEST, quest.getQUESTION());
    values.put(KEY_ANSWER, quest.getANSWER());
    values.put(KEY_OPTA, quest.getOPTA());
    values.put(KEY_OPTB, quest.getOPTB());
    values.put(KEY_OPTC, quest.getOPTC());
    values.put(KEY_OPTD, quest.getOPTD());
    // Εισαγωγή γραμμής
    dbase.insert(TABLE_QUEST, null, values);
}
public List<Question> getAllQuestions()
{
    List<Question> quesList = new ArrayList<Question>();
    // Συλλογή όλων των ερωτημάτων
    String selectQuery = "SELECT * FROM " + TABLE_QUEST;
    dbase=this.getReadableDatabase();
    Cursor cursor = dbase.rawQuery(selectQuery, null);
    // looping through all rows and adding to list
    if (cursor.moveToFirst())
    {
        do {
            Question quest = new Question();
            quest.setID(cursor.getInt(0));
            quest.setQUESTION(cursor.getString(1));
            quest.setANSWER(cursor.getString(2));
            quest.setOPTA(cursor.getString(3));
            quest.setOPTB(cursor.getString(4));
            quest.setOPTC(cursor.getString(5));
            quest.setOPTD(cursor.getString(6));
            quesList.add(quest);
        } while (cursor.moveToNext());
    }
    // επιστρέφει την λίστα των ερωτήσεων
    return quesList;
}
}

```

```

    public int rowcount()
    {
        int row=0;
        String selectQuery = "SELECT * FROM " + TABLE_QUEST;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);
        row=cursor.getCount();
        return row;
    }
} (A Simple Quiz Game in Andriod, 2013)

```

9.5. Δουλεύοντας με την QuizGameActivity.java

Πριν αρχίσουμε να δουλεύουμε με την QuizGameActivity πηγαίνουμε στην QuizActivity και δηλώνουμε τον αριθμό των ερωτήσεων που θα πρέπει ο χρήστης να απαντήσει.

```
public static final int QUESTION_BATCH_SIZE = 10;
```

Επιστρέφουμε τώρα στην QuizGameActivity και δηλώνουμε τον πίνακά μας καθώς και τα στοιχεία που μας είναι απαραίτητα (κουμπιά, id' s, σκορ, TextViews και Question)

```

List<Question> quesList;
    int score=0; //αρχικοποιεί το score
    int qid=0; //αρχικοποιεί τα id' s
    Question curQuestion; //δηλώνει την ερώτηση που είμαστε
    TextView txtQuestion; //δηλώνει το κείμενο της ερώτησης
    RadioButton rda, rdb, rdc, rdd; //είναι οι επιλογές των πιθανών
    απαντήσεων σε μορφή radiobuttons
    Button butNext; // δηλώνει το κουμπί για τη μετάβασή μας στην
    επόμενη ερώτηση #
    private TextView questionNumberTextView; // δείχνει τον αριθμό της
    τρέχουσας ερώτησης #
    private TextView textView_CurrentScore; // δείχνει το score του
    παίχτη μετά την απάντηση της κάθε ερώτησης #

```

Έπειτα μέσα στην onCreate() της QuizGameActivity χρησιμοποιούμε τον παρακάτω κώδικα:

```

DBHelper db=new DBHelper(this);
quesList=db.getAllQuestions();
curQuestion=quesList.get(qid);

// δήλωση των στοιχείων
questionNumberTextView =
    (TextView)findViewById(R.id.questionNumberTextView);

textView_CurrentScore = (TextView)findViewById(R.id.textView_CurrentScore);

txtQuestion=(TextView)findViewById(R.id.textView1);
rda=(RadioButton)findViewById(R.id.radio0);
rdb=(RadioButton)findViewById(R.id.radio1);
rdc=(RadioButton)findViewById(R.id.radio2);
rdd=(RadioButton)findViewById(R.id.radio3);
butNext=(Button)findViewById(R.id.button1);

```

```

setQuestionView();

//ανακάτεμα των ερωτήσεων
Collections.shuffle(quesList);

