



**Σχεδιασμός και Υλοποίηση μιας gamified μαθησιακής εφαρμογής για παιδιά δημοτικού που προωθεί την ενεργή μάθηση**

**Design and Implementation of a gamified educational application in primary education promoting learning engagement**

Διπλωματική Εργασία

των

**Ανδρέα Κωνσταντάκου & Αθανάσιου Παληγιάννη**

Βόλος, Σεπτέμβριος 2016



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και Υλοποίηση μιας  
gamified μαθησιακής εφαρμογής για παιδιά  
δημοτικού που προωθεί την ενεργή μάθηση**

**Design and implementation of a  
gamified educational application  
in primary education promoting  
learning engagement**

Διπλωματική Εργασία

Των

**Ανδρέα Κωνσταντάκου & Αθανάσιου Παληγιάννη**

**Επιβλέποντες:**

<b>Παναγιώτα Τσομπανοπούλου</b>	<b>Δασκαλοπούλου Ασπασία</b>
Αναπληρώτρια Καθηγήτρια, Πανεπιστήμιο Θεσσαλίας	Επίκουρη Καθηγήτρια, Πανεπιστήμιο Θεσσαλίας



Εγκρίθηκε από την διμελή επιτροπή την ημερομηνία εξέτασης

*(Υπογραφή)*

.....

ΚΥΡΙΟΣ ΕΠΙΒΛΕΠΩΝ

Αναπληρώτρια Καθηγήτρια Π.Θ.

*(Υπογραφή)*

.....

ΔΕΥΤΕΡΕΥΩΝ ΕΠΙΒΛΕΠΩΝ

Επίκουρη Καθηγήτρια Π.Θ.

Βόλος, Σεπτέμβριος 2016

*(Υπογραφή)*

.....

**Ανδρέας Κωνσταντάκος**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός  
Υπολογιστών, Πανεπιστημίου Θεσσαλίας

© 2016 – All rights reserved

*(Υπογραφή)*

.....

**Αθανάσιος Παληγιάννης**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός  
Υπολογιστών, Πανεπιστημίου Θεσσαλίας

© 2016 – All rights reserved

Βόλος, Σεπτέμβριος 2016

Στις οικογένειες μας και στους φίλους μας

## Ευχαριστίες

Με την περάτωση της παρούσας εργασίας, θα θέλαμε να ευχαριστήσουμε θερμά τους επιβλέποντες της Διπλωματικής εργασίας κ. Παναγιώτα Τσομπανοπούλου και κ. Ασπασία Δασκαλοπούλου για την εμπιστοσύνη που επέδειξαν στα πρόσωπά μας. Θα θέλαμε επίσης να ευχαριστήσουμε την κ. Χαρίκλεια Τσαλαπάτα για την άριστη συνεργασία, τις ουσιώδεις υποδείξεις και παρεμβάσεις, που διευκόλυναν την εκπόνηση της Διπλωματικής εργασίας.

Επίσης θα θέλαμε να εκφράσουμε την εκτίμησή μας στον συνεργάτη και συνάδελφο Δημήτριο Μπέλλο, με τον οποίο κάναμε πραγματικότητα αυτή την διπλωματική εργασία.

Τέλος, οφείλουμε ένα μεγάλο ευχαριστώ στις οικογένειες μας και στους φίλους μας για την αμέριστη υποστήριξη και την ανεκτίμητη βοήθεια που μας παρείχαν τόσο κατά την διάρκεια των σπουδών μας, όσο και κατά την εκπόνηση της Διπλωματικής εργασίας.

**Ανδρέας Κωνσταντάκος & Αθανάσιος Παληγιάννης**

**Βόλος, 2016**



## Περίληψη

Ο στόχος αυτής της διπλωματικής εργασίας είναι η σχεδίαση και υλοποίηση μίας παιχνιδοποιημένης μαθησιακής εφαρμογής για παιδιά δημοτικού σε λειτουργικό περιβάλλον Android που προωθεί την ενεργή μάθηση.

Συγκεκριμένα, ο χρήστης-μαθητής μπορεί από μία εφαρμογή-παιχνίδι να αποκομίσει γνώσεις πάνω στην ύλη η οποία διδάσκεται στο δημοτικό σχολείο μέσα από μία ευχάριστη ψυχαγωγική διαδικασία.

Ο σχεδιασμός και υλοποίηση αυτής της εφαρμογής έγινε σε λειτουργικό περιβάλλον Android με την χρήση του εργαλείου Android Studio 2.0, αφού προηγουμένως έγινε μία μελέτη της υπάρχουσας κατάστασης στην εκπαίδευση, ανάλυση των αναγκών και αξιοποιώντας τις αρχές της παιχνιδοποίησης.

## Abstract

The purpose of this thesis was the design and implementation of a gamified educational application in primary education promoting learning engagement, using the Android operating system.

Specifically, the user-student can gain knowledge on primary school material with the application-game through a pleasant entertaining process.

The design and implementation of this application was developed in the Android operating system using the Android Studio 2.0 tool. In order to achieve that, a thorough study of the current situation in education, needs' analysis and exploitation of gamifications' principles took part.

## Περιεχόμενα

Εισαγωγή .....	12
1.Ανάλυση της υπάρχουσας κατάστασης στη μάθηση .....	12
1.1 Εφαρμογές που χρησιμοποιούνται σήμερα .....	13
1.1.1 Εφαρμογές σε διαδικτυακές πλατφόρμες .....	13
1.1.2 Εφαρμογές για φορητές συσκευές .....	17
1.2 Τι κενά εντοπίζονται στις παραπάνω εφαρμογές .....	21
1.3 Ποια είναι τα πλεονεκτήματα της δικής μας εφαρμογής .....	22
2. Αρχές της παιχνιδοποίησης .....	22
2.1 Ορισμός της παιχνιδοποίησης .....	22
2.2 Ιστορία της παιχνιδοποίησης .....	23
2.3 Τεχνικές της παιχνιδοποίησης .....	23
2.4 Εφαρμογές της παιχνιδοποίησης .....	23
2.4.1 Μάρκετινγκ .....	24
2.4.2 Έμπνευση .....	24
2.4.3 Υγεία .....	24
2.4.4 Εργασία .....	24
2.4.5 Εκπαίδευση .....	25
3. Λειτουργικό σύστημα Android & το εργαλείο Android Studio .....	26
3.1 Λειτουργικό σύστημα Android .....	26
3.1.1 Γενικά για το Android .....	26
3.1.2 Ιστορική αναδρομή του Android .....	27
3.1.3 Εκδόσεις Android .....	28
3.1.4 Αρχιτεκτονική Android .....	31
3.2 Εργαλείο Android Studio .....	33
3.2.1 Σύντομη επισκόπηση του Android Studio .....	33
4. Υλοποίηση της εφαρμογής Queasy .....	42
4.1 Το Queasy ως εκπαιδευτικό λογισμικό .....	43
4.2 Υλοποίηση της εφαρμογής Queasy .....	43
4.2.1 Δομή της εφαρμογής Queasy .....	44
4.2.2 Περιγραφή των βασικών παραθύρων(Activities) .....	46

4.3 Αλγόριθμοι και τεχνικές προγραμματισμού .....	57
4.3.1 Η σύνδεση της εφαρμογής με τη βάση δεδομένων.....	57
4.3.2 Η υλοποίηση του Mulitplayer Game.....	57
4.3.3 Οι βασικοί αλγόριθμοι της εφαρμογής.....	58
4.4 Στατιστικά στοιχεία για την εφαρμογή .....	66
5. Τελικές σκέψεις-Συμπεράσματα.....	66
Παράρτημα.....	68
Βιβλιογραφία.....	68

## Εισαγωγή

Βρισκόμενοι οι ίδιοι στον χώρο της εκπαίδευσης για πολλά χρόνια(συγκεκριμένα στο τέλος της τριτοβάθμιας εκπαίδευσης) συνειδητοποιήσαμε ότι η συνεργασία τεχνολογίας-εκπαίδευσης είναι, όχι μόνο πολύ μακριά από το επίπεδο όπου θα αξιοποιούταν οι δυνατότητες αυτής της «συνεργασίας» στο μέγιστο, αλλά αντίθετα η τεχνολογία τείνει να «εκπαιδεύει» τους ανθρώπους στην αδράνεια σε πολλές περιπτώσεις. Συγκεκριμένα, ενώ υπάρχουν πολλές εφαρμογές(π.χ. παιχνίδια), ελάχιστες από αυτές έχουν εκπαιδευτικό χαρακτήρα, και όσες το κάνουν, σίγουρα δεν έχουν δείξει ανάλογη επιτυχία με τις υπόλοιπες. Βασιζόμενοι σε αυτές τις παρατηρήσεις και με κίνητρο την ανατροπή αυτού του σκηνικού, επιχειρήσαμε να υλοποιήσουμε μία εφαρμογή-παιχνίδι εκπαιδευτικού χαρακτήρα που να απευθύνεται στη βάση της εκπαιδευτικής πυραμίδας, δηλαδή σε παιδιά δημοτικού.

Σήμερα, σχεδόν όλα τα παιδιά, από μικρή ηλικία, έχουν πρόσβαση στην τεχνολογία την οποία χρησιμοποιούν διαρκώς. Αυτή όμως, με την μορφή διαφόρων εφαρμογών, αντί να τους βοηθά να εξελίσσονται, τους κρατά στάσιμους και τους απομακρύνουν από την εκπαίδευση και την γνώση. Στόχος μας είναι να αξιοποιήσουμε τις δυνατότητες που μας προσφέρει η τεχνολογία και σε συνδυασμό με τις αρχές της παιχνιδοποίησης να δώσουμε κίνητρα στα παιδιά, να αποκομίσουν μαθησιακά οφέλη και γνώσεις μέσα από τη διαδικασία του παιχνιδιού και του συναγωνισμού.

Αξιοποιώντας τις δυνατότητες του Android λογισμικού και λόγω της φορητότητας που φέρουν οι συσκευές που το χρησιμοποιούν, υλοποιήσαμε την εφαρμογή “Queasy”, μία εφαρμογή-παιχνίδι(Quiz γνώσεων) με το περιεχόμενο των ερωτήσεων να έχει αντληθεί από τα σχολικά βιβλία όλων των τάξεων του δημοτικού σχολείου. Λαμβάνοντας υπόψιν ότι απευθυνόμαστε σε παιδιά, η εφαρμογή σχεδιάστηκε αρκετά φιλική προς τους νεαρούς χρήστες(π.χ. ελληνικό μενού, αντίστοιχη εικονογράφηση και ήχος, εύκολος χειρισμός κτλ.)

### 1. Ανάλυση της υπάρχουσας κατάστασης στη μάθηση

Η σύγχρονη ανθρώπινη κουλτούρα έχει ως βασικό χαρακτηριστικό τη διείσδυση της ψηφιακής τεχνολογίας σε απόλυτο σχεδόν βαθμό. Τα ψηφιακά μέσα - το Διαδίκτυο, τα κινητά τηλέφωνα, τα ψηφιακά παιχνίδια, η διαδραστική τηλεόραση - αποτελούν πλέον ένα σημαντικό κομμάτι του ελεύθερου χρόνου των παιδιών και των νέων. Η σχέση των νέων με την ψηφιακή τεχνολογία δε διαμορφώνεται πια κατεξοχήν στο πλαίσιο του σχολείου - όπως συνέβαινε κατά τη διάρκεια της δεκαετίας του 1980 και ακόμη και τη δεκαετία του 1990[1].

Ο αρχαίος Έλληνας φιλόσοφος, Πλάτων, ήταν ίσως ο πρώτος που είχε αναφερθεί σε αυτή τη σχέση, με την εξής φράση του: «Να μη χρησιμοποιείς εξαναγκασμό στα παιδιά όταν διδάσκεις, αλλά να τα μορφώνεις παίζοντας»[2].

Όταν ο φυσικός Willy Higginbotham εφηύρε το πρώτο «ψηφιακό παιχνίδι» στο Εθνικό Εργαστήριο Brookhaven στο Upton της Νέας Υόρκης (Brookhaven National Laboratory in Upton New York) το 1958 – ένα είδος επιτραπέζιας αντισφαίρισης που παιζόταν σε έναν ηλεκτρονικό παλμογράφο, δε θα μπορούσε να φανταστεί σε τι σημείο θα καθόριζε η ανακάλυψή του την παγκόσμια βιομηχανία ψυχαγωγίας τα επόμενα πενήντα χρόνια.

Από την εποχή που τα ψηφιακά παιχνίδια έκαναν τα πρώτα τους βήματα, άρρηκτα συνδεδεμένα με την εξέλιξη της υπολογιστικής τεχνολογίας και την ανάπτυξη νέων μέσων, φτάνουμε στη σημερινή εποχή, όπου τα ψηφιακά παιχνίδια είναι η κατηγορία λογισμικού με την οποία είναι πολύ εξοικειωμένα τα παιδιά από πολύ νωρίς και, μάλιστα, μακριά από το περιβάλλον του σχολείου.

Ήδη από τα τέλη της δεκαετίας του 1980, εξαιτίας της μεγάλης απήχησης που είχαν αποκτήσει τα ψηφιακά παιχνίδια της εποχής (“arcade games”) εμφανίστηκαν διάφοροι εκπαιδευτικοί, ψυχολόγοι και κοινωνιολόγοι, οι οποίοι τόλμησαν τις πρώτες απόπειρες διερεύνησης των μεθόδων και των πεδίων όπου θα μπορούσαν να αξιοποιηθούν αυτά τα παιχνίδια σε μαθησιακά περιβάλλοντα και πλαίσια. Εξάλλου, το παιχνίδι ως δραστηριότητα είναι από τη φύση του ένα μαθησιακό περιβάλλον που οι εκπαιδευτικοί με μαθητοκεντρική προσέγγιση επιδιώκουν να αξιοποιήσουν, αφού το θεωρούν έναν από τους πιο αυθεντικούς και παρωθητικούς τρόπους μάθησης, κάτι το οποίο υποστηρίζουν και οι θεωρητικοί της μάθησης, όπως ο Piaget, ο Bruner και ο Vygotsky.

Ένα από τα βασικότερα ερωτήματα που προκύπτει, λοιπόν, μέσα στα πλαίσια της αξιοποίησης των ψηφιακών μέσων στην εκπαίδευση, είναι με ποιον τρόπο θα μπορούσε το σύγχρονο σχολείο να ανταποκριθεί στο ρόλο που έχουν τα σύγχρονα αυτά μέσα στη ζωή και την καθημερινότητα των παιδιών. Θα έπρεπε να τα αγνοήσει, πρακτική που ακολουθείται ως επί το πλείστον στις ημέρες μας; Θα έπρεπε να τα αξιοποιήσει ως εργαλεία για την επίτευξη των στόχων του Αναλυτικού Προγράμματος Σπουδών, ή, τελικά, να ανακαλύψει τρόπους να τα εντάξει με περισσότερο κριτικό και δημιουργικό τρόπο;

## **1.1 Εφαρμογές που χρησιμοποιούνται σήμερα**

Οι δύο κύριες κατηγορίες μαθησιακών εφαρμογών-παιχνιδιών είναι οι διαδικτυακές πλατφόρμες και οι εφαρμογές για φορητές συσκευές.

### **1.1.1 Εφαρμογές σε διαδικτυακές πλατφόρμες**

Μία διαδικτυακή πλατφόρμα παιχνιδιών είναι μία ιστοσελίδα η οποία λειτουργεί πάνω από ένα server και στην οποία ο χρήστης μπορεί να επιλέξει και να παίξει ένα παιχνίδι από την εκάστοτε γκάμα παιχνιδιών που προσφέρει η πλατφόρμα. Αυτό επιτυγχάνεται με τη χρήση του φυλλομετρητή (browser) είτε στον ηλεκτρονικό υπολογιστή είτε σε κάποια φορητή συσκευή (ταμπλετα, κινητό τηλέφωνο κτλ.).

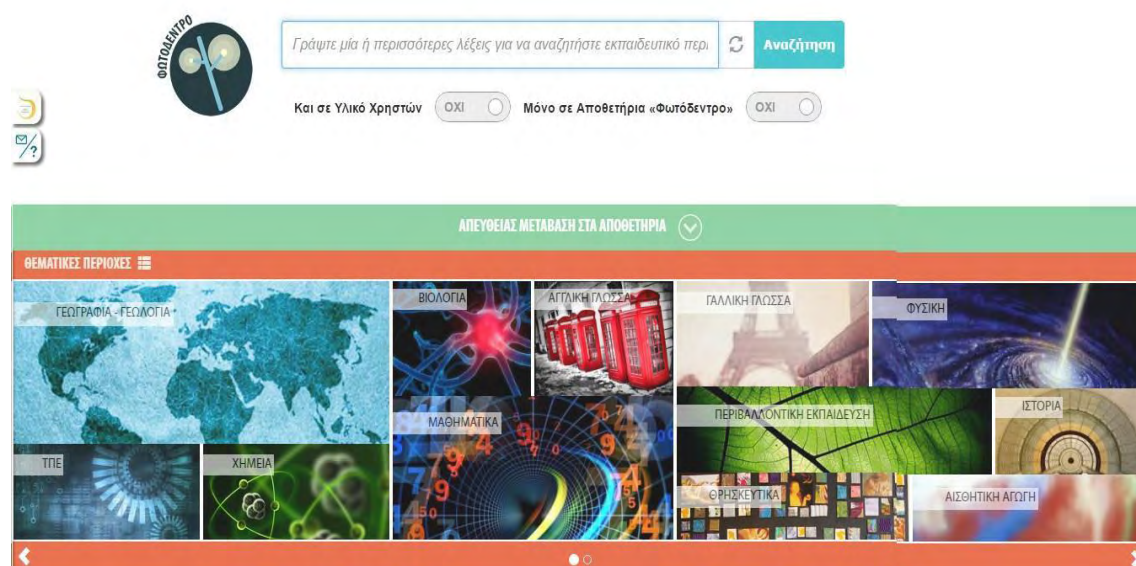
Σε αυτό το σημείο θα αναφέρουμε μερικές γνωστές διακτυακές πλατφόρμες παιχνιδιών που χρησιμοποιούνται ευρέως στην Ελλάδα.

<http://photodentro.edu.gr/>

Το Φωτόδεντρο > Εκπαιδευτικό Λογισμικό (Φωτόδεντρο EDUSOFT) είναι το Πανελλήνιο Αποθετήριο Εκπαιδευτικού Λογισμικού για την πρωτοβάθμια και τη δευτεροβάθμια εκπαίδευση[3].

Φιλοξενεί εκπαιδευτικά λογισμικά και πακέτα με εκπαιδευτικές δραστηριότητες για τοπική μεταφόρτωση.

Το Φωτόδεντρο > Εκπαιδευτικό Λογισμικό σχεδιάστηκε και αναπτύχθηκε από το ΙΤΥΕ – ΔΙΟΦΑΝΤΟΣ στο πλαίσιο του «Ψηφιακού Σχολείου» με στόχο τη συγκέντρωση, ανάρτηση, περιγραφή και ευρεία διάθεση σε σχολεία, εκπαιδευτικούς και μαθητές εκπαιδευτικών λογισμικών και πακέτων που αναπτύχθηκαν, προσαρμόστηκαν ή εξελληνίστηκαν στο πλαίσιο έργων του Υπουργείου Παιδείας και Θρησκευμάτων ή φορέων του (ΙΤΥΕ, ΠΙ κλπ), από το 1998 έως σήμερα, ή άλλων προϊόντων εκπαιδευτικού λογισμικού που διαθέτουν σφραγίδα ποιότητας και διατίθενται ελεύθερα προς την εκπαιδευτική κοινότητα.



