

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Ανάπτυξη Παιχνιδιού Σε Λειτουργικό Android

Game Development In Android

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μάριος Ναζίρης

Επιβλέποντες Καθηγητές:

Ακρίτας Αλκιβιάδης
Καθηγητής Π.Θ.

Σταμούλης Γεώργιος
Καθηγητής Π.Θ.

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Ανάπτυξη Παιχνιδιού Σε Λειτουργικό Android

Game Development In Android

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μάριος Ναζίρης

Επιβλέποντες Καθηγητές:

Ακρίτας Αλκιβιάδης
Καθηγητής Π.Θ.

Σταμούλης Γεώργιος
Καθηγητής Π.Θ.

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 1η Ιουλίου 2015

.....
Ακρίτας Αλκιβιάδης
Καθηγητής Π.Θ.

.....
Σταμούλης Γεώργιος
Καθηγητής Π.Θ.

Διπλωματική εργασία για την απόκτηση του Διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Πανεπιστημίου Θεσσαλίας.

.....
Μάριος Ναζίρης

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας.

© 2015- All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω θερμά τον κ. Ακρίτα Αλκιβιάδη -επιβλέποντα Καθηγητή- και τον κ. Γεώργιο Σταμούλη -συνεπιβλέποντα Καθηγητή- για την εμπιστοσύνη που επέδειξαν στο πρόσωπό μου, για τη συνεχή καθοδήγηση κατά τη διάρκεια της διπλωματικής μου εργασίας, καθώς και για την άψογη συνεργασία που υπήρχε μεταξύ μας.

Εν συνεχεία θα ήθελα να ευχαριστήσω τον φίλο και συνάδελφο Νικόλα για την συνεργασία μας, την αμοιβαία υποστήριξη και την ανεκτίμητη βοήθεια που μου παρείχε κατά την διάρκεια των σπουδών μας.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου για την αμέριστη υποστήριξη και κατανόηση που μου παρείχαν όλα αυτά τα χρόνια και χάρις σ' αυτήν κατάφερα να φέρω εις πέρας τις σπουδές μου.

Μάριος Ναζίρης
Βόλος
Ιούνιος 2015

ΠΕΡΙΛΗΨΗ

Η εξέλιξη της τεχνολογίας έχει αλλάξει δραματικά την ζωή μας. Ως αποτέλεσμα, κάτι που πριν μερικές μόλις δεκαετίες φάνταζε πολυτέλεια ή και σενάριο επιστημονικής φαντασίας να έχει γίνει αναπόσπαστο κομμάτι της καθημερινότητά μας. Η κινητή τηλεφωνία και κατ' επέκταση τα smartphones έχουν αλλάξει τον τρόπο που σκεφτόμαστε και λειτουργούμε. Η χρήση τους δεν περιορίζεται μόνο στις ανάγκες μας αλλά μεγάλο κομμάτι αναλώνεται στον τομέα της διασκέδασής μας.

Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός παιχνιδιού συμβατό για λειτουργικό Android, το δημοφιλέστερο λειτουργικό αυτή τη στιγμή που χρησιμοποιείται στα περισσότερα μοντέλα συσκευών αλλά και στις περισσότερες γενικώς συσκευές παγκοσμίως.

ABSTRACT

The development of technology has changed dramatically our life, As a result, something that few decades ago appeared to be luxury or even script of scientific imagination has become integral part of our everyday routine. The mobile telephony and at extension smartphones have changed the way that we think and we function. Their use is not limited only in our needs but big part is spent in our amusement.

Aim of this diplomatic work is the growth of a game that is compatible for Android operating system, the most popular at the moment that is used in most models of devices but also in most generally devices worldwide.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	9
ΚΕΦΑΛΑΙΟ 1^ο	11
1.1-ΤΙ ΕΙΝΑΙ ΤΟ ANDROID.....	11
1.2-Η ΙΣΤΟΡΙΑ ΤΟΥ ANDROID	12
1.3-ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID.....	12
ΚΕΦΑΛΑΙΟ 2^ο	19
2.1-ΓΙΑΤΙ ANDROID ?.....	19
2.2-ΕΠΙΛΟΓΗ ΕΡΓΑΛΙΩΝ.....	20
2.3-Η ΓΛΩΣΣΑ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	20
2.4-Η ΖΩΗ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ ANDROID	21
2.5-ΟΙ ΜΕΘΟΔΟΙ ΤΟΥ ΚΥΚΛΟΥ ΖΩΗΣ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ	22
ΚΕΦΑΛΑΙΟ 3^ο	23
3.1-Android Virtual Machine(AVD)-Προσομοιωτής Μηχανής Android	23
3.2-Υποστήριξη Πολλών Οθονών	23
3.3-Περιγραφή Κλάσεων και δραστηριοτήτων	24
ΚΩΔΙΚΑΣ.....	32
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	46

ΕΙΣΑΓΩΓΗ

Η τεχνολογία ολοένα και εξελίσσεται και πιο συγκεκριμένα όσον αφορά το χώρο της κινητής τηλεφωνίας. Τα τελευταία χρόνια τα έξυπνα κινητά τηλέφωνα ενσωματώνουν δυνατότητες που μέχρι σήμερα δεν υπήρχαν στα κινητά τηλέφωνα με αποτέλεσμα οι κινητές συσκευές να εισχωρούν όλο και περισσότερο στη ζωή μας. Είναι πλέον αναπόσπαστο κομμάτι στην καθημερινότητα πολλών ανθρώπων καθώς παρέχουν πολλές δυνατότητες στο χρήστη όπως να ψυχαγωγηθούν και να περάσουν ευχάριστα την ώρα τους μέσα από τα ψηφιακά παιχνίδια που μπορούν να ενσωματωθούν στις έξυπνες κινητές συσκευές.

Παλαιότερα, τα παιχνίδια που μπορούσε ο χρήστης να παίξει ήταν απλά και ενσωματωμένα στη συσκευή του τηλεφώνου. Τα γραφικά των παιχνιδιών δεν είχαν χρώματα και δεν ήταν ευχάριστα προς το χρήστη. Με την εξέλιξη όμως των κινητών τηλεφώνων και των υπηρεσιών τους εξελίσσονταν παράλληλα και τα παιχνίδια που μπορούσαν να ενσωματωθούν στις κινητές συσκευές.

Τα παιχνίδια των κινητών συσκευών χρειάζονται μία πλατφόρμα για υποστήριξη. Τα έξυπνα κινητά τηλέφωνα περιλαμβάνουν ολοκληρωμένο λειτουργικό σύστημα. Τα σημαντικότερα λειτουργικά συστήματα είναι: το Android, το Windows Phone, το iOS, το Blackberry και το Bada. Η συγκεκριμένη πτυχιακή εργασία θα ασχοληθεί με το λειτουργικό σύστημα Android.

Το λειτουργικό σύστημα Android είναι μια πλατφόρμα ανοικτού λογισμικού που επιτρέπει στους χρήστες την ανάπτυξη δωρεάν παιχνιδιών αλλά και εφαρμογών. Αυτό συντέλεσε στην ταχεία του διάδοση. Αναπτύχθηκε από την εταιρεία Android Inc. Και εξαγοράστηκε το 2005 από την Google.

Επομένως σκοπός της συγκεκριμένης μελέτης είναι η ανάπτυξη μιας εφαρμογής που προορίζεται για τις έξυπνες κινητές συσκευές με λειτουργικό σύστημα Android. Η εφαρμογή αυτή αφορά την ανάπτυξη του παιχνιδιού της Γάτας με το Ποντίκι «TOM AND JERRY». Με τη συγκεκριμένη εφαρμογή ο χρήστης θα μπορεί να ακονίσει το μυαλό του αφού είναι ένα παιχνίδι στρατηγικής, θα μπορεί να χαλαρώσει, να περάσει δημιουργικά και ευχάριστα τον ελεύθερο του χρόνο και γενικότερα να ψυχαγωγηθεί.

