



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



Εργαστήριο Γνώσης και Αβεβαιότητας

Μεταπτυχιακή Εργασία

**Ανάπτυξη εκπαιδευτικού παιχνιδιού
(ΓΡΙΦΟΜΠΟΤ) για κινητές συσκευές με
λειτουργικό Android**

Γιαζιτζίδου Αρετή
Α.Μ. 2022202102003

Επιβλέποντες:

Μανόλης Γουάλλες
Αναπληρωτής Καθηγητής

Βασίλης Πουλόπουλος
Επίκουρος Καθηγητής

Τρίπολη, Ιούλιος 2023

Συμβουλευτική Επιτροπή:

1. Μανόλης Γουάλλες, Αναπληρωτής Καθηγητής
2. Βασίλης Πουλόπουλος, Επίκουρος Καθηγητής

Εγκρίθηκε από την εξεταστική επιτροπή την 31/07/2023.

- 1.Μανόλης Γουάλλες, Αναπληρωτής Καθηγητής
- 2.Γιώργος Λέπουρας, Καθηγητής
- 3.Κωνσταντίνος Βασιλάκης, Καθηγητής

.....

Γιαζιτζίδου Αρετή
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Πελοποννήσου

Copyright © , Γιαζιτζίδου Αρετή 2023
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πελοποννήσου.

Η παρούσα εργασία εκπονήθηκε σε συνεργασία με το Εργαστήριο Γνώσης και Αβεβαιότητας (ΓΑΒ LAB).

Περίληψη

Τα τελευταία χρόνια η επίδραση της τεχνολογίας σε όλους τους τομείς της καθημερινής μας ζωής είναι μεγάλη. Έχει αλλάξει ο τρόπος εργασίας, επικοινωνίας, διασκέδασης αλλά και εκπαίδευσης με την χρήση όλο και περισσότερων τεχνολογιών πληροφορικής. Τα παιδιά από πολύ μικρή ηλικία εξοικειώνονται με την χρήση υπολογιστή, tablet ή κινητού τηλεφώνου και μέσα από αυτά διασκεδάζουν ή εκπαιδεύονται. Οι δάσκαλοι και οι εκπαιδευτικοί ακόμα και οι ίδιοι οι γονείς προσπαθούν να βρουν νέους αποτελεσματικότερους τρόπους μάθησης με την χρήση των νέων τεχνολογιών πληροφορικής. Αξιοποιώντας όλες τις υπάρχουσες τεχνολογίες της πληροφορίας μπορεί να καταστήσει τους μαθητές ικανούς ώστε να παίρνουν τεκμηριωμένες αποφάσεις και να τις χρησιμοποιούν με τον καλύτερο τρόπο.

Ιδιαίτερα μετά το ξέσπασμα της πανδημίας covid-19 φάνηκε η ανάγκη περισσότερων εκπαιδευτικών εφαρμογών για κινητές συσκευές, αφού είναι οι πιο εύκολα προσβάσιμες συσκευές για την πλειοψηφία των μαθητών όλων των εκπαιδευτικών βαθμίδων. Το Γρίφομποτ αρχικά σχεδιάστηκε ως ένα διαδικτυακό εκπαιδευτικό παιχνίδι για μαθητές. Για τους λόγους όμως που αναφέραμε παραπάνω δημιουργήθηκε η ανάγκη για την ανάπτυξη μιας εφαρμογής για κινητές συσκευές που θα είναι εύχρηστη και ευχάριστη για τους μαθητές. Μέσα από αυτή την εφαρμογή ο μαθητής θα μπορεί να επιλέγει το quiz του μαθήματος που επιθυμεί και μέσα από τα quiz να εξασκεί, να αποκτά αυτοπεποίθηση, ενδιαφέρον για την μάθηση και να ενδυναμώνει την μνήμη του. Στόχος του παιχνιδιού είναι να κεντρίσει το ενδιαφέρον του μαθητή και να εξασκηθεί ευχάριστα πάνω στην ύλη που έχει διδαχθεί στο σχολείο. Παράλληλα εκπαιδεύει στους μαθητές την έννοια του κώδικα με την χρήση διαδοχικών κινήσεων για την επίτευξη του στόχου σε κάθε κουίζ.