Εικόνα 1.1 [3]

Σε αυτή τη δικτυακή πλατφόρμα ο χρήστης βρίσκει μια πληθώρα θεμάτων/κατηγοριών εκπαιδευτικού λογισμικού. Στα αρνητικά του βέβαια είναι ότι

σχεδόν το σύνολο του λογισμικού είναι παλαιότερης τεχνολογίας, ασυμβατό σε πολλές περιπτώσεις με την τεχνολογία που χρησιμοποιείται σήμερα και μη ενημερωμένο ως προς το περιεχόμενό του. Επίσης να συμπληρώσουμε ότι για να χρησιμοποιηθεί το συγκεκριμένο λογισμικό, το οποίο δεν είναι παιχνιδιοποιημένο, πρέπει να το κατεβάσεις και να το εγκαταστήσεις στον υπολογιστή σου, γεγονός που το καθιστά δύσχρηστο.

<http://e-didaskalia.blogspot.gr/>

Η συγκεκριμένη διαδικτυακή πλατφόρμα προσφέρει εκτός των άλλων ένα μεγάλο εύρος παιχνιδιών εκπαιδευτικού χαρακτήρα τόσο γνώσεων όσο και δεξιοτήτων που ξεκινά από ηλικίες 3 έως 7(νηπιαγωγείο) και περιλαμβάνει όλα τα στάδια της εκπαίδευσης(δημοτικό, γυμνάσιο, λύκειο)[4].

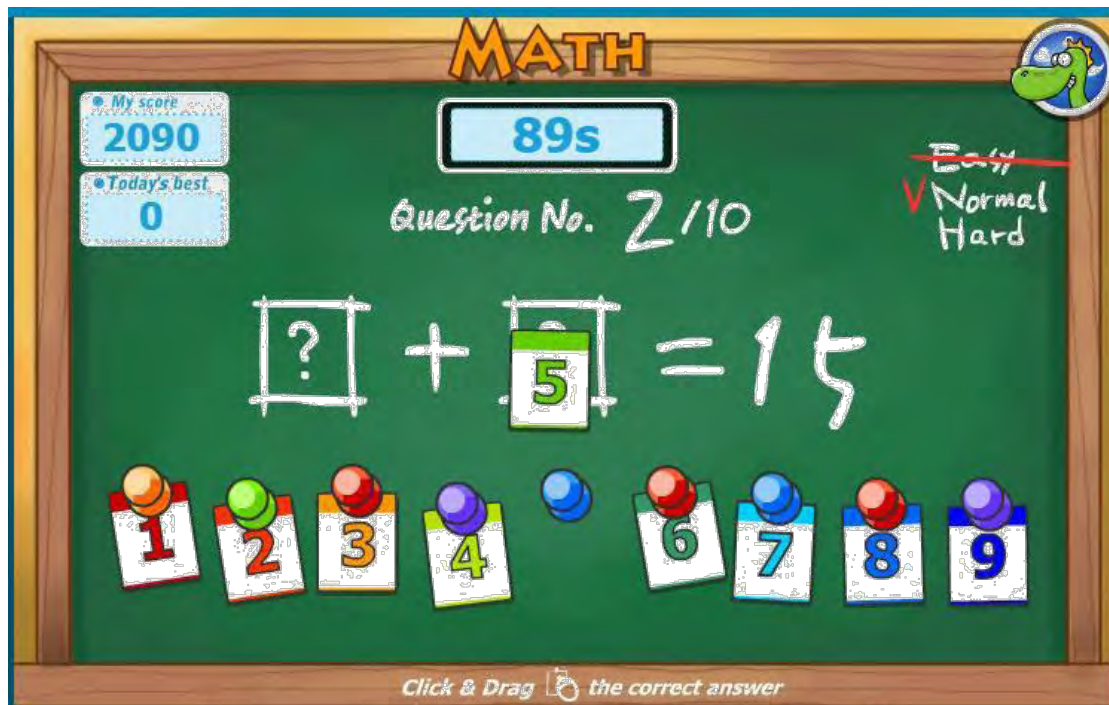


Εικόνα 1.2 [4]

Σε αντίθεση με την προηγούμενη πλατφόρμα, η συγκεκριμένη προσφέρει το πλεονέκτημα ότι η εφαρμογή εκτελείται διαδικτυακά(δλδ δεν χρειάζεται εγκατάσταση). Πολλές από τις εφαρμογές τις είναι παιχνιδιοποιημένες. Ωστόσο δεν στοχεύουν καθαρά προς την εκπαίδευση, αλλά συνδυάζουν γνώσεις και δεξιότητες, χωρίς όμως να δίνουν την δυνατότητα της αλληλεπίδρασης μεταξύ των χρηστών. Εκτός από τις διαδικτυακές της εφαρμογές, προσφέρει λογισμικό προς εγκατάσταση, το οποίο όμως απουσιάζει σε πολλές περιπτώσεις.

[www.games.gr](http://www.games.gr)

Αυτή η διαδικτυακή πλατφόρμα προσφέρει μία τεράστια ποικιλία εφαρμογών-παιχνιδιών μεταξύ των οποίων και την υποκατηγορία αυτών με εκπαιδευτικό χαρακτήρα[5].



Εικόνα 1.3 [5]

Στα πλεονεκτήματά της είναι αναμφίβολα α) οι φιλικές προς τον χρήστη διαδικτυακές εφαρμογές της και β) η ελκυστικότητά τους. Από την άλλη καμία από τις εφαρμογές δεν είναι γραμμένη στην ελληνική γλώσσα, το αντικείμενό τους είναι γενικού περιεχομένου και δεν συνάδει τόσο με την εκπαιδευτική διαδικασία, καθώς απαιτεί περισσότερο δεξιότητες παρά γνώσεις.

<http://ts.sch.gr/software>

Η υπηρεσία της Τεχνικής Στήριξης (ΤΣ) Πληροφοριακών Συστημάτων Σχολικών Μονάδων του Υ.Π.&Θ.Π.&Α. έχει στόχο την υποστήριξη του πληροφοριακού εξοπλισμού των Εκπαιδευτικών και Διοικητικών μονάδων Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης. Στον παρόντα δικτυακό χώρο παρέχεται σε online μορφή το πιστοποιημένο εκπαιδευτικό λογισμικό για την Α'Βάθμια & Β'Βάθμια εκπαίδευση και μπορείτε να το χρησιμοποιήσετε ανεξάρτητα του λειτουργικού σας συστήματος και του φυλλομετρητή που διαθέτει ο Η/Υ σας[6].





Εικόνα 1.4 [6]

Στο συγκεκριμένο ιστότοπο ο χρήστης μπορεί να βρει λογισμικό το οποίο καλύπτει όλη την σχολική ύλη τόσο της Α΄βάθμιας όσο και της Β΄βάθμιας εκπαίδευσης. Το περιεχόμενο του λογισμικού είναι καθαρά εκπαιδευτικού χαρακτήρα αντλημένο από την σχολική ύλη της εκάστοτε τάξης. Η πρόσβαση στις εφαρμογές είναι αρκετά εύκολη και δεν απαιτεί εγκατάσταση του λογισμικού. Σαν αρνητικά της πλατφόρμας θα μπορούσαμε να αναφέρουμε ότι οι εφαρμογές δεν είναι αρκετά παιχνιδοποιημένες, χαρακτηριστικό που κουράζει τον χρήστη. Επίσης απουσιάζει οποιαδήποτε μορφή αλληλεπίδρασης μεταξύ των χρηστών.

### 1.1.2 Εφαρμογές για φορητές συσκευές

Οι εφαρμογές για φορητές συσκευές είναι ως επί το πλείστον εφαρμογές που εκτελούνται σε περιβάλλον Android ή IOS και εγκαθίστανται σε κινητά τηλέφωνα και ταμπλέτες[7].

Σε αυτό το σημείο θα αναφέρουμε μερικές εφαρμογές παιχνιδιών εκπαιδευτικού χαρακτήρα που χρησιμοποιούνται ευρέως στην Ελλάδα. Χωρίζονται σε κάποιες κατηγορίες όπως Κουίζ γνώσεων (QUIZDOM, Το Παιχνίδι των Γνώσεων, Εύρηκα-Quiz Γνώσεων, Σωστό ή λάθος, Εκατομμυριούχος κουίζ κτλ.), Κρεμάλες (Κρεμάλα στα ελληνικά, Greek Hangman κτλ.), Σταυρόλεξα (Σταυρόλεξα στα ελληνικά, Λεξομανία, Γεμίζω σε λόγια κτλ.), Κρυπτόλεξα, Παιχνίδια με λέξεις και αριθμούς (Όνομα Ζώο Φυτό, Βρες τη λέξη, μαθηματικά-προπαίδια κτλ.) κ.α.

### Κρεμάλα στα ελληνικά

Η κρεμάλα είναι ένα κλασικό και διασκεδαστικό παιχνίδι. Παράλληλα μπορεί να χρησιμοποιηθεί ως εκπαιδευτική εφαρμογή μέσω της οποίας μπορεί κανείς να εμπλουτίσει το λεξιλόγιο και της γνώσης του[7].



Εικόνα 1.5 [7]

Παρ' όλα αυτά, οι γνώσεις που αποκομίζει ο χρήστης μέσα από το παιχνίδι περιορίζονται καθαρά στο λεξιλόγιο και την ορθογραφία. Δεν αποκομίζει γνώσεις γραμματικής, συντακτικού, έκφρασης λόγου κτλ., τα οποία είναι όλα πολύ σημαντικό κομμάτι της Γλώσσας.

### QUIZDOM

Το " QUIZDOM " είναι το πιο διαδεδομένο κουίζ γνώσεων στην Ελλάδα. Σου δίνει τη δυνατότητα να τεστάρεις τις γνώσεις σου εναντίον άλλων. Προσφέρει στο χρήστη 18 διαφορετικές κατηγορίες ερωτήσεων. Το παιχνίδι αποτελείται από 4 γύρους τεσσάρων ερωτήσεων. Οι δύο αντίπαλοι απαντούν εναλλάξ στις ίδιες 16 ερωτήσεις. Κάθε παίκτης διαλέγει στην αρχή του παιχνιδιού 2 κατηγορίες. Εκείνος που θα απαντήσει σωστά στις περισσότερες είναι ο νικητής[7].



Εικόνα 1.6 [7]

Ωστόσο το περιεχόμενο των ερωτήσεων είναι γενικού χαρακτήρα και απέχει πολύ από μια εκπαιδευτική εφαρμογή, όπως συμβαίνει και στο σύνολο των κουίζ γνώσεων που έχουν υλοποιηθεί για Android ή IOS περιβάλλοντα. Επίσης τόσο το QUIZDOM όσο και τα υπόλοιπα κουίζ γνώσεων δε προσφέρουν τη δυνατότητα να αναμετρηθείς με τον αντίπαλό σου σε πραγματικό χρόνο. Το QUIZDOM συγκεκριμένα λειτουργεί ασύγχρονα. Δηλαδή, όταν έχεις ένα νέο αίτημα παιχνιδιού από κάποιον παίκτη, έχεις ένα συγκεκριμένο χρονικό όριο να το αποδεχτείς και να παίξεις και το παιχνίδι δεν εξελίσσεται σε πραγματικό χρόνο.

### Σταυρόλεξα στα ελληνικά

Το «Σταυρόλεξα στα ελληνικά» είναι ένα κλασικό παιχνίδι μυαλού και γνώσεων. Η εφαρμογή για Android προσφέρει 120 σταυρόλεξα και πάνω από 2000 λέξεις[7].



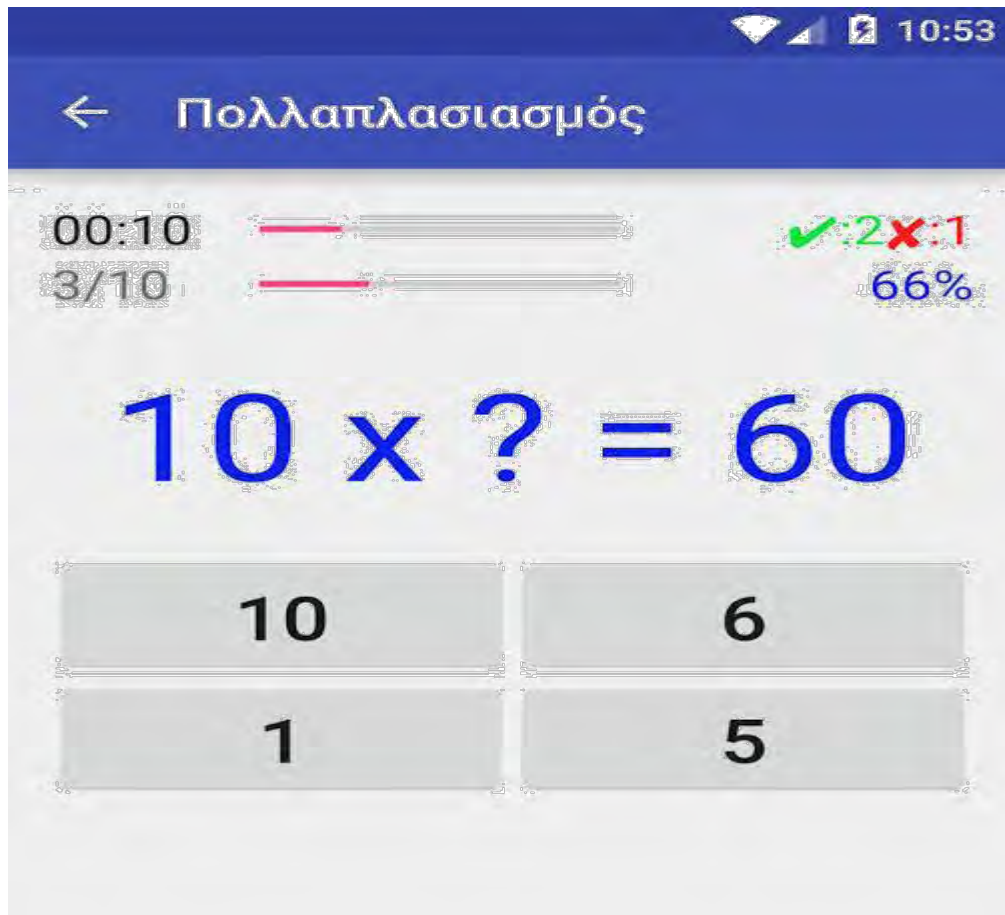
Εικόνα 1.7 [7]

Όπως και στην κρεμάλα έτσι και στις εφαρμογές σταυρόλεξα οι γνώσεις που αποκομίζει ο χρήστης μέσα από το παιχνίδι περιορίζονται καθαρά στο λεξιλόγιο και την ορθογραφία. Δεν αποκομίζει γνώσει γραμματικής, συντακτικού, έκφρασης λόγου κτλ., τα οποία είναι όλα πολύ σημαντικό κομμάτι της Γλώσσας.

### Μάθε την προπαίδεια

Μια απλή εφαρμογή για μαθητές των πρώτων τάξεων του δημοτικού. Αυτή η απλή και πλήρως εξελληνισμένη εφαρμογή βοηθά τα παιδιά να μάθουν την προπαίδεια εξασκούμενα με διαφορετικών τύπων πολλαπλών επιλογών τεστ. Μπορούν να επιλέξουν από 3 τύπους τεστ: α) βρίσκοντας το αποτέλεσμα ενός πολλαπλασιασμού, β) βρίσκοντας τον αριθμό που λείπει σε ένα πολλαπλασιασμό και γ) βρίσκοντας το ζευγάρι των αριθμών που οδηγεί σε ένα πολλαπλασιασμό. Επίσης παρέχει μία λειτουργία μελέτης στην οποία τα παιδιά μπορούν να δουν μόνο τους πίνακες πολλαπλασιασμού και να τους μελετήσουν με το δικό τους ρυθμό[7].





Εικόνα 1.8 [7]

Το κύριο μειονέκτημα αυτής της εφαρμογής είναι ότι ο χρήστης μπορεί να εξασκήσει τις γνώσεις του μόνο στον τομέα της προπαίδειας και δεν καλύπτει όλο το υπόλοιπο φάσμα των Μαθηματικών. Επίσης, τόσο γραφικά όσο και λειτουργικά(αλγοριθμικά) υπάρχουν πολλά κενά στην εφαρμογή, με αποτέλεσμα να μην είναι πολύ ελκυστική για τον χρήστη.

## 1.2 Τι κενά εντοπίζονται στις παραπάνω εφαρμογές

Μετά από μία αναλυτική έρευνα τόσο στις υπάρχουσες διαδικτυακές πλατφόρμες όσο και στις εφαρμογές για φορητές συσκευές που υπήρχαν διαθέσιμες στην Ελλάδα, παρατηρήσαμε μία σειρά από κενά-μειονεκτήματα. Αυτά τα κενά δυσκολεύουν το χρήστη-μαθητή να αξιοποιήσει στο έπακρο την τεχνολογία με απώτερο σκοπό την απόκτηση γνώσεων μέσα από μια ευχάριστη γι' αυτόν διαδικασία.

Αρχικά, κάποιες από αυτές τις εφαρμογές είχαν ασυμβατότητες με την σημερινή τεχνολογία, απαιτούσαν εγκατάσταση της εφαρμογής από τον χρήστη μέσα από μία διαδικασία η οποία είναι κουραστική και πλέον αχρείαστη στην εποχή μας. Ένα επιπλέον αρνητικό κοινό χαρακτηριστικό πολλών εφαρμογών είναι ότι το υλικό(λογισμικό, περιεχόμενο κτλ.) τους δεν ενημερώνεται συχνά ή καθόλου, καθώς επίσης και το γραφικό περιβάλλον δεν έχει τροποποιηθεί καθόλου από την ημέρα που κυκλοφόρησε η εφαρμογή.

Επίσης, όσον αφορά τις διαδικτυακές πλατφόρμες, οι περισσότερες είχαν εκπαιδευτικό περιεχόμενο και χαρακτήρα, τα οποία είχαν αντληθεί από τα σχολικά βιβλία, αλλά δεν ήταν παιχνιδοποιημένες. Δηλαδή η διαδικασία της μάθησης δεν ήταν τόσο διασκεδαστική για τον χρήστη. Αντίθετα, οι περισσότερες εφαρμογές για φορητές συσκευές ήταν ως επί το πλείστον παιχνίδια. Ωστόσο, ο εκπαιδευτικός χαρακτήρας απουσίαζε από αυτές και το γνωστικό αντικείμενο ήταν πολύ πιο γενικό από αυτό που θα προσδοκούσαμε για μια μαθησιακή εφαρμογή.

Κοινό χαρακτηριστικό σχεδόν όλων των παραπάνω εφαρμογών είναι η απουσία της αλληλεπίδρασης και του συναγωνισμού μεταξύ των χρηστών, στοιχείο που κάνει μια εφαρμογή-παιχνίδι πολύ ενδιαφέρουσα και ελκυστική. Ειδικότερα δε, εφαρμογή που οι χρήστες θα μπορούσαν να συναγωνίζονται σε πραγματικό χρόνο(QuizUp), δε συναντήσαμε καμία.

Τέλος, παρόλο που αναφερόμαστε σε ελληνικές πλατφόρμες και εφαρμογές, σε κάποιες από αυτές απουσίαζε η ελληνική γλώσσα τόσο από το περιβάλλον, όσο και από το περιεχόμενο.