Οι ενότητες που ακολουθούν θα κάνουν αναφορά στη βιβλιογραφική ανασκόπηση και θα περιγράψουν την εφαρμογή, τα εργαλεία που

χρησιμοποιήθηκαν και πως αναπτύχθηκε. Η βιβλιογραφική ανασκόπηση θα επικεντρωθεί στο Android και στα παιχνίδια κινητών συσκευών. Το πρακτικό κομμάτι της έρευνας θα περιλαμβάνει την ανάπτυξη εφαρμογής με γλώσσα προγραμματισμού Java για κινητές συσκευές που τρέχουν το λειτουργικό σύστημα Android.

ΚΕΦΑΛΑΙΟ 1^ο

1.1 ΤΙ ΕΙΝΑΙ ΤΟ ANDROID

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, σε ψηφιακές φωτογραφικές μηχανές, σε συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές.

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.

1.2 Η ΙΣΤΟΡΙΑ ΤΟΥ ANDROID

Από την εισαγωγή του Android το 2007, το ανοιχτού κώδικα λειτουργικό σύστημα για κινητά τηλέφωνα, ο τομέας της κινητής τηλεφωνίας έχει αλλάξει ριζικά. Ένα νέο λειτουργικό εξελίχθηκε με ταχύ ρυθμό, φέρνοντας σε δύσκολη θέση τους ηγέτες της αγοράς όπως την Apple, τη Nokia και Blackberry. Η Google έχει μια περίεργη, αν και χιουμοριστική ονομασία για κάθε έκδοση του λειτουργικού της Android. Οι εκδόσεις φέρουν ονομασίες διαφόρων γλυκών επιδορπίων. (ακολουθούν μάλιστα αλφαβητική σειρά).

Android 1.0 και 1.1

Η πρώτη έκδοση κυκλοφόρησε το Σεπτέμβριο του 2008, μαζί με τη συσκευή, «Hero HTC». Τόσο το λειτουργικό σύστημα όσο και η συσκευή HTC έλαβαν θετικές κριτικές. Το όνειρο ενός open-source λειτουργικό για κινητά τελικά έγινε πραγματικότητα!

Έκδοση 1.1 (κυκλοφόρησε το Φεβρουάριο του 2009). Σε αυτό το στάδιο, η Google δεν είχε ακόμη ξεκινήσει να δίνει ονομασίες γλυκών επιδορπίων στις εκδόσεις των λειτουργικών της.



Android 1.5 Cupcake

Κυκλοφόρησε τον Απρίλιο του 2009, αυτή ήταν η πρώτη σημαντική αναβάθμιση του Android, προαναγγέλλοντας την έναρξη της "σειράς dessert". Με το Cupcake, χαρακτηριστικά όπως το ανέβασμα βίντεο, η πρόβλεψη του κειμένου και ασύρματη μετάδοση μουσικής ήταν πλέον διαθέσιμα



Android 1.6 Donut

Κυκλοφόρησε τον Σεπτέμβριο του 2009 Η έκδοση Donut ήρθε με σημαντικές αναβαθμίσεις, το αποκορύφωμα της οποίας ήταν το Google Maps . Επίσης ενίσχυσε τις δυνατότητες φωτογραφίας και βίντεο.



Android 2.0 Eclair

Η έκδοση 2.0 κυκλοφόρησε τον Δεκέμβριο του 2009, ακολουθούμενη από 2,1 τον Ιανουάριο του 2010. Η αναβάθμιση αυτή είχε σημαντικές βελτιώσεις επιτρέποντας διάφορες δυνατότητες Bluetooth, multi-touch υποστήριξη και live wallpapers, μεταξύ των άλλων χαρακτηριστικών



Android 2.2 Froyo

Σύντμηση για το "Frozen Yoghurt", αυτή η έκδοση κυκλοφόρησε το Μάιο του 2010. Έφερε βελτιωμένη ταχύτητα στο android, με την υποστήριξη υψηλής ευκρίνειας σε αναλύσεις οθόνης και Adobe Flash 10.1, επιτρέποντας στους χρήστες να βλέπουν ροή βίντεο στο κινητό τους μέσω browsers. Προστέθηκε και υποστήριξη για Wi-Fi hotspot συνδεσιμότητα



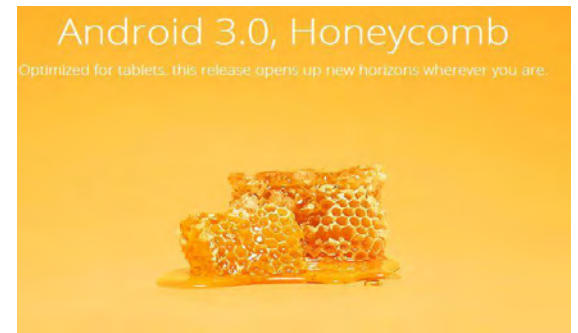
Android 2.3 Gingerbread

Αυτή η έκδοση έκανε το ντεμπούτο της το Δεκέμβριο του 2010. Το βασικό χαρακτηριστικό που έγινε διαθέσιμο με αυτή την έκδοση ήταν το (NFC), επιτρέποντας στους χρήστες να εκτελούν διάφορα, όπως πληρωμές μέσω κινητού τηλεφώνου και την ανταλλαγή δεδομένων μέσω αυτών περνώντας τα κινητά τους τηλέφωνα πάνω από μια ετικέτα. Προσέθεσε επίσης την υποστήριξη για περισσότερες από μία κάμερες και διάφορους άλλους αισθητήρες.



Android 3.0 Honeycomb

Κυκλοφόρησε τον Φεβρουάριο του 2011 και ακολούθησαν γρήγορα οι εκδόσεις 3.1 και 3.2 κατά τη διάρκεια του ίδιου έτους. Αυτή η έκδοση είχε βελτιστοποιηθεί για ταμπλέτες (tablets). Επέτρεψε επίσης στους χρήστες τη δυνατότητα να φορτώνουν τα αρχεία πολυμέσων απευθείας από μια κάρτα SD.



Android 4.0 Ice Cream Sandwich

Κυκλοφόρησε τον Οκτώβριο του 2011, αυτό ήταν μια πολύ σημαντική αναβάθμιση του Android UI, επιτρέποντας την ενίσχυση μενού επικοινωνίας, τη βελτίωση στη διάταξη του πληκτρολογίου και NFC δυνατότητες.



Android 4.1 Jelly Bean

Η τελευταία μεγάλη αναβάθμιση του Android κυκλοφόρησε τον Ιούλιο του 2012. Ακόμη πιο όμορφο το Android UI, αλλά και εξεζητημένο λογισμικό, επιτρέποντας στις Android συσκευές να τρέξουν πιο γρήγορα και. Οι 4.x εκδόσεις επέτρεψαν στους προγραμματιστές να δημιουργήσουν εφαρμογές με ποιότητα, επιβεβαιώνοντας τη θέση της ως το λειτουργικό σύστημα που προτιμούν οι χρήστες.



Android 4.4 KitKat

Η έκδοση του Android 4.4 έχει τόσες σημαντικές και πολλές βελτιώσεις που αξίζει να σταθούμε στα βασικά. Το UI και UX έχουν αλλάξει και έχουν βελτιωθεί στα σημεία. Αυτό σημαίνει ότι τα γραφικά είναι λίγο διαφορετικά και η εμπειρία χρήσης του Android KitKat είναι καλύτερη από ποτέ. Οι διαφορές θα γίνουν αμέσως ορατές όταν οι κάτοχοι Nexus 4 κάνουν αναβάθμιση στο Android 4.4 γιατί για όλους τους άλλους κατόχους Samsung smartphones και LG smartphones με custom skins, τα πράγματα είναι διαφορετικά. Σημαντική αλλαγή που φαίνεται από την πρώτη ματιά είναι η απώλεια των notification bar και control bar που έχουν εξαφανιστεί και έχουν γίνει διάφανες.



Android 5.0 Lollipop

Κυκλοφόρησε τον Νοέμβριο του 2014. Το Material Design, η νέα φιλοσοφία της Google στο σχεδιασμό Android και Chrome, κυριαρχεί στο νέο Android Lollipop, με την πλατφόρμα να έχει πλέον καθαρότερο σχεδιασμό, με αρκετές οπτικές αλλαγές, ανανεωμένο multi-tasking menu και αρκετούς νέους τρόπους αλληλεπίδρασης με τη συσκευή μέσω φωνητικών εντολών.