//Χρησιμοποίηση της μεθόδου OnClickListener για την ενεργοποίηση του
κουμπιού που μας μεταφέρει στην επόμενη ερώτηση

butNext.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        RadioGroup grp=(RadioGroup)findViewById(R.id.radioGroup1);
        RadioButton answer=
            (RadioButton)findViewById(grp.getCheckedRadioButtonId());
        Log.d("yourans", curQuestion.getANSWER()+" "+answer.getText());

        if(curQuestion.getANSWER().equals(answer.getText()))
        {
            score= score + 10; //αν ο παίχτης απαντήσει σωστά τότε
            προστίθενται 10 μονάδες στο σκορ
            Log.d("score", "Your score"+score);

            // δίνει το σκορ εκείνη τη στιγμή
            textView_CurrentScore.setText(
                getResources().getString(R.string.game_score, score));
        }
        else
        {
            score=score - 5; //αν ο παίχτης απαντήσει λάθος
            μειώνεται κατά 5 μονάδες το προηγούμενο σκορ του

            Log.d("score", "Your score"+score);
            textView_CurrentScore.setText(
                getResources().getString(R.string.game_score, score));
        }
        if(qid<10)
        {
            curQuestion=quesList.get(qid);
            setQuestionView();

        }else{
            startActivity(new Intent(QuizGameActivity.this,
                QuizScoresActivity.class));
            QuizGameActivity.this.finish();
        }
    }
});

// Δείχνει το κείμενο της ερώτησης. Στην περίπτωση μας είναι ο
αριθμός της ερώτησης σε συνάρτηση με το σύνολο των
ερωτήσεων(πχ.1/10)
questionNumberTextView.setText(
    getResources().getString(R.string.question, 1,
    QUESTION_BATCH_SIZE));

// δείχνει το κείμενο του σκορ

```

```
        textView_CurrentScore.setText(  
            getResources().getString(R.string.game_score, 0));  
    }  
  
    private void setQuestionView()  
    {  
        txtQuestion.setText(curQuestion.getQUESTION());  
        rda.setText(curQuestion.getOPTA());  
        rdb.setText(curQuestion.getOPTB());  
        rdc.setText(curQuestion.getOPTC());  
        rdd.setText(curQuestion.getOPTD());  
        qid++;  
  