### **1.3 Ποια είναι τα πλεονεκτήματα της δικής μας εφαρμογής**

Λαμβάνοντας υπόψιν όλες τις παραπάνω παρατηρήσεις, στόχος μας είναι να δημιουργήσουμε μία εφαρμογή εκπαιδευτικού χαρακτήρα, φιλική και ελκυστική για τον χρήστη, αξιοποιώντας, ταυτόχρονα, όλες τις δυνατότητες που μας προσφέρει η σύγχρονη τεχνολογία. Ο τρόπος για να το επιτύχουμε αυτό είναι να κρατήσουμε όλα τα θετικά χαρακτηριστικά των εφαρμογών που αναφέραμε παραπάνω, διορθώνοντας τι όποιες αδυναμίες, και στηριζόμενοι στις αρχές της παιχνιδοποίησης(gamification), την οποία θα αναλύσουμε στο επόμενο κεφάλαιο.

## **2. Οι Αρχές της παιχνιδοποίησης**

Σε αυτό το κεφάλαιο θα περιγράψουμε τι είναι η παιχνιδοποίηση. Θα δούμε από που ξεκίνησε, ποιες είναι οι κύριες τεχνικές οι οποίες εφαρμόζονται, και που βρίσκει εφαρμογή στις μέρες μας[8],[9],[10].

### **2.1 Ορισμός της παιχνιδοποίησης**

Παιχνιδοποίηση είναι η εισαγωγή στοιχείων παιχνιδιού(τεχνικές σχεδιασμού παιχνιδιών και αρχές παιχνιδιών) σε διαδικασίες που δεν σχετίζονται με το παιχνίδι, ώστε να γίνουν πιο ελκυστικές προς τον χρήστη και παρακινώντας τον να επιτύχει πιο ευχάριστα το στόχο του μέσα από την διαδικασία του παιχνιδιού.

Μερικοί από τους τομείς τους οποίους ωφελεί η παιχνιδοποίηση είναι η προώθηση της ενεργής συμμετοχής, η παραγωγικότητα, η διαδικασία της αξιολόγησης, η ευκολία χρήσης και η χρησιμότητα συστημάτων, η φυσική εξάσκηση (ο αθλητισμός κτλ.), η καταπολέμηση της απάθειας και της αδράνειας και φυσικά η μάθηση μέσα από τη διαδικασία του παιχνιδιού, το οποίο σχετίζεται άμεσα με την εφαρμογή την οποία θέλουμε να υλοποιήσουμε.

Μία ανασκόπηση των ερευνών σχετικά με την παιχνιδοποίηση δείχνει τα πολλαπλά οφέλη και θετικά αποτελέσματα που αποκομίζουν οι χρήστες μέσα από αυτή τη διαδικασία. Παρ' όλα αυτά, δεν είναι απόλυτο για όλες τις περιπτώσεις.

## **2.2 Ιστορία της παιχνιδοποίησης**

Το παλαιότερο παράδειγμα παιχνιδοποίησης ήταν προγράμματα τακτικών επιβατών τα οποία χρησιμοποιούσαν αεροπορικές εταιρείες τα οποία προσέφεραν στους πιο πιστούς πελάτες τους.

Η παιχνιδοποίηση σαν όρος δημιουργήθηκε το 2003 από τον Nick Pelling, αλλά δεν είχε απήχηση μέχρι το 2010, όπου και άρχισε να κερδίζει το ενδιαφέρον όταν εταιρείες όπως η "BadgeVille" ξεκίνησαν να τη χρησιμοποιούν για να περιγράψουν τη συμπεριφορά από τις διάφορες πλατφόρμες τους. Η εταιρεία "Gartner" ώθησε τη δημοφιλία της παιχνιδοποίησης λέγοντας ότι «περισσότερο από το 50% των Οργανισμών που διαχειρίζονται πληροφοριακές διαδικασίες θα παιχνιδοποιήσουν αυτές τις διαδικασίες» και πρόσθεσε την παιχνιδοποίηση στις διαφημιστικές τους καμπάνιες. Από το 2011 και μετά, πολλές εταιρείες άρχισαν να αναπτύσσουν παιχνιδοποιημένες πλατφόρμες και έγιναν αρκετά διάσημες.

## **2.3 Τεχνικές της παιχνιδοποίησης**

Οι τεχνικές παιχνιδοποίησης προσπαθούν να μοχλεύσουν τις φυσικές επιθυμίες των ανθρώπων για κοινωνικοποίηση, ανταγωνιστικότητα, μάθηση, πρόοδο, τελειοποίηση, αυτοέκφραση, αλτρουϊσμό και κύρος. Πρώιμες στρατηγικές παιχνιδοποίησης χρησιμοποιούν ανταμοιβές για τους παίκτες που φέρουν εις πέρας μία δραστηριότητα. Οι πιο χαρακτηριστικοί τύποι ανταμοιβών είναι πόντοι, εμβλήματα, επίπεδα, επίτευξη προόδου ή εικονικό χρήμα. Κάνοντας τις ανταμοιβές ορατές προς όλους τους παίκτες και χρησιμοποιώντας πίνακες κατάταξης ενθαρρύνονται οι παίκτες για να συναγωνιστούν μεταξύ τους. Εξαιτίας των πιθανών συνεπειών του ανταγωνισμού, οι καλύτερες πρακτικές παιχνιδοποίησης ευνοούν το συναγωνισμό.

Μία άλλη προσέγγιση της παιχνιδοποίησης είναι να μετατρέπονται οι επικείμενες δραστηριότητες σε παιχνίδια. Κάποιες επιπλέον τεχνικές για να επιτευχθεί αυτό είναι η προσθήκη οδηγιών χρήσης, φωνητικές περιγραφές και αυξανόμενα επίπεδα προκλήσεων.

## **2.4 Εφαρμογές της παιχνιδοποίησης**

Σε αυτό το σημείο θα αναφερθούν οι πιο σημαντικοί τομείς που έχει εφαρμοστεί η παιχνιδοποίηση.

### **2.4.1 Μάρκετινγκ**

Η παιχνιδοποίηση χρησιμοποιείται ευρέως στο χώρο του μάρκετινγκ. Περισσότερο από το 70% των 2000 μεγαλύτερων εταιρειών παγκοσμίως, σύμφωνα με το περιοδικό Forbes, για το έτος 2013, κάνουν χρήση της παιχνιδοποίησης για τον τομέα του μάρκετινγκ και τη διατήρηση του πελατολόγιού τους. Για παράδειγμα, το Νοέμβριο του 2011, Αυστραλιανά μέσα ενημέρωσης σε συνεργασία με τη Yahoo λάνσαραν την εφαρμογή Fango mobile app/SAP, μέσω της οποίας οι τηλεθεατές τους μπορούσαν να αλληλεπιδρούν με ζωντανές εκπομπές με τη χρήση τεχνικών, όπως check-in και εμβλημάτων. Από το Φεβρουάριο του 2012, η εφαρμογή είχε κατέβει περισσότερες από 200.000 φορές από τη μέρα που κυκλοφόρησε. Η παιχνιδοποίηση επίσης έχει χρησιμοποιηθεί σε προγράμματα διατήρησης πελατών. Το 2010, τα Starbucks έδιναν εμβλήματα εκπτώσεων σε πελάτες που είχαν κάνει check-in σε πολλαπλά καταστήματα της αλυσίδας. Επίσης έχουν υπάρξει προτάσεις για χρήση της παιχνιδοποίησης σε διαγωνισμούς ευφυΐας, ενθάρρυνση συμπλήρωσης ερωτηματολογίων και ερευνών αγοράς.

### **2.4.2 Έμπνευση**

Η παιχνιδοποίηση μπορεί να χρησιμοποιηθεί για ιδεασμό, δηλαδή την παραγωγή νέων ιδεών. Μία έρευνα του MIT (Τεχνολογικό Ινστιτούτο της Μασαχουσέτης), έδειξε ότι τα παιχνίδια ιδεασμού βοηθούν τους συμμετέχοντες να παράγουν περισσότερες και καλύτερες ιδέες, και συγκριτικά με την επιρροή των ακαδημαϊκών δημοσιεύσεων ο αριθμός των υποδείξεων σε αυτά ήταν πολύ μεγαλύτερος σε μεταγενέστερες έρευνες.

### **2.4.3 Υγεία**

Εφαρμογές όπως η Fitocracy και η QUENTIQ χρησιμοποιούν παιχνιδοποίηση για να ενθαρρύνουν τους χρήστες τους να εξασκούνται πιο αποτελεσματικά και να βελτιώνουν τη συνολική κατάστασή τους υγείας τους. Οι χρήστες επιβραβεύονται με πόντους για δραστηριότητες τις οποίες εκτελούν στη ρουτίνα τους και μέσω των πόντων ανεβαίνουν επίπεδο. Οι χρήστες επίσης μπορούν να ολοκληρώσουν αποστολές και να κερδίσουν παράσημα για τα αθλητικά κατορθώματά τους. Το παιχνίδι Health Month προσφέρει τη δυνατότητα στους χρήστες του να συμμετέχουν σε ένα κοινωνικό παιχνίδι στο οποίο οι πιο επιτυχημένοι παίκτες μπορούν να βοηθήσουν τους πιο αδύναμους να πετύχουν τους στόχους τους.

Μια ανασκόπηση των εφαρμογών υγείας των 100 πιο δημοφιλών στο "Apple app Store" έδειξε μία θετική συσχέτιση μεταξύ των παιχνιδοποιημένων παραμέτρων που χρησιμοποιήθηκαν και των υψηλών αξιολογήσεων των χρηστών. Η εφαρμογή «myfitnesspal» συμπεριλαμβάνει το μεγαλύτερο αριθμό παραμέτρων παιχνιδοποίησης.

### **2.4.4 Εργασία**

Η παιχνιδοποίηση έχει χρησιμοποιηθεί σε μια προσπάθεια βελτίωσης της παραγωγικότητας των εργαζομένων, τις ιατρικές παροχές τους, της οικονομικής υποστήριξής τους, της μεταφοράς τους, πρόσληψής τους και άλλα.



### 2.4.5 Εκπαίδευση

Η άντληση πληροφοριών από διάφορες πληθυσμιακές ομάδες(crowdsourcing) έχει παιχνιδιοποιηθεί σε παιχνίδια όπως το Foldit, ένα παιχνίδι σχεδιασμένο από το πανεπιστήμιο της Ουάσινγκτον ,στο οποίο οι παίκτες ανταγωνίζονται στο διαχωρισμό των πρωτεϊνών σε πιο αποτελεσματικές δομές. Μία δημοσίευση του 2010 στο επιστημονικό περιοδικό «Nature» πίστωσε τους 57.000 ενεργούς παίκτες του Foldit με χρήσιμα αποτελέσματα τα οποία ταίριαζαν ή ξεπερνούσαν αλγοριθμικές λύσεις υπολογιστών. Το ESP game το οποίο χρησιμοποιείται για να παράγει μεταδεδομένα εικόνας. Το Google image labeler είναι μία έκδοση του ESP game την οποία η Google έχει αδειοδοτήσει να παράγει τα δικά της μεταδεδομένα εικόνας. Έρευνα του πανεπιστημίου του Μπον χρησιμοποίησε τη παιχνιδιοποίηση για να αυξήσει τη συνεισφορά της Wikipedia κατά 62%.

Οι τομείς της εκπαίδευσης και της προπόνησης είναι περιοχές στις οποίες υπάρχει μεγάλο ενδιαφέρον για την παιχνιδιοποίηση. Η Microsoft κυκλοφόρησε το παιχνίδι "Ribbon Hero 2" σαν επιπρόσθετο στην εφαρμογή office με σκοπό να βοηθήσει τους ανθρώπους να το χρησιμοποιούν πιο αποτελεσματικά. Η Microsoft περιέγραψε αυτή την επέκταση ως ένα από τα πιο δημοφιλή project του Office που κυκλοφόρησε ποτέ. Το τμήμα εκπαίδευσης της Νέας Υόρκης(NYCDE) με χρηματοδότηση από τα ιδρύματα «McArthur» και «Bill & Melinda Gates» ίδρυσαν ένα σχολείο αποκαλούμενο «Quest to Learn» με επίκεντρο την παιχνιδιοποιημένη εκμάθηση, με απώτερο σκοπό να κάνουν τη διαδικασία της εκπαίδευσης πιο ενεργητική και ελκυστική για τα παιδιά σήμερα. Η SAP(γερμανική εταιρεία ανάπτυξης λογισμικού) έχει χρησιμοποιήσει παιχνίδια για να εκπαιδεύσει τους εργαζομένους του στην βιωσιμότητα του λογισμικού τους. Ο Αμερικανικός στρατός και η Unilever επίσης χρησιμοποίησαν την παιχνιδιοποίηση στη διαδικασία της εξάσκησης. Η Ακαδημία Khan είναι ένα παράδειγμα χρήσης τεχνικών παιχνιδιοποίησης στην εκπαίδευση από απόσταση μέσω διαδικτύου. Τον Αύγουστο του 2009 η Gbanga λάνσαρε το εκπαιδευτικό παιχνίδι Gbanga-ZooH για το ζωολογικό κήπο της Ζυρίχης το οποίο καλεί τους συμμετέχοντες να βοηθήσουν ενεργά τα ζώα που βρίσκονται σε κίνδυνο και να τα επιστρέψουν ασφαλή πίσω σε κάποιο ζωολογικό κήπο. Οι παίκτες διατηρούσαν εικονικούς οικότοπους όπου γύρω από το Canton της Ζυρίχης όπου για προσελκύσουν και να συλλέξουν ζώα που κινδυνεύουν. Το 2014 η εργασία True Life Game ξεκίνησε, με κύριο σκοπό τη συλλογή των καλύτερων τρόπων εφαρμογής εννοιών της παιχνιδιοποίησης και της άντλησης πληροφοριών από διάφορες πληθυσμιακές ομάδες(crowdsourcing) στη δια βίου μάθηση. Το 2015, το πανεπιστήμιο της Αριζόνα πρόσθεσε 5 νέα διαδραστικά παιχνίδια, βασισμένα σε ιστορίες, στο περιβαλλοντικό επιστημονικό του πρόγραμμα. Στα πλαίσια του παιχνιδιού, οι σπουδαστές αναλαμβάνουν ηγετικούς ρόλους και καλούνται να επιλύσουν πολύπλοκα προβλήματα σχετικά με τη βιωσιμότητα του περιβάλλοντος. Τέλος υπάρχουν ενδείξεις ότι η παιχνιδιοποίηση μπορεί να προσφέρει ιδιαίτερα κίνητρα σε μαθητές με δυσλεξία και άλλα προβλήματα.

### 3. Λειτουργικό σύστημα Android & το εργαλείο Android Studio

Σε αυτό το σημείο θα παρουσιάσουμε το λειτουργικό περιβάλλον Android στο οποίο βασίστηκε η εφαρμογή που υλοποιήσαμε. Στην συνέχεια θα παρουσιάσουμε το εργαλείο Android Studio[11,12,13].

#### 3.1 Λειτουργικό σύστημα Android

Το Android είναι ένα λειτουργικό σύστημα που ενσωματώνεται σε συσκευές κινητής τηλεφωνίας, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα kernel του λειτουργικού Linux και ακόμη, επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού της Google.

##### 3.1.1 Γενικά για το Android

Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.



Εικόνα 3.1 Το λογότυπο του Android [12]

Το λογότυπο του Android Συσκευές με Android υπάρχουν πλέον πάρα πολλές, η καθεμία με διαφορετικά χαρακτηριστικά και από διάφορες κατασκευάστριες εταιρίες όπως LG, Samsung, HTC, Sony, Motorola, είναι μερικές από τις εταιρίες που χρησιμοποιούν το λειτουργικό Android για τα smartphones τους. Το πολύ θετικό με τις συσκευές Android είναι ότι είναι αφενός multimedia συσκευές, δηλαδή σας δίνουν τη δυνατότητα να αναπαράγετε πολλαπλά μέσα και multitasking, δηλαδή σας δίνουν τη δυνατότητα εκτέλεσης πολλών εφαρμογών ταυτόχρονα, π.χ. ακούτε τραγούδια ενώ σερφάρετε στο ίντερνετ και ταυτόχρονα απαντάτε σε ένα SMS χωρίς να κλείσετε καμία εφαρμογή ή να χάσετε τη σελίδα που επισκεφτήκατε. Το web browsing στο Android είναι ταχύτατο, υποστηρίζεται από το Adobe Flash Player και υπάρχουν πολλοί browsers για να καλύψουν και τους πλέον απαιτητικούς. Ανεξάρτητα από το κόστος, όλες οι συσκευές Android διαθέτουν GPS και Wi-fi, δικαιώνοντας έτσι το βασικό λόγο δημιουργίας του εν λόγω λειτουργικού συστήματος που δεν είναι άλλος παρά η ανεμπόδιση και εύκολη πρόσβαση στο διαδίκτυο, σε συνδυασμό με ένα πλήθος εφαρμογών, όπως χάρτες, αναζήτηση, chat και e-mail, που πραγματικά επιτρέπουν στο χρήστη να μένει διαρκώς δικτυωμένος και ενημερωμένος. Βασικό χαρακτηριστικό του Android, επίσης, είναι η πληθώρα εφαρμογών που υπάρχουν στο Play Store δωρεάν και μη, όπου μπορεί ο χρήστης να

κατεβάσει. Ακόμη, το Android σας δίνει τη δυνατότητα να προσθέσετε widgets, δηλαδή 12 εικονίδια για την ταχύτερη πρόσβαση στα προγράμματα, τα οποία τοποθετούνται στη home screen του κινητού. Επιπλέον η notification bar είναι εξαιρετικά χρήσιμη, καθώς με ένα απλό drag - κατέβασμα βλέπετε όλες τις ειδοποιήσεις για τη συσκευή σας, μπορείτε να ανοιγοκλείσετε απευθείας τις υπηρεσίες χωρίς να χρειάζεται να πάτε στις ρυθμίσεις ή να προσθέσετε και άλλες λειτουργίες αναλόγως την συσκευή σας. Όσον αφορά το hardware οι διπύρνηνοι, τετραπύρνηνοι ή και οκταπύρνηνοι πλέον επεξεργαστές και οι διακεκριμένες GPU είναι πλέον γεγονός κάνοντας την εμπειρία χρήσης όλο και καλύτερη.

### 3.1.2 Ιστορική αναδρομή

Η **Android Inc.** ιδρύθηκε από τέσσερα άτομα, τους Andy Rubin, Rich Miner, Nick Sears και Chris White, ενώ η ονομασία προήλθε από το παρατσούκλι του Rubin, που ήταν βέβαια "Android", λόγω της εμμονής του με τα ρομπότ. Αρχικά το Android στόχευε να γίνει μια πλατφόρμα για ψηφιακές κάμερες, των οποίων η αγορά τότε έδειχνε πως θα ανθίσει σύντομα, όπως και έγινε. Ωστόσο, πολύ σύντομα η προσοχή των ιδρυτών στράφηκε προς τα smartphones, που τότε κατείχαν πολύ μικρό μερίδιο της αγοράς τηλεφώνων, θεωρώντας πως δεν τους αρκούσε η αγορά των ψηφιακών μηχανών.

Αποδείχθηκε σοφή επιλογή, αφού το 2005 η Google ενδιαφέρθηκε και αγόρασε την startup. Σοφή ήταν η επιλογή και από μεριάς της Google, καθώς μέσω του Android ενισχύθηκε σημαντικά η κυριαρχία της σε κάθε επιχείρηση και σε κάθε σπίτι, πάνω σε κάθε χρήστη. Ωστόσο, από το 2005 πέρασαν 2 χρόνια για να αρχίσει η ανάπτυξη του λειτουργικού και ακόμη 1 χρόνος για να κυκλοφορήσει το Android 1.0 OS, το Σεπτέμβριο του 2008. Το **T-Mobile G1** ήταν το πρώτο Android smartphone, το οποίο πωλείτο και ως **HTC Dream** σε πολλές αγορές.