Πρόκειται για ένα πολύ μεγάλο release για τη Google, που αποτελεί το πρώτο βήμα προς το μέλλον του λειτουργικού Android ως μία πλατφόρμα που δε θα αφορά μόνο smartphones και tablets.

1.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID

Λειτουργίες Οθόνης

Η πλατφόρμα είναι προσαρμόσιμη σε μεγαλύτερη ανάλυση (VGA), δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες, τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 1.0 έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.

Αποθήκευση Δεδομένων

Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης

Συνδεσιμότητα

Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, και Wi-Fi.

Αποστολή μηνυμάτων

SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.

Περιήγηση στον Ιστό

Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή βασισμένο στην ανοιχτή τεχνολογία WebKit.

Υποστήριξη Java

Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik, η οποία αποτελεί εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένης για χρήση σε φορητές συσκευές, παρόλο που δεν είναι πρότυπη εικονική μηχανή Java.

Υποστήριξη Πολυμέσων

Το λειτουργικό Android υποστηρίζει τις ακόλουθα μορφές ήχου, στατικής και κινούμενης εικόνας: H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.

Επιπλέον υποστήριξη υλικού

Το λειτουργικό Android μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.

Περιβάλλον Ανάπτυξης Λογισμικού

Περιλαμβάνει έναν προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Eclipse IDE.

Αγορά και Εγκατάσταση Εφαρμογών

Παρόμοια με το App Store του iPhone OS, το Android Market είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και να εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή. Αρχικά μόνο δωρεάν εφαρμογές ήταν δυνατόν να εγκατασταθούν. Εφαρμογές επί πληρωμή ήταν μετέπειτα διαθέσιμες στο Android Market στις ΗΠΑ ύστερα από τις 19 Φεβρουαρίου 2009.

Οθόνη Αφής Πολλαπλών Σημείων

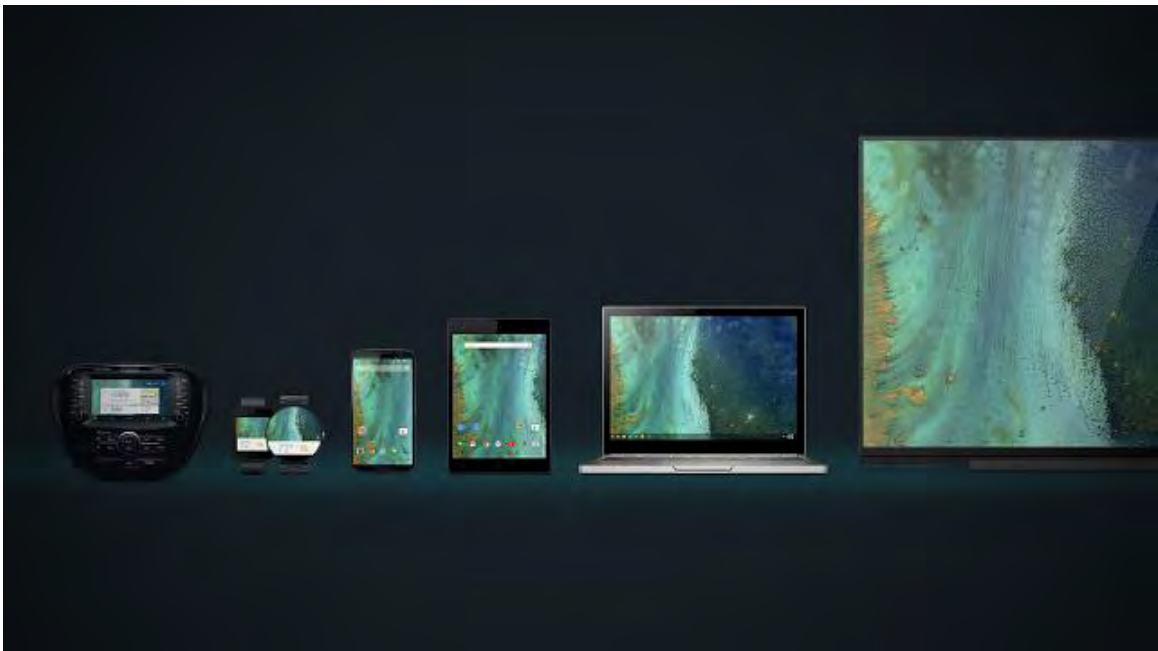
Το λειτουργικό Android είχε εξ ορισμού υποστήριξη για οθόνες πολλαπλών σημείων αλλά η δυνατότητα αυτή έχει κλειδωθεί σε επίπεδο πυρήνα (πιθανόν για αποφυγή παραβιάσεων των πατεντών λογισμικού της Apple στις τεχνολογίες οθονών αφής). Κυκλοφορεί μια ανεπίσημη τροποποίηση (mod) που έχει αναπτυχθεί για να υποστηρίζει πολλαπλή επαφή (multi-touch), αλλά απαιτεί δικαιώματα πρόσβασης υπερχρήστη (superuser) στη συσκευή για να γραφεί στη μνήμη flash ένας πυρήνας που να μην είναι υπογεγραμμένος (unsigned kernel).

ΚΕΦΑΛΑΙΟ 2^ο

2.1 ΓΙΑΤΙ ANDROID ?

Πλέον η ανάπτυξη εφαρμογών για κινητές επικοινωνίες γίνεται με εύκολο και απλό τρόπο. Αντίθετα η ανάπτυξη εφαρμογών για ασύρματες επικοινωνίες, με τους πάρα πολύ ακριβούς μεταγλωττιστές και τα προνομιακά προγράμματα προγραμματιστών, είναι πολύ ακριβή για να μπορεί να χρησιμοποιηθεί σε επιτραπέζιους υπολογιστές. Εδώ, το *Android* σπάει το μοντέλο αποκλειστικής εκμετάλλευσης. Σε σχέση με τις άλλες πλατφόρμες κινητών επικοινωνιών, στην πραγματικότητα δεν υπάρχει κόστος για ανάπτυξη εφαρμογών για το *Android*.

Η ευελιξία του *Android* και το μικρό κόστος γρήγορα οδήγησαν στην εξάπλωση αυτής της νέας πλατφόρμας και το λειτουργικό έχει πάψει να χρησιμοποιείται αποκλειστικά για smartphones. Πληθώρα συσκευών χρησιμοποιούν πλέον το *Android* που έχει χρησιμοποιηθεί σε ταμπλέτες, ρολόγια, τηλεοράσεις, φορητούς και επιτραπέζιους υπολογιστές και πρόσφατα ακόμη και σε αυτοκίνητα με το *Android Auto*. Τα σχέδια της Google δεν σταματούν εκεί και εικάζετε πως σύντομα ίσως έχουμε ακόμη και κονσόλα παιχνιδιών ικανή να συναγωνιστεί το μονοπώλιο των σειρών PlayStation της Sony και Xbox της Microsoft.



2.2 ΕΠΙΛΟΓΗ ΕΡΓΑΛΙΩΝ

Το *Android SDK* και τα εργαλεία του *Android* διατίθενται δωρεάν στο site του προγραμματιστή του *Android*, <http://developer.android.com>. Το δωρεάν διατιθέμενο πρόγραμμα *Eclipse* έγινε το δημοφιλέστερο ολοκληρωμένο περιβάλλον ανάπτυξης (*Integrated Development Environment, IDE*) για ανάπτυξη εφαρμογών *Android*.

Παρόλα αυτά, η *Android* προσπαθεί να προωθήσει το δικό της ολοκληρωμένο περιβάλλον ανάπτυξης *Android Studio* για την δημιουργία *Android* εφαρμογών και πρόσφατα έχει σταματήσει να προσφέρει στη σελίδα της το *ADT BUNDLE* που περιλάμβανε τα δημοφιλή IDE, το *Eclipse*, το *Android SDK*, τα *ADT* πρόσθετα, τα εργαλεία πλατφόρμας και εικονικά συστήματα για προσομοιωτές.

Λόγω της υπάρχουσας εξοικειώσής μου με το *Eclipse* θεωρήθηκε σκόπιμο όπως αναπτύξω την εφαρμογή μου σε αυτό, με τη βοήθεια του οδηγού για εφαρμογές *Android* του *Chunyen Liu* <<*Android Development: A Guide to Getting Started*>>.