Πίνακας περιεχομένων

Εισαγωγή	1
1 "Γρίφομποτ"	3
1.1 Εισαγωγή στο "Γρίφομποτ"	3
1.2 API "Γρίφομποτ"	3
1.3 EndPoints "Γρίφομποτ"	4
1.3.1 Λήψη λίστας των public quiz	5
1.3.2 Λήψη του ID ενός Quiz από ένα PIN	5
1.3.3 Λήψη των στοιχείων ενός Quiz από το ID του	6
1.3.4 Καταγραφή score ενός χρήστη σύμφωνα με το ID	7
1.3.5 Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του	8
1.4 Ροή "Γρίφομποτ"	8
1.5 Οδηγός "Γρίφομποτ"	10
1.5.1 Αρχική Οθόνη	10
1.5.2 Οθόνη "Πώς παίζεται"	11
1.5.3 Οθόνη "Έχω PIN"	11
1.5.4 Οθόνη "Λίστα Quiz"	12
1.5.5 Οθόνη "Όνομα Χρήστη"	14
1.5.6 Οθόνη "Παιχνιδιού"	14
1.5.7 Οθόνη "Λίστα αποτελεσμάτων"	22
2 Χρήση Τεχνολογιών & Τεχνολογικών Εργαλείων	25
2.1 Εισαγωγή	25
2.2 Android studio	25
2.3 JAVA	26
2.4 XML(Extensible Markup Language)	26
2.5 JSON (JavaScript Object Notation)	27
2.6 Adobe Photoshop	28
2.7 Adobe Illustrator	28
2.8 HTTP Protocol(Πρωτόκολλο μεταφοράς υπερκειμένου)	29
2.9 API (Application Programming Interface)	29
3 Παρουσίαση παιχνιδιού "Γρίφομποτ"	31
3.1 Κλάσεις MainActivity και MainFragment	31
3.2 Κλάση ListQuizItemFragment	33
3.3 Κλάση SetPinFragment	34
3.4 Κλάση ResultFragment	35
3.5 Κλάση GameBoardFragment	36
3.6 Κλάση HowToFragment	40
3.7 Κλάση OnGameClass	40

4	Συμπεράσματα	43
	Βιβλιογραφία	45

Πίνακας σχημάτων

Εικόνα 1.1.	Αρχική Οθόνη	4
Εικόνα 1.2.	Αρίθμηση του ταμπλό του παιχνιδιού	9
Εικόνα 1.3.	Αρχική Οθόνη	10
Εικόνα 1.4.	Menu Αρχικής Οθόνης	10
Εικόνα 1.5.	Πως παίζεται	11
Εικόνα 1.6.	Έχω PIN	12
Εικόνα 1.7.	Λίστα Quiz	13
Εικόνα 1.8.	Δήλωση ονόματος χρήστη	14
Εικόνα 1.9.	Ταμπλό παιχνιδιού-Οδηγίες	15
Εικόνα 1.10.	Οθόνη παιχνιδιού	16
Εικόνα 1.11.	Ερώτηση quiz	17
Εικόνα 1.12.	Μήνυμα Λάθους	18
Εικόνα 1.13.	Μήνυμα Σωστού	19
Εικόνα 1.14.	Σχεδίαση διαδρομής	20
Εικόνα 1.15.	Μήνυμα λάθους διαδρομής	21
Εικόνα 1.16.	Μήνυμα τέλους παιχνιδιού	22
Εικόνα 1.17.	Οθόνη Λίστα Αποτελεσμάτων	23
Εικόνα 2.1.	Παράδειγμα ενός αρχείου XML	27
Εικόνα 2.2.	Παράδειγμα ενός αρχείου JSON	28
Εικόνα 3.1.	MainActivity.java	32
Εικόνα 3.2.	mainfragment.java	32
Εικόνα 3.3.	mainfragment.java	33
Εικόνα 3.4.	ListQuizItemFragment.java : Κλήση endPoint "Λήψη λίστας των public quiz"	34
Εικόνα 3.5.	SetPinFragment.java : Κλήση endPoint "Λήψη του ID ενός Quiz από ένα PIN" 1	35
Εικόνα 3.6.	ListQuizItemFragment.java : Κλήση endPoint "Λήψη του ID ενός Quiz από ένα PIN" 2	35
Εικόνα 3.7.	ResultFragment.java : Κλήση endPoint "Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του"	36
Εικόνα 3.8.	GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-1	37
Εικόνα 3.9.	GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-2	37
Εικόνα 3.10.	GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-3	37
Εικόνα 3.11.	GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-4	38