Εικόνα 3.2. Το πρώτο Android smartphone T-Mobile G1 [12]

Το 2009 εμφανίστηκε το **Motorola Droid** που πλασαρίστηκε στην αγορά σαν εναλλακτική λύση απέναντι στο iPhone, πουλώντας μισό εκατομμύριο συσκευές στον πρώτο μήνα κυκλοφορίας.



Εικόνα 3.3 Motorola Droid [12]

Στις 28 Αύγουστου 2008 η Google ανακοινώσε το Android Market ( όπου από 6 Μάρτιου 2012 ονομάζετε Google Play ) και σταδιακά άρχισε να υποστηρίζει εφαρμογές επί πληρωμή σε Οι Android συσκευές άρχιζαν να γίνονται γνωστές και να αρέσουν στο κοινό, με αποτέλεσμα όλο και περισσότεροι διαφορεές χώρες ξεχωριστά. κατασκευαστές να στραφούν προς την πλατφόρμα της Google.



Εικόνα 3.4 Android Market [12]

### 3.1.3 Εκδόσεις Android

Όπως αναφέραμε παραπάνω το Σεπτέμβριο του 2008, τρία δηλαδή χρόνια μετά την πρώτη του παρουσίαση, κυκλοφόρησε το Android **1.0 OS** με το T-Mobile G1 που ήταν το πρώτο Android smartphone.

Πρώτο μεγάλο update ήταν η έκδοση **1.5 “Cupcake”**, τον Απρίλιο του 2009, φέρνοντας τα widgets και τους φακέλους στην homescreen. Και η έκδοση **1.6 “Donut”** δεν άργησε να την αντικαταστήσει, μόλις 5 μήνες αργότερα, υποστηρίζοντας οθόνες μεγαλύτερων αναλύσεων και με αρκετές βελτιώσεις. Βελτιώσεις που συνεχίστηκαν με ραγδαίο ρυθμό μέσω της επόμενης έκδοσης, με κωδική ονομασία **“Eclair”**, ένα μήνα αργότερα! Το Android προσπαθούσε να ωριμάσει γρήγορα για να προλάβει τις εξελίξεις, στην πραγματικότητα όμως, το ίδιο δημιουργούσε ταυτόχρονα αυτές τις εξελίξεις.



Εικόνα 3.5 Android 1.5 Cupcake



Εικόνα 3.6 Widgets στην homescreen



Εικόνα 3.7 Android 1.6 Donat [12]

Μέχρι που φτάνουμε στο καιρίο έτος 2010, το οποίος σημάδεψαν δύο μεγάλα updates **2.2 “Froyo”** τον Ιούνιο και **2.3 “Gingerbread”** στην εκπονή του χρόνου, το Δεκέμβριο και η κυκλοφορία του **Google Nexus One**. Η συσκευή, αν και δεν αποτέλεσε εμπορική επιτυχία, έδειξε το δρόμο για το μέλλον και προσέφερε πολύτιμη εμπειρία στους κατασκευαστές.



Εικόνα 3.8 Android 2.2 Froyo



Εικόνα 3.9 Google Nexus One





Εικόνα 3.10 Android 2.3 Gingerbread [12]

Απόδειξη ήταν η αύξηση του μεριδίου της αγοράς του Android στις ΗΠΑ, αφού ξεπέρασε μέσα στο 2010 το **iOS**, χάρη κυρίως στο **πρώτο Galaxy S** της Samsung και το **Evo 4G** της HTC.



Εικόνα 3.11 Διαμάχη μεταξύ Android και iOS [12]

Από εκεί και πέρα, η γιγάντωση και η εξάπλωση του Android γίνονται με τρομακτικούς ρυθμούς. Το Φεβρουάριο του 2011 κάνει την εμφάνισή της η **έκδοση 3.0 “Honeycomb”**, με κύριο στόχο τα ανερχόμενα **tablets**, η οποία γρήγορα όμως έδωσε τη θέση της στην πολύ επιτυχημένη και δημοφιλή έκδοση **4.0**, γνωστή και ως **“Ice Cream Sandwich”**.



Εικόνα 3.12 Android 3.0 HoneyComb



Εικόνα 3.13 Android 4.0 Ice Cream Sandwich [12]

Βασιζόμενη πάνω στην Honeycomb αλλά με βελτιώσεις στα σημεία, κυριάρχησε εύκολα για ολόκληρο τον επόμενο χρόνο. Έτσι, η **4.1 “Jelly Bean”** αναπτύχθηκε πιο άνετα και αβίαστα, για να κυκλοφορήσει παράλληλα με το best

seller της Samsung, το **Galaxy S III** και μετά από λίγο ακολούθησε και η **4.2** με την ίδια ονομασία. Προηγουμένως το Android είχε ξεπεράσει και τη **Blackberry** σε μερίδιο αγοράς στις ΗΠΑ, δείχνοντας πως ήρθε για να μείνει. Και τώρα έχουμε φτάσει στην έκδοση **4.4 “Kit Kat”** που διατηρεί σταθερά στην κορυφή το πολυαγαπημένο λειτουργικό.



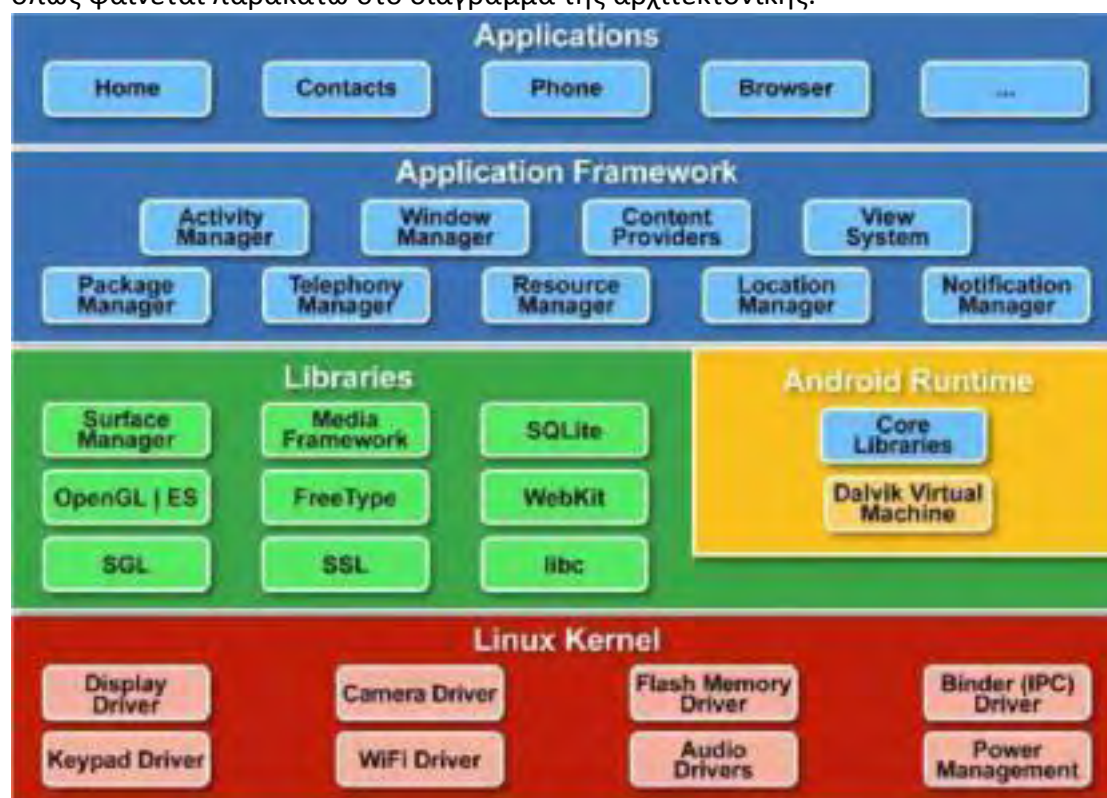
Εικόνα 3.14 Android 4.2 Jelly bean



Εικόνα 3.15 Android 4.4 Kit Kat

### 3.1.4 Αρχιτεκτονική Android

Το λειτουργικό σύστημα Android είναι μια στοίβα στοιχείων λογισμικού που σε γενικές γραμμές χωρίζονται σε πέντε ενότητες και τέσσερα κύρια στρώματα, όπως φαίνεται παρακάτω στο διάγραμμα της αρχιτεκτονικής.



Εικόνα 3.16 Διάγραμμα αρχιτεκτονικής του Android [11]

Ο **πυρήνας (kernel) Linux** είναι το χαμηλότερο επίπεδο του συστήματος και είναι υπεύθυνο για την αλληλεπίδραση με το hardware. Επίσης προσφέρει την βασική διεργασία στο σύστημα όπως επεξεργασία δεδομένων, διαχείριση της μνήμης και διαχείριση των συσκευών – περιφερειακών όπως την όπως την κάμερα, το πληκτρολόγιο, την οθόνη κλπ. Τέλος ο πυρήνας των Linux ασχολείται και με την

διαχείριση σε ότι σχέση με το δίκτυο καθώς προσφέρει και όλους τους απαραίτητους drivers που απαιτούνται.

Οι **βιβλιοθήκες (libraries)** του συστήματος βρίσκονται όπως φαίνεται στο σχεδιάγραμμα πάνω από τον πυρήνα. Εκεί περιλαμβάνονται και οι πολύ γνωστές libc, το WebKit και το SQLite Database που χρησιμοποιείται για την αποθήκευση και διαμοιρασμό δεδομένων της εφαρμογής. Συνήθως είναι προγραμματισμένες σε γλώσσα C/C++ και η πρόσβαση σε αυτές δεν είναι δυνατή καθώς χρησιμοποιούνται αυτές του ανώτερου επιπέδου.

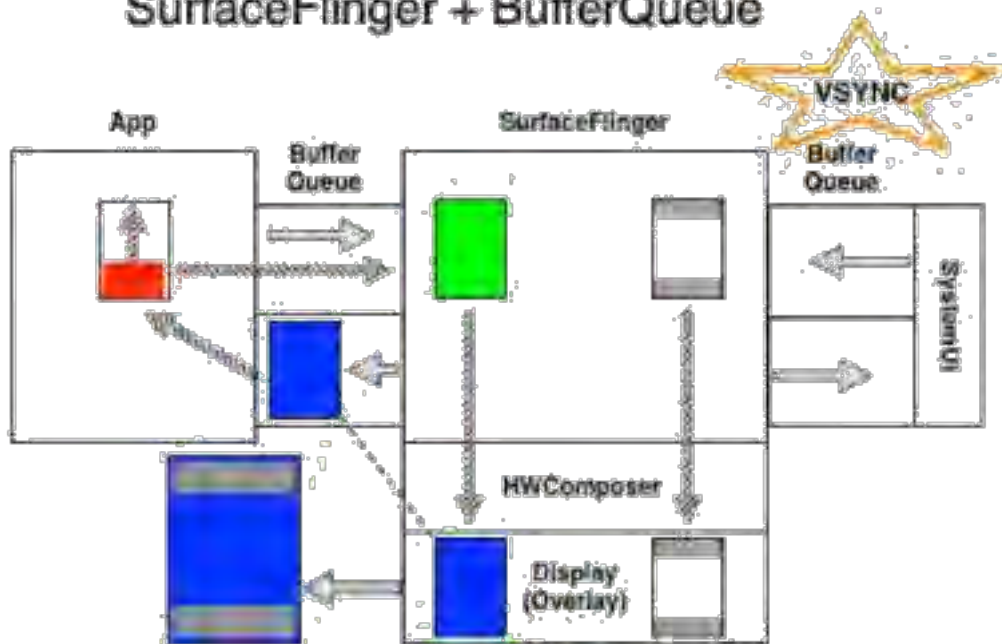
Τρίτο στη σειρά είναι το **Android Runtime** που περιλαμβάνει ένα πολύ σημαντικό στοιχείο το **Dalvik Virtual Machine** που είναι σαν ένα είδος Java εικονικής μηχανής ειδικά σχεδιασμένη και βελτιστοποιημένη για Android. Το τελευταίο χρησιμοποιεί τα χαρακτηριστικά του πυρήνα του Linux όπως την διαχείριση της μνήμης και το multi-threading που είναι εγγενή για την γλώσσα προγραμματισμού Java. Ακόμη αυτό που το κάνει ξεχωριστό είναι ο σχεδιασμός του, που δίνει την δυνατότητα σε κάθε εφαρμογή να τρέχει σε δικό της εικονικό μηχανήμα. Έτσι βελτιώνεται και η ασφάλεια και η ταχύτητα στο λειτουργικό.

Το **πλαίσιο εφαρμογών (Application Framework)** περιέχει ένα πλαίσιο ανάπτυξης που τυποποιεί και διευκολύνει τον προγραμματισμό των Android εφαρμογών. Αυτές οι βιβλιοθήκες είναι γραμμένες σε Java και μπορούν να χρησιμοποιηθούν από τους developers. Κύριο στοιχείο του είναι η επαναχρησιμοποίηση των αρχείων κάθε εφαρμογής καθώς μπορούν να δημοσιεύσουν τις δυνατότητές τους και να χρησιμοποιηθούν από κάποια άλλη στο μέλλον.

Οι **εφαρμογές (applications)** είναι το βασικό στοιχείο κάθε λειτουργικού και το Android από προεπιλογή διαθέτει μια γκάμα προεγκατεστημένων εφαρμογών όπως το Ημερολόγιο, Επαφές, Μηνύματα SMS κλπ. Μέσα από το Google Play Store και άλλα markets που κυκλοφορούν ο χρήστης όμως μπορεί να εγκαταστήσει όποια εφαρμογή θέλει.



## SurfaceFlinger + BufferQueue



Εικόνα 3.17 Λειτουργία του Android [11]

### 3.2 Το εργαλείο Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android.

#### 3.2.1 Σύνοψη επισκόπηση του Android Studio

Το Android Studio έχει ως βάση το IntelliJ IDEA. Το Android Studio είναι ελεύθερα διαθέσιμα υπό την άδεια χρήσης Apache 2.0.

#### Gradle

Το Android Studio χρησιμοποιεί Gradle (ένα λογισμικό ανοικτού κώδικα που χρησιμοποιείται για την αυτοματοποίηση της διαδικασίας “χτισίματος” λογισμικού) ως το θεμέλιο του συστήματος κατασκευής, με περισσότερα δυνατότητες για Android που παρέχονται από το Android Plugin για Gradle. Αυτό το σύστημα κατασκευής “τρέχει” ως ένα ολοκληρωμένο εργαλείο από το μενού του Android Studio, και ανεξάρτητα από τη γραμμή εντολών. Κάποια από τα πλεονεκτήματα που προσφέρει το συγκεκριμένο σύστημα κατασκευής (build system) είναι ότι ο χρήστης μπορεί να προσαρμόσει, να ρυθμίσει και να επεκτείνει τη διαδικασία κατασκευής (το build process της εφαρμογής), να δημιουργήσει πολλαπλά APKs για την εφαρμογή του με διαφορετικά χαρακτηριστικά χρησιμοποιώντας στην ουσία το ίδιο project, και τέλος να επαναχρησιμοποιήσει εύκολα τον κώδικα και τους πόρους της εφαρμογής του.

Τα αρχεία του build της εφαρμογής ονομάζονται build.gradle. Είναι απλά αρχεία κειμένου που χρησιμοποιούν το συντακτικό της γλώσσας Groovy για να διαμορφώσουν τη διαδικασία του build με στοιχεία που παρέχονται από το Android plugin για Gradle. Κάθε project στο Android Studio έχει ένα topLevel αρχείο build,

για όλο το project, έχει και ξεχωριστά modulelevel αρχεία build αρχεία για το κάθε module του project. Όταν ο χρήστης κάνει import κάποιο project το Android Studio παράγει αυτόματα τα απαραίτητα αρχεία build.

## Modules

Ένα module είναι μια συλλογή αρχείων πηγαίου κώδικα της εφαρμογής και ρυθμίσεις για το build process της εφαρμογής. Τα modules επιτρέπουν στο προγραμματιστή να χωρίσει το project του σε διακριτά κομμάτια το καθένα από τα οποία παρέχει συγκεκριμένη λειτουργία και όλα μαζί συνδυασμένα καθορίζουν τη συνολική συμπεριφορά της εφαρμογής. Ένα project μπορεί να έχει ένα ή πολλά modules, και κάποιο module μπορεί να χρησιμοποιήσει άλλα modules, και να δηλωθεί ως εξαρτημένο από αυτά. Κάθε module μπορεί να κατασκευαστεί, να δοκιμαστεί ή να αποσφαλματωθεί ως ανεξάρτητη οντότητα.

Τα modules είναι ιδιαίτερα χρήσιμα όταν ο προγραμματιστής θέλει να δημιουργήσει δικές του βιβλιοθήκες στο project που της χρησιμοποιεί η εφαρμογή, όπως επίσης και όταν θέλει να δημιουργήσει διαφορετικές υλοποιήσεις (διαφορετικά κομμάτια κώδικα και resources) για διαφορετικούς τύπους συσκευών(κινητά τηλέφωνα, ρολόγια android κ.α.) και να κρατήσει όλα τα αρχεία αυτά μαζί, στα πλαίσια ενός ενιαίου project και όχι πολλών, ώστε να μην χρειαστεί να επαναλάβει κομμάτια κώδικα ή resources που χρησιμοποιούνται από περισσότερες από μια υλοποιήσεις.

Ο χρήστης μπορεί να προσθέσει modules στο Android Studio επιλέγοντας File > New > New Module. Το Android Studio παρέχει αρκετά διαφορετικά είδη modules.

### Android app module

Αποτελεί ένα container για τον πηγαίο κώδικα της εφαρμογής,, τα αρχεία των πόρων, και τις ρυθμίσεις της εφαρμογής, όπως το αρχείο κατασκευής (build file) σε επίπεδο module και το αρχείο Android Manifest. Όταν δημιουργείτε ένα νέο project, το προεπιλεγμένο όνομα του module είναι "app". Στο παράθυρο Create New Module του Android Studio διατίθενται αρκετά Modules τα οποία παρέχουν απαραίτητα αρχεία και κάποια πρότυπα κώδικα που είναι κατάλληλα για την αντίστοιχη εφαρμογή ή τύπο συσκευής. Τα modules αυτά είναι: Phone & Table Module, Android Wear Module, Android TV Module, Glass Module.

### Library module

Αποτελεί ένα container για τον επαναχρησιμοποιήσιμο κώδικα, τον οποίο ο προγραμματιστής μπορεί να χρησιμοποιήσετε ως εξάρτηση(dependency) σε άλλα modules της εφαρμογής(δηλ. άλλα Modules να εξαρτώνται από αυτό για την λειτουργία τους) ή την εισαγωγή(import) του σε άλλα projects. Δομικά, το module της βιβλιοθήκης είναι το ίδιο με ένα app module, αλλά όταν "χτίζεται"(γίνεται build), δημιουργείται ένα συμπιεσμένο αρχείο κώδικα αντί ενός APK, με

αποτέλεσμα να μην γίνεται η εγκατάστασή του στη συσκευή ως αυτόνομη εφαρμογή.