        // display current question number  
        questionNumberTextView.setText(  
            getResources().getString(R.string.question, qid,  
            QUESTION_BATCH_SIZE));  
    }  
}
```

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

1. Συμπεράσματα

Στην εργασία αυτή ήρθαμε σε επαφή για πρώτη φορά με το λειτουργικό σύστημα Android. Δεν γνωρίζαμε πολλά γι' αυτό, παρά μόνο ότι χρειαζόταν γνώσεις Java για να προγραμματίσεις σε αυτό.

Μετά από όλον αυτόν τον καιρό που δουλέψαμε πάνω σε αυτό μπορούμε να πούμε ότι:

Πρώτα απ' όλα καταφέραμε να υλοποιήσουμε εφαρμογή σε Android. Αυτό σημαίνει ότι κατανοήσαμε σε μεγάλο βαθμό πως λειτουργεί η όλη διαδικασία.

Δεύτερον, εμβαθύνσαμε στον προγραμματισμό σε Java (γιατί πράγματι το Android έχει άμεση σχέση με αυτή τη γλώσσα προγραμματισμού). Επίσης εξοικειωθήκαμε στον προγραμματισμό σε xml ο οποίος ήταν σχεδόν άγνωστος σε μας, αλλά απαραίτητος για την υλοποίηση εφαρμογών σε Android.

Ήρθαμε σε επαφή και με τα εργαλεία που βοηθούν στην υλοποίηση μιας εφαρμογής σε Android. Το ότι αυτά τα εργαλεία αλλά και μεγάλο κομμάτι του κώδικα βρίσκεται ελεύθερο στο Internet χωρίς κόστος, μας ενθάρρυνε ακόμη περισσότερο στην ενασχόλησή μας με το συγκεκριμένο αντικείμενο.

Όλα αυτά μας βοήθησαν στην εξοικείωσή μας με το Android. Τώρα πια δεν είναι κάτι άγνωστο, αλλά κάτι πολύ οικείο σε μας, τόσο ώστε να θέλουμε να συνεχίσουμε την ενασχόλησή μας με αυτό.

2. Μελλοντική Εργασία

Στην εργασία αυτή προσπαθήσαμε να υλοποιήσουμε ένα παιχνίδι ερωτήσεων σε Android. Στο χρονικό διάστημα που είχαμε στη διάθεσή μας καταφέραμε να το αναπτύξουμε στην πιο βασική του μορφή, δηλαδή, να έχει μια αρχική οθόνη που να μας οδηγεί σε ένα μενού επιλογών, με κάθε επιλογή να μας παραπέμπει και στην αντίστοιχη οθόνη και τέλος η κεντρική οθόνη του παιχνιδιού όπου εμφανίζονται διαδοχικά 10 ερωτήσεις, τυχαία από ένα σύνολο ερωτήσεων κι όσο ο παίχτης απαντά σωστά ή λάθος, να αλλάζει και το σκορ ανάλογα.

Το παιχνίδι, όμως, έχει δυνατότητες επέκτασης κι εξέλιξης. Αρχικά και μελλοντική μας επιδίωξη είναι η πρόσθεση φωτογραφιών στην κύρια οθόνη του παιχνιδιού. Δηλαδή, με κάθε αλλαγή της ερώτησης να υπάρχει και η αντίστοιχη φωτογραφία, η σχετική με την ερώτηση. Αυτή η φωτογραφία μπορεί να είναι αποθηκευμένα τοπικά μέσα στο παιχνίδι ή ακόμη να είναι από κάποια ιστοσελίδα. Δηλαδή, προγραμματιστικά να έχουμε συνδέσει το πρόγραμμα ώστε να τραβά την φωτογραφία της ερώτησης από την ιστοσελίδα όπου είναι ανεβασμένη, χωρίς να βαραίνει έτσι το παιχνίδι το ίδιο.

