



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση εφαρμογής Android

**Βαμβακάς Αλέξιος
Κολλάρου Μαργαρίτα**

Εισηγητής: : Δρ Ιωάννης Έλληνας, Καθηγητής

**ΑΘΗΝΑ
ΦΕΒΡΟΥΑΡΙΟΣ 2019**

(Κενό φύλλο)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση εφαρμογής Android

**Βαμβακάς Αλέξιος
Α.Μ. 42725**

**Κολλάρου Μαργαρίτα
Α.Μ. 43205**

Εισηγητής:

Δρ Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

**Ιωάννης Έλληνας
Πάρις Μαστροκώστας
Κωνσταντίνος Κουκουλέτσος**

Ημερομηνία εξέτασης 27/2/2019

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αλέξιος Βαμβακάς του Δημητρίου με αριθμό μητρώου 42725 και η κάτωθι υπογεγραμμένη Μαργαρίτα Κολλάρου του Ιωάννη , με αριθμό μητρώου 43205 φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

(Κενό φύλλο)

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της ανάπτυξης εφαρμογής Android. Την προσπάθειά μας αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα θέλαμε να ευχαριστήσουμε για την υπομονή του και για τις πολύτιμες συμβουλές του.

(Κενό φύλλο)

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη μίας εφαρμογής Android ώστε να βοηθήσει τους ασθενείς που ανακάμπτουν από εγκεφαλικό επεισόδιο να εξασκήσουν το λόγο και την ομιλία μέσω της εκφοράς γραμμάτων και λέξεων καθώς και να παίξουν και να διασκεδάσουν μαθαίνοντας. Μέσω της αναγνώρισης φωνής και της σύγκρισης, ο χρήστης μπορεί να έχει μια ένδειξη για την επίδοση του και να ξέρει πότε να προχωρήσει στην επόμενη λέξη η επίπεδο.

ABSTRACT

The present thesis concerns the development of an Android application that focuses on helping patients that recover from a stroke to practice their speech and cognitive procedures through articulating letters and words or offers the opportunity to play and have fun while learning. Through the process of speech recognition and the comparison with the user input, the user can have a clear indication of his effort and whether he should proceed to the next word or level.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Android, Software Development

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: android, εφαρμογή, αναγνώριση φωνής

ΠΕΡΙΕΧΟΜΕΝΑ

1ο ΜΕΡΟΣ - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ	11
ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΕΦΑΡΜΟΓΕΣ ANDROID	11
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας	11
1.2 Ιστορική αναδρομή.....	11
1.3 Τρόπος λειτουργίας Android Εφαρμογής.....	12
ΚΕΦΑΛΑΙΟ 2	15
2.1 Ορισμός Android.....	15
2.2 Ιστορία	16
2.3 Ιστορικό Ενημερώσεων	19
2.4 Εφαρμογές.....	22
2.5 Οι απαιτήσεις υλικού	23
2.6 Αρχιτεκτονική του Android.....	24
2.7 Android Studio.....	25
2.8 JAVA.....	26
2.9 XML	32
2.10 Ανατομία μιας Android εφαρμογής.....	36
2ο ΜΕΡΟΣ - ΑΝΑΛΥΣΗ ΠΕΡΙΠΤΩΣΗΣ	46
ΚΕΦΑΛΑΙΟ 3 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	46
3.1 Οι ανάγκες που οδήγησαν στη δημιουργία της εφαρμογής.....	46
3.2 Υλοποίηση.....	46
3.3 Εγκατάσταση Android Studio.....	46
ΚΕΦΑΛΑΙΟ 4 Παρουσίαση Εφαρμογής	54
4.1 Αρχική Σελίδα	54
4.2 Λειτουργία LEARN Εκμάθησης	57
4.3 Λειτουργία PLAY Παιχνιδιού.....	61
ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα – Προοπτικές	63
5.1. Συμπεράσματα.....	63
5.2. Προοπτικές	63
ΒΙΒΛΙΟΓΡΑΦΙΑ	64

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1:	11
Σχήμα 1.2:	12
Σχήμα 1.3:	13
Σχήμα 2.1:	18
Σχήμα 2.2:	20

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1:	18
--------------------	----

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

IDE Interactive Development Environment
ADT Android Device Tools
JDK Java Development Kit
XML Extensible Markup Language
W3C World Wide Web Consortium
XHTML eXtensible HyperText Markup Language
HTML HyperText Markup Language
API Application Programming Interface
GPL General Public Language
VM Virtual Machine
JIT Just in Time
SOAP Simple Object Access Protocol
RSS Reach Site Summary
UTF Unicode Transformation Format

1ο ΜΕΡΟΣ - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΕΦΑΡΜΟΓΕΣ ANDROID

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια ιστορική αναδρομή γύρω από την εξέλιξη του Android

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής Android. Σύμφωνα με τις οδηγίες του Εθνικού Κέντρου Αποκατάστασης ώστε να βοηθήσει τους ασθενείς που ανακάμπτουν από εγκεφαλικό να εξασκήσουν την ομιλία τους.

1.2 Ιστορική αναδρομή

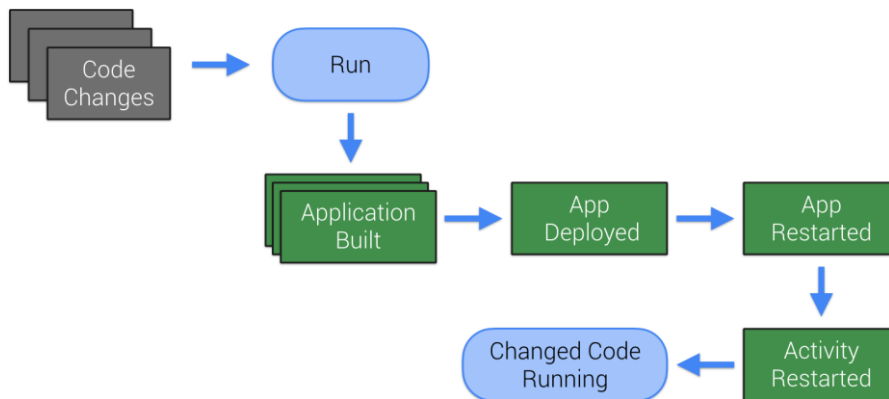
Οι πρώτες εφαρμογές Android δημιουργήθηκαν με τη δημιουργία της πλατφόρμας Google Play τον Οκτώβριο του 2008 με πρώτη εφαρμογή το Φιδάκι (Snake) όπου ο παίχτης ελέγχει μια τελεία ή τετράγωνο σε ένα οριοθετημένο επίπεδο. Καθώς αυτό κινείται, αφήνει ένα ίχνος πίσω του που μοιάζει με κινούμενο φίδι. Μέσα στο οριοθετημένο επίπεδο εμφανίζονται αντικείμενα ή ζώα και στόχος του κινούμενου φιδιού είναι να τα φάει.



Σχήμα 1. 1

Όσο περισσότερα αντικείμενα τρώει τόσο μακραίνει η ουρά του. Δεύτερος στόχος είναι να μην ακουμπήσει το τέλος της ουράς του με το κεφάλι του και να μην ακουμπήσει πουθενά πάνω στο οριοθετημένο επίπεδο κάνοντας ελιγμούς μέχρι να τερματίσει.

1.3 Τρόπος λειτουργίας Android Εφαρμογής



Σχήμα 1. 2

Τι είναι μια Android Εφαρμογή:

Μια Android εφαρμογή είναι μια software εφαρμογή που τρέχει στην πλατφόρμα Android. Αφού η πλατφόρμα Android έχει χτιστεί για συσκευές κινητών τηλεφώνων. Μια τυπική εφαρμογή Android είναι σχεδιασμένη για ένα smartphone ή ένα tablet PC το οποίο εκτελεί και υποστηρίζει το λειτουργικό σύστημα του Android.

Παρόλο που μια εφαρμογή Android μπορεί να διατεθεί στους προγραμματιστές μέσω της ιστοσελίδας του εκάστοτε προγραμματιστή, οι περισσότερες Android εφαρμογές ανεβαίνουν και δημοσιεύονται στο Android Market, μια ψηφιακή πλατφόρμα που αγοράζεις εφαρμογές . Το Android Market υποστηρίζει δωρεάν και επί πληρωμή εφαρμογές .

Οι εφαρμογές Android είναι γραμμένες σε γλώσσα Java και χρησιμοποιούν τις Java core βιβλιοθήκες. Πρώτα μεταγλωττίζονται σε εκτελέσιμα Dalvik ώστε να τρέχουν στην Ψηφιακή μηχανή (Dalvik virtual machine), η οποία είναι μια ψηφιακή μηχανή σχεδιασμένη ειδικά για κινητές συσκευές.

Οι Προγραμματιστές μπορούν να κατεβάσουν το λογισμικό του Android software development kit (SDK) από την ιστοσελίδα του Android. Το SDK περιέχει εργαλεία ,δείγματα κώδικα και σχετικά έγγραφα για την δημιουργία εφαρμογών Android.

Οι αρχάριοι προγραμματιστές που απλά θέλουν να πειραματιστούν με τον προγραμματισμό σε Android programming μπορούν να χρησιμοποιήσουν την Online εφαρμογή App Inventor. Χρησιμοποιώντας αυτήν την εφαρμογή που

βρίσκουν στο διαδίκτυο ο χρήστης μπορεί να κατασκευάσει μια εφαρμογή Android βήμα βήμα σαν τη διαδικασία κατασκευής ενός παζλ.

Ξεκινώντας από το κάτω κάτω επίπεδο όπου βρίσκεται το Linux Kernel. Το Android χρησιμοποιεί το Linux λόγω των device drivers, της διαχείρισης της μνήμης, εργασιών και του networking. Δεν θα προγραμματίσεις σε αυτό το επίπεδο ποτέ απευθείας.

Το επόμενο επίπεδο περιλαμβάνει τις Android native βιβλιοθήκες. Αυτές είναι γραμμένες σε C/C++ εσωτερικά. αλλά θα τις καλείς μέσω Java interfaces. Σε αυτό το επίπεδο μπορείς να βρεις τον Surface Manager, 2D and 3D γραφικά, Media codecs (MPEG-4, H.264, MP3, etc.), την βάση SQL(SQLite), και ένα native web browser engine (WebKit).

Το επόμενο είναι το Android runtime, που περιέχει το Dalvik Virtual Machine. Το Dalvik τρέχει αρχεία dex, τα οποία μετατρέπονται κατά τη διάρκεια της μεταγλώττισης από κανονικές κλάσεις σε αρχεία jar . Τα αρχεία Dex είναι πιο συμπαγή και αποτελεσματικά από ότι τα αρχεία class, και μια σημαντική βελτίωση για την περιορισμένη μνήμη και τις συσκευές που λειτουργούν με μπαταρία που είναι και ο στόχος του Android .

Οι core Java βιβλιοθήκες είναι μέρος του Android runtime. Είναι γραμμένες σε Java, όπως και το κάθε τι σε αυτό το επίπεδο. Εδώ, το Android παρέχει ένα σημαντικό [subset](#) από τα πακέτα της [Java 5](#) , που περιλαμβάνουν συλλογές [Collections](#), [I/O](#).

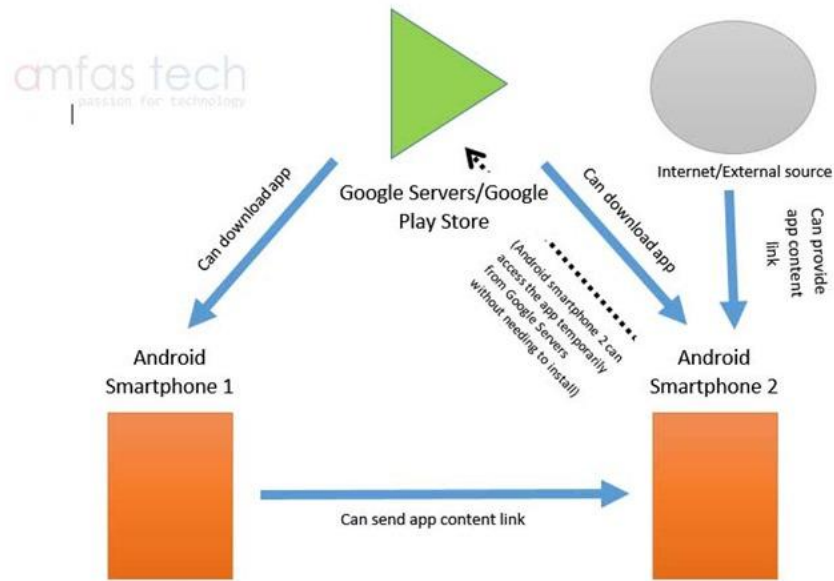
Στο επόμενο επίπεδο είναι το Application Framework . Κομμάτια αυτού του toolkit παρέχονται από την Google, και κομμάτια της αποτελούν οι επεκτάσεις ή οι υπηρεσίες που γράφεις. Το πιο σημαντικό component του framework είναι ο Activity Manager, ο οποίος διαχειρίζεται τον κύκλο ζωής των εφαρμογών και ένα κοινό μέτωπο "back-stack" για την πλοήγηση του χρήστη.

Τέλος, το ανώτερο επίπεδο είναι το επίπεδο των εφαρμογών (Applications layer).

Η πλειοψηφία του κώδικα υπάρχει εδώ όπως built-in applications εφαρμογές κινητών και φυλλομετρητών.

Μια από τις μοναδικές και δυνατές ιδιότητες του Android είναι όλες οι εφαρμογές που έχουν level playing field. Αυτό περιλαμβάνει τις εφαρμογές που η

Google γράφει και πρέπει να περάσουν μέσα από το δημόσιο API που χρησιμοποιείται.



Σχήμα 1. 3

ΚΕΦΑΛΑΙΟ 2

2.1 Ορισμός 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, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) 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 Ιρίνα Μπλόκ.