2.3 Η ΓΛΩΣΣΑ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Οι εφαρμογές *Android* γράφονται σε *Java*, μία απ' τις δημοφιλέστερες γλώσσες προγραμματισμού στον κόσμο. Οι προγραμματιστές της *Java* θα είναι εξοικειωμένοι με πολλά από τα πακέτα, που παρέχονται ως τμήμα του *Android SDK*. Έτσι κάποιος προγραμματιστής μπορεί να χρησιμοποιήσει το δημοφιλέστερο περιβάλλον ανάπτυξης, το *Eclipse*, που είναι δωρεάν και υποστηρίζεται επαρκώς από την ομάδα του *Android*, μπορεί να γράφει τις εντολές στο συνηθέστερο λειτουργικό σύστημα *Windows* και μπορεί να επιλέξει για ποιες συσκευές θα αναπτύξει την εφαρμογή του (με βάση την έκδοση του *Android*, στην οποία τρέχουν οι συσκευές).

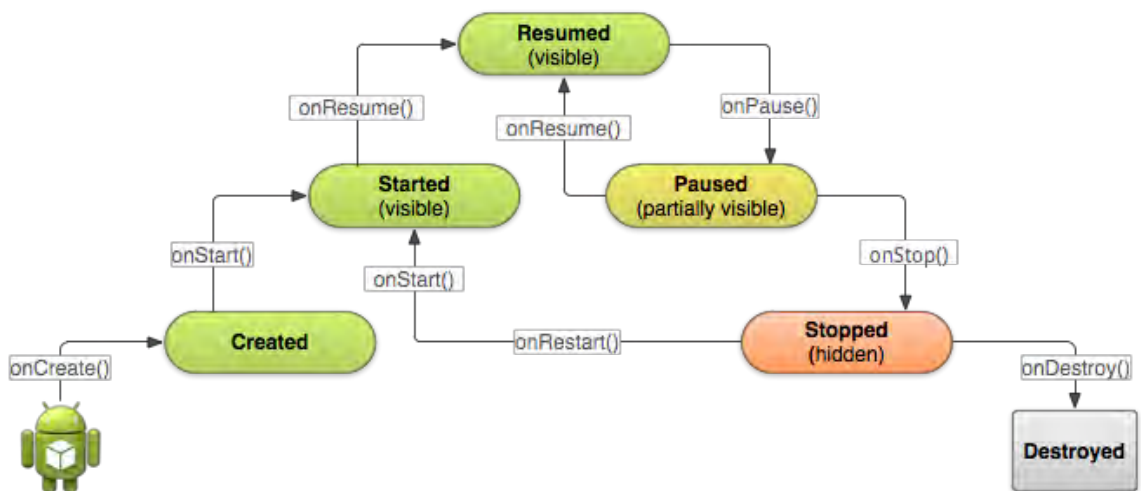
Η ανάπτυξη εφαρμογών *Android* ήταν μια πρόκληση καθώς ήταν κάτι καινούριο όμως η χρήση της *Java* διευκόλυνε την δημιουργία της εφαρμογής, λόγω της εμπειρίας που απέκτησα από προηγούμενες εργασίες που ήταν αποκλειστικά γραμμένες σε αυτή τη γλώσσα προγραμματισμού.

2.4 Η ΖΩΗ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ ANDROID

Αντίθετα από άλλα παραδείγματα προγραμματισμού στα οποία η εφαρμογή ξεκινάει με μια κύρια μέθοδο (main()), στο Android το σύστημα αρχίζει τον κώδικα σε μια δραστηριότητα (Activity) με την κλήση συγκεκριμένων μεθόδων επανάκλησης (callback) που αντιστοιχούν στα συγκεκριμένα στάδια του κύκλου της ζωής της. Υπάρχει μια ακολουθία μεθόδων επανάκλησης που ξεκινούν μια δραστηριότητα και μια ακολουθία μεθόδων επανάκλησης που καταστρέφουν μια δραστηριότητα.

Κατά τη διάρκεια της ζωής μιας δραστηριότητας, το σύστημα καλεί ένα σύνολο μεθόδων κύκλου ζωής σε μια ακολουθία παρόμοια με μια πυραμίδα βημάτων. Δηλαδή κάθε στάδιο του κύκλου ζωής της δραστηριότητας είναι ένα ξεχωριστό βήμα στην πυραμίδα. Δεδομένου ότι το σύστημα δημιουργεί μια νέα περίπτωση δραστηριότητας, κάθε μέθοδος επανάκλησης μετακινεί τη κατάσταση της δραστηριότητας ένα βήμα προς την κορυφή. Η κορυφή της πυραμίδας είναι το σημείο στο οποίο η δραστηριότητα τρέχει στο πρώτο πλάνο και ο χρήστης μπορεί να αλληλεπιδράσει μαζί της.

Δεδομένου ότι ο χρήστης αρχίζει να αφήνει τη δραστηριότητα, το σύστημα καλεί άλλες μεθόδους που μετακινούν τη κατάσταση της δραστηριότητας ένα βήμα προς τα κάτω στην πυραμίδα προκειμένου να αποσυναρμολογηθεί η δραστηριότητα. Σε ορισμένες περιπτώσεις, η δραστηριότητα θα κινηθεί μερικώς κάτω από την πυραμίδα και θα περιμένει (όπως όταν μεταπηδά ο χρήστης σε μια άλλη εφαρμογή), από ποιο σημείο η δραστηριότητα μπορεί να κινηθεί πίσω στην κορυφή (εάν ο χρήστης επιστρέψει στη δραστηριότητα) και να συνεχίσει από εκεί όπου ο χρήστης έφυγε.



2.5 ΟΙ ΜΕΘΟΔΟΙ ΤΟΥ ΚΥΚΛΟΥ ΖΩΗΣ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ

OnCreate ()

Μόλις το σύστημα καλέσει την onCreate() η μέθοδος αυτή καλεί αμέσως την onStart().

OnStart ()

Μόλις το σύστημα καλέσει την onStart() η μέθοδος αυτή καλεί αμέσως την onResume().

OnResume ()

Σε αυτή τη κατάσταση η δραστηριότητα τρέχει στο πρώτο πλάνο και ο χρήστης μπορεί να αλληλεπιδράσει μαζί της.

OnPause ()

Σε αυτή τη κατάσταση η δραστηριότητα κρύβεται μερικώς από άλλη δραστηριότητα- άλλη δραστηριότητα που είναι στο πρώτο πλάνο είναι ημιδιάφανη ή δεν καλύπτει ολόκληρη την οθόνη. Η σταματημένη δραστηριότητα δεν λαμβάνει δεδομένα εισόδου και δεν μπορεί να εκτελέσει οποιοδήποτε κώδικα.

OnStop ()

Σε αυτή τη κατάσταση η δραστηριότητα είναι εντελώς κρυμμένη και μη ορατή στο χρήστη και θεωρείται ότι είναι στο υπόβαθρο. Όταν η δραστηριότητα είναι σταματημένη όλες οι πληροφορίες κατάστασης όπως οι μεταβλητές διατηρούνται αλλά δεν μπορεί να εκτελέσει τον οποιοδήποτε κώδικα.

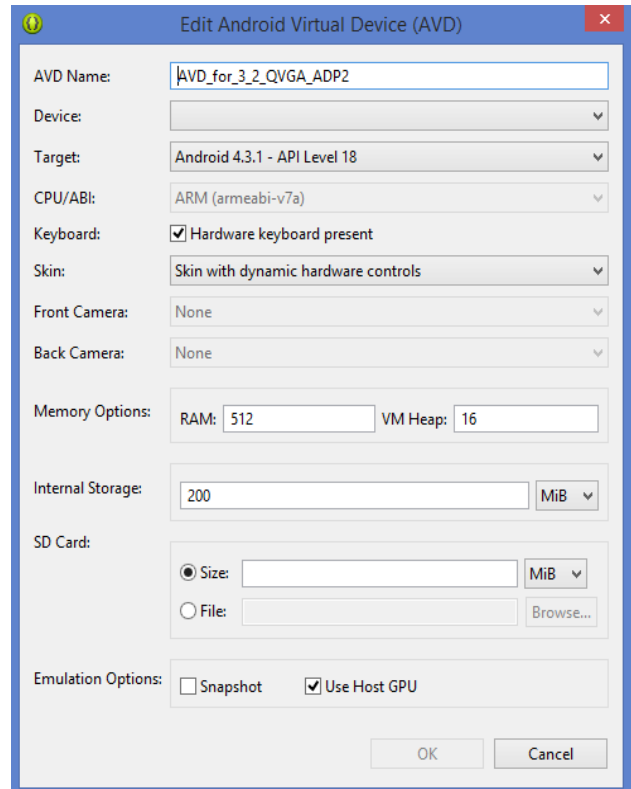
OnDestroy ()

Το σύστημα καλεί αυτή τη μέθοδο για τη δραστηριότητά, ως το τελικό σήμα όταν η δραστηριότητα πρόκειται να αφαιρεθεί τελείως από τη μνήμη του συστήματος.