Στην προσπάθειά μας αυτή, για την ανάπτυξη ενός παιχνιδιού με ερωτήσεις, χρησιμοποιήσαμε έναν περιορισμένο αριθμό ερωτήσεων. 40 ερωτήσεις από τις οποίες η εφαρμογή επέλεγε 10 ερωτήσεις με τυχαίο τρόπο για να εμφανίσει στον χρήστη. Μελλοντικός στόχος, λοιπόν, είναι η αύξηση του αριθμού των ερωτήσεων, τόσο ώστε να χρειαστεί ο χρήστης να επαναλάβει πολλές φορές το παιχνίδι για να υπάρξει πιθανότητα να εμφανιστεί η ίδια ερώτηση.

Υπάρχει δυνατότητα προσθήκης ήχου. Θα μπορούσαμε όταν ξεκινά το παιχνίδι να δώσουμε στο πρόγραμμα ένα ήχο εισόδου, τέτοιον ώστε να προδιαθέτει ευχάριστα τον παίχτη να συνεχίσει παρακάτω. Έπειτα ένα διαφορετικό ήχο για κάθε οθόνη. Στην οθόνη εξέλιξης του παιχνιδιού θα μπορούσαμε να προσθέσουμε μία μουσική που να ταξιδεύει τους παίχτες νοερά στην Ευρώπη.

Επίσης, θα μπορούσαμε να δίνουμε τη δυνατότητα στο χρήστη μέσα από τις ρυθμίσεις, να εναλλάσσει τις παραπάνω επιλογές. Να μπορεί, δηλαδή, να έχει τη δυνατότητα να αποφασίζει αυτός αν θέλει να έχει τον ήχο ανοιχτό ή όχι. Ακόμη, μια άλλη δυνατότητα, πιθανόν να είναι, να μπορεί να επιλέγει το φόντο της κύριας οθόνης του παιχνιδιού, της οθόνης που εξελίσσεται το παιχνίδι.

Τέλος ένα σημαντικό κομμάτι του παιχνιδιού είναι η σύνδεση του σκορ που πετυχαίνει ο χρήστης με τον πίνακα των σκόρερ. Αυτό μπορεί να γίνει με τη βοήθεια ενός server, όπου εκεί θα φορτώνονται οι βαθμολογίες και θα περνούν έπειτα στον πίνακα των σκόρερ. Προϋπόθεση γι' αυτό θα είναι να βάζει ο χρήστης πρώτα το όνομά του ή το ψευδώνυμο που θέλει, είτε πριν την έναρξη του παιχνιδιού, είτε στη λήξη του, προκειμένου έπειτα να χρησιμοποιηθεί για την καταχώρηση της βαθμολογίας στον πίνακα των σκόρερ.

Υπάρχουν ακόμη πολλές κι αναρίθμητες δυνατότητες εξέλιξης και αναβάθμισης ενός προγράμματος, αν κάποιος θελήσει να το ερευνήσει σε βάθος. Η ασφάλεια του παιχνιδιού, η δυνατότητα διεθνοποίησής του (να μεταφραστεί και σε άλλες γλώσσες), η δυνατότητα δημοσίευσής του μέσω του Google Play, η ενοποίησή του με Υπηρεσίες Κοινωνικής Δικτύωσης είναι μερικά από τα πράγματα που μπορούν να γίνουν επιπρόσθετα στο μέλλον.

Τα παραπάνω βήματα φαίνονται απλά, αλλά προϋποθέτουν αρκετή έρευνα και προσπάθεια για τη σωστή κωδικοποίησή τους, έτσι ώστε το πρόγραμμα να λειτουργεί σωστά. Εμείς θελήσαμε για αρχή, να εστιάσουμε την προσοχή μας στις βασικές λειτουργίες μιας εφαρμογής, να μπορέσουμε να καταλάβουμε τον τρόπο που μπορεί να υλοποιηθεί μία Android εφαρμογή κι όχι στο να την τελειοποιήσουμε. Αυτό για την ώρα το αφήνουμε για το άμεσο μέλλον.

Πίνακας εικόνων

ΕΙΚΟΝΑ 1: ΛΟΓΟΤΥΠΟ ANDROID	9