Η τελευταία έκδοση καλείται Android 9.0 Pie και φέρνει σημαντικές αλλαγές, όπως το Adaptive Battery και το Adaptive Brightness. Επίσης, στο Android 9.0 Pie υπάρχει ένα νέο σύστημα πλοήγησης με ένα μεμονωμένο κουμπί home.

2.2 Ιστορία

Το λειτουργικό σύστημα Android ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Ο Rubin χαρακτήρισε το έργο Android ως "τεράστιο δυναμικό για την ανάπτυξη έξυπνων κινητών συσκευών που έχουν μεγαλύτερη επίγνωση της θέσης και των προτιμήσεων του ιδιοκτήτη του". Οι πρώτες προθέσεις της εταιρείας ήταν να αναπτύξει ένα προηγμένο λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές και αυτή ήταν η βάση του πεδίου για τους επενδυτές τον Απρίλιο του 2004. Η εταιρεία αποφάσισε τότε ότι η αγορά των φωτογραφικών μηχανών δεν ήταν αρκετά μεγάλη για τους στόχους της, πέντε μήνες αργότερα είχε εκτρέψει τις προσπάθειές της και ανέβαζε το Android ως λειτουργικό σύστημα χειρός που θα ανταγωνιζόταν τα Symbian και τα Microsoft Windows Mobile.

Το λογότυπο του Android από το 2007 έως το 2014.



Ο Ρούμπιν είχε δυσκολία να προσελκύσει επενδυτές από νωρίς και το Android αντιμετώπιζε έξωση από το χώρο γραφείων του. Ο Steve Perlman, ένας στενός φίλος του Ρούμπιν, του έφερε 10.000 δολάρια σε μετρητά σε ένα φάκελο, και έπειτα εισπράχθηκε ένα μη ανακοινωθέν ποσό ως χρηματοδότηση σπόρων. Ο Perlman αρνήθηκε ένα μερίδιο στην εταιρεία και δήλωσε ότι "το έκανα επειδή πίστευα στο πράγμα και ήθελα να βοηθήσω τον Andy".

Τον Ιούλιο του 2005, η Google απέκτησε το Android Inc. για τουλάχιστον 50 εκατομμύρια δολάρια. Οι βασικοί υπάλληλοί της, όπως οι Rubin, Miner και White, εντάχθηκαν στο Google ως μέρος της εξαγοράς. Δεν γνώριζαν πολλά για το μυστικό Android την εποχή εκείνη, καθώς η εταιρεία παρέσχε λίγες λεπτομέρειες εκτός από το ότι έκανε λογισμικό για κινητά τηλέφωνα. Στο Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια πλατφόρμα κινητών συσκευών που τροφοδοτείται από τον πυρήνα του Linux. Η Google προώθησε την πλατφόρμα στους κατασκευαστές και τους μεταφορείς με την υπόσχεση παροχής ενός ευέλικτου, αναβαθμίσιμου συστήματος. Η Google είχε "παρατάξει μια σειρά από εξαρτήματα υλικού και συνεργάτες λογισμικού και σημείωσε στους μεταφορείς ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας".

Η Google άλλαξε αργότερα τα έγγραφα προδιαγραφών Android για να δηλώσει ότι θα υποστηρίζονται οι "Οθόνες αφής", αν και "το Προϊόν σχεδιάστηκε με την παρουσία διακριτών φυσικών κουμπιών ως υπόθεση, επομένως μια οθόνη αφής δεν μπορεί να αντικαταστήσει πλήρως τα φυσικά κουμπιά". Μέχρι το 2008, τόσο η Nokia όσο και το BlackBerry ανακοίνωσαν έξυπνα τηλέφωνα με βάση το touch για να ανταγωνιστεί το iPhone 3G και η εστίαση του Android τελικά μετατράπηκε σε απλές οθόνες αφής. Το πρώτο εμπορικά διαθέσιμο smartphone που τρέχει το Android ήταν το HTC Dream, επίσης γνωστό ως T-Mobile G1, που ανακοινώθηκε στις 23 Σεπτεμβρίου 2008.

Στις 5 Νοεμβρίου 2007, η Open Handset Alliance, μια κοινοπραξία εταιρειών τεχνολογίας όπως η Google, κατασκευαστές συσκευών όπως HTC, Motorola και Samsung, ασύρματες εταιρείες όπως η Sprint και η T-Mobile και κατασκευαστές chipset όπως η Qualcomm και η Texas Instruments με στόχο την ανάπτυξη της "πρώτης πραγματικά ανοικτής και ολοκληρωμένης πλατφόρμας για κινητές συσκευές". Μέσα σε ένα χρόνο η Open Handset Alliance αντιμετώπισε δύο άλλους ανταγωνιστές ανοικτής πηγής, το Symbian Foundation και το LiMo Foundation, το τελευταίο αναπτύσσοντας επίσης ένα κινητό λειτουργικό σύστημα όπως το Google. Τον Σεπτέμβριο του 2007, το Information Week κάλυψε μια μελέτη Evaluateserve που ανέφερε ότι η Google είχε καταθέσει αρκετές αιτήσεις ευρεσιτεχνίας στον τομέα της κινητής τηλεφωνίας.

Από το 2008, το Android έχει δει πολλές αναβαθμίσεις, οι οποίες έχουν βελτιώσει σταδιακά το λειτουργικό σύστημα, προσθέτοντας νέα χαρακτηριστικά και διορθώνοντας σφάλματα σε προηγούμενες κυκλοφορίες. Κάθε μεγάλη κυκλοφορία ονομάζεται με αλφαβητική σειρά μετά από επιδόρπιο ή ζαχαρούχο σκεύασμα, με τις πρώτες εκδόσεις του Android να ονομάζονται "Cupcake", "Donut", "Eclair" και "Froyo", με αυτή τη σειρά. Κατά την ανακοίνωση του Android KitKat το 2013, η Google εξήγησε ότι "Δεδομένου ότι αυτές οι συσκευές κάνουν τη ζωή μας τόσο γλυκιά, κάθε έκδοση Android ονομάζεται μετά από ένα επιδόρπιο", παρόλο που ένας εκπρόσωπος της Google δήλωσε στο CNN σε συνέντευξή του ότι "Είναι σαν μια εσωτερική ομάδα κάτι που προτιμάμε να είμαστε λίγο - πώς θα πρέπει να πω - λίγο ασυμβίβαστο στο θέμα, θα πω".

Το 2010, η Google παρουσίασε τη σειρά συσκευών Nexus, μια σειρά στα οποία η Google συνεργάστηκε με διαφορετικούς κατασκευαστές συσκευών για

την παραγωγή νέων συσκευών και την εισαγωγή νέων εκδόσεων Android. Η σειρά περιεγράφηκε ότι έχει "διαδραματίσει βασικό ρόλο στο ιστορικό του Android εισάγοντας νέες επαναλήψεις λογισμικού και πρότυπα υλικού σε όλο το πλάτος" και έγινε γνωστός για το λογισμικό "free-bloat" με "έγκαιρες [...] ενημερώσεις". Στη διάσκεψη προγραμματιστών τον Μάιο του 2013, η Google ανακοίνωσε μια ειδική έκδοση του Samsung Galaxy S4, όπου, αντί να χρησιμοποιήσει την προσαρμογή του Android, το τηλέφωνο έτρεξε "stock Android" και υποσχέθηκε να λάβει γρήγορα νέες ενημερώσεις συστήματος. Η συσκευή θα γίνει η αρχή του προγράμματος Google Play και θα ακολουθήσει άλλες συσκευές, όπως η έκδοση HTC One Google Play και η έκδοση MotoG Google Play. Το 2015, ο Ars Technica έγραψε ότι "Νωρίτερα αυτή την εβδομάδα, το τελευταίο από τα τηλέφωνα Android για την έκδοση Google Play στην ηλεκτρονική προθήκη της Google αναφέρθηκε ως" δεν είναι πλέον διαθέσιμο προς πώληση "και ότι" Τώρα όλα έχουν φύγει και φαίνεται όπως και το πρόγραμμα έχει ολοκληρωθεί".

Από το 2008 έως το 2013, ο Hugo Barra υπηρέτησε ως εκπρόσωπος των προϊόντων, εκπροσωπώντας το Android σε συνεντεύξεις Τύπου και το Google I / O, το ετήσιο συνέδριο της Google για τους προγραμματιστές. Έφυγε από την Google τον Αύγουστο του 2013 για να ενώσει την κινεζική εταιρεία κατασκευής τηλεφώνων Xiaomi. Λιγότερο από έξι μήνες νωρίτερα, ο τότε διευθύνων σύμβουλος της Google, Larry Page ανακοίνωσε σε blog, ότι ο Andy Rubin είχε μετακομίσει από το τμήμα Android για να αναλάβει νέα έργα στο Google και ότι η Sundar Pichai θα γίνει το νέο Android. Ο ίδιος ο Pichai θα αλλάξει τις θέσεις του και θα γίνει ο νέος CEO της Google τον Αύγουστο του 2015 μετά την αναδιάρθρωση της εταιρίας στο αλφαβητάρι αλφαβήτου, καθιστώντας τον Hiroshi Lockheimer νέο επικεφαλής του Android.

Τον Ιούνιο του 2014, η Google ανακοίνωσε το Android One, ένα σύνολο «μοντέλων αναφοράς υλικού» που θα «επιτρέψουν στους κατασκευαστές συσκευών να δημιουργήσουν εύκολα τηλέφωνα υψηλής ποιότητας με χαμηλό κόστος», σχεδιασμένα για καταναλωτές στις αναπτυσσόμενες χώρες. Τον Σεπτέμβριο, η Google ανακοίνωσε το πρώτο σετ Android One phones για κυκλοφορία στην Ινδία. Ωστόσο, η Recode ανέφερε τον Ιούνιο του 2015 ότι το έργο ήταν "απογοήτευση", αναφέροντας "διστακτικούς καταναλωτές και συνεργάτες κατασκευής" και "σφάλματα από την εταιρεία αναζήτησης που ποτέ

δεν έχει σπάσει το υλικό". Τα σχέδια για την επανενεργοποίηση του Android One εμφανίστηκαν τον Αύγουστο του 2015, με την Αφρική να ανακοινώθηκε ως επόμενη θέση για το πρόγραμμα μια εβδομάδα αργότερα. Μια αναφορά από τις πληροφορίες τον Ιανουάριο του 2017 δήλωσε ότι η Google επεκτείνει το πρόγραμμα Android One με χαμηλό κόστος στις Ηνωμένες Πολιτείες, αν και η The Verge σημειώνει ότι η εταιρεία πιθανότατα δεν θα παράγει τις ίδιες τις ίδιες τις συσκευές.

Η Google εισήγαγε τα smartphone Pixel και Pixel XL τον Οκτώβριο του 2016, τα οποία διατέθηκαν στο εμπόριο ως τα πρώτα τηλέφωνα που έκανε η Google και παρουσίαζαν αποκλειστικά ορισμένες λειτουργίες λογισμικού, όπως ο Βοηθός Google (Google Assistant), πριν από την ευρύτερη ανάπτυξη. Τα τηλέφωνα Pixel αντικατέστησαν τη σειρά Nexus, με μια νέα γενιά τηλεφώνων Pixel που ξεκίνησε τον Οκτώβριο του 2017.

2.3 Ιστορικό Ενημερώσεων



Σχήμα 2 1

Έκδοση :	Κωδική Ονομασία :	Ημερομηνία	API level :	Διανομή ^[15] :
1.6	<i>Donut</i>	15 Σεπτεμβρίου 2009	4	0 %
2.0	<i>Eclair</i>	26 Οκτωβρίου 2009	7	0 %
2.2	<i>Froyo</i>	20 Μαΐου 2010	8	0%

2.3	<i>Gingerbread</i>	9 Φεβρουαρίου 2011	10	0.2%
3.2	<i>Honeycomb</i>	15 Ιουλίου 2011	13	0%
4.0	<i>Ice Cream Sandwich</i>	16 Δεκεμβρίου 2011	15	0.3%
4.1	<i>Jelly Bean</i>	9 Ιουλίου 2012	16	1.2%
4.2	<i>Jelly Bean</i>	13 Νοεμβρίου 2012	17	1.9%
4.3	<i>Jelly Bean</i>	24 Ιουλίου 2013	18	0.5%
4.4	<i>KitKat</i>	31 Οκτωβρίου 2013	19	9.1%
5.0	<i>Lollipop</i>	3 Νοεμβρίου 2014	21	4.2%
5.1	<i>Lollipop</i>	9 Μαρτίου 2015	22	16.2%
6.0	<i>Marshmallow</i>	5 Οκτωβρίου 2015	23	23.5%
7.0	<i>Nougat</i>	22 Αυγούστου 2016	24	21.2%
7.1	<i>Nougat</i>	4 Οκτωβρίου 2016	25	9.6%
8.0	<i>Oreo</i>	21 Αυγούστου 2017	26	10.1%
8.1	<i>Oreo</i>	5 Δεκεμβρίου 2017	27	2.0%
9.0	<i>Pie</i>	6 Αυγούστου 2018	28	

Πίνακας 2 1

Παρόλο που το Android είναι ένα προϊόν ελεύθερου λογισμικού, ένα κομμάτι της ανάπτυξης του λογισμικού συνεχίζεται σε ιδιωτικό παρακλάδι.

Για να έρθει αυτό το λογισμικό σε κοινή θέαση δημιουργήθηκε ένα παρακλάδι του μόνο ανάγνωσης, εν ονόματι "Cupcake". Το Cupcake συνήθως συγχέεται με τον τίτλο μιας ενημέρωσης, σε αντίθεση με όσα δηλώνει η ίδια η Google στην ιστοσελίδα ανάπτυξης του Android: "το Cupcake αποτελεί ακόμη ένα έργο σε εξέλιξη, όχι μια επίσημη έκδοση." Αξιοσημείωτες αλλαγές στο λειτουργικό Android θα παρουσιαστούν στο cupcake και περιλαμβάνουν αλλαγές στο σύστημα διαχείρισης των μεταφορτώσεων (download manager), το framework, Bluetooth, το λογισμικό συστήματος, το ραδιόφωνο και το σύστημα τηλεφωνίας, εργαλεία προγραμματισμού, το κυρίως σύστημα και διάφορες εφαρμογές, καθώς και πληθώρα διορθώσεις σφαλμάτων.