ΚΕΦΑΛΑΙΟ 3^ο

3.1 Android Virtual Machine(AVD)-Προσομοιωτής Μηχανής Android

Για τους σκοπούς της εργασίας χρειάστηκε η δημιουργία ενός προσομοιωτή μηχανής Android στον οποίο δοκιμάστηκε η εφαρμογή μέχρι να βγει το τελικό επιθυμητό αποτέλεσμα. Τα δεδομένα που χρησιμοποιήθηκαν εμφανίζονται στην εικόνα δίπλα και ο προσομοιωτής πιο κάτω.



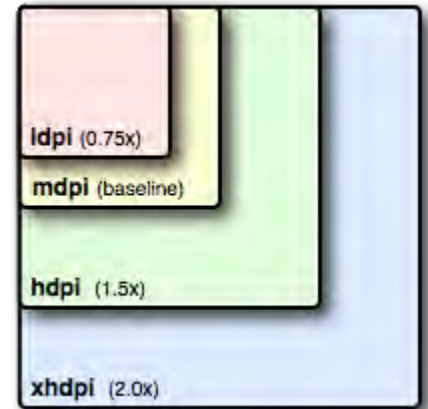
3.2 Υποστήριξη Πολλών Οθονών

Το Android τρέχει σε μια ποικιλία συσκευών που προσφέρουν διαφορετικά μεγέθη οθόνης και πυκνότητες. Για εφαρμογές, το σύστημα Android παρέχει ένα συνεπές περιβάλλον ανάπτυξης σε όλες τις συσκευές και χειρίζεται το μεγαλύτερο μέρος της εργασίας για την προσαρμογή του περιβάλλοντος εργασίας του χρήστη για κάθε αίτηση για την οθόνη στην οποία εμφανίζεται. Ταυτόχρονα, το σύστημα παρέχει API που επιτρέπει τον έλεγχο UI της

εφαρμογής για συγκεκριμένα μεγέθη οθόνης και πυκνότητες, με στόχο τη βελτιστοποίηση του σχεδιασμού UI για διαφορετικές ρυθμίσεις παραμέτρων της οθόνης.

Για τους σκοπούς της εργασίας ώστε να είναι συμβατή με πολλές συσκευές χρειάστηκε να μετατραπούν οι ίδιες εικόνες σε διάφορα μεγέθη και πυκνότητες.

Ένα σύνολο έξι γενικευμένων πυκνοτήτων φαίνεται πιο κάτω:



- *ldpi* (χαμηλό) ~ 120dpi
- *mdpi* (μέσο) ~ 160dpi
- *hdpi* (υψηλό) ~ 240dpi
- *xhdpi* (υπερυψηλής) ~ 320dpi
- *xxhdpi* (έξτρα-υπερυψηλής) ~ 480dpi

3.3 Περιγραφή Κλάσεων και δραστηριοτήτων

Κλάσεις

Οι απλές κλάσεις είναι ο κώδικας ο οποίος είναι γραμμένος αποκλειστικά σε Java και δημιουργεί τα αντικείμενα του παιχνιδιού.

Coordinates.java

Η κλάση αυτή καθορίζει ανά πάσα στιγμή τη θέση του κάθε αντικειμένου. Καλείται από την κλάση *Contents.java* κατά τη δημιουργία της, αρχικοποιείται και ενημερώνεται μετά από κάθε αλλαγή θέσης του αντικειμένου.

Contents.java

Η κλάση *Contents.java* είναι η κύρια κλάση στη δημιουργία του παιχνιδιού αλλά και η πιο γενική καθώς οι υπόλοιπες κλάσεις την επεκτείνουν είτε άμεσα είτε μέσω άλλων κλάσεων και χρησιμοποιούν τις μεθόδους τις ή τις ξαναγράφουν.

TerrainFeature.java

Η κλάση *TerrainFeature.java* επεκτείνει την κλάση *Contents.java* και μέσω αυτής ορίζονται τα αντικείμενα που βρίσκονται πάνω στην επιφάνεια του παιχνιδιού.

Trap.java

Η κλάση επεκτείνει την κλάση *TerrainFeature.java* και μέσω αυτής δημιουργούνται οι παγίδες που βρίσκονται διάσπαρτες τυχαία στην επιφάνεια του παιχνιδιού. Ο αριθμός των παγίδων είναι τυχαίος αλλά κυμαίνεται σε ένα εύρος από 1 μέχρι 10.

Tunnel.java

Η κλάση επεκτείνει την κλάση *TerrainFeature.java* και μέσω αυτής δημιουργούνται τα τούνελ που βρίσκονται στα 4 άκρα της επιφάνειας παιχνιδιού. Όταν το ποντίκι (Jerry) εισέλθει στο τούνελ θα εμφανιστεί κάπου τυχαία στην επιφάνεια παιχνιδιού, σε κάποιο σημείο που δεν κατοικείται ήδη.

Animal.java

Η κλάση *Animal.java* επεκτείνει την κλάση *Contents.java* και μέσω αυτής ορίζονται τα ζώα που βρίσκονται πάνω στην επιφάνεια του παιχνιδιού.

Enemy.java

Η κλάση αυτή επεκτείνει την κλάση *Animal.java* και μέσω αυτής ορίζονται τα εχθρικά ζώα που βρίσκονται πάνω στην επιφάνεια του παιχνιδιού.

Cat.java

Η κλάση αυτή επεκτείνει την κλάση *Enemy.java* και μέσω αυτής δημιουργούνται οι γάτες (Tom) που βρίσκονται διάσπαρτες τυχαία στην επιφάνεια του παιχνιδιού. Ο αριθμός των γάτων διαφέρει ανάλογα με το επίπεδο δυσκολίας που έχει επιλέξει ο χρήστης.

Friend.java

Η κλάση αυτή επεκτείνει την κλάση *Animal.java* και μέσω αυτής ορίζονται τα φιλικά ζώα που βρίσκονται πάνω στην επιφάνεια του παιχνιδιού.

Mouse.java

Η κλάση αυτή επεκτείνει την κλάση *Friend.java* και μέσω αυτής δημιουργείται το ποντίκι (Jerry) που εμφανίζεται σε μη καθορισμένη θέση στην επιφάνεια του παιχνιδιού.

Δραστηριότητες(Activities)

Κάθε εφαρμογή που δημιουργούμε αποτελείται από ένα τουλάχιστον activity ενώ επαγγελματικές εφαρμογές περιέχουν πολλά περισσότερα activities. Όσα activities και να έχουμε όμως πάντα ένα θεωρείται το κύριο activity το οποίο παρουσιάζεται στον χρήστη όταν ξεκινάει η εφαρμογή. Από εκεί και πέρα, κάθε activity μπορεί να καλέσει ένα άλλο activity αν το απαιτεί η εφαρμογή μας.

Κάθε φορά που ένα καινούργιο activity ξεκινάει το προηγούμενο activity σταματάει και το σύστημα σπρώχνει αυτό το activity στο stack. Το stack έχει τη λογική του “last in, first out”. Αυτό σημαίνει όταν το τρέχον activity ολοκληρώσει την λειτουργία του και εμείς πατήσουμε το back κουμπί, τότε το activity από το stack ξαναμπαίνει σε λειτουργία.

MainActivity.java

Η κύρια δραστηριότητα του παιχνιδιού. Όταν ο χρήστης επιλέξει το παιχνίδι ενεργοποιείται η δραστηριότητα αυτή και εμφανίζεται στην οθόνη το κυρίως παράθυρο. Αυτό περιλαμβάνει το κουμπί Play και Rate στο μέσο της οθόνης καθώς και τα κουμπιά info και exit στη γραμμή εργαλείων.