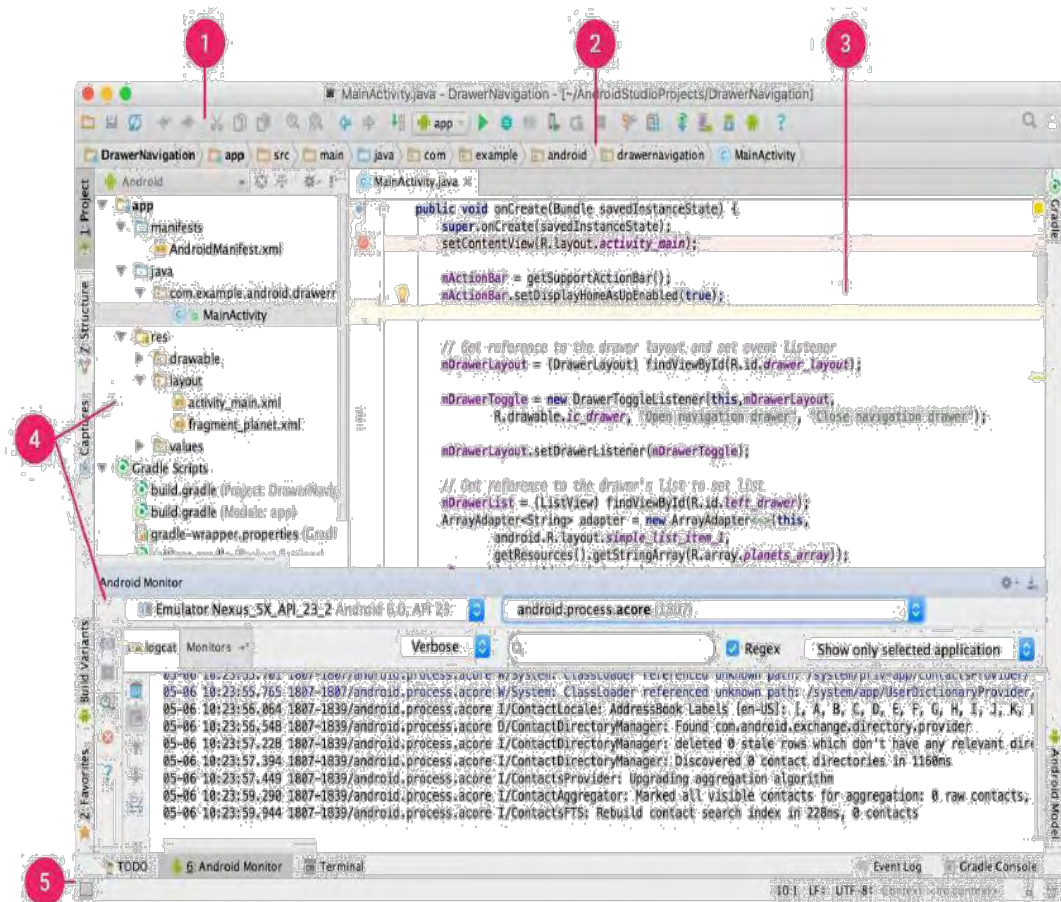
Στο παράθυρο Create New Module του Android Studio διατίθενται δύο library modules, Android Library και Java Library. Το Android Library module μπορεί να περιέχει όλα τα είδη αρχείων που υποστηρίζονται σε ένα Android Project συμπεριλαμβανομένου και του πηγαίου κώδικα, των πόρων, και του αρχείου android manifest. Το αποτέλεσμα του build process ενός τέτοιου module είναι ένα αρχείο Android Archive (AAR) το οποίο μπορεί να προστεθεί ως dependency(εξάρτηση) για άλλα app modules. Το Java Library module μπορεί να περιέχει μόνο πηγαίο κώδικα Java. Το αποτέλεσμα του build process είναι ένα αρχείο Java ARchive (JAR) το οποίο μπορεί να προστεθεί ως dependency(εξάρτηση) για άλλα app modules ή άλλα Java projects.

### **Google Cloud module**

Αποτελεί ένα container για το Google Cloud backend κώδικα της εφαρμογής. Το module αυτό προσθέτει τον απαιτούμενο κώδικα και εξαρτήσεις για ένα λειτουργικό, εκτελέσιμο backend που ο προγραμματιστής μπορεί να επεκτείνει με χαρακτηριστικά που επιθυμεί, όπως η αποθήκευση δεδομένων.

### **User Interface**

Όπως φαίνεται και στην παρακάτω εικόνα, το κεντρικό παράθυρο του Android Studio αποτελείται από τα παρακάτω επιμέρους λογικά τμήματα.



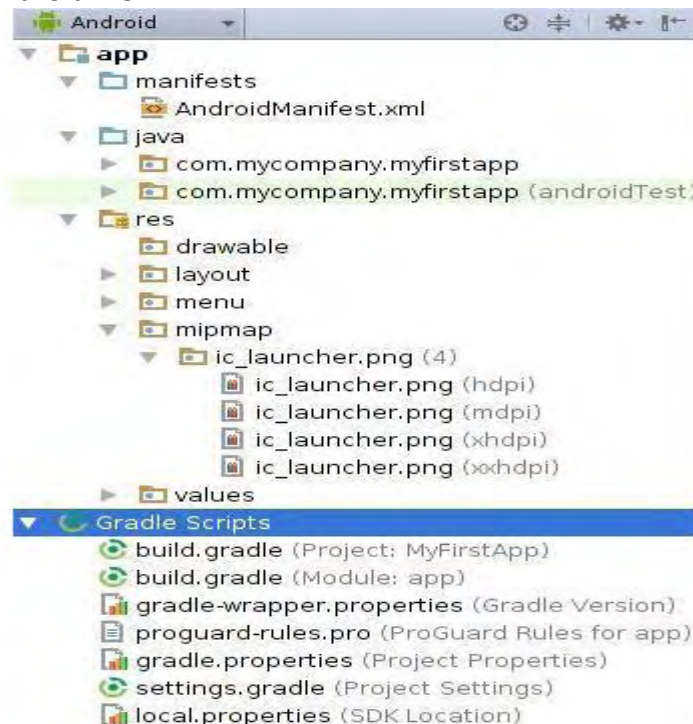
**Εικόνα 3.18 – Περιβάλλον Android Studio**

1. Η γραμμή εργαλείων επιτρέπει πληθώρα ενεργειών όπως το “τρέξιμο” της εφαρμογής και τη χρήση διάφορων εργαλείων android.
2. Η μπάρα πλοήγησης παρέχει εύκολη πρόσβαση σε όλα τα επιμέρους τμήματα του project, κάνοντας χρήση μιας ιεραρχικής δόμησης των φακέλων, όπως αυτά φαίνονται και στο παράθυρο του εργαλείου “Project”.
3. Στο παράθυρο επεξεργασίας(editor window) ο χρήστης μπορεί να δημιουργεί και να τροποποιεί κώδικα. Το παράθυρο αυτό αλλάζει, με βάση το είδος του αρχείου που έχει ανοίξει για επεξεργασία.
4. Τα παράθυρα εργαλείων (Tool windows) παρέχουν πρόσβαση σε συγκεκριμένες ενέργειες όπως είναι η διαχείριση του project, η αναζήτηση, ο έλεγχος έκδοσης (version control) και άλλα. Τα παράθυρα αυτά μπορούν να συμπυκνωθούν και να αναπτυχθούν.
5. Η γραμμή κατάστασης(status bar) εμφανίζει την κατάσταση του project καθώς και του ίδιου του IDE και τυχόν προειδοποιήσεων και μηνυμάτων.

Ο χρήστης έχει τη δυνατότητα να οργανώσει όπως θέλει την εμφάνιση του κεντρικού παραθύρου του Android Studio κρύβοντας ή μετακινώντας γραμμές

εργαλείων και παράθυρα. Μπορεί επίσης να χρησιμοποιήσει πολλές συντομεύσεις πληκτρολογίου για άμεση πρόσβαση σε πολλά χαρακτηριστικά του IDE. Ο χρήστης μπορεί ανά πάσα στιγμή, να πραγματοποιήσει αναζήτηση σε όλο τον πηγαίο κώδικα, τις βάσεις δεδομένων, τα στοιχεία της διεπαφής χρήστη(user interface), και ούτω καθεξής, με διπλό πάτημα του πλήκτρου Shift, ή κάνοντας κλικ στο μεγεθυντικό φακό στην επάνω δεξιά γωνία του παραθύρου του Android Studio. Η δομή των αρχείων ενός project στο Android Studio και η παρουσίασή τους στο User Interface

### Android view



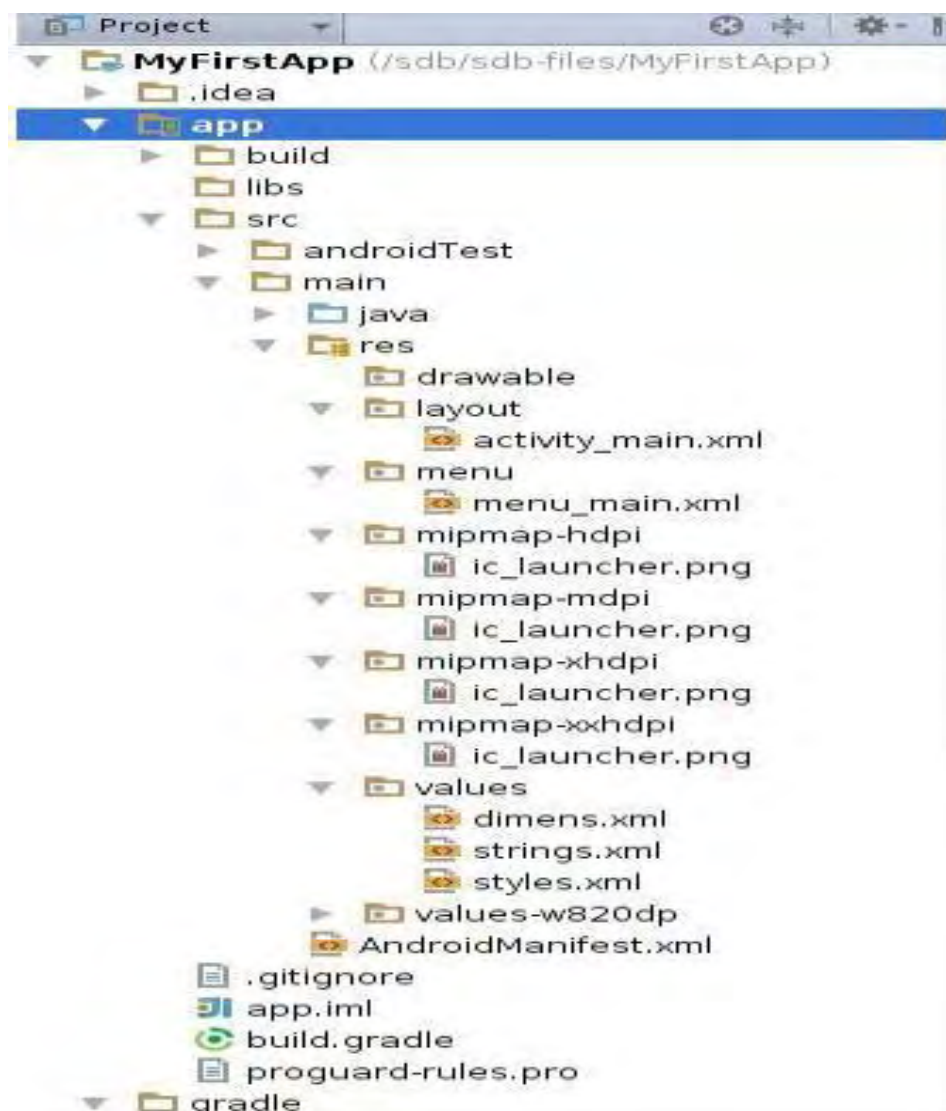
Εικόνα 3.19 – Android view

Από προεπιλογή το Android Studio εμφανίζει παρουσιάζει τα αρχεία του project σε Android view. Αυτό δεν αντικατοπτρίζει την πραγματική ιεραρχία των φακέλων στο σκληρό δίσκο, αλλά είναι οργανωμένο με βάση τα επιμέρους modules και τους διαφορετικούς τύπους αρχείων με σκοπό την απλοποίηση της πλοήγησης του χρήστη μεταξύ σημαντικών αρχείων κλειδιών του project, κρύβοντας ορισμένα αρχεία ή φακέλους που συνήθως δεν χρησιμοποιούνται. Ο χρήστης, έχοντας επιλέξει το Android View, μπορεί να βρει τα αρχεία ρυθμίσεων για τη διαδικασία “χτισίματος”(build) της εφαρμογής, τα manifest αρχεία για κάθε module, όπως και όλα τα εναλλακτικά resources με βάση το είδος τους, ομαδοποιημένα στο ίδιο μέρος και όχι σε διαφορετικούς φακέλους όπως αυτά βρίσκονται πραγματικά στον σκληρό δίσκο.

Μέσα σε κάθε Android app module, τα αρχεία βρίσκονται ομαδοποιημένα στις ομάδες “manifests”, “java” και “res”. Η ομάδα “manifests” περιέχει το αρχείο AndroidManifest.xml, η ομάδα “java” περιέχει τα αρχεία πηγαίου κώδικα Java, διαχωρισμένα με βάση το package name τους και η ομάδα “res” περιέχει όλους

τους πόρους που δεν έχουν μορφή κώδικα, όπως είναι τα XML layouts, User interface strings, εικόνες bitmap κ.α., διαχωρισμένους σε κατάλληλες υποομάδες.

### Project view



Εικόνα 3.20 – Project view

Για να δει ο χρήστης την πραγματική δομή των αρχείων του project, όπως αυτά βρίσκονται στο σκληρό, συμπεριλαμβανομένων και κάποιων αρχείων που αποκρύπτονται στο Android view, πρέπει να επιλέξει το "Project" στο παράθυρο του



project, εκεί που ήταν προεπιλεγμένο το “Android”(είχαμε δηλαδή το Android view ως προεπιλογή). Με την επιλογή του Project view ο χρήστης έχει τη δυνατότητα να δει πολλά περισσότερα αρχεία και φακέλους, από τα οποία τα πιο σημαντικά είναι:

#### **modulename/**

##### **build/**

Περιέχει διαφορετικά build outputs.

##### **libs/**

Περιέχει ιδιωτικές βιβλιοθήκες της εφαρμογής.

##### **src/**

Περιέχει όλα τα αρχεία κώδικα και πόρων για το συγκεκριμένο module στους εξής υποφακέλους:

##### **androidTest/**

Περιέχει κώδικα για δοκιμές της εφαρμογής σε μια συσκευή Android.

##### **main/**

Περιέχει τα “κύρια” αρχεία: τον κώδικα και τους πόρους που είναι κοινά για όλα τα διαφορετικά builds.

##### **AndroidManifest.xml**

Περιγράφει τη φύση της εφαρμογής και κάθε ένα από τα συστατικά της μέρη.

##### **java/**

Περιέχει τα αρχεία πηγαίου κώδικα Java.

##### **jni/**

Περιέχει εγγενή κώδικα χρησιμοποιώντας το Java Native Interface (JNI).

##### **gen/**

Περιέχει τα αρχεία Java που δημιουργούνται από το Android Studio, όπως το αρχείο R.java.

##### **res/**

Περιέχει τους πόρους της εφαρμογής όπως τα drawable αρχεία, τα layout αρχεία, τα User Interface strings (κείμενα) κ. α.

#### **assets/**

Περιέχει αρχεία τα οποία θα πρέπει να μεταγλωττίσουν σε αρχείο .apk όπως είναι, χωρίς να υποστούν αλλαγές.

#### **test/**

Περιέχει κώδικα για τις τοπικές δοκιμές που εκτελούνται σε τοπική JVM.

#### **build.gradle (module)**

Αυτό καθορίζει το build configuration (διαμόρφωση κατασκευής) για το συγκεκριμένο module.

#### **build.gradle (project)**

Αυτό καθορίζει το build configuration (διαμόρφωση κατασκευής) για όλα τα modules. Αυτό το αρχείο είναι αναπόσπαστο μέρος του project και θα πρέπει να ελέγχεται συχνά για τυχόν αναθεωρήσεις όπως και όλος ο πηγαίος κώδικας της εφαρμογής.

#### **Project Structure Settings**

Ο χρήστης μπορεί να έχει πρόσβαση σε σημαντικές ρυθμίσεις του project στο File > Project Structure. Το Project Structure περιέχει τους εξής τομείς:

- **SDK Location:** Θέτει την τοποθεσία του σκληρού δίσκου, όπου βρίσκονται το JDK, το Android SDK, και το Android NDK που χρησιμοποιούνται από το project.
- **Project:** Ορίζει την έκδοση του Gradle, το plugin του Android για Gradle, και το όνομα της τοποθεσίας αποθήκευσης.
- **Developer Services:** Περιέχει ρυθμίσεις για τα Android Studio addin components.
- **Modules:** Επιτρέπει στο χρήστη να επεξεργαστεί τα build configurations σε επίπεδο module, όπως είναι οι εξαρτήσεις(dependencies), το target και το minimum SDK, και το signature της εφαρμογής.



Το τμήμα Developer Services του Project Structure περιέχει σελίδες ρυθμίσεων(configuration pages) για αρκετές υπηρεσίες που ο χρήστης μπορεί να χρησιμοποιήσει στην εφαρμογή του. Αυτές είναι:

- AdMob: Σας επιτρέπει να ενεργοποιήσετε το AdMob component της Google, το οποίο βοηθά τον προγραμματιστή να “κατανοήσει” τους χρήστες της εφαρμογής ώστε να τους προβάλει προσαρμοσμένες διαφημίσεις.
- Analytics: Επιτρέπει την εκκίνηση του Google Analytics το οποίο μετρά τις αλληλεπιδράσεις των χρηστών με την εφαρμογή σε διάφορες συσκευές.
- Authentication: Επιτρέπει στους χρήστες της εφαρμογής να κάνουν sign in στην εφαρμογή χρησιμοποιώντας τον Google λογαριασμό τους.
- Cloud: Επιτρέπει την ενεργοποίηση των cloudbased υπηρεσιών Firebase για την εφαρμογή.
- Notifications: Επιτρέπει τη χρήση Google Cloud Messaging για την επικοινωνία μεταξύ της εφαρμογής και του διακομιστή της.

Ενεργοποιώντας οποιαδήποτε από αυτές τις υπηρεσίες μπορεί να προκαλέσει το Android Studio να βάλει επιπρόσθετες εξαρτήσεις και να δώσει επιπλέον δικαιώματα στην εφαρμογή. Κάθε μια από τις παραπάνω σελίδες ρυθμίσεων(configuration pages) παραθέτει όλες τις εξαρτήσεις και τα δικαιώματα αυτά, καθώς και άλλες ενέργειες που το Android Studio θα πραγματοποιήσει αυτόματα, εφόσον ο προγραμματιστής θα επιλέξει να ενεργοποιήσει τη σχετική υπηρεσία.

Το τμήμα Modules του Project Structure επιτρέπει την αλλαγή των ρυθμίσεων για κάθε ένα από τα modules του project μας. Κάθε μια από τις σελίδες ρυθμίσεων του τμήματος Modules αποτελείται από τις παρακάτω καρτέλες ρυθμίσεων(tabs):

- Properties: Καθορίζει τις εκδόσεις του SDK και τα build tools που πρέπει να χρησιμοποιηθούν για την μεταγλώττιση του συγκεκριμένου module.
- Signing: Καθορίζει το πιστοποιητικό που πρέπει να χρησιμοποιηθεί ως υπογραφή του APK της εφαρμογής μας.
- Flavors: Επιτρέπει τη δημιουργία πολλαπλών build flavors(“γεύσεις κατασκευής”), όπου κάθε flavor καθορίζει μια σειρά από ρυθμίσεις, όπως το ελάχιστο SDK και το SDKστόχου module, τον κωδικό έκδοσης και το όνομα έκδοσης. Για παράδειγμα, ο προγραμματιστής μπορεί να ορίσει ένα flavor που έχει ελάχιστο SDK το 15 και SDKστόχο το 21, και ένα δεύτερο flavor που έχει ένα ελάχιστο SDK το 19 και SDKστόχο το 23.