Στις 7 Αυγούστου 2018, κυκλοφόρησε η επίσημη ενημέρωση έκδοσης 9.0 για το Android. Αποτελείται από πολλά νέα χαρακτηριστικά και βελτιώσεις στο γραφικό περιβάλλον:



Σχήμα 2 2

- Τη λειτουργία **Adaptive Battery**, η οποία δίνει προτεραιότητα στην κατανάλωση της μπαταρίας μόνο για τις εφαρμογές και τις υπηρεσίες που χρησιμοποιεί περισσότερο ο χρήστης, κλείνοντας ταυτόχρονα όλες τις υπόλοιπες εφαρμογές που "τρέχουν" στο background.
- Τη λειτουργία **Adaptive Brightness**, η οποία μαθαίνει πως να προσαρμόζει την φωτεινότητα της οθόνης ανάλογα με το περιβάλλον.
- Τα **App Actions** θα σε βοηθούν να πραγματοποιείς την επόμενη ενέργεια σου πιο γρήγορα επειδή θα αντιλαμβάνονται τι ακριβώς θέλεις να κάνεις.

Η εμφάνιση και η εμπειρία χρήσης γίνεται πιο εύκολη με το νέο σύστημα πλοήγησης.

- Το **Android Dashboard**, το οποίο θα δείχνει στον χρήστη αναλυτικά πόσο χρόνο ξοδεύει σε συγκεκριμένες εφαρμογές, πόσες φορές, ξεκλείδωσε το smartphone και πόσες ειδοποιήσεις έλαβε κατά τη διάρκεια της ημέρας.
- Το **App Timer** είναι ένα εργαλείο που θα του επιτρέπει να θέτει χρονικό περιορισμό στην χρήση μιας εφαρμογής, θα τον ειδοποιεί όταν φτάνει στο όριο που έχει θέσει και θα μετατρέπει το εικονίδιο της εφαρμογής σε ασπρόμαυρο για να του υπενθυμίσει.
- Η λειτουργία **Wind Down** θα ενεργοποιεί το **Night Light** όταν σκοτεινιάζει, θα βάζει την συσκευή σε **Do Not Disturb mode** και θα μετατρέπει την οθόνη σε ασπρόμαυρη την ώρα που έχεις δηλώσει ότι βρίσκεσαι στο κρεβάτι.

2.4 Εφαρμογές

Οι εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών, γράφονται χρησιμοποιώντας το κιτ ανάπτυξης λογισμικού Android (SDK) και, συχνά, τη γλώσσα προγραμματισμού Java. Η Java μπορεί να συνδυαστεί με το C / C ++, μαζί με μια επιλογή μη προεπιλεγμένων χρόνων εκτέλεσης που επιτρέπουν καλύτερη υποστήριξη C ++. Η γλώσσα προγραμματισμού Go υποστηρίζεται επίσης, αν και με περιορισμένο σύνολο διεπαφών προγραμματισμού εφαρμογών (API). Τον Μάιο του 2017, η Google ανακοίνωσε την υποστήριξη για την ανάπτυξη εφαρμογών Android στη γλώσσα προγραμματισμού Kotlin.

Το SDK περιλαμβάνει ένα πλήρες σύνολο εργαλείων ανάπτυξης, συμπεριλαμβανομένου ενός προγράμματος εντοπισμού σφαλμάτων, βιβλιοθηκών λογισμικού, ενός εξομοιωτή ακουστικών βασισμένου σε QEMU, τεκμηρίωσης, κώδικα δείγματος και σεμινάρια. Αρχικά, το υποστηριζόμενο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) της Google ήταν το Eclipse χρησιμοποιώντας το πρόσθετο εργαλείων ανάπτυξης Android (ADT). τον Δεκέμβριο του 2014, η Google κυκλοφόρησε το Android Studio, βασισμένο στην IntelliJ IDEA, ως πρωταρχικό IDE για ανάπτυξη εφαρμογών Android. Άλλα εργαλεία ανάπτυξης είναι διαθέσιμα, συμπεριλαμβανομένου ενός κιτ φυσικής ανάπτυξης (NDK) για εφαρμογές ή επεκτάσεις σε C ή C ++, Google App Inventor, ένα οπτικό περιβάλλον για αρχάριους προγραμματιστές και διάφορες πλατφόρμες εφαρμογών διαδικτυακών εφαρμογών. Τον Ιανουάριο του 2014, η Google παρουσίασε ένα πλαίσιο βασισμένο στο Apache Cordova για τη μεταφορά εφαρμογών ιστού HTML 5 Chrome στο Android, τυλιγμένο σε ένα κέλυφος εφαρμογής.

Το Android έχει μια αυξανόμενη ποικιλία εφαρμογών τρίτου μέρους που μπορούν να αποκτηθούν από χρήστες, μεταφορτώντας και εγκαθιστώντας το αρχείο APK της εφαρμογής (πακέτο εφαρμογών Android) ή μεταφορτώνοντάς τα χρησιμοποιώντας ένα πρόγραμμα καταστήματος εφαρμογών που επιτρέπει στους χρήστες να εγκαταστήσουν, να ενημερώσουν και να καταργήσουν εφαρμογές από τις συσκευές τους. Το Google Play Store είναι το κύριο κατάστημα εφαρμογών που είναι εγκατεστημένο σε συσκευές Android που συμμορφώνονται με τις απαιτήσεις συμβατότητας της Google και διαθέτουν άδεια χρήσης του λογισμικού

Google Mobile Services. Το Google Play Store επιτρέπει στους χρήστες να περιηγούνται, να πραγματοποιούν λήψη και να ενημερώνουν εφαρμογές που δημοσιεύονται από την Google και τρίτους προγραμματιστές. από τον Ιούλιο του 2013, υπάρχουν διαθέσιμα περισσότερα από ένα εκατομμύριο εφαρμογές για το Android στο Play Store. Από τον Ιούλιο του 2013 έχουν εγκατασταθεί 50 δισεκατομμύρια αιτήσεις. Ορισμένοι μεταφορείς προσφέρουν άμεση χρέωση για μεταφορείς εφαρμογών Google Play, όπου το κόστος της εφαρμογής προστίθεται στο μηνιαίο λογαριασμό του χρήστη. Από τον Μάιο του 2017, υπάρχουν περισσότεροι από ένα δισεκατομμύριο ενεργοί χρήστες το μήνα για το Gmail, το Android, το Chrome, το Google Play και τους Χάρτες.

Λόγω της ανοικτής φύσης του Android, υπάρχουν και άλλες αγορές εφαρμογών τρίτων για το Android, είτε για να αντικαταστήσουν συσκευές που δεν επιτρέπεται να μεταφέρονται στο Google Play Store, να παρέχουν εφαρμογές που δεν μπορούν να προσφερθούν στο Google Play Store λόγω σε παραβιάσεις πολιτικής ή για άλλους λόγους. Παραδείγματα αυτών των καταστημάτων τρίτων έχουν συμπεριλάβει τα Amazon Appstore, GetJar και SlideMe. Το F-Droid, μια άλλη εναλλακτική αγορά, επιδιώκει να παρέχει μόνο εφαρμογές που διανέμονται υπό ελεύθερες και ανοιχτές πηγές.

2.5 Οι απαιτήσεις υλικού

Η κύρια πλατφόρμα υλικού για το Android είναι η αρχιτεκτονική ARM, ενώ επίσημα υποστηρίζονται επίσης οι x86 και MIPS αρχιτεκτονικές. Και οι δύο 64-bit και 32-bit εκδόσεις των τριών αρχιτεκτονικών που υποστηρίζονται από την απελευθέρωση του Android 5.0, ανεπίσημο έργο του Android-x86, που είχε παράξει υποστήριξη για την x86 και MIPS αρχιτεκτονικές πριν από την επίσημη υποστήριξη. Από το 2012, οι συσκευές Android με επεξεργαστές Intel άρχισαν να εμφανίζονται, συμπεριλαμβανομένων των τηλεφώνων και των ταμπλετών. Αν κερδίσουμε την υποστήριξη για πλατφόρμες 64-bit, το Android έγινε για πρώτη φορά να τρέχει σε 64-bit x86 και, στη συνέχεια, στο ARM64.

Οι ελάχιστες απαιτήσεις υλικού έχουν αναβαθμιστεί σε βήματα με την πάροδο του χρόνου, με τις νέες κυκλοφορίες εκδόσεων του Android. Αρχικά το ελάχιστο ήταν 32 MB μνήμης RAM αλλά λιγότερο από 128 MB δεν συνιστάται, με το πρώτο κινητό τηλέφωνο HTC Dream ("ναυαρχίδα") που χρησιμοποιούσε 192

MB RAM με 32 MB μνήμης flash, και αρχιτεκτονική ARM με επεξεργαστή 200 MHz (ARMv5). Από το Νοέμβριο του 2013 και με την νέα έκδοση Android 4.4, που είναι για συσκευές ARM-based που απαιτούν ARMv7 επεξεργαστή (το Android 5.0 υποστηρίζει επίσης ARMv8-A), ενώ η ελάχιστη ποσότητα της μνήμης RAM είναι 512 MB. Η απαιτούμενη ελάχιστη μνήμη που είναι διαθέσιμη για το Android 4.4 είναι 340 MB, και όλες οι συσκευές με λιγότερα από 512 MB μνήμης RAM, πρέπει να αναφέρονται ως "χαμηλής μνήμης RAM" συσκευές.

Τον Οκτώβριο του 2011, με το Android 4.0, η μονάδα επεξεργασίας γραφικών (GPU) υποστηρίζει πλέον το OpenGL ES 2.0 (και το ES 1.0), και η επιτάχυνση υλικού έγινε υποχρεωτική, ανεξάρτητα από το εάν οι εφαρμογές χρησιμοποιούν άμεσα το OpenGL ES ή όχι. Αργότερα, το Android 4.3 πρόσθεσε υποστήριξη για το OpenGL ES 3.0 αν χρησιμοποιείται, με παράλληλη την υποστήριξη και για τις δύο παλαιότερες εκδόσεις (ES 2.0 και 1.0) που εξακολουθεί να είναι υποχρεωτική.

Εκτός από την εκτέλεση απευθείας σε x86 με βάση το υλικό, το Android μπορεί επίσης να τρέχει σε x86 αρχιτεκτονική με τη χρήση ενός εξομοιωτή Android το οποίο είναι μέρος του Android SDK, ή με τη χρήση BlueStacks ή του Andy.

2.6 Αρχιτεκτονική του Android

Όπως μπορείτε να δείτε στο σχήμα , στον πυρήνα της πλατφόρμας Android βρίσκεται ένα Linux kernel το οποίο είναι υπεύθυνο για τη διαχείριση των device drivers, τον έλεγχο πρόσβασης στους πόρους του συστήματος, τη διαχείριση μνήμης και τις λοιπές υπηρεσίες που παρέχει ένα λειτουργικό σύστημα. Στους device drivers συγκαταλέγονται αυτοί της οθόνης, του WiFi, της κάμερας, του ήχου κ.α. Ένα επίπεδο επάνω βρίσκονται οι native βιβλιοθήκες του συστήματος που είναι γραμμένες σε C++ και περιλαμβάνουν το OpenGL, την SQLite, την Media library κ.α. Οι εφαρμογές που τρέχουν στο κινητό μπορούν να έχουν πρόσβαση στις βιβλιοθήκες αυτές μέσω της Dalvik JVM. Όπως έχει ήδη αναφερθεί, οι εφαρμογές Android είναι γραμμένες σε Java και άρα για να τρέξουν χρειάζονται το αντίστοιχο περιβάλλον. Όπως λοιπόν για να εκτελέσουμε μία εφαρμογή σε ένα PC είναι απαραίτητο να είναι εγκατεστημένο το κατάλληλο JRE

(Java Runtime Environment), για τις εφαρμογές Android τον ρόλο του JRE παίζει η Dalvik VM

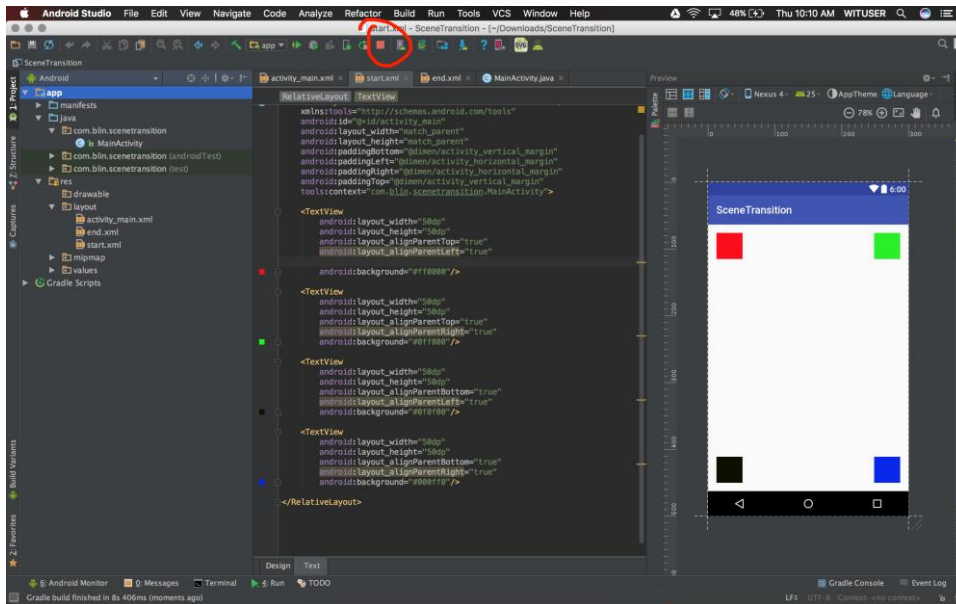
Δεδομένης της σαφώς πιο περιορισμένης επεξεργαστικής ισχύος καθώς και της ποσότητας διαθέσιμης μνήμης που έχουν οι κινητές συσκευές σε σχέση με τους υπολογιστές, η συγκεκριμένη VM είναι βελτιστοποιημένη να χρησιμοποιεί μικρότερα σε μέγεθος αρχεία ενδιάμεσου κώδικα, τα οποία σε αντίθεση με αυτά της Java SE έχουν κατάληξη `.dex` αντί για `.class`. Επίσης, σύμφωνα με την Google, διαθέτει και πιο καλά γραμμένο garbage collector. Αν και υπάρχει έκδοση της Java για κινητά τηλέφωνα, η Java ME, η Google έκρινε σωστό να χρησιμοποιήσει τη δική της υλοποίηση και έτσι γεννήθηκε η Dalvik. Κάθε εφαρμογή λοιπόν που γράφετε και εκτελείτε στο Android κινητό σας, χρησιμοποιεί τη Dalvik και τρέχει σε ξεχωριστό instance του VM. Στο αμέσως επόμενο επίπεδο βρίσκεται το Android SDK που περιέχει όλες τις απαραίτητες βιβλιοθήκες για τη συγγραφή εφαρμογών. Ο κώδικας που θα γράφετε για την εκτέλεση κάποιας λειτουργίας, π.χ. για την πραγματοποίηση μιας κλήσης, την αποστολή ενός SMS, τον εντοπισμό της τρέχουσας θέσης κλπ. στην ουσία θα καλεί κάποια από τις μεθόδους που παρέχονται από το συγκεκριμένο SDK. Έτσι λοιπόν, οι εφαρμογές βρίσκονται στην κορυφή του application stack που στο σχήμα 1.1 ονομάζεται Application Layer.