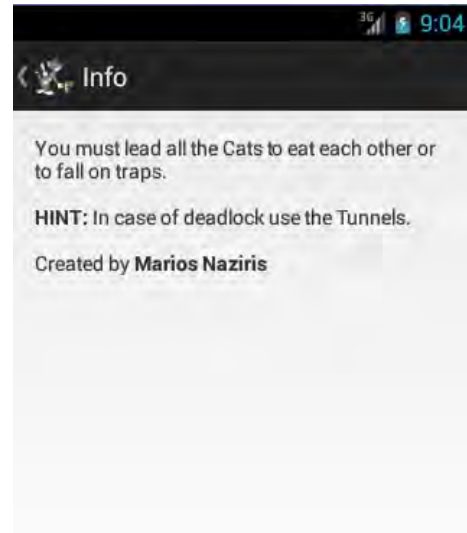
DifficultyActivity.java

Όταν επιλεγεί το κουμπί Play, ενεργοποιείται η δραστηριότητα *DifficultyActivity.java* και εμφανίζεται στην οθόνη ένα νέο παράθυρο που περιλαμβάνει 3 κουμπιά, ένα για κάθε επίπεδο δυσκολίας. Τα κουμπιά αυτά είναι το easy, για εύκολο παιχνίδι, το medium, για μέτριας δυσκολίας παιχνίδι, και το hard για δύσκολο παιχνίδι.



Info.java

Στη γραμμή εργαλείων υπάρχει το εικονίδιο πληροφοριών. Όταν ο χρήστης πατήσει το κουμπί αυτό εμφανίζεται ένα νέο παράθυρο στην οθόνη στο οποίο υπάρχουν οδηγίες για το παιχνίδι και γενικές πληροφορίες.



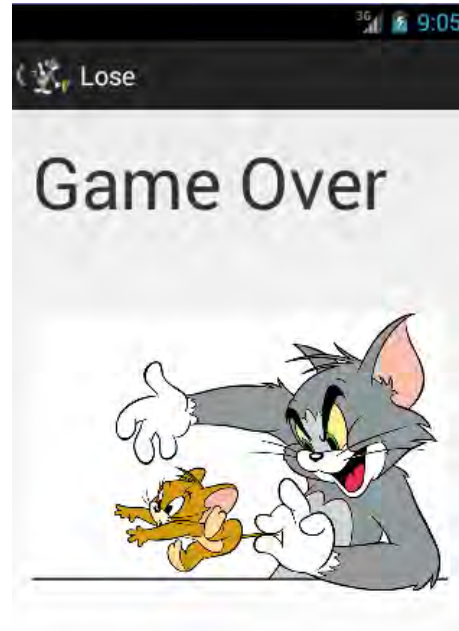
Level.java

Όταν επιλεγεί το επίπεδο δυσκολίας ενεργοποιείται η δραστηριότητα αυτή και στην οθόνη εμφανίζεται η επιφάνεια του παιχνιδιού και από κάτω τα κουμπιά για την κίνηση του παιχνιδιού `up`(πάνω), `down`(κάτω), `right`(δεξιά) και `left`(αριστερά).



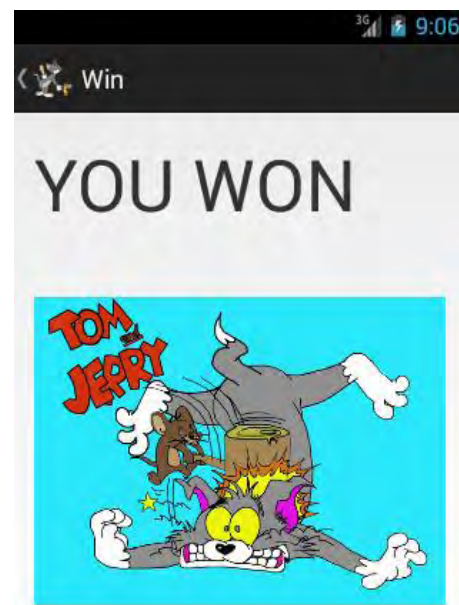
Lose.java

Η δραστηριότητα αυτή ενεργοποιείται όταν ο Jerry πέσει σε παγίδα ή έχει φαγωθεί από γάτα και ο χρήστης έχει χάσει. Με την κλήση αυτής της δραστηριότητας τερματίζεται το παιχνίδι και εμφανίζεται μήνυμα ήττας.



Win.java

Η δραστηριότητα αυτή ενεργοποιείται όταν ο Jerry καταφέρει να σκοτώσει όλες τις γάτες και κερδίσει το παιχνίδι. Με την κλήση αυτής της δραστηριότητας τερματίζεται το παιχνίδι και εμφανίζεται μήνυμα νίκης.



Αρχεία XML

AndroidManifest.xml

Στο αρχείο αυτό καθορίζεται η κύρια δραστηριότητα καθώς και η γονική δραστηριότητα κάθε δραστηριότητας. Το αρχείο περιλαμβάνει επίσης ορισμένες πληροφορίες και δεδομένα για κάθε δραστηριότητα ξεχωριστά.

Activity_main.xml

Το *Activity_main.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου της κυρίας οθόνης που περιλαμβάνει το κουμπί Play και Rate στο μέσο της οθόνης καθώς και τα κουμπιά info και exit στη γραμμή εργαλείων.

Activity_info.xml

Το *Activity_info.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου πληροφοριών στο οποίο υπάρχουν οδηγίες για το παιχνίδι και γενικές πληροφορίες.

Activity_difficulty.xml

Το *Activity_difficulty.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου επιλογής δυσκολίας στην οποία υπάρχουν τα κουμπιά easy, medium και hard.

Activity_level.xml

Το *Activity_level.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου επιφάνειας παιχνιδιού. Στο παράθυρο αυτό υπάρχουν εκατό αντικείμενα εικόνες, κάθε ένα από αυτά αντιπροσωπεύει ένα αντικείμενο του παιχνιδιού. Είναι επίσης υπεύθυνο για τον σχηματισμό των κουμπιών για τις κινήσεις up(πάνω), down(κάτω), right(δεξιά) και left(αριστερά).

Activity_lose.xml

Το *Activity_lose.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου που εμφανίζεται όταν ο παίκτης ηττηθεί και περιέχει ουσιαστικά μόνο το μήνυμα ήττας.

Activity_win.xml

Το *Activity_win.xml* είναι υπεύθυνο για τον σχηματισμό του παραθύρου που εμφανίζεται όταν ο παίκτης νικήσει και περιέχει ουσιαστικά μόνο το μήνυμα νίκης.

Strings.xml

Είναι το αρχείο το οποίο περιέχει τους πόρους συμβολοσειρών των εφαρμογών.

ΚΩΔΙΚΑΣ

Coordinates.java

```
Coordinates.java ☒
package com.developer.catandmousegame;

public class Coordinates{
    private int x;
    private int y;

    public Coordinates(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public Coordinates(Coordinates location) {
        this.x = location.x;
        this.y = location.y;
    }

    public int getX() {
        return this.x;
    }

    public int getY() {
        return this.y;
    }

    @Override
    public String toString() {
        return "";
    }

    @Override
    public boolean equals(Object obj) {
        if(obj.getClass().equals(this.getClass())){
            Coordinates ant = ((Coordinates) obj);
            if(this.x == ant.x && this.y == ant.y){
                return true;
            }
        }
        return false;
    }
}
```

Contents.java

```
Contents.java ☒
package com.developer.catandmousegame;

public abstract class Contents {
    private Coordinates location;

    public Contents(Coordinates location) {
        this.location = new Coordinates(location);
    }

    public Contents(int x, int y) {
        this.location = new Coordinates(x, y);
    }

    public Coordinates getLocation() {
        return new Coordinates(location);
    }

    public void setLocation(Coordinates newLocation) {
        this.location = new Coordinates(newLocation);
    }

    public abstract boolean canBeOccupied();
    public abstract boolean canBeKilled();
    public abstract boolean canKill();
}
```