- **Build Types:** Επιτρέπει τη δημιουργία και επεξεργασία διάφορων build configurations. Κάθε module έχει από προεπιλογή δύο build types, debug και release. Ο προγραμματιστής μπορεί να ορίσει περισσότερα build types με βάση τις ανάγκες του.
- **Dependencies:** Εδώ παραθέτονται οι εξαρτήσεις του συγκεκριμένου module από άλλα modules, βιβλιοθήκες και αρχεία. Ο προγραμματιστής μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει εξαρτήσεις από αυτό το πλαίσιο.

#### 4. Η εφαρμογή Queasy

Αφού μελετήσαμε την υπάρχουσα κατάσταση στον τομέα του εκπαιδευτικού λογισμικού και θέλοντας να εισάγουμε σε αυτό κάποιες βασικές αρχές της παιχνιδοποίησης οι οποίες απουσιάζουν από τις περισσότερες εφαρμογές, αποφασίσαμε την υλοποίηση μιας εκπαιδευτικής εφαρμογής σε μορφή κουίζ. Το όνομα αυτής της εφαρμογής είναι *Queasy* και προέρχεται ετυμολογικά από τα συνθετικά Quiz+easy, δηλαδή το εύκολο κουίζ. Στην παρακάτω εικόνα 4.1 φαίνεται το λογότυπο της εφαρμογής.



Εικόνα 4.1 *Queasy* Logo

#### 4.1 Το Queasy ως εκπαιδευτικό λογισμικό

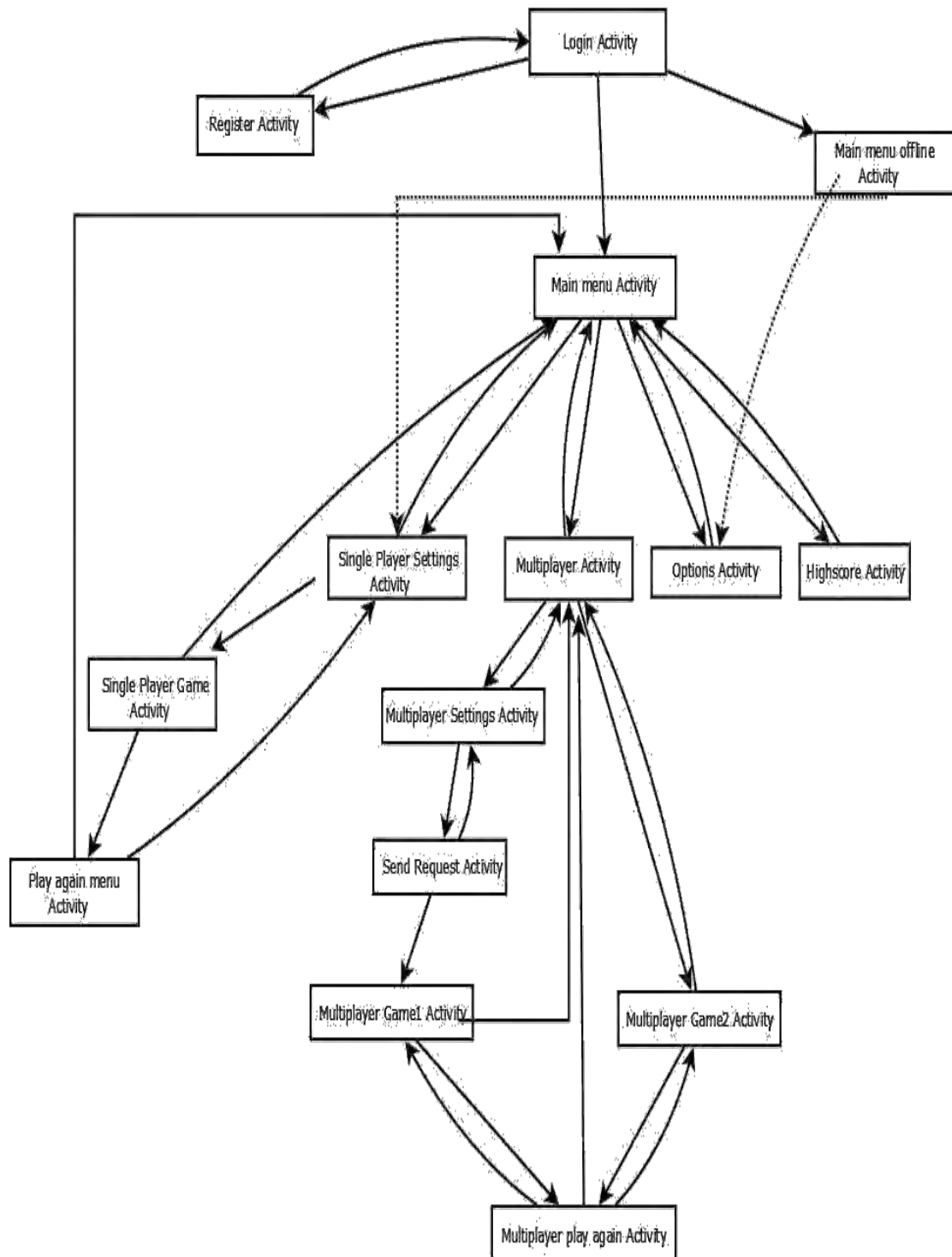
Το Queasy είναι μία εφαρμογή-παιχνίδι κουίζ που απευθύνεται σε παιδιά δημοτικού. Το παιχνίδι προσφέρει στο χρήστη-παιδί τη δυνατότητα να παίξει μόνος του(Single Player) όσο και να συναγωνιστεί με κάποιον φίλο του, συμμαθητή του, τους γονείς του ή και κάποιον άγνωστο(Multiplayer). Το αντικείμενο των ερωτήσεων έχει εμπνευστεί και αντληθεί από τα κύρια μαθήματα που διδάσκεται ένα παιδί στο δημοτικό σχολείο. Περιλαμβάνει τις εξής κατηγορίες: Λέξεις(Γλώσσα), Αριθμοί(Μαθηματικά), Ιστορία, Κόσμος(Μελέτη Περιβάλλοντος, Φυσική, Γεωγραφία, Γενικές). Οι ερωτήσεις προέρχονται από τα σχολικά βιβλία. Το παιχνίδι προσφέρει στο χρήστη 3 διαφορετικά επίπεδα δυσκολίας. Το Εύκολο επίπεδο απευθύνεται σε ερωτήσεις των Α' και Β' τάξεων δημοτικού. Το Μέτριο επίπεδο απευθύνεται σε ερωτήσεις των Γ' και Δ' τάξεων δημοτικού. Τέλος, το Δύσκολο επίπεδο απευθύνεται σε ερωτήσεις των Ε' και ΣΤ' τάξεων δημοτικού. Επίσης η εφαρμογή δίνει στο χρήστη τη δυνατότητα να επιλέξει τον αριθμό των ερωτήσεων στον οποίο θα απαντήσει(10,15 ή 20 ερωτήσεις), καθώς και τον διαθέσιμο χρόνο που θα έχει στη διάθεση του να απαντήσει(10, 15 ή 20 δευτερόλεπτα). Κάθε χρήστης δημιουργεί ένα προσωπικό λογαριασμό και αναλόγως την επιτυχία και τους πόντους που θα συγκεντρώσει κάθε φορά που θα παίξει κάποιο παιχνίδι κερδίζει φήμη(ανεβαίνει επίπεδο), μπορεί να κερδίσει μετάλλια και να σκαρφαλώσει όσο πιο ψηλά γίνεται στην συνολική κατάταξη. Και τα 4 αυτά στοιχεία που προαναφέρθηκαν(πόντοι-points, μετάλλια-badges, επίπεδο-level, κατάταξη-ranking,leaderboard) αποτελούν βασικές αρχές στις οποίες στηρίζεται η παιχνιδοποίηση-gamification, όπως αναφέραμε αναλυτικά και στο κεφάλαιο 2. Καθώς η εφαρμογή απευθύνεται σε παιδιά δημοτικού, όλα τα μενού και τα κουμπιά είναι γραμμένα στην ελληνική γλώσσα. Τα χρώματα που έχουν χρησιμοποιηθεί στο γραφικό κομμάτι της εφαρμογής είναι όλα απαλά και διακριτικά και το φόντο του γραφικού περιβάλλοντος έχει χαρακτηριστικά κινουμένων σχεδίων. Με αυτό το τον τρόπο η εφαρμογή γίνεται πιο φιλική και εύχρηστη προς τα παιδιά.

#### 4.2 Υλοποίηση της εφαρμογής Queasy

Όπως αναφέραμε και στο κεφ.3 το βασικό εργαλείο που χρησιμοποιήσαμε για να υλοποιήσουμε το «Queasy» είναι το *Android Studio 2.0*. Όσον αφορά το γραφικό κομμάτι, η γλώσσα προγραμματισμού που χρησιμοποιήσαμε είναι η xml, μία δυνατότητα που προσφέρει το *Android Studio*. Όσον αφορά τις υπόλοιπες λειτουργίες της εφαρμογής και το αλγοριθμικό κομμάτι, η γλώσσα προγραμματισμού που χρησιμοποιήσαμε είναι η Java. Εκτός από το περιβάλλον του *Android Studio*, για την υλοποίηση της εφαρμογής χρειάστηκε να χρησιμοποιήσουμε ένα server([www.queasyga.com](http://www.queasyga.com))[16] πάνω στον οποίο χειριζόμαστε τη δική μας βάση δεδομένων(μέσω του php\_my\_admin). Για να επιτύχουμε την επικοινωνία(ανάγνωση, εγγραφή κτλ.) μεταξύ του κώδικα Java, στον οποίο είναι γραμμένη η εφαρμογή, και της βάσης δεδομένων, χρησιμοποιήσαμε κάποιες βιβλιοθήκες(JSON, Volley), μία σειρά από php script(χρησιμοποιήσαμε το εργαλείο Eclipse), καθώς και κώδικα SQL ο οποίος εκτελείται στα php αρχεία[14,15].

### 4.2.1 Δομή της εφαρμογής Queasy

Σε αυτό το σημείο θα παρουσιάσουμε αναλυτικά, ξεκινώντας από το αρχικό παράθυρο, τη ροή εκτέλεσης της εφαρμογής. Κάθε εφαρμογή Android, όπως αναφέραμε και πιο πάνω, αποτελείται από το γραφικό κομμάτι το οποίο είναι γραμμένο σε κώδικα Xml και το αλγοριθμικό κομμάτι το οποίο είναι γραμμένο σε κώδικα Java. Ο προγραμματιστικός όρος για τον συνδυασμό αυτών των δύο ονομάζεται Activity, και στην ουσία πρόκειται για ένα παράθυρο που εμφανίζεται στην συσκευή του κινητού μας τηλεφώνου. Στο διάγραμμα ροής που ακολουθεί(εικόνα 4.2) φαίνεται αναλυτικά η ροή εκτέλεσης από το ένα Activity στο επόμενο για όλη τη διάρκεια ζωής της εφαρμογής, ξεκινώντας από το Login Activity, το οποίο είναι το πρώτο Activity που εκτελείται όταν ο χρήστης εκκινεί την εφαρμογή.

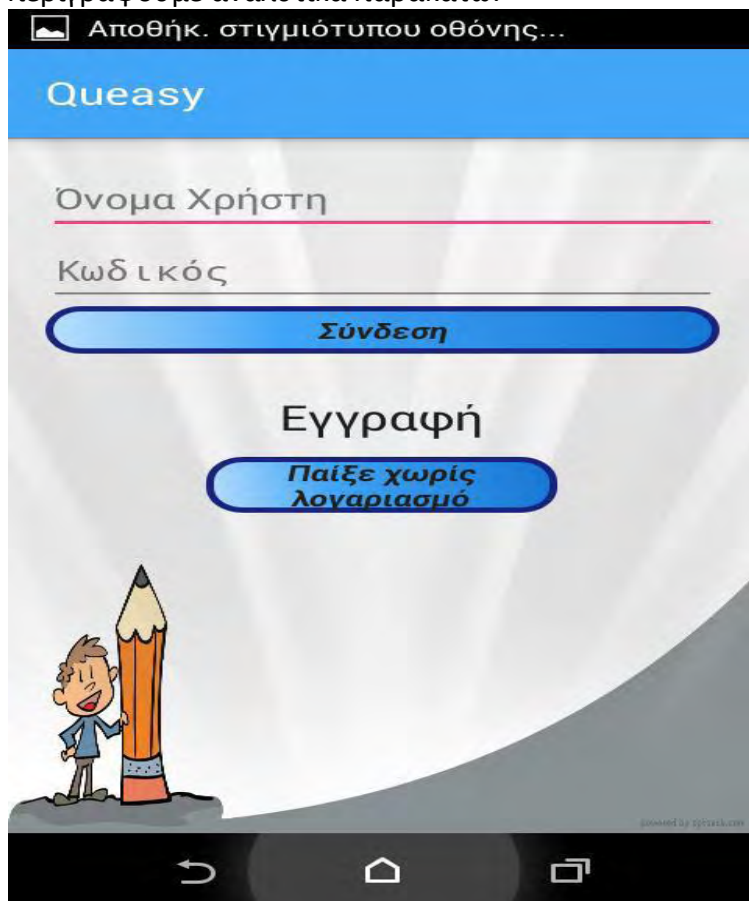


**Εικόνα 4.2** Διάγραμμα ροής

## 4.2.2 Περιγραφή των βασικών παραθύρων(Activities)

### Login Activity

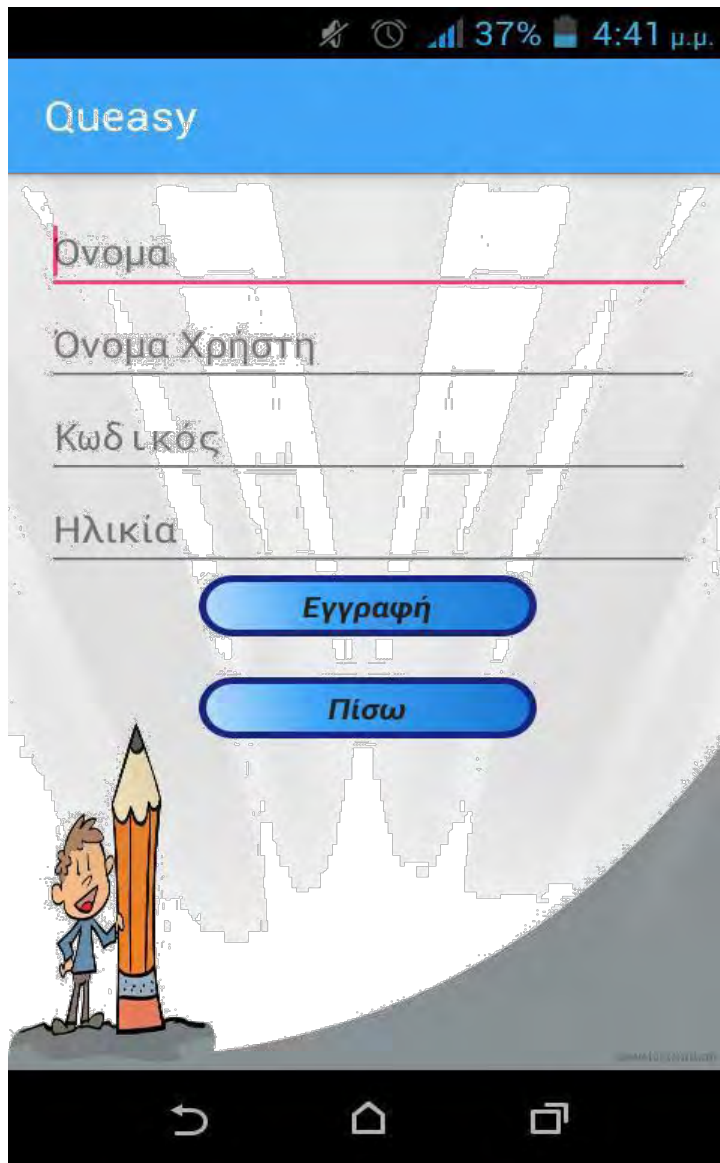
Είναι το πρώτο Activity που εκτελείται κατά την εκκίνηση της εφαρμογής. Σε αυτό το παράθυρο ο χρήστης μπορεί να δώσει τα στοιχεία του λογαριασμού του για σύνδεση στην εφαρμογή, να δημιουργήσει ένα νέο λογαριασμό, εάν δεν έχει, και τέλος υπάρχει δυνατότητα να παίξει χωρίς λογαριασμό(offline mode) σε περίπτωση που δεν έχει πρόσβαση στο διαδίκτυο. Στην τελευταία περίπτωση οι πόντοι που θα συγκεντρώσει δεν αποθηκεύονται στη βάση δεδομένων, συνεπώς η θέση του στην κατάταξη παραμένει ως έχει και προσφέρεται για εξάσκηση και μόνο. Πίσω από τη διαδικασία της σύνδεσης στην εφαρμογή κρύβεται μια επικοινωνία με τη βάση δεδομένων για εξακρίβωση των στοιχείων του χρήστη(read). Η διαδικασία αυτής της σύνδεσης με τη βάση χρησιμοποιείται κατά κόρον στην εφαρμογή και θα την περιγράψουμε αναλυτικά παρακάτω.



Εικόνα 4.3

### Register Activity

Σε αυτό το Activity ο χρήστης μπορεί να δημιουργήσει το δικό του λογαριασμό δίνοντας τα στοιχεία του, δηλαδή: όνομα, όνομα χρήστη(username), κωδικό πρόσβασης(password) και ηλικία. Εάν το όνομα χρήστη είναι διαθέσιμο και δεν είναι κατειλημμένο από κάποιον άλλο χρήστη ολοκληρώνεται η διαδικασία της εγγραφής. Και σε αυτό το σημείο γίνεται σύνδεση με τη βάση δεδομένων(write).



Εικόνα 4.4

### Main Menu Activity

Αφού ολοκληρώθηκε η σύνδεση του χρήστη, εισέρχεται στο κεντρικό μενού της εφαρμογής. Σε αυτό το μενού υπάρχουν πέντε επιλογές: *Παίξε μόνος σου*(Single Player), *Παίξε με αντίπαλο*(Multiplayer), *Ρυθμίσεις*(Options), *Κατάταξη*(Highscore) και *Έξοδος*(Quit).





**Εικόνα 4.5**

### **Single Player Settings Activity**

Σε αυτό το Activity ο χρήστης επιλέγει την κατηγορία των ερωτήσεων στην οποία θέλει να παίξει(Λέξεις, Αριθμοί, Ιστορία, Κόσμος, Γενικές), το επίπεδο δυσκολίας(Εύκολο, Μέτριο, Δύσκολο), τον αριθμό των ερωτήσεων(10, 15, 20) και το διαθέσιμο χρόνο απάντησης σε δευτερόλεπτα(10, 15, 20).