ΕΙΚΟΝΑ 2 ΤΟ ΛΟΓΟΤΥΠΟ ΤΗΣ GOOGLE PLAY	11
ΕΙΚΟΝΑ 3: IOS ENANTION ANDROID	11
ΕΙΚΟΝΑ 4: (DEVICE DATA, 2014).....	12
ΕΙΚΟΝΑ 5: ΣΤΟΙΧΕΙΑ ΠΟΥ ΣΥΛΛΕΧΘΗΚΑΝ ΜΕΤΑ ΑΠΟ ΜΙΑ ΕΒΔΟΜΑΔΙΑΙΑ ΕΡΕΥΝΑ ΜΕΧΡΙ ΤΗΝ 4Η ΙΟΥΝΙΟΥ 2014. ΟΙ ΕΚΔΟΣΕΙΣ ΜΕ ΠΟΣΟΣΤΟ ΚΑΤΩ ΤΟΥ 0,1% ΔΕΝ ΕΜΦΑΝΙΖΟΝΤΑΙ. (DEVELOPERS, DASHBOARDS/ANDROID DEVELOPERS, 2014).....	13
ΕΙΚΟΝΑ 6: ΈΚΔΟΣΗ GINGERBREAD.....	14
ΕΙΚΟΝΑ 7: ΈΚΔΟΣΗ HONEYCOMB.....	14
ΕΙΚΟΝΑ 8: ΈΚΔΟΣΗ ICE CREAM SANDWICH	15
ΕΙΚΟΝΑ 9: ΈΚΔΟΣΗ JELLY BEAN	16
ΕΙΚΟΝΑ 10: ΈΚΔΟΣΗ ΚΙΤΚΑΤ	16
ΕΙΚΟΝΑ 11: ΚΕΝΤΡΙΚΗ ΣΕΛΙΔΑ ΤΩΝ ECLIPSE DOWNLOADS.....	17
ΕΙΚΟΝΑ 12: Η ΣΕΛΙΔΑ ΓΙΑ ΤΟ ΚΑΤΕΒΑΣΜΑ ΤΟΥ ECLIPSE INDIGO	18
ΕΙΚΟΝΑ 13: ECLIPSE DOWNLOAD FOR WINDOWS 32BIT	18
ΕΙΚΟΝΑ 14: Η ΣΕΛΙΔΑ ΤΗΣ ORACLE ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΈΚΔΟΣΗΣ JAVA SE DEVELOPMENT KIT 7 UPDATE 55	19
ΕΙΚΟΝΑ 15: ΑΠΟΔΟΧΗ ΤΗΣ ΑΔΕΙΑΣ ΚΑΙ ΚΑΤΕΒΑΣΜΑ ΤΗΣ ΕΚΔΟΣΗΣ ΤΗΣ JAVA ΓΙΑ ΤΑ WINDOWS 32BIT (WINDOWS X86)	20
ΕΙΚΟΝΑ 16: WINDOWS INSTALLER	20
ΕΙΚΟΝΑ 17: SETUP JDK 7U55	21
ΕΙΚΟΝΑ 18: ΕΠΙΛΟΓΗ ΤΩΝ ΠΡΟΣΘΕΤΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ (DEVELOPMENT TOOLS) ΚΑΙ ΤΗΣ ΕΠΙΛΟΓΗΣ ΤΟΥ ΦΑΚΕΛΟΥ ΣΤΟΝ ΟΠΟΙΟ ΕΓΚΑΘΙΣΤΑΝΤΑΙ	21
ΕΙΚΟΝΑ 19: ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΦΑΚΕΛΟΣ ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	22
ΕΙΚΟΝΑ 20: ΓΙΝΕΤΑΙ Η ΕΓΚΑΤΑΣΤΑΣΗ... ..	22
ΕΙΚΟΝΑ 21: ΕΠΙΤΥΧΗΣ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΟΥ JAVA SE DEVELOPMENT KIT 7 UPDATE 55	23
ΕΙΚΟΝΑ 22: INSTALLER R22.6.2-WINDOWS.EXE - Ο ΣΥΝΔΕΣΜΟΣ ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ΣΤΗ ΣΕΛΙΔΑ ΤΩΝ DEVELOPER.ANDROID.COM	23
ΕΙΚΟΝΑ 23: ΌΡΟΙ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΙ ΑΠΟΔΟΧΗ ΤΟΥΣ ΓΙΑ ΤΗΝ ΣΥΝΕΧΙΣΗ ΤΟΥ ΚΑΤΕΒΑΣΜΑΤΟΣ ΤΩΝ ADT PLUGIN	24
ΕΙΚΟΝΑ 24: ANDROID SDK TOOLS SETUP WIZARD.....	25
ΕΙΚΟΝΑ 25: ΕΠΙΤΥΧΗΣ ΕΥΡΕΣΗ ΕΓΚΑΤΕΣΤΗΜΕΝΗΣ ΕΚΔΟΣΗΣ JAVA.....	25
ΕΙΚΟΝΑ 26: ΕΠΙΛΟΓΗ ΧΡΗΣΤΗ	26
ΕΙΚΟΝΑ 27: ΕΠΙΛΟΓΗ ΦΑΚΕΛΟΥ ΠΡΟΟΡΙΣΜΟΥ ΤΩΝ ΑΡΧΕΙΩΝ	26
ΕΙΚΟΝΑ 28: ΕΠΙΛΟΓΗ START MENU FOLDER	27
ΕΙΚΟΝΑ 29: ΕΞΑΓΩΓΗ ΑΡΧΕΙΩΝ... ..	27
ΕΙΚΟΝΑ 30: ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΩΝ SDK TOOLS	28
ΕΙΚΟΝΑ 31: ΈΝΑΡΞΗ ΤΟΥ ECLIPSE	28
ΕΙΚΟΝΑ 32: ΚΑΘΟΡΙΣΜΟΣ ΤΗΣ ΔΙΑΔΡΟΜΗΣ ΤΟΥ ΦΑΚΕΛΟΥ WORKSPACE	29
ΕΙΚΟΝΑ 33: ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ECLIPSE	29
ΕΙΚΟΝΑ 34: HELP →INSTALL NEW SOFTWARE.....	30
ΕΙΚΟΝΑ 35: ADD SOFTWARE	30
ΕΙΚΟΝΑ 36:ΕΙΣΑΓΩΓΗ NAME ΚΑΙ LOCATION ΤΟΥ ΙΣΤΟΤΟΠΟΥ.....	31
ΕΙΚΟΝΑ 37: PENDING	31
ΕΙΚΟΝΑ 38: ΔΙΑΘΕΣΙΜΑ DEVELOPER TOOLS	32
ΕΙΚΟΝΑ 39: ΑΝΑΣΚΟΠΗΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ ΠΟΥ ΘΑ ΚΑΤΕΒΟΥΝ	32