2.7 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη

σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.



Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android . Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

2.8 JAVA

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την Sun, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες (C++ ++, που μετέπειτα ονομάστηκε C#) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στην Sun. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα Oak. Το όνομά της το πήρε από

το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα.

Από την Oak στη Java

Η Oak ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με την C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστραφή (object oriented) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα Oak ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που εκτός των άλλων ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η επίσημη εμφάνιση της Java αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Ο πρώτος μεταγλωττιστής (compiler) της ήταν γραμμένος στη γλώσσα C από τον James Gosling. Το 1994, ο A.Van Hoff ξαναγράφει τον μεταγλωττιστή της γλώσσας σε Java, ενώ το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη Java για την δημιουργία λογισμικού. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά το μεταγλωττιστή (javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit).

Η εξαγορά από την Oracle και το μέλλον της Java

Στις 27 Απριλίου 2010 η εταιρία λογισμικού Oracle Corporation ανακοίνωσε ότι μετά από πολύμηνες συζητήσεις ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών (πνευματικά δικαιώματα/ πατέντες) που η δεύτερη είχε στην κατοχή της ή δημιουργήσει. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της Java και του γενικότερου οικοσυστήματος τεχνολογιών γύρω από αυτή μιας και ο έμμεσος έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια.

Εκδόσεις

Σήμερα οι υποστηριζόμενες εκδόσεις είναι οι Java 8 και 9. Οι σημαντικότερες που έχουν κυκλοφορήσει, με τις αντίστοιχες ημερομηνίες κυκλοφορίας, είναι οι εξής :

- JDK 1.0 (23 Ιανουαρίου 1996)[1]
- JDK 1.1 (19 Φεβρουαρίου 1997)
- J2SE 1.2 (8 Δεκεμβρίου 1998)
- J2SE 1.3 (8 Μαΐου 2000)
- J2SE 1.4 (6 Φεβρουαρίου 2002)
- J2SE 5.0 (30 Σεπτεμβρίου 2004)
- Java SE 6 (11 Δεκεμβρίου 2006)
- Java SE 7 (28 Ιουλίου 2011)
- Java SE 8 (18 Μαρτίου 2014)
- Java SE 9 (21 Σεπτεμβρίου 2017)

Τα χαρακτηριστικά της Java

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

Η εικονική μηχανή της Java

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή `javac`, ο οποίος παράγει έναν αριθμό από αρχεία `.class` (κώδικας `byte` ή `bytecode`). Ο κώδικας `byte` είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία `.class`. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί. Αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine).[3] Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα `bytecode` απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή `native code`) με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

Ο συλλέκτης απορριμμάτων (Garbage Collector)

Ακόμα μία ιδέα που βρίσκεται πίσω από τη Java είναι η ύπαρξη του συλλέκτη απορριμμάτων (Garbage Collector). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι

αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ όπου αποθηκεύονται κυρίως στη στοίβα) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι κοινά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

Η προεπιλεγμένη μέθοδος για συλλογή απορριμμάτων μοιράζει το σωρό σε δύο κομμάτια. Το πρώτο, χρησιμοποιείται για νέα αντικείμενα και λέγεται μαιευτήριο. Αφού επιβιώσουν ένα προεπιλεγμένο αριθμό εκκαθαρίσεων στο μαιευτήριο, ώριμα αντικείμενα μεταφέρονται στο δεύτερο κομμάτι του σωρού. Το μαιευτήριο είναι μοιρασμένο σε δύο κομμάτια, τον από-χώρο και τον προς-χώρο. Νέα αντικείμενα τοποθετούνται αρχικά στον από-χώρο. Όταν αυτός γεμίσει, ζωντανά αντικείμενα αντιγράφονται στον προς-χώρο και οι δύο χώροι ανταλλάσσουν ρόλο.

Επιδόσεις

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η Java αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η C++. Εμπειρικές μετρήσεις στο παρελθόν είχαν δείξει ότι η C++ μπορούσε να είναι αρκετές φορές γρηγορότερη από την Java. Ωστόσο γίνονται προσπάθειες από τη Oracle για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της IBM), οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη C++ έχει μικρύνει κατά πολύ.

Οι τελευταίες εκδόσεις του javac με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα.

Ως αντικειμενοστραφής γλώσσα, η Java βασίζεται σε χρήση αντικειμένων. Τα αντικείμενα είναι συλλογές πεδίων πληροφορίας και μεθόδων επεξεργασίας και προβολής πληροφορίας. Τα διάφορα αντικείμενα ανήκουν σε κλάσεις, οι οποίες δηλώνουν τον τύπο ομοειδών αντικειμένων.

Κάθε μέλος μίας κλάσης, είτε πεδίο είτε μέθοδος, προσδιορίζεται από έναν μετατροπέα ορατότητας. Υπάρχουν τέσσερις μετατροπείς ορατότητας: `private`, `package-private` (με κενό μετατροπέα), `protected` και `public`. Τα `private` μέλη είναι ορατά μόνο από την ίδια κλάση, τα `package-private` είναι ορατά από κλάσεις του ίδιου πακέτου, τα `protected` από κλάσεις του ίδιου πακέτου και από κλάσεις εκτός πακέτου που επεκτείνουν (`extends`) αυτήν την τάξη. Τα `public` μέλη είναι ορατά από όλες τις κλάσεις της εφαρμογής.

Εργαλεία ανάπτυξης

Όλα τα εργαλεία που χρειάζεται κάποιος για να γράψει Java προγράμματα έρχονται δωρεάν, από το περιβάλλον ανάπτυξης μέχρι εργαλεία `build` όπως το Apache Ant και βιβλιοθήκες, ενώ υπάρχουν πολλές διαφορετικές υλοποιήσεις της Εικονικής Μηχανής και του μεταγλωττιστή (πχ the GNU Compiler for Java) της Java.

Πολλά εργαλεία και τεχνολογίες σε Java μπορούν να βρεθούν στο Apache Software Foundation αλλά και στο Jakarta Project.

Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)

Για να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο `vi` (γνωστός στο χώρο του Unix). Παρ'όλ'αυτά, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (`debugging`). Υπάρχουν αρκετά διαθέσιμα, ενώ πολλά από αυτά έρχονται δωρεάν.

2.9 XML

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

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

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

Η XML (αγγλ. αρκτ. από το eXtensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο.[Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προελαύνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork.

Χαρακτήρας Unicode

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Επεξεργαστής και Εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

Σήμανση και Περιεχόμενο



Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";". Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

Ετικέτα

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας:

- ετικέτες-αρχής, για παράδειγμα: `<section>`,
- ετικέτες-τέλους, για παράδειγμα: `</section>` και
- ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα: `<line-break/>`.

Στοιχείο

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά.

Ένα παράδειγμα ενός στοιχείου είναι το:

```
<Greeting>Hello, world</Greeting>
```

Ένα άλλο είναι το:

```
<line-break/>
```

Χαρακτηριστικό

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο.

Στο παράδειγμα παρακάτω, το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`:

```

```

Ένα άλλο παράδειγμα θα ήταν το

```
<step number="3">Connect A to B.</step>
```

όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

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

Παράδειγμα:

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που χρησιμοποιεί όλες τις παραπάνω έννοιες και στοιχεία.

```
<?xml version="1.0" encoding='UTF-8'?><painting>  
  
<caption>This is Raphael's "Foligno" Madonna, painted in <date>1511</date>-  
<date>1512</date>.</caption></painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα painting, img, caption, και δύο date. Τα στοιχεία date, είναι παιδιά του στοιχείου caption, το οποίο είναι παιδί του στοιχείου-ρίζας painting. Το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt.

Χαρακτήρες και διαφυγή

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο. Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

2.10 Ανατομία μιας Android εφαρμογής

Υπάρχουν τέσσερα διαφορετικά είδη building blocks μιας Android εφαρμογής. Κάθε ένα είδος έχει σαν στόχο να εξυπηρετήσει ένα συγκεκριμένο σκοπό και έχει σαφώς ορισμένο κύκλο ζωής που ορίζει πότε το συστατικό θα δημιουργηθεί και θα καταστραφεί. Τα τέσσερα αυτά building blocks είναι τα: Activities, Services, Content Providers, Broadcast receivers και αναλύονται στην συνέχεια.

Συστατικά εφαρμογής

i) **Activities**

Μια activity αντιπροσωπεύει μια οθόνη με ένα user interface. Για παράδειγμα, μια εφαρμογή για email μπορεί να έχει μια activity που εμφανίζει τη λίστα με τα νέα emails, μια άλλη για τη σύνθεση νέου email και μια διαφορετική για την ανάγνωση των emails. Παρόλο που οι activities συνεργάζονται για να δημιουργήσουν μια συνεκτική εμπειρία για τον χρήστη, κάθε μια είναι ανεξάρτητη από τις υπόλοιπες. Έτσι, μια διαφορετική εφαρμογή μπορεί να θέσει σε λειτουργία οποιαδήποτε από αυτές τις activities (αν η εφαρμογή για email το επιτρέπει). Για παράδειγμα, η εφαρμογή της κάμερας έχει τη δυνατότητα να ενεργοποιήσει την activity σύνθεσης email, έτσι ώστε ο χρήστης να μπορέσει να την διαμοιραστεί.

ii) **Services**

Ένα service είναι ένα συστατικό που τρέχει στο background για να εκτελέσει διαδικασίες μεγάλης χρονικής διάρκειας, ή για να εκτελέσει εργασίες απομακρυσμένων διεργασιών. Ένα service δεν παρέχει user interface. Για παράδειγμα, ένα service μπορεί να παίζει μουσική στο background ενώ ο χρήστης έχει ανοιχτή μια άλλη εφαρμογή. Ένα άλλο συστατικό, όπως μια activity, μπορεί να θέσει σε λειτουργία ένα service και να την αφήσει να τρέχει ή να την δεσμεύσει ώστε να επικοινωνεί μαζί της.

iii) **Content providers**

Ένας διαμοιραστής περιεχομένου (content provider) διαχειρίζεται ένα σύνολο από διαμοιραζόμενα δεδομένα της εφαρμογής. Έτσι γίνεται δυνατή η αποθήκευση των δεδομένων αυτών είτε στο σύστημα αρχείων (δηλαδή στην μνήμη της εκάστοτε εφαρμογής), είτε σε μια τοπική βάση δεδομένων (SQLite), είτε σε απομακρυσμένη βάση δεδομένων, είτε σε οποιοδήποτε άλλο χώρο αποθήκευσης έχει πρόσβαση η εφαρμογή. Μέσω του content provider, μπορεί ακόμη εφόσον ζητηθεί να γίνει ακόμα και τροποποίηση των δεδομένων, όπως για παράδειγμα η δυνατότητα διαχείρισης των πληροφοριών των επαφών του εκάστοτε χρήστη. Εφόσον δηλαδή δοθεί η κατάλληλη άδεια μπορεί κάποιο κομμάτι του content provider να αλλάξει το περιεχόμενο κάποιας επαφής. Οι content providers είναι επίσης χρήσιμοι για την ανάγνωση και εγγραφή δεδομένων που είναι ιδιωτικά για την εφαρμογή μας και όχι διαμοιραζόμενα. Για 62 παράδειγμα μια εφαρμογή Note Pad χρησιμοποιεί τον content provider για να αποθηκεύσει σημειώσεις.

iv) **Broadcast receivers**

Ένας broadcast receiver είναι ένα συστατικό που απαντά σε system-wide broadcast ανακοινώσεις. Πολλά broadcasts προέρχονται από το σύστημα, για παράδειγμα μια broadcast ανακοίνωση ότι η οθόνη έκλεισε, η μπαταρία είναι χαμηλή, ή ότι τραβήχτηκε μια φωτογραφία. Τα broadcasts μπορούν ακόμα να αρχικοποιηθούν από εφαρμογές, για παράδειγμα το να επιτρέπεται άλλες εφαρμογές να ξέρουν ότι κάποια δεδομένα έχουν φορτωθεί στην συσκευή και είναι διαθέσιμα για χρήση. Παρόλο που οι broadcast receivers δεν έχουν user interface, μπορούν να δημιουργήσουν μια ειδοποίηση στο status bar, για να ενημερώσουν το χρήστη όταν ένα broadcast συμβεί.

Ενεργοποίηση Συστατικών

Τρία από τα τέσσερα building blocks - activities, services, και broadcast receivers - ενεργοποιούνται μέσω ενός ασύγχρονου μηνύματος γνωστό και ως intent. Κατά τη διάρκεια εκτέλεσης μιας εφαρμογής ένα intent επιτρέπει τη δέσμευση μεμονωμένων συστατικών αυτής και τη μεταφορά τους σε μια άλλη, είτε προέρχονται από την ίδια είτε από κάποια άλλη. Για τις activities και τις services,

ένα intent ορίζει την μια ενέργεια που πρέπει να εκτελεστεί (για παράδειγμα να "δεις" ή να "στείλεις" κάτι) και μπορεί να διευκρινίζει το URI των δεδομένων πάνω στο οποίο θα γίνει η ενέργεια. Για παράδειγμα, ένα intent μπορεί να μεταβιβάσει μια αίτηση σε μια activity για να δείξει μια εικόνα ή να ανοίξει μια ιστοσελίδα. Σε μερικές περιπτώσεις υπάρχει η δυνατότητα να τεθεί σε λειτουργία μια activity για να λάβουμε ένα αποτέλεσμα το οποίο στη συγκεκριμένη περίπτωση μπορεί να επιστραφεί σε ένα intent από την activity. Για τους broadcast receivers, το intent ορίζει την ανακοίνωση που πρόκειται να γίνει broadcast.

To αρχείο Manifest

Όταν μια εφαρμογή εκκινείτε ,το πρώτο πράγμα που κάνει το λειτουργικό σύστημα είναι να επιβεβαιώσει ότι τα συστατικά της εφαρμογής είναι δηλωμένα στο αρχείο 63 AndroidManifest.xml αυτής, το οποίο πρέπει να βρίσκεται στο αρχικό φάκελο (root) της εφαρμογής.

Το αρχείο manifest προκειμένου να ορίσει τα συστατικά των εφαρμογών κάνει τα εξής:

- Προσδιορίζει κάθε άδεια του χρήστη που απαιτεί η εφαρμογή, όπως πρόσβαση στο Internet ή αναγνωσιμότητα των επαφών του χρήστη.

- Ορίζει το μικρότερο API Level που πρέπει να έχει το λειτουργικό που τρέχει το κινητό για να λειτουργήσει την εφαρμογή, και βασίζεται σε ποια APIs αυτή χρησιμοποιεί.

- Ορίζει ποιους εξωτερικούς αισθητήρες και ποιες επιπλέον άδειες θα χρησιμοποιηθούν ή θα ζητηθούν από την εφαρμογή, όπως κάμερα, bluetooth services, ή multitouch οθόνη.

- Δηλώνει τις API βιβλιοθήκες με τις οποίες η εφαρμογή χρειάζεται να συνδεθεί, για παράδειγμα η Google Maps βιβλιοθήκη.

Resources

Μια Android εφαρμογή δεν είναι μόνο κώδικας αλλά και άλλους πόρους σχετικούς με την εμφάνιση της εφαρμογής όπως εικόνες, αρχεία ήχου κ.λπ. Για παράδειγμα, κάποια από τα πιο βασικά πράγματα που πρέπει να επεξεργαστεί ο προγραμματιστής είναι τα διάφορα χρώματα, styles, animations, menus και την εμφάνιση της activity με XML αρχεία. Χρησιμοποιώντας application resources είναι εύκολο να αναβαθμιστούν πολλά χαρακτηριστικά μιας εφαρμογής χωρίς να τροποποιηθεί ο κώδικας. Όλα τα resources γίνονται compile μέσα στο application και αποθηκεύονται στον φάκελο res μέσα στο project. Οι εικόνες, τα strings, τα xml και άλλα αποθηκεύονται σε ξεχωριστούς φακέλους. Στόχος του resource compiler είναι να συμπιέζει και πακετάρει τους πόρους και μετά να δημιουργεί μια κλάση R που περιέχει τους identifiers που χρησιμοποιούνται στο πρόγραμμα για να μπορούν τα resources να είναι διαθέσιμα οποιαδήποτε στιγμή. Έτσι το Android με αυτό τον τρόπο εξοικονομεί μνήμη στο δίσκο.

Κύκλος ζωής Activity

Οι activities στο σύστημα, διαχειρίζονται υπό τη μορφή στοίβας. Όταν μια activity ξεκινήσει, τότε τοποθετείται στην κορυφή της στοίβας και γίνεται η τρέχουσα activity και η προηγούμενη activity παραμένει στην από κάτω θέση και έρχεται στο προσκήνιο μόνο αν η καινούρια activity τελειώσει.

Μια activity μπορεί ουσιαστικά να βρεθεί σε τέσσερις καταστάσεις:

- Αν μια διεργασία βρίσκεται στο προσκήνιο, εμφανίζεται δηλαδή στην οθόνη (είναι στην κορυφή της στοίβας), τότε βρίσκεται στην κατάσταση active ή running.
- Αν μια activity βρίσκεται στο background της εφαρμογής αλλά είναι ακόμα ορατή (δηλαδή μια νέα activity που δεν είναι full-sized είναι focused πάνω από την πρώτη activity), τότε βρίσκεται στην κατάσταση paused. Σε αυτήν την κατάσταση η activity είναι ζωντανή, αλλά μπορεί το σύστημα να την κλείσει σε περίπτωση χαμηλής μνήμης.

- Αν μια activity είναι πλήρως κρυμμένη από άλλη activity, τότε βρίσκεται στην κατάσταση stopped. Διατηρεί όλη την κατάσταση και τις πληροφορίες, ωστόσο δεν είναι πλέον ορατή στον χρήστη, έτσι το παράθυρό της είναι κρυμμένο και συνήθως κλείνει από το σύστημα όταν η υπάρχει ανάγκη για μνήμη.

- Αν κάποια διεργασία είναι στην κατάσταση paused ή stopped, το σύστημα μπορεί να την απομακρύνει από την μνήμη, είτε ζητώντας άδεια για να το κάνει είτε από μόνο του. Όταν ξαναεμφανιστεί στο χρήστη, πρέπει να εκκινήσει πάλι και να αποκαταστήσει την προηγούμενη της κατάσταση. Τα τετραγωνισμένα παραλληλόγραμμα δείχνουν τις callback μεθόδους που μπορούν να εφαρμοστούν για τις διάφορες διεργασίες όταν μια από αυτές θέλει να κινηθεί μεταξύ διαφόρων καταστάσεων. Τα χρωματιστά στρογγυλεμένα παραλληλόγραμμα δείχνουν τις κύριες καταστάσεις στις οποίες μπορεί να βρεθεί μια Activity.

Υπάρχουν τρία loops που μπορείτε να διαχειριστείτε μέσα σε μια activity:

- Το **loop entire lifetime** μιας activity διενεργείται μεταξύ της πρώτης κλήσης onCreate(Bundle) μέχρι μιας τελικής κλήσης onDestroy(). Μια activity θα κάνει όλες τις 65 αρχικοποιήσεις στην κλήση onCreate(), και θα απελευθερώσει όλα τα resources στην onDestroy().

- Το **loop visible lifetime** μιας διεργασίας ορίζεται μεταξύ μιας κλήσης onStart() και της αντίστοιχης κλήσης onStop(). Κατά την διάρκεια αυτού του κύκλου δίνεται η δυνατότητα στο χρήστη να δει την activity στην οθόνη, παρόλο που μπορεί να μην είναι στο προσκήνιο και να αλληλοεπιδρά με τον χρήστη. Μεταξύ των δυο αυτών μεθόδων μπορούν να διατηρούνται τα resources που είναι απαραίτητα προκειμένου να μπορεί ο χρήστης να δει την activity. Αυτές οι δυο μέθοδοι μπορούν αν καλούνται πολλές φορές, καθώς μια activity εμφανίζεται ή είναι κρυμμένη στο χρήστη.

- Το **foreground lifetime loop** μιας activity διενεργείται μεταξύ μιας κλήσης onResume() και της αντίστοιχης κλήσης onPause(). Κατά τη διάρκεια αυτή η activity βρίσκεται μπροστά από όλες τις υπόλοιπες activities και αλληλοεπιδρά με τον χρήστη. Μια activity μπορεί να συχνά να μεταβαίνει μεταξύ των δυο αυτών

μεθόδων, όταν για παράδειγμα η συσκευή τίθεται σε sleep mode ή όταν φτάνει ένα αποτέλεσμα μιας activity μέσω ενός intent. Για αυτό το λόγο ο κώδικας σε αυτές τις μεθόδους πρέπει να είναι αρκετά “ελαφρύς”.

ΑΝΑΓΝΩΡΙΣΗ ΦΩΝΗΣ

Αναγνώριση ομιλίας

Στην επιστήμη της πληροφορικής, η αναγνώριση ομιλίας αναφέρεται στην μετάφραση προφερόμενων λέξεων σε κείμενο. Είναι επίσης αλλιώς γνωστή και ως "αυτόματη αναγνώριση ομιλίας", "υπολογιστική αναγνώριση ομιλίας" ή "speech-to-text" (STT), δηλαδή (μετατροπή) "από-ομιλία-σε-κείμενο".

Κάποια συστήματα αναγνώρισης ομιλίας χρησιμοποιούν "αναγνώριση ομιλίας ανεξάρτητη από τον ομιλητή", ενώ άλλα χρησιμοποιούν "εξάσκηση", όπου ένα άτομο διαβάζει κομμάτια κειμένου σε ένα σύστημα αναγνώρισης ομιλίας. Τότε, αυτού του είδους τα συστήματα αναλύουν τη φωνή ενός ομιλητή και την χρησιμοποιούν για να προσαρμόσουν την αναγνώριση της ομιλίας του συγκεκριμένου ατόμου από τον υπολογιστή, με αποτέλεσμα την πιο ακριβή καταγραφή της. Συστήματα που δεν χρησιμοποιούν εξάσκηση ονομάζονται συστήματα "ανεξάρτητα από τον ομιλητή".

Οι εφαρμογές αναγνώρισης ομιλίας συμπεριλαμβάνουν τα Φωνητικά Περιβάλλοντα Χρήστη (Voice User Interfaces) όπως η φωνητική πληκτρολόγηση (π.χ. "Κάλεσε σπίτι"), ο έλεγχος των οικιακών ηλεκτρονικών συσκευών και συστημάτων (π.χ. "Άναψε φως"), η διαδικτυακή αναζήτηση, η απλή τοποθέτηση δεδομένων, η προετοιμασία δομημένων εγγράφων (π.χ. μία αναφορά ραδιολογίας), η επεξεργασία "από-ομιλία-σε-κείμενο" και τα αεροσκάφη.

Ο όρος αναγνώριση ομιλίας δεν αναφέρεται τόσο στο "ποιος" μιλάει, αλλά στο "τι" λέει. Αναγνωρίζοντας επιπλέον όμως και την ταυτότητα του ομιλητή είναι δυνατόν να διευκολυνθεί η διαδικασία της μετάφρασης της ομιλίας του σε συστήματα που έχουν προηγουμένως εξασκηθεί στην φωνή του συγκεκριμένου ατόμου ή ακόμη είναι δυνατόν να πιστοποιηθεί ή αναγνωριστεί η ταυτότητα του ομιλητή σε συστήματα ασφαλείας.

GOOGLE API

Είναι ένα σύνολο εφαρμογών (API) που αναπτύσσονται από την Google. Οι εφαρμογές αυτές επιτρέπουν την επικοινωνία με τις Υπηρεσίες Google και την ενσωμάτωσή τους σε άλλες υπηρεσίες. Παραδείγματα αυτών είναι η Αναζήτηση, το Gmail, η Μετάφραση και οι Χάρτες Google οι οποίες μπορούν να χρησιμοποιηθούν από προγραμματιστές για να αξιοποιήσουν ή να επεκτείνουν τη λειτουργικότητα των εφαρμογών τους.

Στην συγκεκριμένη εφαρμογή έχουμε χρησιμοποιήσει την εφαρμογή “Google Cloud Speech-to-Text” η οποία επιτρέπει στους προγραμματιστές να μετατρέπουν τον ήχο σε κείμενο εφαρμόζοντας ισχυρά μοντέλα νευρωνικών δικτύων. Το API αναγνωρίζει 120 γλώσσες και παραλλαγές για να υποστηρίξει την παγκόσμια βάση χρηστών. Ο χρήστης μπορεί να ενεργοποιήσει τη φωνητική εντολή και τον έλεγχο, να μεταγράψει ήχο από τα τηλεφωνικά κέντρα και πολλά άλλα. Μπορεί να επεξεργαστεί ροή σε πραγματικό χρόνο ή προεγγραμμένο ήχο, χρησιμοποιώντας την τεχνολογία εκμάθησης μηχανών της Google.

Παρακάτω θα βρείτε τον κώδικα που χρησιμοποιήσαμε στην εφαρμογή μας. Ηχογραφεί την απάντηση του χρήστη, ελέγχει αν είναι η σωστή, εμφανίζει το αντίστοιχο μήνυμα και ήχο επιτυχίας ή αποτυχίας και τέλος εμφανίζει στην οθόνη τί ακριβώς είπε ο χρήστης. Την απάντηση που έδωσε δηλαδή.

Ξεκινώντας από το αρχείο java.