Εικόνα 4.6

### Single Player Game Activity

Αυτό το Activity είναι το κυρίως ταμπλό του παιχνιδιού. Το γραφικό περιβάλλον παραπέμπει στο σχολικό πίνακα πάνω στον οποίο απεικονίζονται η αντίστοιχη ερώτηση κάθε φορά, μία μπάρα που δείχνει στο χρήστη το διαθέσιμο χρόνο που έχει για να απαντήσει στην ερώτηση, οι 4 δυνατές επιλογές απαντήσεων, καθώς και το σύνολο των πόντων που έχει συγκεντρώσει.

Αφού ο χρήστης επιλέξει την απάντηση που επιθυμεί αναλόγως με το αν απάντησε σωστά ή λάθος, η επιλογή του γίνεται πράσινη ή κόκκινη αντίστοιχα. Αν ο χρήστης δεν προλάβει να απαντήσει μέσα στο διαθέσιμο χρόνο, δε συγκεντρώνει πόντους γι' αυτό το γύρο και του εμφανίζεται αυτόματα η σωστή απάντηση. Επίσης, υπάρχουν

και τα αντίστοιχα ηχητικά εφέ που συνοδεύουν τη σωστή απάντηση, τη λάθος απάντηση, τη λήξη του χρόνου και την πρόοδο της μπάρας εξέλιξης του χρόνου.

Σε αυτό το Activity εκτελούνται παρασκηνακά, μια σειρά από αλγορίθμους, για τον υπολογισμό των πόντων, την φόρτωση κάθε φορά της αντίστοιχης ερώτησης ανάλογα με τις επιλογές του χρήστη, τα οποία θα περιγράψουμε αναλυτικά παρακάτω.



**Εικόνα 4.7**

**Εικόνα 4.8**

**Εικόνα 4.9**

### **Play again menu Activity**

Αφού τελειώσει το παιχνίδι, ο χρήστης βλέπει τους πόντους που έχει συγκεντρώσει και του προσφέρεται η δυνατότητα να ξεκινήσει άμεσα ένα νέο παιχνίδι ή να επιστρέψει στο αρχικό μενού. Σε αυτό το Activity εκτελείται παρασκηνακά ένας αλγόριθμος υπολογισμού του επιπέδου του χρήστη, με βάση τους προηγούμενους πόντους που είχε συγκεντρώσει συνολικά προστίθοντας σε αυτούς τους πόντους που συγκέντρωσε στο τελευταίο παιχνίδι. Επίσης γίνεται σύνδεση με τη βάση δεδομένων και αποθηκεύονται σε αυτή το καινούριο συνολικό άθροισμα των πόντων και το νέο επίπεδο του χρήστη. Ο αλγόριθμος για τον υπολογισμό του επιπέδου του χρήστη θα παρουσιαστεί αναλυτικά παρακάτω.



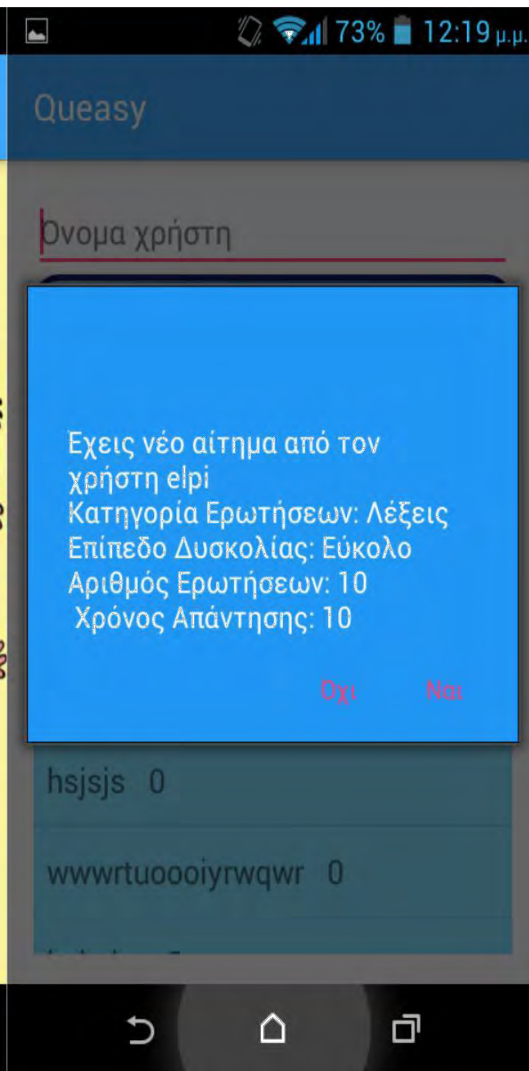
Εικόνα 4.10

### Multiplayer Activity

Σε αυτό το Activity ο χρήστης μπορεί να δεχτεί νέα αιτήματα παιχνιδιού από άλλους χρήστες που είναι συνδεδεμένοι στην εφαρμογή ή να στείλει ο ίδιος ένα νέο αίτημα σε κάποιον από τους συνδεδεμένους χρήστες. Παρασκηνιακά τρέχει συνεχώς, με τη μορφή αναδρομής ένας αλγόριθμος που «ακούει» στη βάση δεδομένων για νέα αιτήματα παιχνιδιού. Είναι μια μορφή προσομοίωσης της λειτουργίας ενός server. Περισσότερα γι' αυτό θα δούμε αναλυτικά παρακάτω.



Εικόνα 4.11



Εικόνα 4.12

### Multiplayer Settings Activity

Όπως και στο Single player Settings Activity, έτσι ακριβώς και στο Multiplayer Settings Activity, ο χρήστης καθορίζει τις επιλογές παιχνιδιού για το νέο αίτημα που θέλει να αποστείλει.





Εικόνα 4.13

### Send Request Activity

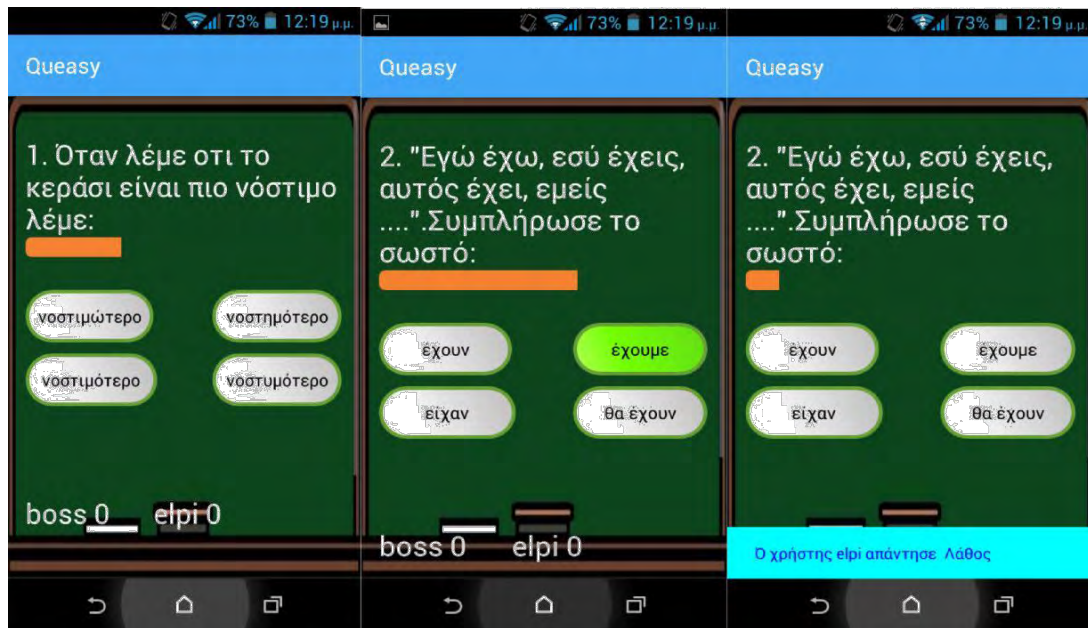
Σε αυτό το Activity ο χρήστης βλέπει όλους τους συνδεδεμένους χρήστες στην εφαρμογή, μπορεί επίσης να αναζητήσει κάποιον συγκεκριμένο χρήστη χρησιμοποιώντας την επιλογή αναζήτησης και να αποστείλει ένα νέο αίτημα παιχνιδιού σε κάποιον από τους συνδεδεμένους χρήστες που είναι διαθέσιμοι εκείνη τη στιγμή και δε βρίσκονται ήδη σε παιχνίδι. Επίσης, αν ο χρήστης στείλει ένα νέο αίτημα σε κάποιον άλλο χρήστη που είναι μεν συνδεδεμένος αλλά βρίσκεται σε διαδικασία παιχνιδιού, ενημερώνεται από την εφαρμογή.



**Εικόνα 4.14**

### **Multiplayer Game1, Game2 Activities**

Πρόκειται για το ίδιο περιβάλλον που παρουσιάσαμε και στο Single Player Game Activity, με κάποιες επιπλέον διαφορές. Αρχικά, η καινούρια ερώτηση παράγεται σε κάθε γύρο του παιχνιδιού από τον χρήστη που έστειλε το αίτημα(user1) και αποστέλλεται στον χρήστη που δέχτηκε το αίτημα(user2). Επιπλέον, η μόνη διαφορά στο γραφικό κομμάτι είναι ότι κάθε ένας από τους δύο χρήστες εκτός από το δικό του άθροισμα των πόντων, βλέπει το άθροισμα των πόντων του αντιπάλου, καθώς και αν απάντησε σωστά ή λάθος σε κάθε ερώτηση. Για το συγχρονισμό των δύο χρηστών έχει υλοποιηθεί μία σειρά αλγορίθμων προσομοίωσης server κατά την οποία ο ένας χρήστης «ακούει» συνεχώς αν ο αντίπαλός του έχει απαντήσει στην τρέχουσα ερώτηση και όταν και οι δύο χρήστες απαντήσουν φορτώνεται η επόμενη ερώτηση. Και αυτός ο αλγόριθμος θα περιγραφεί αναλυτικά παρακάτω.



Εικόνα 4.15

Εικόνα 4.16

Εικόνα 4.17

### Multiplayer play again Activity

Σε αυτό το Activity οι δύο χρήστες (user1, user2) οι οποίοι αναμετρήθηκαν ενημερώνονται για τους συνολικούς πόντους που συγκέντρωσαν και για το ποιος τελικά είναι ο νικητής. Όπως και στο Play again menu Activity, γίνεται υπολογισμός του νέου αθροίσματος των πόντων των δύο χρηστών, καθώς και του επιπέδου και αποθήκευση αυτών στη βάση δεδομένων.



Εικόνα 4.18

### Options Activity

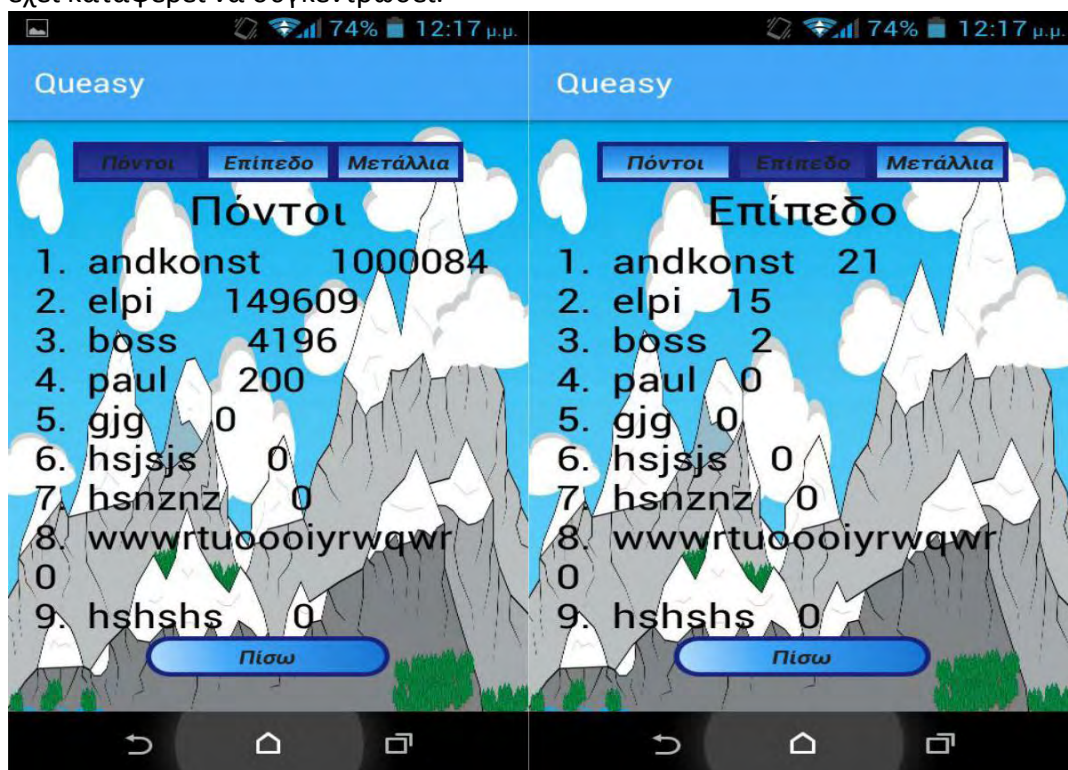
Στο μενού Ρυθμίσεις ο χρήστης έχει τη δυνατότητα να ενεργοποιήσει ή απενεργοποιήσει τόσο τον ήχο όσο και την μουσική.



Εικόνα 4.19

### Highscore Activity

Στο μενού Κατάταξη ο χρήστης μπορεί να ενημερωθεί για τη θέση του στη συνολική κατάταξη όλων των χρηστών της εφαρμογής με βάση το συνολικό άθροισμα των πόντων που έχει συγκεντρώσει παίζοντας. Επίσης, μπορεί να ενημερωθεί για το επίπεδο στο οποίο έχει καταφέρει να ανέβει-πρόοδος, συγκριτικά με τους υπόλοιπους χρήστες, όσο και για τα διάφορα μετάλλια τα οποία έχει καταφέρει να συγκεντρώσει.



Εικόνα 4.20

Εικόνα 4.21



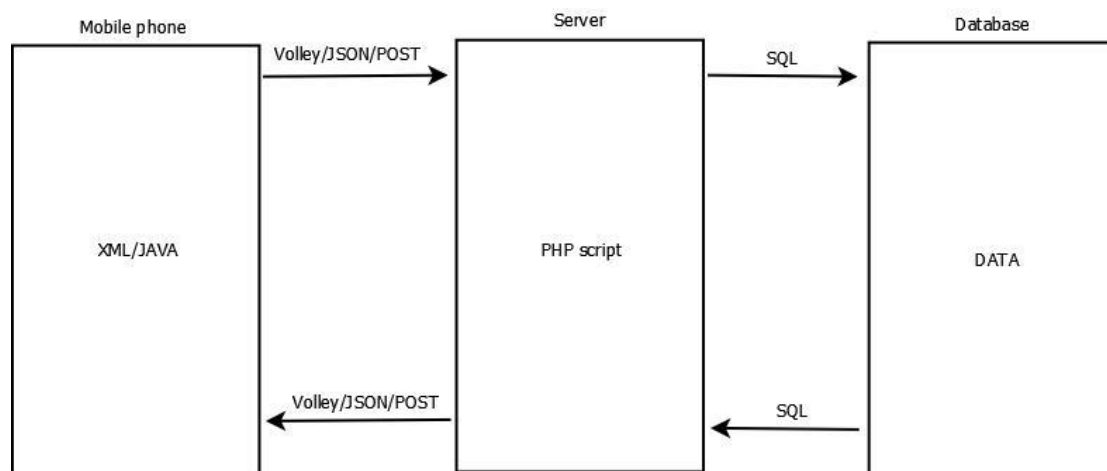
### 4.3 Αλγόριθμοι και τεχνικές προγραμματισμού

Όπως αναφέραμε στο υποκεφάλαιο 4.2.2, η εφαρμογή εκτελεί μία σειρά από αλγόριθμους τους οποίους θα αναλύσουμε σε αυτό το υποκεφάλαιο.

#### 4.3.1 Η σύνδεση της εφαρμογής με τη βάση δεδομένων

Για να καταφέρουμε να έχουμε πρόσβαση στη βάση δεδομένων, η οποία είναι υλοποιημένη στον server(<http://queasygame.esy.es/>), χρησιμοποιήσαμε τη βιβλιοθήκη Android Volley, αντικείμενα τύπου JSON και την μέθοδο POST για να ανταλλάξουμε πληροφορία μεταξύ της εφαρμογής και κάποιων ενδιάμεσων php script τα οποία εκτελούνται στον server. Μέσω των php script επιτύχαμε σύνδεση με τη βάση δεδομένων με την εκτέλεση κώδικα SQL και πραγματοποιήσαμε λειτουργίες εγγραφής, ανάγνωσης ή τροποποίησης των πεδίων της βάσης. Με την ακριβώς αντίστροφη διαδικασία η απαραίτητη κάθε φορά πληροφορία επέστρεφε στην εφαρμογή.