ΕΙΚΟΝΑ 40: ΑΠΟΔΟΧΗ ΤΩΝ ΌΡΩΝ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	33
ΕΙΚΟΝΑ 41: INSTALLING SOFTWARE... ..	33
ΕΙΚΟΝΑ 42: ΕΙΔΟΠΟΙΗΣΗ ΓΙΑ ΕΠΑΝΕΚΚΙΝΗΣΗ ΤΟΥ ECLIPSE.....	34
ΕΙΚΟΝΑ 43:ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ANDROID SDK MANAGER (WINDOWS → ANDROID SDK MANAGER) ..	34
ΕΙΚΟΝΑ 44: SDK PACKAGES	35
ΕΙΚΟΝΑ 45: ΈΛΕΓΧΟΣ ΓΙΑ ΕΝΗΜΕΡΩΣΕΙΣ ΣΤΟ ECLIPSE	35

ΕΙΚΟΝΑ 46: ΈΛΕΓΧΟΣ ΓΙΑ UPDATES	36
ΕΙΚΟΝΑ 47: ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ PROJECT	39
ΕΙΚΟΝΑ 48: ΕΠΙΛΟΓΗ ΔΗΜΙΟΥΡΓΙΑΣ ANDROID APPLICATION PROJECT	40
ΕΙΚΟΝΑ 49: ΟΝΟΜΑΤΟΔΟΤΗΣΗ PROJECT ΚΙ ΕΠΙΛΟΓΗ SDK	40
ΕΙΚΟΝΑ 50: ΕΠΙΛΟΓΗ ΤΗΣ ΔΙΑΔΡΟΜΗΣ ΤΟΥ ΦΑΚΕΛΟΥ WORKSPACE, ΤΗΣ ΔΗΜΙΟΥΡΓΙΑΣ Η ΜΗ ΤΗΣ ΑΡΧΙΚΗΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ ΚΑΙ ΕΠΙΛΟΓΗ ΔΗΜΙΟΥΡΓΙΑΣ ΕΙΚΟΝΙΔΙΟΥ.....	41
ΕΙΚΟΝΑ 51: CONFIGURE LAUNCHER ICON (ΜΠΟΡΟΥΜΕ ΝΑ ΔΙΑΛΕΞΟΥΜΕ ΤΟ ΛΟΓΟΤΥΠΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ)	42
ΕΙΚΟΝΑ 52: CREATE ACTIVITY (ΕΠΙΛΟΓΗ ΤΟΥ ΕΙΔΟΥΣ ΤΗΣ ACTIVITY).....	42
ΕΙΚΟΝΑ 53: ΔΗΛΩΣΗ ΤΟΥ ΟΝΟΜΑΤΟΣ ΤΗΣ ΑΡΧΙΚΗΣ ACTIVITY ΚΑΙ ΤΟΥ LAYOUT ΠΟΥ ΤΗ ΣΥΝΟΔΕΥΕΙ..	43
ΕΙΚΟΝΑ 54: Η ACTIVITY ΠΟΥ ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΣΤΗΝ ΑΡΧΙΚΗ ΤΗΣ ΜΟΡΦΗ	43
ΕΙΚΟΝΑ 55: ΤΟ ΛΟΓΟΤΥΠΟΥ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ	44
ΕΙΚΟΝΑ 56: ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΑΡΧΕΙΟΥ	45
ΕΙΚΟΝΑ 57: ΔΗΛΩΣΗ ΤΟΥ ΟΝΟΜΑΤΟΣ, ΤΗΣ ΚΑΤΑΛΗΞΗΣ ΚΑΙ ΤΟΥ ΦΑΚΕΛΟΥ ΠΟΥ ΘΑ ΑΠΟΘΗΚΕΥΤΕΙ .	45
ΕΙΚΟΝΑ 58: ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΚΛΑΣΗΣ.....	48
ΕΙΚΟΝΑ 59: ΔΗΛΩΣΗ ΟΝΟΜΑΤΟΣ ΤΗΣ ΝΕΑΣ ACTIVITY ΚΑΙ ΤΗΣ ACTIVITY ΠΟΥ ΘΑ ΚΛΗΡΟΝΟΜΕΙ	48
ΕΙΚΟΝΑ 60: ΔΗΜΙΟΥΡΓΙΑ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ	51
ΕΙΚΟΝΑ 61: ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ.....	52
ΕΙΚΟΝΑ 62: ΕΠΙΛΟΓΗ ΟΝΟΜΑΤΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΤΗΣ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ	52
ΕΙΚΟΝΑ 63: ΤΟ ΣΥΝΟΛΟ ΤΩΝ ΔΙΑΘΕΣΙΜΩΝ ΕΙΚΟΝΙΚΩΝ ΜΗΧΑΝΩΝ ΚΑΙ Η ΔΥΝΑΤΟΤΗΤΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΟΥΣ.....	53
ΕΙΚΟΝΑ 64: ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ.....	53
ΕΙΚΟΝΑ 65: ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ ΠΟΥ ΔΗΜΙΟΥΡΓΟΥΜΕ	54
ΕΙΚΟΝΑ 66: RUN → RUN CONFIGURATIONS.....	55
ΕΙΚΟΝΑ 67: CREATE, MANAGE AND RUN CONFIGURATIONS	55
ΕΙΚΟΝΑ 68: ΕΠΙΛΟΓΗ TARGET ΓΙΑ ΤΟ ΠΡΟΓΡΑΜΜΑ ΜΑΣ.....	56
ΕΙΚΟΝΑ 69: Η ΕΦΑΡΜΟΓΗ ΜΑΣ ΜΕΤΑ ΤΗΝ ΠΡΩΤΗ ΦΟΡΤΩΣΗ	56
ΕΙΚΟΝΑ 70: ΤΟ ΛΟΓΟΤΥΠΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΤΟΝ ΚΑΤΑΛΟΓΟ ΤΩΝ ΕΦΑΡΜΟΓΩΝ ΤΟΥ ΕΙΚΟΝΙΚΟΥ ΚΙΝΗΤΟΥ ΜΑΣ	57
ΕΙΚΟΝΑ 71: Η ΕΦΑΡΜΟΓΗ ΜΕ ΤΗΝ ΤΡΟΠΟΠΟΙΗΣΗ ΤΗΣ MAIN.XML.....	61