```

1 package com.example.katerina.degree_paper;
2
3 import ...
4
5 public class Animals extends AppCompatActivity {
6
7     MediaPlayer mySong, success, nosuccess;
8     //ViewFlipper viewFlipper;
9     //Button next;
10    //Button previous;
11
12    private static final int REQ_CODE_SPEECH_INPUT = 100;
13    private TextView mVoiceInputTv;
14    private ImageButton mSpeakBtn;
15
16    @Override
17    protected void onCreate(Bundle savedInstanceState) {
18        super.onCreate(savedInstanceState);
19        setContentView(R.layout.activity_animals);
20
21        mySong = MediaPlayer.create( context: com.example.katerina.degree_paper.Animals.this, R.raw.ciger);
22        success = MediaPlayer.create( context: Animals.this, R.raw.success);
23        nosuccess = MediaPlayer.create( context: Animals.this, R.raw.nosuccess);
24
25        mVoiceInputTv = (TextView) findViewById(R.id.voiceInput);
26        mSpeakBtn = (ImageButton) findViewById(R.id.btnSpeak);
27        mSpeakBtn.setOnClickListener( () -> startVoiceInput() );
28    }
29
30    onActivityResult()
    
```


Αρχείο Xml

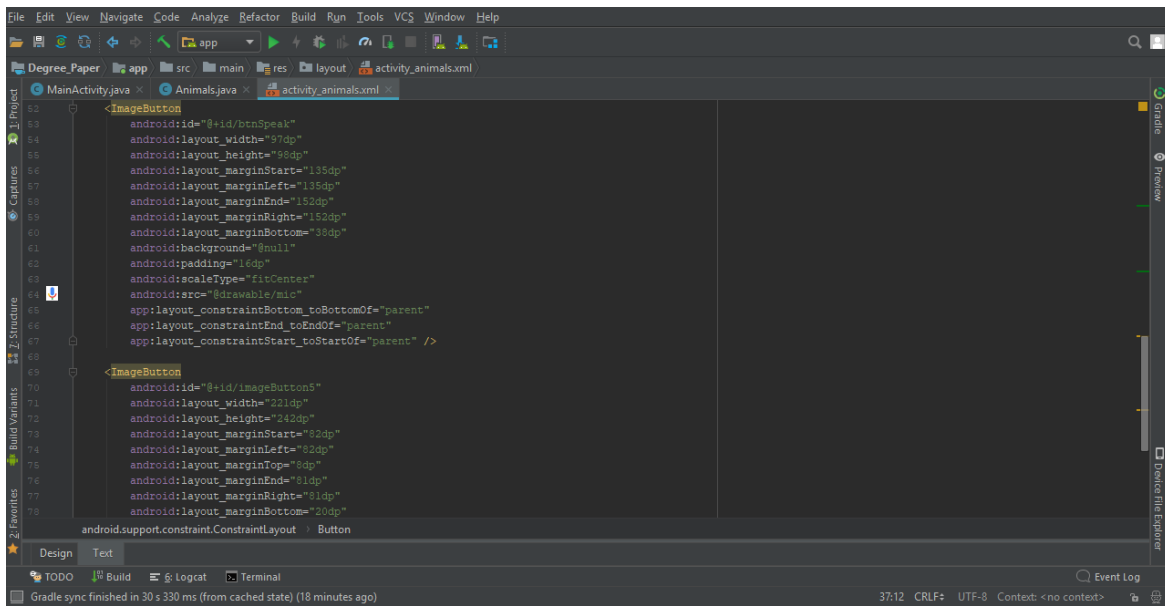
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".Animals">
8
9     <TextView
10        android:id="@+id/voiceInput"
11        android:layout_width="96dp"
12        android:layout_height="wrap_content"
13        android:layout_marginStart="144dp"
14        android:layout_marginLeft="144dp"
15        android:layout_marginTop="16dp"
16        android:layout_marginEnd="144dp"
17        android:layout_marginRight="144dp"
18        app:layout_constraintEnd_toEndOf="parent"
19        app:layout_constraintStart_toStartOf="parent"
20        app:layout_constraintTop_toTopOf="parent" />
21
22     <Button
23        android:id="@+id/button6"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:layout_marginStart="80dp"
27        android:layout_marginLeft="80dp"
28
29        android.support.constraint.ConstraintLayout > Button
    
```

```

28        android:layout_height="wrap_content"
29        android:layout_marginStart="80dp"
30        android:layout_marginLeft="80dp"
31        android:layout_marginEnd="16dp"
32        android:layout_marginRight="16dp"
33        android:layout_marginBottom="158dp"
34        android:onClick="cat"
35        android:text=">"
36        app:layout_constraintBottom_toBottomOf="parent"
37        app:layout_constraintEnd_toEndOf="parent"
38        app:layout_constraintStart_toEndOf="@+id/button7" />
39
40     <Button
41        android:id="@+id/button7"
42        android:layout_width="wrap_content"
43        android:layout_height="wrap_content"
44        android:layout_marginStart="6dp"
45        android:layout_marginLeft="6dp"
46        android:layout_marginEnd="80dp"
47        android:layout_marginRight="80dp"
48        android:layout_marginBottom="158dp"
49        android:onClick="animals"
50        android:text="<"
51        app:layout_constraintBottom_toBottomOf="parent"
52        app:layout_constraintEnd_toStartOf="@+id/button6"
53        app:layout_constraintStart_toStartOf="parent" />
54
55        android.support.constraint.ConstraintLayout > Button
    
```

Σχεδίαση Εφαρμογής Android



```
52 <ImageButton
53     android:id="@+id/btnSpeak"
54     android:layout_width="97dp"
55     android:layout_height="96dp"
56     android:layout_marginStart="136dp"
57     android:layout_marginLeft="136dp"
58     android:layout_marginEnd="152dp"
59     android:layout_marginRight="152dp"
60     android:layout_marginBottom="38dp"
61     android:background="@null"
62     android:padding="16dp"
63     android:scaleType="fitCenter"
64     android:src="@drawable/mic"
65     app:layout_constraintBottom_toBottomOf="parent"
66     app:layout_constraintEnd_toEndOf="parent"
67     app:layout_constraintStart_toStartOf="parent" />
68
69 <ImageButton
70     android:id="@+id/imageButton5"
71     android:layout_width="221dp"
72     android:layout_height="242dp"
73     android:layout_marginStart="82dp"
74     android:layout_marginLeft="82dp"
75     android:layout_marginTop="8dp"
76     android:layout_marginEnd="81dp"
77     android:layout_marginRight="81dp"
78     android:layout_marginBottom="20dp"
79 />
```

Στο αρχείο xml θα προσθέσουμε ένα κουμπί το οποίο όταν πατηθεί από τον χρήστη θα καλέσει την μέθοδο `startVoiceInput`. Με αυτήν τη μέθοδο δημιουργούμε το `RecognizerIntent`.

2ο ΜΕΡΟΣ - ΑΝΑΛΥΣΗ ΠΕΡΙΠΤΩΣΗΣ

ΚΕΦΑΛΑΙΟ 3 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

3.1 Οι ανάγκες που οδήγησαν στη δημιουργία της εφαρμογής

Αυτή η εφαρμογή δημιουργήθηκε διότι το Εθνικό Κέντρο Αποκατάστασης είχε ανάγκη από μια εφαρμογή που να μπορεί να συγκρίνει αυτό που εκφέρεται από τους ασθενείς με το επιθυμητό αποτέλεσμα και έτσι αυτό να κινητοποιεί τους ασθενείς να μην βαριούνται και να παρατάνε κατευθείαν την εξάσκηση αλλά και να έχουν μια πραγματική και αντικειμενική ένδειξη της προόδου τους.

3.2 Υλοποίηση

Για να υλοποιήσουμε αυτή την εφαρμογή χρησιμοποιήσαμε το Android Studio. Οι γλώσσες που χρησιμοποιήθηκαν είναι Java και XML. Μέσω online μαθημάτων και tutorials άρθρων από το support Page του Android και βοήθεια από τον καθηγητή μας και έμπειρους φίλους προγραμματιστές στο Android καταφέραμε να υλοποιηθεί αυτή η εφαρμογή ανταποκρινόμενοι πλήρως σε όλα τα κριτήρια που είχαν τεθεί ώστε να θεωρείται πετυχημένη. Τα βήματα της υλοποίησης αναφέρονται και περιγράφονται στη συνέχεια με αναλυτικά βήματα καθώς και screenshot ώστε να είναι πιο ξεκάθαρη η διαδικασία.

3.3 Εγκατάσταση Android Studio

Εγκαταστήσαμε το Android Studio από την επίσημη ιστοσελίδα <https://developer.android.com/studio/>

Μεταβήκαμε στην καρτέλα Design όπου υλοποιήθηκε η XML δηλαδή η δομή της εφαρμογής και στη συνέχεια υλοποιήσαμε όλη τη λογική την προγραμματιστική σε Java.

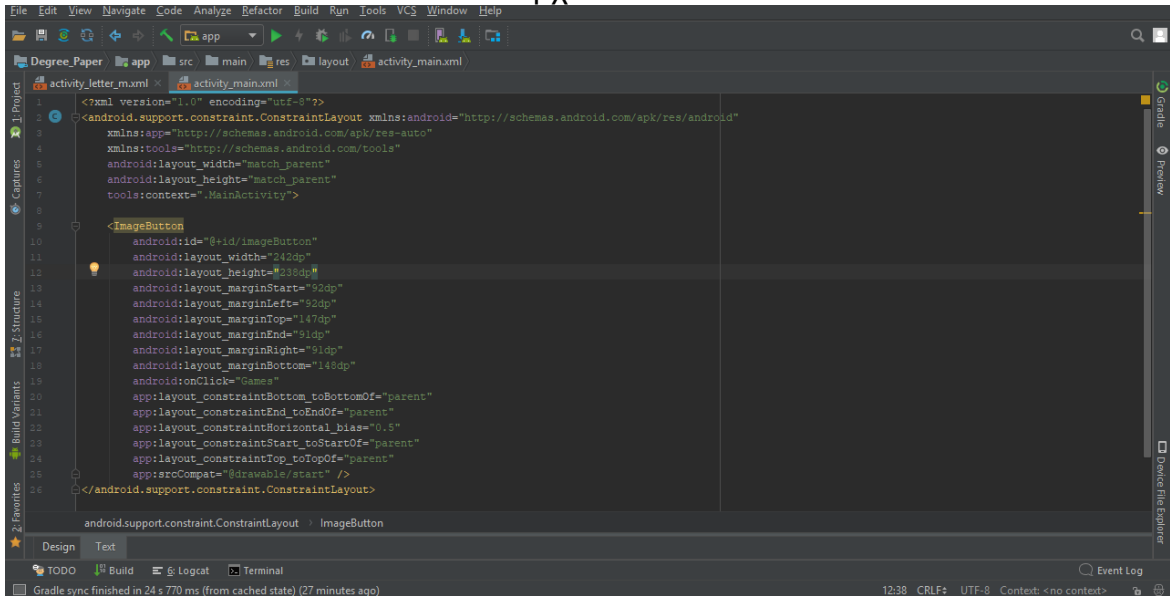
Πιο αναλυτικά κάθε activity αναπαριστά πχ ένα γράμμα ή εικόνα όπου συγκρίνεται το φωνητικό input του χρήστη με αυτό που είναι αποθηκευμένο στην εφαρμογή και αν υπάρχει ταύτιση εμφανίζεται μήνυμα σωστού και ακούγεται ήχος επιτυχίας. Ο χρήστης μπορεί να προηγηθεί σε διαφορετικά activities από τα κουμπιά home button, τα κουμπιά πλοήγησης ή από το μενού. Δηλαδή τα μενού περιέχουν αλλά activities και γενικά η λογική είναι ότι φορτώνεται η αρχική σελίδα

Σχεδίαση Εφαρμογής Android

και ανάλογα με το που θα κατευθυνθεί ο χρήστης, φορτώνεται το αντίστοιχο activity και ανάλογα με τα actions του χρήστη πχ click υλοποιούνται αντίστοιχες κλάσεις και events στην Java και φορτώνονται αλλά services ή activities.

Ενδεικτικά κομμάτια κώδικα:

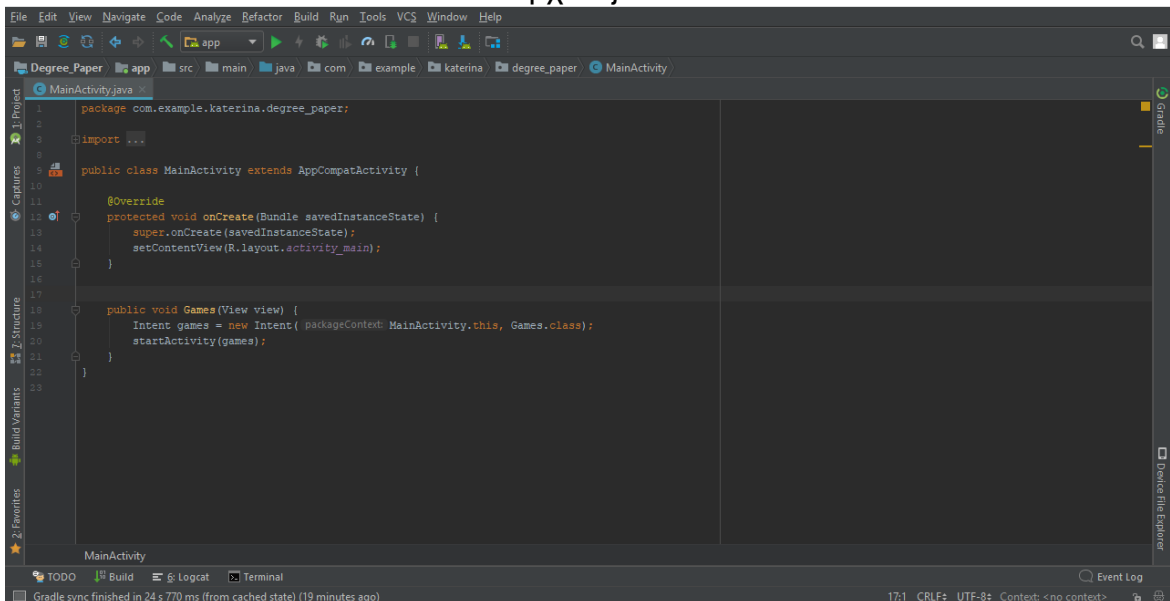
Main activity: Αρχείο xml



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageButton
        android:id="@+id/imageButton"
        android:layout_width="242dp"
        android:layout_height="238dp"
        android:layout_marginStart="92dp"
        android:layout_marginLeft="92dp"
        android:layout_marginTop="147dp"
        android:layout_marginEnd="91dp"
        android:layout_marginRight="91dp"
        android:layout_marginBottom="146dp"
        android:onClick="Games"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/start" />
</android.support.constraint.ConstraintLayout>
```

Στο αρχείο xml προσθέτουμε ένα imagebutton, ;έχουμε αποθηκεύσει στο φάκελο drawable την εικόνα που θέλουμε να χρησιμοποιήσουμε(start) και ορίζουμε , «καλούμε», την μέθοδο Games όταν κλικάρει ο χρήστης στην εικόνα.

Αρχείο java