TerrainFeature.java

```
TerrainFeature.java ✕
package com.developer.catandmousegame;

public class TerrainFeature extends Contents{

    public TerrainFeature() {
        super(0, 0);
    }

    public TerrainFeature(int x, int y) {
        super(x, y);
    }

    public TerrainFeature(Coordinates point) {
        super(point);
    }

    @Override
    public String toString() {
        return(" ");
    }

    public boolean canBeOccupied() {
        return true;
    }

    public boolean canBeKilled(){
        return false;
    }

    public boolean canKill(){
        return (false);
    }
}
```

Trap.java

```
Trap.java ✕
package com.developer.catandmousegame;

public class Trap extends TerrainFeature {

    public Trap(int x, int y) {
        super(x, y);
    }

    public Trap(Coordinates point) {
        super(point);
    }

    public boolean canBeOccupied() {
        return(false);
    }

    public boolean canKill(){
        return (true);
    }

    public String toString() {
        return("TR");
    }
}
```

Tunnel.java

```
Tunnel.java ✖
package com.developer.catandmousegame;

import java.util.Random;

public class Tunnel extends TerrainFeature {

    Random tixeos = new Random();
    Coordinates telos = new Coordinates(tixeos.nextInt(9), tixeos.nextInt(9));

    public Tunnel(int x, int y) {
        super(x, y);
    }

    public Tunnel(Coordinates point) {
        super(point);
    }

    public String toString() {
        return "T";
    }

    public boolean canBeOccupied() {
        return false;
    }
}
```

Animal.java

```
Animal.java ✖
package com.developer.catandmousegame;

public abstract class Animal extends Contents {

    private boolean isDead;

    public Animal(Coordinates location) {
        super(location);
        isDead = false;
    }

    public Animal(int x, int y) {
        super(x, y);
        isDead = false;
    }

    public boolean isDead() {
        return isDead;
    }

    public void kill() {
        isDead = true;
    }

    public void move(Coordinates newLocation) {
        setLocation(newLocation);
    }

    public boolean canBeOccupied() {
        return false;
    }

    public boolean canBeKilled(){
        return true;
    }

    public boolean canKill(){
        return false;
    }
}
```

Enemy.java

```
Enemy.java ☒
package com.developer.catandmousegame;

public abstract class Enemy extends Animal {

    public Enemy(int x, int y) {
        super(x, y);
    }

    public Enemy(Coordinates location) {
        super(location);
    }

    public abstract Coordinates chase(Coordinates target);
}
```

Cat.java

```
Cat.java ☒
package com.developer.catandmousegame;

public class Cat extends Enemy {

    public Cat(int x, int y) {
        super(x, y);
    }

    public Cat(Coordinates location) {
        super(location);
    }

    @Override
    public String toString() {
        return "C";
    }

    public boolean canKill() {
        return true;
    }
}

public Coordinates chase(Coordinates target){

    int CatX = this.getLocation().getX();
    int CatY = this.getLocation().getY();
    Coordinates newLocation;
    newLocation = this.getLocation();
    if(this.getLocation().getX() > target.getX()){
        CatX--;
    }
    else if(this.getLocation().getX() < target.getX()){
        CatX++;
    }

    if(this.getLocation().getY() > target.getY()){
        CatY--;
    }
    else if(this.getLocation().getY() < target.getY()) {
        CatY++;
    }
    newLocation = new Coordinates(CatX,CatY);
    return newLocation;
}
```

Friend.java

```
Friend.java ✕  
package com.developer.catandmousegame;  
  
public abstract class Friend extends Animal {  
    public Friend(int x, int y) {  
        super(x, y);  
    }  
    public Friend(Coordinates location) {  
        super(location);  
    }  
}
```

Mouse.java

```
Mouse.java ✕  
package com.developer.catandmousegame;  
  
public class Mouse extends Friend {  
    public Mouse(int x, int y) {  
        super(x, y);  
    }  
    public Mouse(Coordinates location) {  
        super(location);  
    }  
    @Override  
    public String toString() {  
        return "M";  
    }  
}
```

MainActivity.java

```
MainActivity.java ✕
public class MainActivity extends Activity {

    public final static String EXTRA_MESSAGE = "com.developer.catandmousegame.MESSAGE";

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.

        switch (item.getItemId()) {
            case R.id.action_exit:
                exit();
                return true;
            case R.id.action_info:
                info();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    private void info() {
        Intent intent = new Intent(this, Info.class);
        startActivity(intent);
    }

    private void exit() {
        System.exit(0);
    }

    public void play(View view) {

        Intent intent = new Intent(this, DifficultyActivity.class);
        startActivity(intent);
    }
}
```

DifficultyActivity.java

```
DifficultyActivity.java ✕
import android.app.Activity;

public class DifficultyActivity extends Activity {
    public final static String EXTRA_MESSAGE2 = "com.developer.catandmousegame.MESSAGE2";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_difficulty);
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.difficulty, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    public void easy(View view) {

        Intent intent = new Intent(this, Level.class);
        String message = "easy";
        intent.putExtra(EXTRA_MESSAGE2, message);
        startActivity(intent);
    }

    public void medium(View view) {

        Intent intent = new Intent(this, Level.class);
        String message = "medium";
        intent.putExtra(EXTRA_MESSAGE2, message);
        startActivity(intent);
    }

    public void hard(View view) {

        Intent intent = new Intent(this, Level.class);
        String message = "hard";
        intent.putExtra(EXTRA_MESSAGE2, message);
        startActivity(intent);
    }
}
```

Info.java

```
Info.java ✕  
  
public class Info extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_info);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.info, menu);  
        return true;  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // Handle action bar item clicks here. The action bar will  
        // automatically handle clicks on the Home/Up button, so long  
        // as you specify a parent activity in AndroidManifest.xml.  
        int id = item.getItemId();  
        if (id == R.id.action_settings) {  
            return true;  
        }  
        return super.onOptionsItemSelected(item);  
    }  
}
```

Level.java

```
Level.java ✕  
  
public class Level extends Activity {  
  
    public static final int ROWS = 10;  
    public static final int COLS = 10;  
    private Contents[][] world;  
    private Enemy[] enemies;  
    private Mouse hero;  
    private int numberOfEnemies;  
    private int NumTrap;  
    private Trap[] pagida;  
    private Tunnel[] tunnel = new Tunnel[4];  
    ImageView image;  
    enum Direction {LEFT, RIGHT, UP, DOWN};  
    String message;  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_level);  
        getActionBar().setDisplayHomeAsUpEnabled(true);  
        Intent intent = getIntent();  
        message = intent.getStringExtra(DifficultyActivity.EXTRA_MESSAGE2);  
        createWorld();  
        draw();  
    }  
}
```

```
Level.java ✕
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.level, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

public void draw(){
    int number = R.id.image00;
    for (int i=0; i<COLS; i++) {
        for (int j=0; j<ROWS; j++) {
            image = (ImageView) findViewById(number);
            number++;
            if(world[i][j].toString().equals("C")){
                image.setImageResource(R.drawable.cat);

            }else if(world[i][j].toString().equals("M")){
                image.setImageResource(R.drawable.mouse);
            }else if(world[i][j].toString().equals("TR")){
                image.setImageResource(R.drawable.trap);
            }else if(world[i][j].toString().equals("T")){
                image.setImageResource(R.drawable.tunnel);
            }else{ image.setImageResource(R.drawable.black);
            }

        }
    }
}

public void up(View view) {
    move(Direction.UP);
    if (isGameOver()) {
        displayWinner();
        System.exit(0);
    }
}
```