Εικόνα 3.12.	GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-5	38
Εικόνα 3.13.	GameBoardFragment.java : Αναζήτηση Ερώτησης	39
Εικόνα 3.14.	GameBoardFragment.java : Κλήση endPoint "Καταγραφή score ενός χρήστη σύμφωνα με το ID"	39
Εικόνα 3.15.	GameBoardFragment.java : κίνηση του πιονιού	40
Εικόνα 3.16.	HowToFragment.java	40
Εικόνα 3.17.	OnGameClass.java - Συνάρτηση για την εμφάνιση μηνυμάτων	41
Εικόνα 3.18.	OnGameClass.java	41
Εικόνα 3.19.	OnGameClass.java - Συνάρτηση για το ξεκίνημα του χρόνου	41
Εικόνα 3.20.	OnGameClass.java - Συνάρτηση για τον υπολογισμό της βαθμολογίας	42
Εικόνα 3.21.	OnGameClass.java - Συνάρτηση για την ενεργοποίηση των κουμπιών	42
Εικόνα 3.22.	OnGameClass.java - Συνάρτησεις για τον έλεγχο αν ένα κουτακι περιέχει "βόμβα" και συνάρτηση για την κίνηση του πιονιού	42

Εισαγωγή

Η ενσωμάτωση της τεχνολογίας στην εκπαιδευτική διαδικασία μετά την εποχή του COVID-19 προσφέρει μια πληθώρα οφελών. Βελτιώνει την πρόσβαση στην εκπαίδευση, προσαρμόζεται στις ατομικές ανάγκες, ενθαρρύνει τη συνεργατική μάθηση, ενισχύει τις ψηφιακές δεξιότητες και προετοιμάζει τους μαθητές για το μέλλον. Η συνεχής εξέλιξη της τεχνολογίας καθιστά αναγκαίο να εκμεταλλευτούμε αυτό το δυναμικό για να προάγουμε μια πιο σύγχρονη, δυναμική και αποτελεσματική εκπαιδευτική διαδικασία.

Ειδικότερα τα εκπαιδευτικά παιχνίδια για κινητές συσκευές προσφέρουν διαδραστικές εμπειρίες μάθησης. Οι μαθητές αλληλεπιδρούν με το παιχνίδι, επιλύοντας προβλήματα και ανακαλύπτοντας νέες πληροφορίες. Με αυτόν τον τρόπο η μάθηση μπορεί να γίνει πιο ευχάριστη και ενδιαφέρουσα, ενθαρρύνοντας τη συμμετοχή και την ενασχόληση των μαθητών. Επίσης οι κινητές συσκευές είναι τόσο διαδομένες σε αυτή την εποχή επιτρέποντας στην πλειοψηφία των μαθητών να έχουν πρόσβαση σε εκπαιδευτικά παιχνίδια. Επιπλέον, λόγω της φορητότητας των συσκευών να μπορούν να χρησιμοποιούν τα εκπαιδευτικά παιχνίδια οπουδήποτε και όποια στιγμή το επιθυμούν.

Το υπάρχον εκπαιδευτικό παιχνίδι "ΓΡΙΦΟΜΠΟΤ" σχεδιάστηκε να είναι προσβάσιμο μέσω φυλλομετρητή και φυσικά απαιτεί την χρήση του διαδικτύου. Το μειονέκτημα που αντιμετωπίζουν πολλές ιστοσελίδες και κυρίως ιστοσελίδες με παιχνίδια είναι η μη αυτόματη προσαρμογή σε διάφορες οθόνες και μεγέθη συσκευών. Αυτό σημαίνει ότι δεν είναι σχεδιασμένη η ιστοσελίδα του εκπαιδευτικού παιχνιδιού "ΓΡΙΦΟΜΠΟΤ" ειδικά για κινητές συσκευές. Αποτέλεσμα αυτού είναι πολλά στοιχεία να μην εμφανίζονται ή να εμφανίζονται παραμορφωμένα, τα γράμματα να είναι πολύ μικρά και να μην είναι ευανάγνωστα κα. Έτσι ο μαθητής που θα θέλει να παίξει το παιχνίδι από κινητή συσκευή θα δυσκολευτεί στην περιήγηση και στην αλληλεπίδραση με το ίδιο το παιχνίδι.