```
package com.example.katerina.degree_paper;
import ...
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void Games(View view) {
        Intent games = new Intent(packageContext MainActivity.this, Games.class);
        startActivity(games);
    }
}
```

Στο αρχείο java ορίζουμε την μέθοδο Games, η οποία «ενεργοποιείται» μόλις ο χρήστης κλικάρει την εικόνα Start. Και ουσιαστικά εισέρχεται στην κεντρική εφαρμογή, όπου θα επιλέξει αν θέλει να μάθει(“Learn”) ή να παίξει (“Play”).

Activity animals Αρχείο xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".Animals">
8
9     <TextView
10        android:id="@+id/voiceInput"
11        android:layout_width="96dp"
12        android:layout_height="wrap_content"
13        android:layout_marginStart="144dp"
14        android:layout_marginLeft="144dp"
15        android:layout_marginTop="16dp"
16        android:layout_marginEnd="144dp"
17        android:layout_marginRight="144dp"
18        app:layout_constraintEnd_toEndOf="parent"
19        app:layout_constraintStart_toStartOf="parent"
20        app:layout_constraintTop_toTopOf="parent" />
21
22     <Button
23        android:id="@+id/button6"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:layout_marginStart="80dp"
27        android:layout_marginLeft="80dp"
28        android.support.constraint.ConstraintLayout /> Button
    
```

```

28        android:layout_marginEnd="16dp"
29        android:layout_marginRight="16dp"
30        android:layout_marginBottom="150dp"
31        android:onClick="cat"
32        android:text=">"
33        app:layout_constraintBottom_toBottomOf="parent"
34        app:layout_constraintEnd_toEndOf="parent"
35        app:layout_constraintStart_toEndOf="@+id/button7" />
36
37     <Button
38        android:id="@+id/button7"
39        android:layout_width="wrap_content"
40        android:layout_height="wrap_content"
41        android:layout_marginStart="6dp"
42        android:layout_marginLeft="6dp"
43        android:layout_marginEnd="80dp"
44        android:layout_marginRight="80dp"
45        android:layout_marginBottom="150dp"
46        android:onClick="animals"
47        android:text="<"
48        app:layout_constraintBottom_toBottomOf="parent"
49        app:layout_constraintEnd_toStartOf="@+id/button6"
50        app:layout_constraintStart_toStartOf="parent" />
51
52     <ImageButton
53        android:id="@+id/btnSpeak"
54        android:layout_width="97dp"
55        android.support.constraint.ConstraintLayout /> Button
    
```

```

55     android:layout_height="98dp"
56     android:layout_marginStart="135dp"
57     android:layout_marginLeft="135dp"
58     android:layout_marginEnd="152dp"
59     android:layout_marginRight="152dp"
60     android:layout_marginBottom="38dp"
61     android:background="@null"
62     android:padding="16dp"
63     android:scaleType="fitCenter"
64     android:src="@drawable/mic"
65     app:layout_constraintBottom_toBottomOf="parent"
66     app:layout_constraintEnd_toEndOf="parent"
67     app:layout_constraintStart_toStartOf="parent" />
68
69     <ImageButton
70         android:id="@+id/imageButton5"
71         android:layout_width="221dp"
72         android:layout_height="242dp"
73         android:layout_marginStart="82dp"
74         android:layout_marginLeft="82dp"
75         android:layout_marginTop="8dp"
76         android:layout_marginEnd="81dp"
77         android:layout_marginRight="81dp"
78         android:layout_marginBottom="20dp"
79         android:onClick="playIT"
80         android:padding="00dp"
81         android:scaleType="centerCrop"
82     />

```

```

82     app:layout_constraintBottom_toTopOf="@+id/button6"
83     app:layout_constraintEnd_toEndOf="parent"
84     app:layout_constraintStart_toStartOf="parent"
85     app:layout_constraintTop_toBottomOf="@+id/voiceInput"
86     app:srcCompat="@drawable/tiger2" />
87
88 </android.support.constraint.ConstraintLayout>

```

Σε αυτό το αρχείο δημιουργούμε δύο buttons και δύο imagebuttons.

Τα buttons είναι τα κουμπιά με τα οποία ο χρήστης πηγαίνει από τη μία εικόνα στην άλλη. Κάθε φορά ορίζουμε διαφορετική μέθοδο όταν κλικάρει ο χρήστης τα κουμπιά, έτσι ώστε να μην υπάρχει επανάληψη και ο χρήστης να μην ξαναδεί την ίδια εικόνα.

Το ένα imagebutton είναι η κεντρική εικόνα στην οποία καλούμε την μέθοδο playIT . Έτσι ο χρήστης μόλις κλικάρει την εικόνα θα ακούσει πως προφέρεται αυτό που απεικονίζεται στην εικόνα.

Το δεύτερο imagebutton είναι το μικρόφωνο της Google όπου ο χρήστης μόλις κλικάρει την εικόνα θα έχει την δυνατότητα να ηχογραφήσει την απάντησή του και μέσω της εφαρμογής της Google θα λάβει και το ανάλογο αποτέλεσμα.

Τέλος δημιουργούμε και ένα textview όπου θα εμφανίζονται τα αποτελέσματα των απαντήσεων που δίνει ο χρήστης.

Αρχείο java

```

package com.example.katerina.degree_paper;

import androidx.appcompat.app.AppCompatActivity;

public class Animals extends AppCompatActivity {

    MediaPlayer mySong,success,nosuccess;

    private static final int REQ_CODE_SPEECH_INPUT = 100;
    private TextView mVoiceInputTv;
    private ImageButton mSpeakBtn;

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

        mySong = MediaPlayer.create(this, R.raw.tiger);
        success = MediaPlayer.create(this, R.raw.success);
        nosuccess = MediaPlayer.create(this, R.raw.nosuccess);

        mVoiceInputTv = (TextView) findViewById(R.id.voiceInput);
        mSpeakBtn = (ImageButton) findViewById(R.id.btnSpeak);
        mSpeakBtn.setOnClickListener(() -> startVoiceInput());
    }
}

```

```

mSpeakBtn.setOnClickListener(() -> startVoiceInput());

private void startVoiceInput() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Μπαράβα το κατόπρησ!");
    try {
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    } catch (ActivityNotFoundException a) {
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case REQ_CODE_SPEECH_INPUT: {
            if (resultCode == RESULT_OK && null != data) {
                ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                mVoiceInputTv.setText(result.get(0));
                if (mVoiceInputTv.getText().toString().equals("τιφωνη")) {
                    Toast.makeText(getApplicationContext(), "Μπαράβα το κατόπρησ!", Toast.LENGTH_SHORT).show();
                    Right(R.raw.success);
                }
            }
        }
    }
}

```


manifest

Στο συγκεκριμένο αρχείο είναι δηλωμένες όλες οι δραστηριότητες που έχουμε χρησιμοποιήσει στην εφαρμογή μας , το όνομα της εφαρμογής και το logo της. Σε κάθε δραστηριότητα μπορούμε να επιλέξουμε διαφορετικό θέμα. Στη συγκεκριμένη εφαρμογή έχουμε αλλάξει το θέμα σε δύο δραστηριότητες (Splash_screen και MainActivity) διότι δεν θέλαμε να εμφανίζεται η μπάρα με το menu.