ΕΙΚΟΝΑ 72: Η ΚΥΡΙΑ ΟΘΟΝΗ ΕΠΙΛΟΓΩΝ ΜΕΤΑ ΑΠΟ ΤΙΣ ΠΑΡΑΠΑΝΩ ΡΥΘΜΙΣΕΙΣ	72

Βιβλιογραφία – Δικτυογραφία

- Android Developers: Adding platforms and packages.* (n.d.). Retrieved 5 10, 2014, from Τοποθεσία web των Android Developers: <http://developer.android.com/sdk/installing/adding-packages.html>
- A Simple Quiz Game in Android.* (2013, Φεβρουάριος 13). Retrieved Σεπτέμβριος 3, 2014, from Developerfeed: <http://www.developerfeed.com/simple-quiz-game-android>
- Android Developers. (n.d.). Service.* Retrieved Οκτώβριος 6, 2014, from Android Developers: <http://developer.android.com/reference/android/app/Service.html>
- Android Developers. (n.d.). Widgets.* Retrieved Σεπτέμβριος 2014, 2014, from Android Developers: <http://developer.android.com/design/patterns/widgets.html>
- Android Developers: Setting Up an Existing IDE.* (n.d.). Retrieved Απρίλιος 10, 2014, from Τοποθεσία web των Android Developers: <http://developer.android.com/sdk/installing/index.html>
- Building a TODO List app in Android using SQLite.* (2013, Ιανουάριος 22). Retrieved Σεπτέμβριος 2, 2014, from DeveloperFeed: <http://www.developerfeed.com/building-todo-list-app-android-using-sqlite>
- Conder, D. (2012). Μάθετε την ανάπτυξη Εφαρμογών για το Android σε 24 ώρες - 2η Έκδοση. In L. Darcey, & S. Conder, *Sams Teach Yourself Android Application Development in 24 Hours - 2nd Edition*. Αθήνα: Μ. Γκιούρδας.
- Developers, A. (2014, Ιουνίου 4). *Dashboards/Android Developers.* Retrieved Ιούνιος 12, 2014, from Android Developers: https://developer.android.com/about/dashboards/index.html?utm_source=android.net
- Developers, A. (n.d.). *Ice Cream Sandwich.* Retrieved Σεπτέμβριος 14, 2014, from Android Developers: <http://developer.android.com/about/versions/android-4.0-highlights.html>
- Device Data.* (2014). Retrieved Μάιος 6, 2014, from Device Atlas - A dotMobi Product: <https://deviceatlas.com/device-data/explorer>
- dotMobi. (2014, Μάρτιος 18). *Apple Devices Dominate Global Web Browsing, while Android Gains Ground.* Retrieved Απρίλιος 18, 2014, from dotMobi.Internet made mobile.Everywhere, every device.: <http://dotmobi.com/content/apple-devices-dominate-global-web-browsing-while-android-gains-ground-0>
- Elpidis, C. (2014). *Android vs iOS: Πως διαμορφώνεται ο Παγκόσμιος Χάρτης.* Retrieved Μάρτιος 30, 2014, from Techgear: <http://www.techgear.gr/android-vs-ios-web-traffic-infographic-86635/>
- Kitsati, C. (2013, Οκτώβριος 31). *Android 4.4 KitKat. Αυτά είναι τα χαρακτηριστικά του.* Retrieved Σεπτέμβριος 20, 2014, from DigitalLife: <http://www.digitallife.gr/android-4-4-kit-kat-ayta-einai-ta-xarakteristika-tou-95605>