Με αυτό το σκεπτικό αποφασίστηκε η ανάπτυξη του εκπαιδευτικού παιχνιδιού "ΓΡΙΦΟΜΠΟΤ" για κινητές συσκευές. Οι mobile εφαρμογές είναι σχεδιασμένες ειδικά για κινητές συσκευές. Σχεδιάζονται έτσι ώστε να προσαρμόζονται αυτόματα στο μέγεθος και τις ανάγκες της κινητής συσκευής. Εκμεταλλεύονται πλήρως τις δυνατότητες της συγκεκριμένης συσκευής, προσφέροντας καλύτερη απόδοση από μια ιστοσελίδα. Ακόμη, προσφέρουν πιο ομαλή και ευχάριστη εμπειρία πλοήγησης σε μια κινητή συσκευή, με βελτιστοποιημένες λειτουργίες αφής και αλληλεπίδρασης. Έτσι, οι μαθητές θα μπορούν να αλληλεπιδράσουν με το εκπαιδευτικό παιχνίδι και να έχουν μια ευχάριστη εμπειρία.

Στα κεφάλαια που ακολουθούν θα περιγράψουμε όσο καλύτερα γίνεται τον τρόπο με τον οποίο αναπτύχθηκε το εκπαιδευτικό παιχνίδι "Γριφομποτ" για κινητές συσκευές. Συγκεκριμένα, στο Κεφάλαιο 1 παρουσιάζεται η βασική ροή του παιχνιδιού, το API (Application Programming Interface) του διαδικτυακού παιχνιδιού το οποίο είναι υπεύθυνο για την διασύνδεση με την βάση δεδομένων και καθορίζει την σειρά των βασικών λειτουργιών του παιχνιδιού. Γι' αυτό τον λόγο ο σχεδιασμός της mobile εφαρμογής στηρίχθηκε σε αυτό το API (Application Programming Interface). Επίσης στο κεφάλαιο αυτό παρουσιάζεται ο οδηγός του εκπαιδευτικού παιχνιδιού "Γριφομποτ" για κινητές συσκευές. Στο Κεφάλαιο 2 παρουσιάζονται όλα τα τεχνολογικά εργαλεία αλλά και γλώσσες προγραμματισμού που απαιτήθηκαν για την ανάπτυξη της εφαρμογής, στο Κεφάλαιο 3 παρου-

ζιάζονται τα σημαντικότερα στοιχεία απο την εφαρμογή Γρίφομποτ (κλάσεις, κώδικας), και τέλος στο Κεφάλαιο 4 παρουζιάζονται τα συμπεράσματά μας.

Κεφάλαιο 1

”Γρίφομποτ”

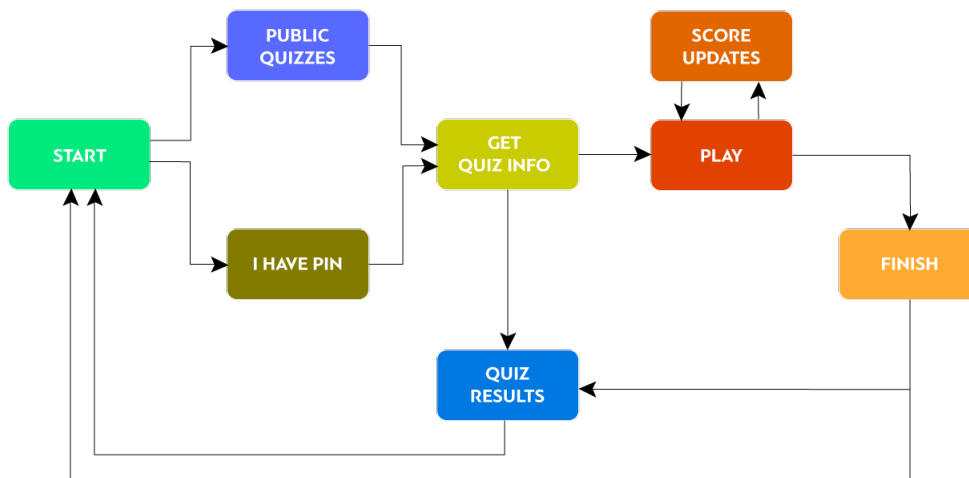
1.1 Εισαγωγή στο ”Γρίφομποτ”