menu

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3        xmlns:app="http://schemas.android.com/apk/res-auto">
4
5      <item
6          android:id="@+id/home"
7          android:enabled="true"
8          android:icon="@drawable/ic_home"
9          android:title="Home"
10         app:showAsAction="ifRoom" />
11
12     <item android:id="@+id/fruits"
13         android:title="#φρούτα"
14         app:showAsAction="never" />
15
16     <item android:id="@+id/animals"
17         android:title="Ζώα"
18         app:showAsAction="never" />
19
20     <item android:id="@+id/vegetables"
21         android:title="Μαγούλες"
22         app:showAsAction="never" />
23
24 </menu>

```

Εδώ δημιουργούμε το menu. Δηλώνουμε όνομα για κάθε αντικείμενο και σε ποια δραστηριότητα αναφέρεται το καθένα.

ΚΕΦΑΛΑΙΟ 4 Παρουσίαση Εφαρμογής

4.1 Αρχική Σελίδα

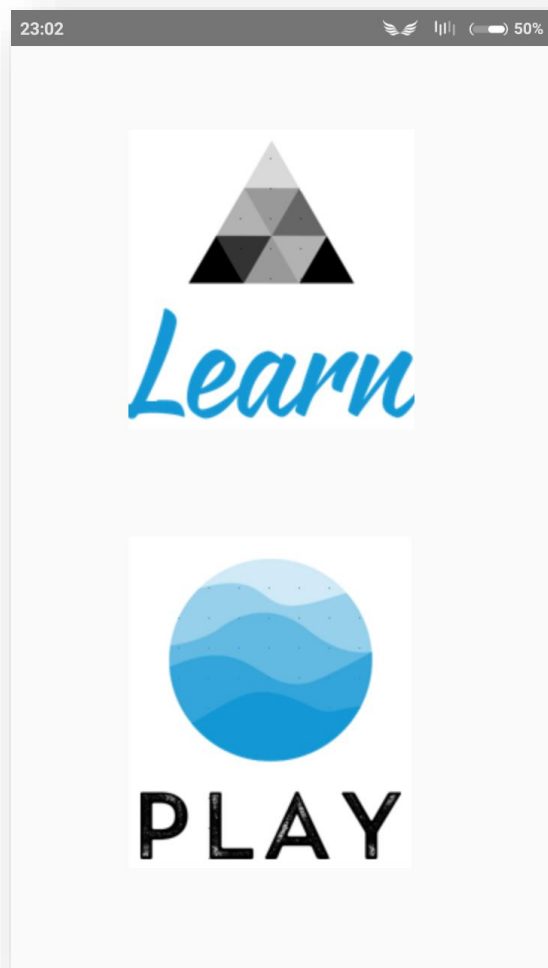
Όταν ο χρήστης κλικάρει στο εικονίδιο της εφαρμογής αρχικά βλέπει το λογότυπο του πανεπιστημίου.



Έπειτα κατευθύνεται στην αρχική σελίδα που καλείται να πατήσει το πλήκτρο start,

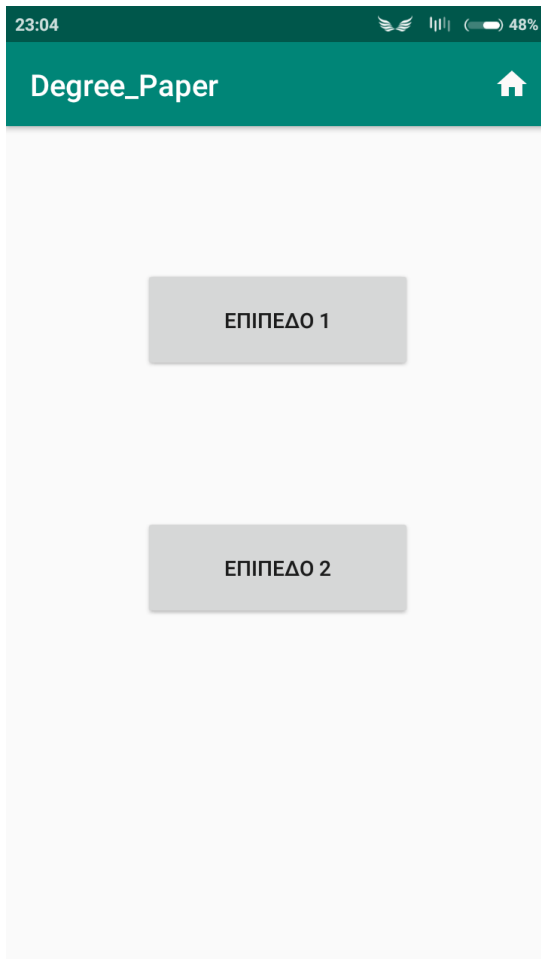


ώστε να ξεκινήσει και να διαλέξει την επιθυμητή λειτουργία Learn ή Play ,
δηλαδή αν θέλει να παίξει ή να μάθει.

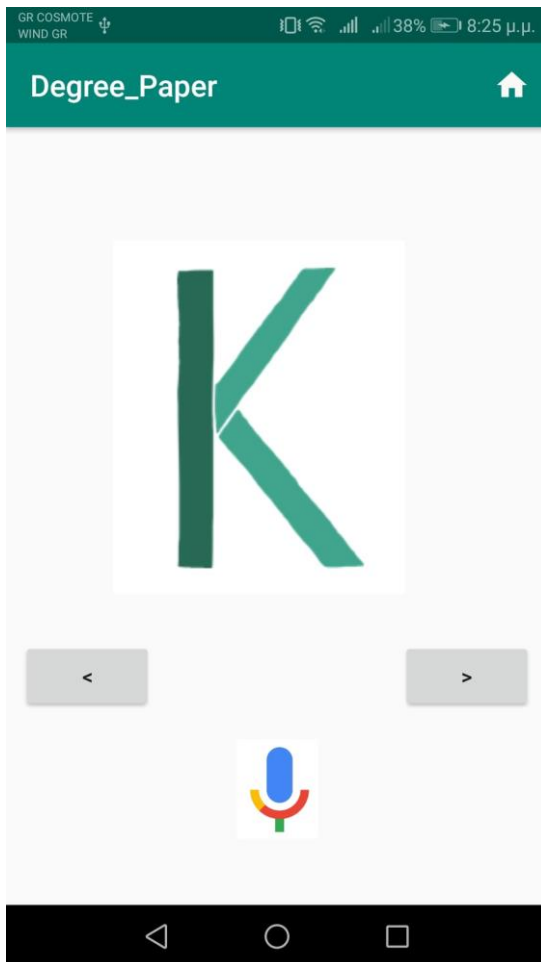


4.2 Λειτουργία LEARN Εκμάθησης

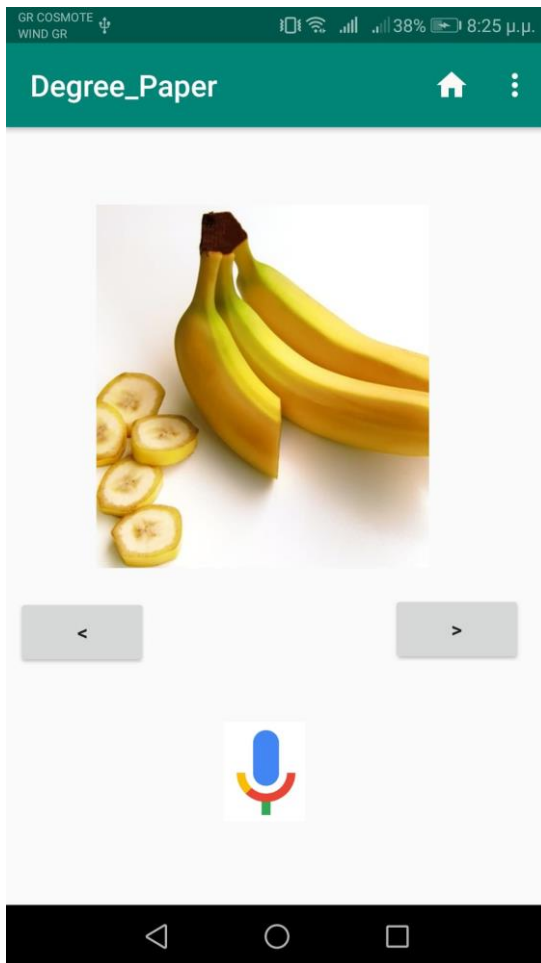
Όταν διαλέξει την λειτουργία Learn, καλείται να διαλέξει επίπεδο 1 που περιέχει γράμματα ή επίπεδο 2 που περιέχει λέξεις από τις κατηγορίες φρούτα ζώα λαχανικά.



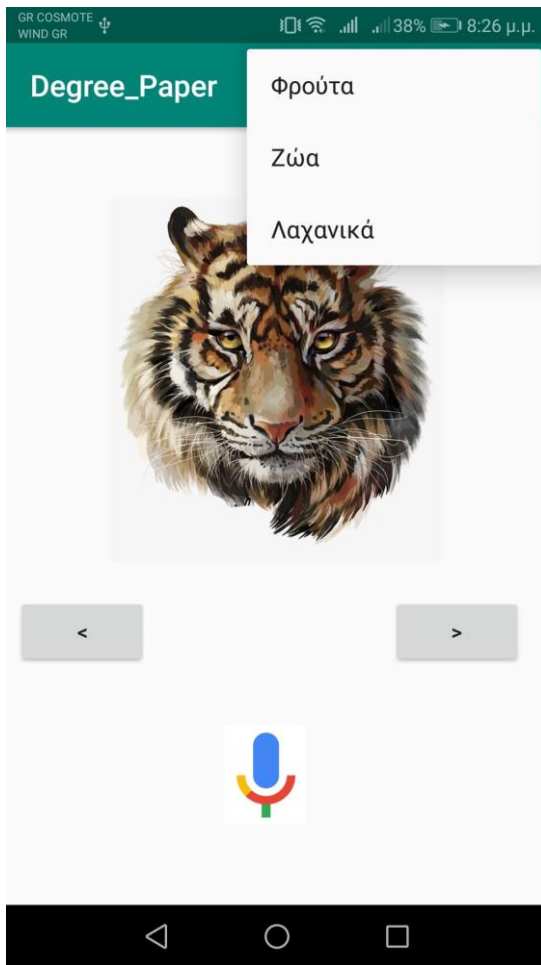
Το πρώτο γράμμα που εμφανίζεται είναι το κ και μπορείς να προηγηθείς και σε άλλα γράμματα πατώντας τα βελάκια, να ακούσεις την σωστή εκφορά του πατώντας πάνω στο γράμμα ή να εξασκήσεις την εκφορά του πατώντας το μικρόφωνο. Ότι ειπωθεί θα εμφανιστεί στην οθόνη ακριβώς πάνω στην εικόνα και ανάλογα αν είναι σωστό ή λάθος θα εμφανιστεί το αντίστοιχο μήνυμα και ήχος επιτυχίας ή αποτυχίας.



Στο δεύτερο επίπεδο εμφανίζεται η μπανάνα όπου επικρατεί η ίδια λογική πλοήγησης και χρήσης με πριν.

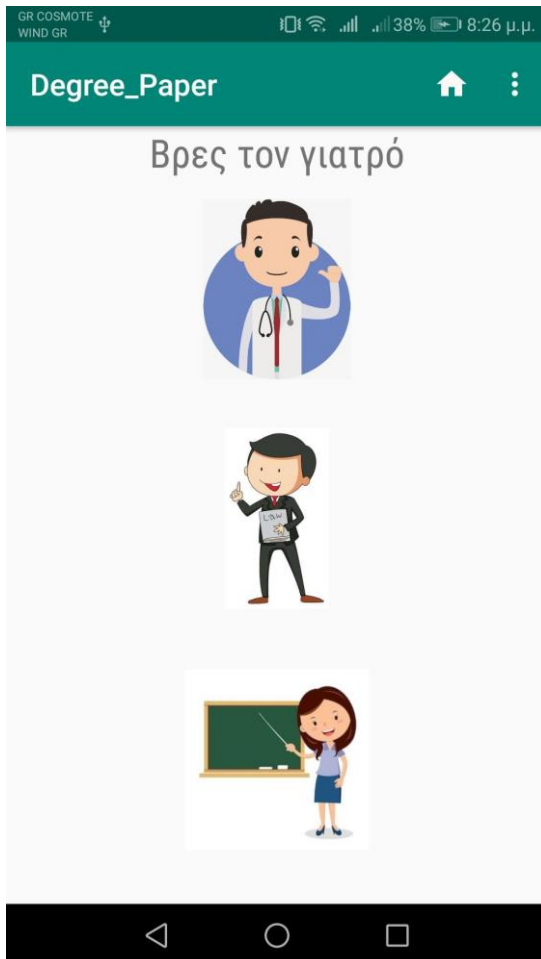


Πατώντας τις τρεις τελείες ο χρήστης μπορεί να μεταβεί στις υπόλοιπες κατηγορίες λέξεων.

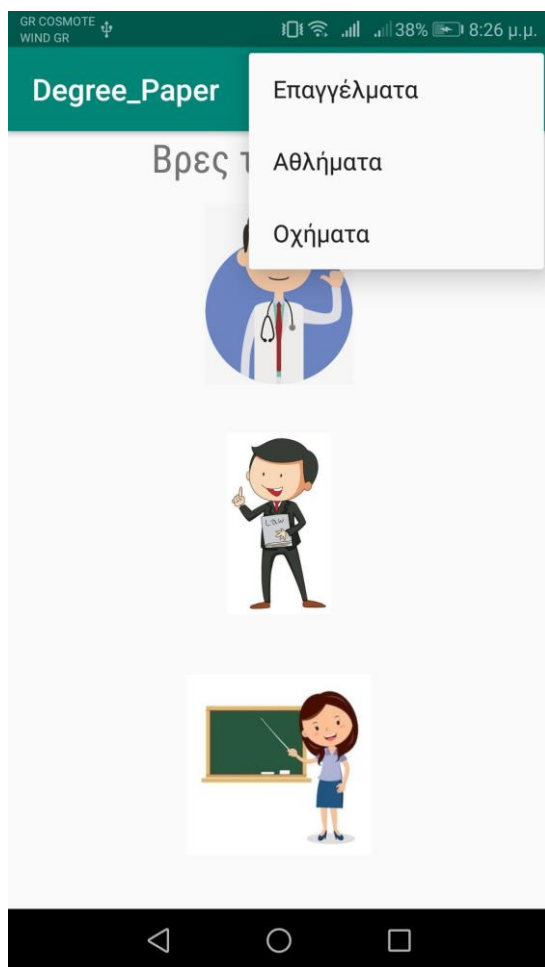


4.3 Λειτουργία PLAY Παιχνιδιού

Σε αυτό το παιχνίδι ο χρήστης καλείται να βρει την σωστή απάντηση δίνοντας του τρεις επιλογές. Αμέσως μετά την απάντηση του θα εμφανιστεί το κατάλληλο μήνυμα, επιτυχίας ή αποτυχίας, και ο αντίστοιχος ήχος.



Πατώντας τις τρεις τελείες ο χρήστης μπορεί να μεταβεί στις υπόλοιπες κατηγορίες.



ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα – Προοπτικές

5.1. Συμπεράσματα

Η παρούσα πτυχιακή εργασία πραγματεύεται την δημιουργία μιας εφαρμογής Android ώστε να βοηθήσει τους ασθενείς που ανακάμπτουν από εγκεφαλικό επεισόδιο να εξασκήσουν το λόγο και την ομιλία μέσω της εκφοράς γραμμάτων και λέξεων καθώς και να παίξουν και να διασκεδάσουν μαθαίνοντας Έτσι βοηθούνται οι ασθενείς που ανακάμπτουν από εγκεφαλικό να εξασκηθούν και οι γιατροί λογοθεραπευτές να έχουν ένα χρήσιμο εργαλείο στη διάθεση τους. Με αυτή την εφαρμογή τους δόθηκε η δυνατότητα να συγκρίνουν αυτό που εκφέρουν και να βλέπουν αν είναι σωστό ή όχι.

5.2. Προοπτικές

Σε μελλοντικό επίπεδο θα μπορούσαν να υπάρχουν περισσότεροι βαθμοί σύγκρισης και ποσοστά, στατιστικά για τις επιδόσεις του χρήστη και που χρειάζεται βελτίωση, αναφορά προόδου στο γιατρό με email ή ειδοποιήσεις ακόμα και καρτέλα ασθενή. Επίσης μέσω της αναγνώρισης φωνής να γίνεται και αναγνώριση συναισθημάτων και από εκεί να βγαίνουν στατιστικά των επιδόσεων του χρήστη ανάλογα με την συναισθηματική του κατάσταση και να προσαρμόζεται ανάλογα και το περιβάλλον της εφαρμογής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Blog, A. D. (n.d.). *Android Developers Blog*. Ανάκτηση από <https://android-developers.googleblog.com/2009/02/android-market-update-support-for.html>
- Developer.android. (n.d.). *Developer.android*. Ανάκτηση από <https://developer.android.com/guide/components/fundamentals>
- DGA. (n.d.). *DGA*. Ανάκτηση από <http://www.dga.gr/web/publications/files/android.pdf>
- <https://el.wikipedia.org/wiki/Android>. (n.d.). *Wikipedia*. Ανάκτηση από <https://el.wikipedia.org/wiki/Android>
- Jackrutorial. (n.d.). *Jackrutorial*. Ανάκτηση από <https://www.jackrutorial.com/2018/03/how-to-create-image-button-in-android.html>
- Stacktips. (n.d.). *Stacktips*. Ανάκτηση από <https://stacktips.com/tutorials/android/speech-to-text-in-android>
- Teamtreehouse. (n.d.). *Teamtreehouse*. Ανάκτηση από <https://teamtreehouse.com/community/add-sound-to-button-onclick>
- Technobyte. (n.d.). *Technobyte*. Ανάκτηση από <https://www.technobyte.org/using-mediaplayer-play-pause-stop-audio-file-android/>
- Techopedia. (n.d.). *Techopedia*. Ανάκτηση από <https://www.techopedia.com/definition/25099/android-app>
- Tutlane. (n.d.). *Tutlane*. Ανάκτηση από <https://www.tutlane.com/tutorial/android/android-audio-media-player-with-examples>
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από https://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD_%CF%84%CE%BF%CF%85_Android
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από <https://el.wikipedia.org/wiki/Android>
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από https://el.wikipedia.org/wiki/Android_Studio
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από] <https://el.wikipedia.org/wiki/Java>
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από <https://el.wikipedia.org/wiki/XML>
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από https://el.wikipedia.org/wiki/%CE%91%CE%BD%CE%B1%CE%B3%CE%BD%CF%8E%CF%81%CE%B9%CF%83%CE%B7_%CE%BF%CE%BC%CE%B9%CE%BB%CE%AF%CE%B1%CF%82
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από https://en.wikipedia.org/wiki/Mobile_app
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από [https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από [https://el.wikipedia.org/wiki/Snake_\(%CE%B2%CE%B9%CE%BD%CF%84%CE%B5%CE%BF%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%AF%CE%B4%CE%B9\)](https://el.wikipedia.org/wiki/Snake_(%CE%B2%CE%B9%CE%BD%CF%84%CE%B5%CE%BF%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%AF%CE%B4%CE%B9))
- Wikipedia. (n.d.). *Wikipedia*. Ανάκτηση από https://en.wikipedia.org/wiki/Android_software_development
- Zdnet. (n.d.). *Zdnet*. Ανάκτηση από <https://www.zdnet.com/article/how-android-works-the-big-picture/>