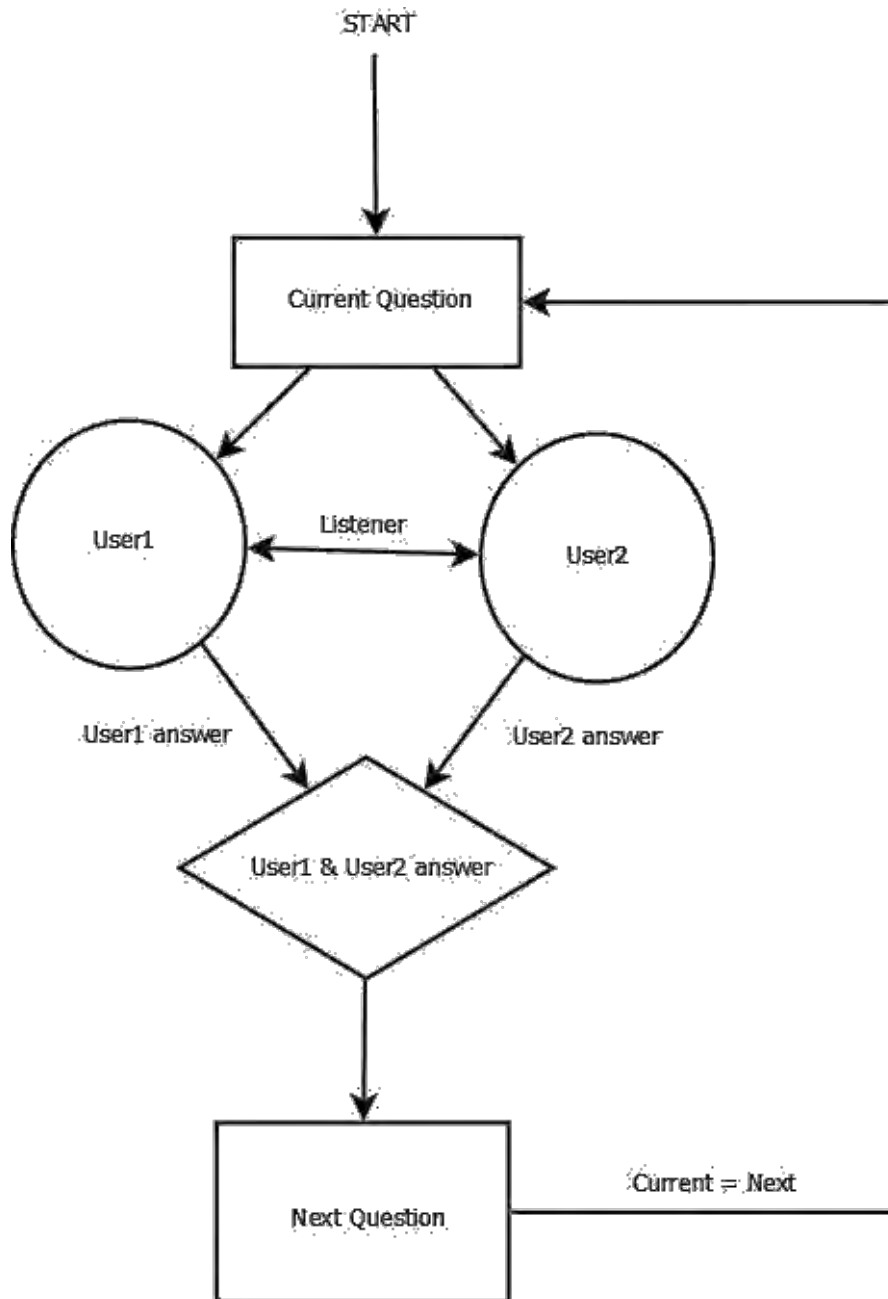
Η διαδικασία αυτής της σύνδεσης με τη βάση δεδομένων χρησιμοποιήθηκε κατά κόρον σχεδόν σε όλα τα Activities(Login, Register, Save, Highscore, Quit, Multiplayer etc.) μέσα στην εφαρμογή με τις απαραίτητες κάθε φορά τροποποιήσεις. Στο παρακάτω διάγραμμα παρουσιάζεται η διαδικασία που περιγράψαμε(Εικόνα 4.23)



Εικόνα 4.22

#### 4.3.2 Η υλοποίηση του Multiplayer Game

Αξιοποιώντας τη μέθοδο που περιγράψαμε στο υποκεφάλαιο 4.3.1 και τις δυνατότητες της αναδρομικής κλήσης συναρτήσεων καταφέραμε να υλοποιήσουμε μία συνεχή ανταλλαγή πληροφοριών, απαραίτητων για το συγχρονισμό των δύο χρηστών, δημιουργώντας έτσι ένα multiplayer περιβάλλον στο οποίο οι δύο χρήστες ανταγωνίζονται σε πραγματικό χρόνο. Ο κάθε ένας από τους χρήστες «ακούει» διαρκώς εάν ο αντίπαλός του έχει απαντήσει και όταν και οι δύο χρήστες απαντήσουν μεταφέρονται και οι δύο στην επόμενη ερώτηση. Στο διάγραμμα που ακολουθεί παρουσιάζεται αυτή η διαδικασία(Εικόνα 4.24)



**Εικόνα 4.23**

### 4.3.3 Οι βασικοί αλγόριθμοι της εφαρμογής

Όπως αναφέρθηκε και πιο πάνω, στην εφαρμογή εκτελούνται μια σειρά από αλγόριθμους. Οι πιο βασικοί από αυτούς είναι οι εξής: ο αλγόριθμος για τον υπολογισμό των κερδισμένων πόντων, ο αλγόριθμος για τον υπολογισμό του επιπέδου του χρήστη, ο αλγόριθμος για την παραγωγή της επόμενης ερώτησης και πολλοί άλλοι.

#### **Αλγόριθμος υπολογισμού των κερδισμένων πόντων**

Ο αλγόριθμος για τον υπολογισμό των κερδισμένων πόντων λαμβάνει υπόψιν του τις εξής 3 παραμέτρους: το επίπεδο δυσκολίας, τον διαθέσιμο χρόνο

που έχει ο χρήστης για να απαντήσει και το πόσο γρήγορα απάντησε μέσα σε αυτό το διαθέσιμο χρόνο. Το επίπεδο δυσκολίας Εύκολο πολλαπλασιάζει τους κερδισμένους πόντους του χρήστη με συντελεστή βαρύτητας 1. Αντίστοιχα, το επίπεδο δυσκολίας Μέτριο πολλαπλασιάζει τους κερδισμένους πόντους με συντελεστή βαρύτητας 2 και το επίπεδο δυσκολίας Δύσκολο με συντελεστή βαρύτητας 3. Η επιλογή χρόνου απάντησης 10 δευτερολέπτων δίνει στο χρήστη το μέγιστο όφελος των 10 πόντων ανά ερώτηση. Αντίστοιχα, η επιλογή χρόνου απάντησης 15 δευτερολέπτων δίνει στο χρήστη το μέγιστο όφελος των 8 πόντων ανά ερώτηση και η επιλογή χρόνου απάντησης 20 δευτερολέπτων δίνει στο χρήστη το μέγιστο όφελος των 5 πόντων ανά ερώτηση. Τέλος, ο χρήστης συγκεντρώνει το 100% των δυνατών πόντων αν απάντησε στο χρονικό διάστημα 80-100% του διαθέσιμου χρόνου, το 80% των δυνατών πόντων αν απάντησε στο χρονικό διάστημα 60-80% του διαθέσιμου χρόνου, το 60% των δυνατών πόντων αν απάντησε στο χρονικό διάστημα 40-60% του διαθέσιμου χρόνου, το 40% των δυνατών πόντων αν απάντησε στο χρονικό διάστημα 20-40% του διαθέσιμου χρόνου και το 20% των δυνατών πόντων αν απάντησε στο χρονικό διάστημα 0-20% του διαθέσιμου χρόνου. Ο συνδυασμός αυτών των τριών παραμέτρων, μέσα από τον αλγόριθμο που παρουσιάζεται παρακάτω, υπολογίζει τους κερδισμένους πόντους του χρήστη για κάθε σωστή απάντηση.

#### Ψευδοκώδικας του αλγορίθμου

```
switch (time_of_answer) {
  case 10: {
    if (time >= 0 && time <= 2) {

      last_game_score = last_game_score + 10 * dif_level;
    } else if (time > 2 && time <= 4) {

      last_game_score = last_game_score + 8 *
      dif_level; } else if (time > 4 && time <= 6) {

      last_game_score = last_game_score + 6 *
      dif_level; } else if (time > 6 && time <= 8) {

      last_game_score = last_game_score + 4 *
      dif_level; } else if (time > 8 && time < 10) {

      last_game_score = last_game_score + 2 * dif_level;
    }
    break;
  }
  case 15: {
    if (time >= 0 && time <= 3) {

      last_game_score = last_game_score + 8 *
      dif_level; } else if (time > 3 && time <= 6) {
```

```

        last_game_score = last_game_score + 6 *
dif_level; } else if (time > 6 && time <= 9) {

        last_game_score = last_game_score + 4 *
dif_level; } else if (time > 9 && time <= 12) {

        last_game_score = last_game_score + 2 *
dif_level; } else if (time > 12 && time < 15) {

        last_game_score = last_game_score + 1 * dif_level;
    }
    break;
}
case 20: {
    if (time >= 0 && time <= 4) {

        last_game_score = last_game_score + 5 *
dif_level; } else if (time > 4 && time <= 8) {

        last_game_score = last_game_score + 4 *
dif_level; } else if (time > 8 && time <= 12) {

        last_game_score = last_game_score + 3 *
dif_level; } else if (time > 12 && time <= 16) {

        last_game_score = last_game_score + 2 *
dif_level; } else if (time > 16 && time < 20) {

        last_game_score = last_game_score + 1 * dif_level;
    }
    break;}
}

```

### **Αλγόριθμος υπολογισμού του επιπέδου**

Ο αλγόριθμος για τον υπολογισμό του επιπέδου(level) του κάθε χρήστη λαμβάνει υπόψιν του το συνολικό άθροισμα των πόντων που έχει συγκεντρώσει ένας χρήστης. Είναι ένας απλός αλγόριθμος ο οποίος όσο ανεβαίνει το επίπεδο, ανεβαίνει εκθετικά και το άθροισμα των απαιτούμενων πόντων που χρειάζεται να έχει συγκεντρώσει ο χρήστης. Ο χρήστης ξεκινάει από το επίπεδο ένα το οποίο επιτυγχάνεται στους 1000 κερδισμένους πόντους και μπορεί να φτάσει μέχρι το επίπεδο 21 το οποίο επιτυγχάνεται στους 253.000 κερδισμένους πόντους.

Ψευδοκώδικας του αλγορίθμου

```
if(total_score>=0 && total_score<1000) {  
    level=0;  
}  
  
if(total_score >=1000 && total_score< 3000) {  
    level=1;  
}  
if(total_score >=3000 && total_score<6000) {  
    level=2;  
}  
  
if(total_score>=6000 && total_score<10000)  
    { level=3;  
}  
  
if(total_score>=10000 && total_score<15000)  
    { level=4;  
}  
  
if(total_score>=15000 && total_score<21000)  
    { level=4;  
}  
  
if(total_score>=21000 && total_score<28000)  
    { level=5;  
}  
  
if(total_score>=28000 && total_score<36000)  
    { level=6;  
}  
  
if(total_score>=36000 && total_score<45000)  
    { level=7;  
}  
  
if(total_score>=45000 && total_score<55000)  
    { level=8;  
}  
  
if(total_score>=55000 && total_score<66000)  
    { level=9;  
}
```

```
if(total_score>=66000 && total_score<78000)
{ level=10;
}

if(total_score>=78000 && total_score<91000)
{ level=11;
}

if(total_score>=91000 && total_score<105000) {
    level=12;
}

if(total_score>=105000 && total_score<120000)
{ level=13;
}

if(total_score>=120000 && total_score<136000)
{ level=14;
}

if(total_score>=136000 && total_score<153000)
{ level=15;
}

if(total_score>=153000 && total_score<171000)
{ level=16;
}

if(total_score>=171000 && total_score<190000)
{ level=17;
}

if(total_score>=190000 && total_score<210000)
{ level=18;
}

if(total_score>=210000 && total_score<231000)
{ level=19;
}

if(total_score>=231000 && total_score<253000)
{ level=20;
}
```

```

if(total_score>=253000 ) {
    level=21;
}

```

### **Αλγόριθμος παραγωγής της επόμενης ερώτησης**

Ανάλογα με τις επιλογές του χρήστη για την κατηγορία των ερωτήσεων και το επίπεδο δυσκολίας, η εφαρμογή διαβάζει τις ερωτήσεις από 14 διαφορετικά αρχεία(Λέξεις-Εύκολο, Λέξεις-Μέτριο, Λέξεις-Δύσκολο, Αριθμοί-Εύκολο, Αριθμοί-Μέτριο, Αριθμοί-Δύσκολο, Ιστορία-Μέτριο, Ιστορία-Δύσκολο, Κόσμος-Εύκολο, Κόσμος-Μέτριο, Κόσμος-Δύσκολο, Γενικές-Εύκολο, Γενικές-Μέτριο, Γενικές-Δύσκολο). Κάθε αρχείο προσφέρει 200 διαφορετικές ερωτήσεις(εκτός από την κατηγορία Γενικές η οποία προσφέρει 800 ερωτήσεις ανά επίπεδο δυσκολίας). Εάν μία ερώτηση επιλεγεί μία φορά από τον αλγόριθμο αποκλείεται να επιλεγεί ξανά. Αυτό εξασφαλίζει ότι όλες οι ερωτήσεις στις οποίες θα διαγωνιστεί ο χρήστης θα είναι διαφορετικές μεταξύ τους. Για να το επιτύχουμε αυτό χρησιμοποιήσαμε μεταξύ άλλων μία γεννήτρια παραγωγής ενός τυχαίου αριθμού.

#### Ψευδοκώδικας του αλγορίθμου

```

public void Find_The_Question(int number_of_question ,int difficulty_level, int
question_category) {

```

```

    switch(question_category)
    { case 1: {
        switch (difficulty_level)
        { case 1:
            input = new
Scanner(getResources().openRawResource(R.raw.vocab_easy));
            break;

            case 2:

                input = new
Scanner(getResources().openRawResource(R.raw.vocab_medium));
                break;

            case 3:

                input = new
Scanner(getResources().openRawResource(R.raw.vocab_hard));
                break;
        }
        break;
    }
    case 2: {

```

```

switch (difficulty_level)
{ case 1:

    input = new
Scanner(getResources().openRawResource(R.raw.maths_easy));
    break;

    case 2:

    input = new
Scanner(getResources().openRawResource(R.raw.maths_medium));
    break;

    case 3:

    input = new
Scanner(getResources().openRawResource(R.raw.maths_hard));
    break;
}
break;
}
case 3: {
    switch (difficulty_level)
    { case 1:

        input = new
Scanner(getResources().openRawResource(R.raw.cosmos_easy));
        break;

        case 2:

        input = new
Scanner(getResources().openRawResource(R.raw.cosmos_medium));
        break;

        case 3:

        input = new
Scanner(getResources().openRawResource(R.raw.cosmos_hard));
        break;
    }
    break;
}
case 4: {
    switch (difficulty_level)
    { case 1:

```



```

        input = new
Scanner(getResources().openRawResource(R.raw.history_easy));
        break;

    case 2:

        input = new
Scanner(getResources().openRawResource(R.raw.history_medium));
        break;

    case 3:

        input = new
Scanner(getResources().openRawResource(R.raw.history_hard));
        break;
    }
    break;
}
case 5: {
    switch (difficulty_level)
    { case 1:

        input = new
Scanner(getResources().openRawResource(R.raw.general_easy));
        break;

    case 2:

        input = new
Scanner(getResources().openRawResource(R.raw.general_medium));
        break;

    case 3:

        input = new
Scanner(getResources().openRawResource(R.raw.general_hard));
        break;
    }
    break;
}
} /// end of switch

```

```

public int Generator_Function() {
    int num1=0;
    Random generator=new Random();

```

```
num1 = 1 + (int)(Math.random() * ((200 - 1) + 1));    ////200
// Min + (int)(Math.random() * ((Max - Min) + 1))
// And there you have it. A random integer value in the range [Min,Max]
return num1;
}
```

#### 4.4 Στατιστικά στοιχεία για την εφαρμογή

Για την υλοποίηση της εφαρμογής χρειάστηκε η παραγωγή κώδικα XML, JAVA, PHP, SQL.

Χρειάστηκε να προγραμματίσουμε 29 διαφορετικά αρχεία XML με συνολικό όγκο κώδικα περισσότερο από 100KB.

Χρειάστηκε επίσης να προγραμματίσουμε 40 διαφορετικά αρχεία JAVA(τόσο Activities όσο και JAVA classes) με συνολικό όγκο κώδικα πάνω από 540KB.

Επίσης, χρειάστηκε να προγραμματίσουμε 23 διαφορετικά αρχεία PHP(μέσα στα οποία εκτελείται και κομμάτι κώδικα SQL) με συνολικό όγκο κώδικα 40KB.

Τέλος, για την υλοποίηση της εφαρμογής χρειάστηκε να αγοράσουμε ένα server(<http://queasygame.esy.es/>) με ενοίκιο 25 ευρώ ανά τρίμηνο.

### 5. Τελικές σκέψεις-συμπεράσματα

Σε αυτό το σημείο θεωρούμε σημαντικό να αναφέρουμε ότι το περιεχόμενο των ερωτήσεων έχει αντληθεί αποκλειστικά από την ύλη που διδάσκεται στο δημοτικό σχολείο αυτή την εποχή. Πιο συγκεκριμένα, οι κατηγορίες 'Λέξεις', 'Αριθμοί' και 'Ιστορία' αντιστοιχίζονται άμεσα με τα μαθήματα του δημοτικού σχολείου, όλων των τάξεων ανά κατηγορία και ανά επίπεδο(κάθε επίπεδο δύο τάξεις), Γλώσσα, Μαθηματικά και Ιστορία. Η κατηγορία 'Κόσμος' είναι μια μικτή κατηγορία και περιλαμβάνει ερωτήσεις από όλα τα υπόλοιπα μαθήματα, δηλαδή Εικαστικά, Μελέτη Περιβάλλοντος, Φυσική, Φυσική Αγωγή, Μουσική, Γεωγραφία, Αγγλικά και Θρησκευτικά. Τέλος, η κατηγορία 'Γενικές' περιέχει ερωτήσεις από όλες τις παραπάνω κατηγορίες. Με λίγα λόγια, το πρόγραμμα σπουδών του δημοτικού σχολείου είναι άμεσα συνδεδεμένο με το περιεχόμενο των ερωτήσεων της εφαρμογής. Επίσης, καμία άλλη μαθησιακή εφαρμογή(τόσο διαδικτυακές πλατφόρμες όσο και εφαρμογές για φορητές συσκευές) δεν συνδέεται άμεσα με το πρόγραμμα σπουδών του δημοτικού σχολείου και δεν χρησιμοποιείται σε επίσημο επίπεδο στην εκπαίδευση.

Για τους λόγους που αναφέρθηκαν παραπάνω, συνίσταται η χρήση της εφαρμογής μέσα στα πλαίσια του σχολείου, αλλά και έξω από αυτό. Όσον αφορά τις γνώσεις και τις ψηφιακές δεξιότητες που χρειάζεται να έχει τόσο ο δάσκαλος όσο και οι μαθητές για να χρησιμοποιήσουν την εφαρμογή είναι βασικές γνώσεις χειρισμού φορητών συσκευών(κινητό τηλέφωνο, ταμπλέτα) και δεν απαιτείται κανενός είδους εκπαίδευση ως προς την χρήση της. Στις μέρες μας, τα περισσότερα παιδιά ηλικίας δημοτικού είναι εξοικειωμένα με τη χρήση τέτοιων συσκευών.

Τέλος, θα θέλαμε να αναφέρουμε κάποιες προτάσεις για χρήση της εφαρμογής στα πλαίσια ενός σχολικού μαθήματος υπό την επίβλεψη του δασκάλου. Μία ιδέα είναι η εφαρμογή να δίνει τη δυνατότητα στο δάσκαλο να επιλέγει το δικό του σύνολο ερωτήσεων, μία μορφή τεστ όχι σε χαρτί αλλά σε μία φορητή συσκευή, και να συναγωνίζονται σε αυτό οι μαθητές μιας τάξης. Κατά βάση ο ρόλος της θα έπρεπε να είναι επιβραβευτικός(και όχι εξεταστικός) για τον-τους νικητή-νικητές είτε με την μορφή κάποιου δώρου, όπως ένα λογοτεχνικό βιβλίο, είτε με κάποια συμβολική διάκριση(μετάλλια γνώσεων). Ο συγκεκριμένος διαγωνισμός θα μπορούσε να επεκταθεί, πέρα από τα πλαίσια μίας τάξης, σε διαγωνισμό μεταξύ όλων των τάξεων ενός σχολείου ή ακόμα και σε ευρύτερα πλαίσια(σχολείο εναντίον σχολείου).

## Παράρτημα

### Βιβλιογραφία

- [1] <http://www.digitalgames.edu.gr/>
- [2] <http://apothetirio.teiep.gr/xmlui/handle/123456789/4959?show=full>
- [3] <http://photodentro.edu.gr/>
- [4] <http://e-didaskalia.blogspot.gr/>
- [5] [www.games.gr](http://www.games.gr)
- [6] <http://ts.sch.gr/software>
- [7] <https://play.google.com/store>
- [8] <https://en.wikipedia.org/wiki/Gamification>
- [9] <https://badgeville.com/wiki/Gamification>
- [10] <http://www.bunchball.com/gamification>
- [11] <https://developer.android.com/studio/index.html>
- [12] <https://developer.android.com/develop/index.html>
- [13] [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)
- [14] <http://stackoverflow.com/>
- [15] [https://www.youtube.com/results?search\\_query=login+activity+android+studio](https://www.youtube.com/results?search_query=login+activity+android+studio)
- [16] <https://www.hosting24.com/>