Το ”Γρίφομποτ” είναι ένα εκπαιδευτικό παιχνίδι για μαθητές. Αρχικά το ”Γρίφομποτ” σχεδιάστηκε ως ένα διαδικτυακό παιχνίδι. Όμως η εξοικίωση που διαθετούν τα παιδιά με τις κινητές συσκευές και λόγω του ότι είναι πιο εύκολα προσβάσιμες αναπτύχθηκε η ανάγκη της δημιουργίας μιας εκπαιδευτικής εφαρμογής για κινητά τηλέφωνα με λειτουργικό σύστημα Android. Δάσκαλοι και καθηγητές που θέλουν να εκπαιδεύσουν με ένα διασκεδαστικό τρόπο τους μαθητές τους μπορούν να σχεδιάσουν ένα κουίζ ερωτήσεων και να το υποβάλλουν στην πλατφόρμα. Το ”Γρίφομποτ” απευθύνεται σε μαθητές δημοτικού και γυμνασίου. Είναι ευκολο στην χρήση του μιας και δεν χρειάζεται κάποια εγγραφή για δημιουργία λογαριασμού με την χρήση email. Ένας από τους στόχους αυτού του παιχνιδιού είναι να κεντρίσει το ενδιαφέρον του μαθητή και να εξασκηθεί ευχάριστα πάνω στην ύλη που έχει διδαχθεί στο σχολείο.

1.2 API ”Γρίφομποτ”

Η ροή του ”Γρίφομποτ” καθορίστηκε από το API(Application Programming Interface) που δημιουργήθηκε ώστε να μπορέσει το παιχνίδι να επικοινωνήσει με την υπάρχουσα online βάση δεδομένων. Αυτό το API(Application Programming Interface) θα χρησιμοποιηθεί για την λήψη των quiz, για την ενημέρωση των αποτελεσμάτων καθώς και την λήψη των στατιστικών στοιχείων των παικτών.

Παρακάτω μπορούμε να δούμε την βασική ροή των δεδομένων σύμφωνα με το API και την οποία θα ακολουθήσει και η εφαρμογή.



Εικόνα 1.1. Αρχική Οθόνη

Ετσι αρχικά ο μαθητής μπαίνοντας στην εφαρμογή θα μπορεί να επιλέξει είτε να δει "Free Quiz" είτε να βάλει ένα "PIN" το οποίο του το παρείχε ο εκπαιδευτικός απο κάποιο ιδιωτικό quiz. Στη συνέχεια εαν επιλέξει "Free Quiz" θα επιστρέψει την λίστα των διαθέσιμων quiz. Όταν επιλέξει το quiz που επιθυμεί να παίξει ο μαθητής είτε από την λίστα είτε από την αρχή της εφαρμογής δηλώνοντας ένα PIN λαμβάνει τα στοιχεία του quiz χρησιμοποιώντας το quiz id. Μαζί με τα στοιχεία του quiz επιστρέφεται και μια παράμετρος uuid η οποία είναι ένας μοναδικός αριθμός καταγραφής score για το quiz. Εν συνεχεία ο μαθητής παίζει το παιχνίδι σύμφωνα με τα στοιχεία που διαμορφώνουν το quiz και σύμφωνα με τις ενέργειές του στέλνονται στον server score updates. Με τη λήξη του παιχνιδιού ο μαθητής μπορεί να δει τα αποτελέσματα για το συγκεκριμένο quiz.

1.3 EndPoints "Γρίφομποτ"

EndPoints είναι μια συγκεκριμένη διεύθυνση (URL) που ένα API παρέχει για να εκτελέσει συγκεκριμένες λειτουργίες και να ανταλλάξει δεδομένα με άλλες εφαρμογές ή διακομιστές.

Οι εφαρμογές μπορούν να κάνουν αιτήματα στα τελικά σημεία του API για να ζητήσουν πληροφορίες ή να εκτελέσουν κάποια ενέργεια. Αυτά τα αιτήματα μπορούν να είναι σε μορφή HTTP (όπως GET, POST, PUT, DELETE) και να περιλαμβάνουν παράμετρους ή δεδομένα που απαιτούνται για την εκτέλεση της λειτουργίας.