- Open Handset Alliance. (n.d.). *Android Overview.* Retrieved Απρίλιος 15, 2014, from Open Handset Alliance: http://www.openhandsetalliance.com/android_overview.html
- softy-warely.* (2011). Retrieved 2014, from package com.android.cse400;: <http://softy-warely.googlecode.com/svn-history/r59/trunk/projects/WordBox/src/com/android/cse400/WordBoxScores.java>
- Tutorials Point. (n.d.). *Android Services.* Retrieved Οκτώβριος 6, 2014, from tutorialspoint. Simply easy learning: http://www.tutorialspoint.com/android/android_services.htm

- Tutorialspoint. (n.d.). *Android Broadcast Receivers*. Retrieved Οκτώβριος 5, 2014, from tutorialspoint. Simply Easy Learning: http://www.tutorialspoint.com/android/android_broadcast_receivers.htm
- Tutorialspoint. (n.d.). *Android Content Providers*. Retrieved Οκτώβριος 5, 2014, from tutorialspoint. Simply Easy Learning: http://www.tutorialspoint.com/android/android_content_providers.htm
- Tutorialspoint. (n.d.). *Android Intents and Filters*. Retrieved Οκτώβριος 5, 2014, from tutorialspoint. Simply Easy Learning: http://www.tutorialspoint.com/android/android_intents_filters.htm
- Wikipedeia. (2014, Ιουλίου 29). *Software repository*. Retrieved Οκτώβριος 10, 2014, from Wikipedia. The free Encyclopedia: http://en.wikipedia.org/wiki/Software_repository
- wikipedia. (2014, Σεπτέμβριος 17). *Android Jelly Bean*. Retrieved Σεπτέμβριος 26, 2014, from Wikipedia: http://en.wikipedia.org/wiki/Android_Jelly_Bean
- ΒΙΚΙΠΑΙΔΕΙΑ. (n.d.). *Διεπαφή προγραμματισμού εφαρμογών*. Retrieved Σεπτέμβριος 25, 2014, from ΒΙΚΙΠΑΙΔΕΙΑ, η ελεύθερη εγκυκλοπαίδεια: http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B5%CF%80%CE%B1%CF%86%CE%AE_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D_%CE%B5%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CF%8E%CE%BD
- ΔΟΥΚΑΣ, Ε. (2011). " ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΟΥ ΤΗΛΕΦΩΝΟΥ ΜΕ ΧΡΗΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ GOOGLE ANDROID ". In Ε. ΔΟΥΚΑΣ, *Πτυχιακή Εργασία*. ΛΑΡΙΣΑ: ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ.
- Trivia Quiz App for Android. Retrieved Σεπτέμβριος 6, 2014, from Hupamobile: <http://www.chupamobile.com/android-full-games/trivia-quiz-app-for-android-527>
- Thornsby Jessica (2013, Ιανουάριος 8). *Android UI Workshop: Build an Interactive Quiz App from tuts+*: <http://code.tutsplus.com/tutorials/android-ui-workshop-build-an-interactive-quiz-app--mobile-14208>