Level.java

```
public void down(View view) {
    move(Direction.DOWN);
    if (isGameOver()) {
        displayWinner();
        System.exit(0);
    }
}

public void right(View view) {
    move(Direction.RIGHT);
    if (isGameOver()) {
        displayWinner();
        System.exit(0);
    }
}

public void left(View view) {
    move(Direction.LEFT);
    if (isGameOver()) {
        displayWinner();
        System.exit(0);
    }
}

public void lose() {

    Intent intent = new Intent(this, Lose.class);
    startActivity(intent);
}

public void win() {

    Intent intent = new Intent(this, Win.class);
    startActivity(intent);
}

private void displayWinner() {
    if (hero.isDead()) {
        lose();
    }
    else if (numberOfEnemies == 0){
        win();
    }
}

private boolean isGameOver() {
    if(hero.isDead()){
        return true;
    }
    if(numberOfEnemies == 0){
        return true;
    }
    return false;
}
```

```

private void createWorld() {
    Random randomizer = new Random();
    world = new Contents[ROWS][COLS];
    for (int i=0; i<ROWS; i++) {
        for (int j=0; j<COLS; j++) {
            world[i][j] = new TerrainFeature(i, j);
        }
    }
    Coordinates Tloc;
    Tloc = new Coordinates(0,0);
    tunnel[0] = new Tunnel(Tloc);
    Tloc = new Coordinates(0,9);
    tunnel[1] = new Tunnel(Tloc);
    Tloc = new Coordinates(9,0);
    tunnel[2] = new Tunnel(Tloc);
    Tloc = new Coordinates(9,9);
    tunnel[3] = new Tunnel(Tloc);

    world[0][0] = tunnel[0];
    world[0][COLS-1] = tunnel[1];
    world[ROWS-1][0] = tunnel[2];
    world[ROWS-1][COLS-1] = tunnel[3];
    do{
        NumTrap = randomizer.nextInt(10);
    }while(NumTrap == 0);
    pagida = new Trap[NumTrap];
    for (int i=0; i<NumTrap; i++) {
        Coordinates NewTrap;
        int x, y;
        do {
            x = randomizer.nextInt(ROWS);
            y = randomizer.nextInt(COLS);
            NewTrap = new Coordinates(x, y);
        } while(!world[x][y].canBeOccupied());
        pagida[i] = new Trap(NewTrap);
        world[NewTrap.getX()][NewTrap.getY()] = pagida[i];
    }
    do{
        if(message.equalsIgnoreCase("easy")){
            numberOfEnemies =randomizer.nextInt(4)+1;
        }else if(message.equalsIgnoreCase("medium")){
            numberOfEnemies =randomizer.nextInt(4)+4;
        }else if(message.equalsIgnoreCase("hard")){
            numberOfEnemies =randomizer.nextInt(4)+8;
        }else{numberOfEnemies=10;}
    }while(numberOfEnemies == 0);
    enemies = new Cat[numberOfEnemies];
}

```

```

    for (int i=0; i<numberOfEnemies; i++) {
        Coordinates NewCat;
        int x, y;
        do {
            x = randomizer.nextInt(ROWS);
            y = randomizer.nextInt(COLS);
            NewCat = new Coordinates(x, y);
        } while(!world[x][y].canBeOccupied());
        enemies[i] = new Cat(NewCat);
        world[NewCat.getX()][NewCat.getY()] = enemies[i];
    }

    //place mouse
    Coordinates loc;
    int x, y;
    do {
        x = randomizer.nextInt(ROWS);
        y = randomizer.nextInt(COLS);
        loc = new Coordinates(x, y);
    } while(!world[x][y].canBeOccupied());
    hero = new Mouse(loc);
    world[loc.getX()][loc.getY()] = hero;

```

```

private void move(Direction direction) {

    // try to move the hero
    Coordinates currentLocation = hero.getLocation();
    Coordinates newLocation;
    switch(direction) {
        case UP:
            newLocation = new Coordinates(currentLocation.getX()-1, currentLocation.getY());
            break;
        case DOWN:
            newLocation = new Coordinates(currentLocation.getX()+1, currentLocation.getY());
            break;
        case RIGHT:
            newLocation = new Coordinates(currentLocation.getX(), currentLocation.getY()+1);
            break;
        case LEFT:
            newLocation = new Coordinates(currentLocation.getX(), currentLocation.getY()-1);
            break;
        default:
            newLocation = new Coordinates(currentLocation);
    }

    // make sure the hero is not trying to leave the board
    if (!isValid(newLocation)) {
        newLocation = new Coordinates(currentLocation);
    }
}

```

```

for (int i=0; i<NumTrap; i++){
    if(newLocation.equals(pagida[i].getLocation())){
        hero.kill();
    }
}
for (int i=0; i<numberOfEnemies; i++){
    if(newLocation.equals(enemies[i].getLocation())){
        hero.kill();
    }
}
for (int i=0; i<numberOfEnemies; i++){
    if(newLocation.equals(enemies[i].chase(hero.getLocation()))){
        hero.kill();
    }
}
for (int i=0; i<4; i++){
    if(newLocation.equals(tunnel[i].getLocation())){
        newLocation = tunnel[i].telos;
    }
}
hero.move(newLocation);
world[currentLocation.getX()][currentLocation.getY()] = new TerrainFeature(currentLocation);
world[newLocation.getX()][newLocation.getY()] = hero;

Coordinates[] MoveCat = new Coordinates[numberOfEnemies];
for (int i=0; i<numberOfEnemies; i++){
    MoveCat[i] = enemies[i].chase(currentLocation);
}

```

```

for (int i=0; i<numberOfEnemies; i++){

    world[enemies[i].getLocation().getX()][enemies[i].getLocation().getY()] = new Terr
    enemies[i].move(MoveCat[i]);

    if(world[MoveCat[i].getX()][MoveCat[i].getY()].canKill()){
        MoveCat[i] = MoveCat[numberOfEnemies-1];
        enemies[i] = enemies[numberOfEnemies-1];
        enemies[numberOfEnemies-1].kill();
        MoveCat[numberOfEnemies-1] = null;
        numberOfEnemies--;
        i--;
    }
    else{
        world[MoveCat[i].getX()][MoveCat[i].getY()] = enemies[i];
    }
    draw();
}

private boolean isValid(Coordinates loc) {
    return loc.getX() >=0 && loc.getX() < COLS && loc.getY() >=0 && loc.getY() < ROWS;
}
}

```

Lose.java

```
Lose.java
public class Lose extends Activity {

    ImageView image;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lose);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        image = (ImageView) findViewById(R.id.lose);
        image.setImageResource(R.drawable.lose);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.lose, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Win.java

```
Win.java
public class Win extends Activity {

    ImageView image;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_win);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        image = (ImageView) findViewById(R.id.win);
        image.setImageResource(R.drawable.win);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        getMenuInflater().inflate(R.menu.win, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Android Developers . *What is Android?* Ανακτήθηκε 21 Ιουνίου 2015, από http://www.openhandsetalliance.com/android_overview.html.
2. Android Developers . *Supporting Multiple Screens* . Ανακτήθηκε 21 Ιουνίου, 2015 από http://developer.android.com/guide/practices/screens_support.html.
3. Android Developers. *Android Supported Media Formats* . Ανακτήθηκε 21 Ιουνίου, 2015 από <http://developer.android.com/guide/appendix/media-formats.html>
4. Android Developers. *Starting an Activity*. Ανακτήθηκε 21 Ιουνίου, 2015 από <http://developer.android.com/training/basics/activity-lifecycle/starting.html>
5. Android Official Blog. *Android Wear: New watch faces from around the world*. Ανακτήθηκε 21 Ιουνίου, 2015 από <http://officialandroid.blogspot.gr/>
6. Open Handset Alliance. *Android Overview*. Ανακτήθηκε 21 Ιουνίου, 2015 από <http://www.greekdroids.eu/h-iotasigmatauomicronrho943alpha-tauomicronupsilon-android.html>.
7. Open Handset Alliance. 2007. *Industry Leaders Announce Open Platform for Mobile Devices*. Ανακτήθηκε 21 Ιουνίου, 2015 από http://www.openhandsetalliance.com/press_110507.html
8. Topolsky, Joshua. *Google's Android OS early look SDK now available* Ανακτήθηκε 21 Ιουνίου, 2015 από <http://www.engadget.com/2007/11/12/googles-android-os-early-look-sdk-now-available/>
9. Κασάπογλου, Μ. 2013. *Android lesson 7: understanding activities*. Ανακτήθηκε 21 Ιουνίου, 2015 από <http://www.kassapoglou.com/android-lesson-7-understanding-activities/>
10. Μανέττας, Α. (2014). *Ανάπτυξη Android εφαρμογής: Δίαιτα(DietDroid)* (Μεταπτυχιακή διατριβή, Πανεπιστήμιο Πειραιώς)., 13, Ανακτήθηκε 21 Ιουνίου, 2015 από <http://digilib.lib.unipi.gr/dspace/bitstream/unipi/6473/1/Manettas.pdf>