Το API(Application Programming Interface) στη συνέχεια ανταποκρίνεται στο αίτημα και στέλνει πίσω μια απάντηση, που μπορεί να περιλαμβάνει τα αποτελέσματα του αιτήματος ή άλλες πληροφορίες που ζητήθηκαν.

Τα τελικά σημεία του API ορίζουν τον τρόπο με τον οποίο οι εφαρμογές μπορούν να αλληλεπιδρούν με το API και να αξιοποιούν τις δυνατότητες που παρέχει.

Το API αυτό αποτελείται από πέντε endPoints

1. Λήψη της λίστας των public quiz
2. Λήψη του ID ενός Quiz από ένα PIN

3. Λήψη των στοιχείων ενός Quiz από το ID του
4. Καταγραφή score ενός χρήστη σύμφωνα με το ID
5. Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του

1.3.1 Λήψη λίστας των public quiz

Το endpoints αυτό χρησιμοποιεί το URL: `"/api/quiz/public"` και την μέθοδο GET για να εξάγει τα δεδομένα.

- URL: `/api/quiz/public`
- Method: GET
- Parameters (body): -
- Header info: -
- Reply: JSON formatted

Επιστρέφει ως απάντηση τα στοιχεία των διαθέσιμων ελευθερων quiz σε μορφή JSON. Για κάθε ένα quiz επιστρέφει τα παρακάτω στοιχεία:

- `"id"`: Αριθμός id quiz
- `"name"`: Τίτλος Quiz
- `"description"`: Περιγραφή του Quiz
- `"size"`: Μέγεθος ταμπλό
- `"level"`: Επίπεδο τάξης
- `"excercises"`: Αριθμό ερωτήσεων
- `"pin"`: pin quiz

1.3.2 Λήψη του ID ενός Quiz από ένα PIN

Το endpoints αυτό χρησιμοποιεί το URL: `"/api/quiz/id/pin"` και την μέθοδο POST γιατί στέλνει μία παράμετρο το pin για να επιστρέψει το id. Τα δεδομένα αυτά έρχονται σε μορφή JSON.

- URL: `/api/quiz/id/pin`
- Method: POST
- Parameters (body): pin
- Header info: content-type: application/x-www-form-urlencoded
- Reply: JSON formatted

1.3.3 Λήψη των στοιχείων ενός Quiz από το ID του

Το Endpoint καλείται για να πραγματοποιηθεί λήψη των στοιχείων ενός quiz από το id του. Το endpoint αυτό χρησιμοποιεί το URL: `"/api/quiz/play/game"` και την μέθοδο POST γιατί στέλνει μία παράμετρο το id.

- Method: POST
- Parameters (body): id
- Header info: content-type: application/x-www-form-urlencoded
- Reply: JSON formatted

Επιστρέφει ως απάντηση τα στοιχεία ενός συγκεκριμένου quiz σε μορφή JSON.

- `"id"`: Αριθμός id quiz
- `"name"`: Όνομα quiz
- `"description"`: Περιγραφή quiz
- `"size"`: Μέγεθος ταμπλό
- `"level"`: Επίπεδο τάξης
- `"exercises"`: Αριθμό ερωτήσεων
- `"pin"`: Pin του Quiz
- `"uuid"`: Επιστρέφει έναν μοναδικό αριθμό που χρησιμοποιείται για την καταγραφή της βαθμολογίας
- `"code"`: Επιστρέφει έναν πίνακα με τα στοιχεία του ταμπλό του παιχνιδιού
- `"maths"`: Επιστρέφει έναν πίνακα με τα στοιχεία των ερωτήσεων

Πίνακας code

- `"id"`: Αριθμός id που δηλώνει τον αριθμό στο κουτάκι που αντιστοιχεί στο ταμπλό
- `"blocked"`: μια μεταβλητή boolean που δηλώνει εάν το τετράγωνο περιέχει βόμβα
- `"player"`: μια μεταβλητή boolean που δηλώνει εάν το τετράγωνο περιέχει το πιόνι(σκύλο)
- `"solved"`: μια μεταβλητή boolean που δηλώνει εάν το τετράγωνο είναι απαντημένο
- `"math"`: μια μεταβλητή boolean που δηλώνει εάν το τετράγωνο περιέχει ερώτηση
- `"matheq"`: Έναν αριθμό που δηλώνει την ερώτηση

Πίνακας maths

- `"id"`: Αριθμός id ερώτησης(matheq)
- `"question"`: Ερώτηση
- `"correct"`: Σωστή απάντηση

- "alt1": Πρώτη επιλογή απάντησης
- "alt2": Δεύτερη επιλογή απάντησης
- "alt3": Τρίτη επιλογή απάντησης
- "alt4": Τέταρτη επιλογή απάντησης
- "image": Εικόνα εαν υπάρχει
- "story": Εξήγηση εαν υπάρχει

1.3.4 Καταγραφή score ενός χρήστη σύμφωνα με το ID

Το EndPoints καλείτε για να καταγράψει τη βαθμολογία του μαθητή. Το endpoints αυτό εκτελείτε κάθε φορά που ο μαθητής δίνει μία σωστή απάντηση.

Η βαθμολογία υπολογίζεται χρησιμοποιώντας τον ακόλουθο αλγόριθμο. Για κάθε ερώτηση που ο μαθητής απαντά σωστά και μετακινεί σωστά τον χαρακτήρα του παιχνιδιού (σκύλο) οι βαθμοί που κερδίζει είναι ο ακέραιος αριθμός αν διαιρεσουμε τον αριθμό 100 με τον αριθμό των κινήσεων που πραγματοποιήσε. Δηλαδή
(100 / αριθμοί κινήσεων) βαθμοί.

Π.χ. Αν ο μαθητής έπρεπε να μετακινήσει 4 για να πάει στον στόχο παίρνει $100/4 = 25$ πόντους. Εάν ο χρήστης έπρεπε να μετακινήσει 3 για να πάει στον στόχο, παίρνει $100/3 = 33$ πόντους.

Το τελικό σημείο πρέπει να καλείται κάθε φορά που ο χρήστης σκοράρει και όχι μόνο στο τέλος.

Το endpoints αυτό χρησιμοποιεί το URL: "/api/record/score" και την μέθοδο POST γιατί στέλνει τις παραμέτρους το id του quiz, το όνομα χρήστη, την βαθμολογία που κάθε φορά απαντάει σωστά σε μία ερώτηση, τις κινήσεις που πραγματοποίησε για να φτάσει στο κουτάκι της απάντησης, το uuid που λήφθηκε από το endpoints "Λήψη των στοιχείων ενός Quiz από το ID του".

- URL: /api/record/score
- Method: POST
- Parameters (body): id – το id του quiz
- username – the username of the user (can be provided once when the app is initiated)
- score – the score to add to the user (the score of each time the user scores)
- movements – the movements required to go to the answer
- uuid – the uuid provided by the previous call
- Header info: content-type: application/x-www-form-urlencoded
- Reply: JSON formatted

Επιστρέφει σε μορφή JSON τους συνολικούς βαθμούς, τις συνολικές κινήσεις και το σύνολο των σωστών απαντήσεων καθώς και το uuid.

- "uuid": το uuid που λήφθηκε από το endpoints "Λήψη των στοιχείων ενός Quiz από το ID του"
- "score": Συνολικοί βαθμοί
- "movements": Συνολικές κινήσεις
- "questions": Σύνολο σωστών απαντημένων ερωτήσεων

1.3.5 Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του

Το Endpoint καλείται για να προβληθούν τα αποτελέσματα όλων των παικτών σε ένα συγκεκριμένο quiz. Το endpoint αυτό χρησιμοποιεί το URL: `"/api/quiz/scores"` και την μέθοδο POST γιατί στέλνει μία παράμετρο το id του quiz για να επιστρέψει τα αποτελέσματα ενός συγκεκριμένου quiz.

- URL: `/api/quiz/scores`
- Method: POST
- Parameters (body): id – το id του quiz
- Header info: content-type: application/x-www-form-urlencoded
- Reply: JSON formatted

Για κάθε μαθητή που έχει καταγραφή αποτελέσμα επιστρέφει τα παρακάτω στοιχεία σε μορφή JSON.

- `"username"`: Όνομα Χρήστη
- `"score"`: Συνολική Βαθμολογία
- `"questions"`: Σύνολο σωστών απαντημένων ερωτήσεων
- `"movements"`: Σύνολο κινήσεων
- `"seconds"`: χρόνος

1.4 Ροή "Γρίφομποτ"

Μέχρι τώρα αναφέραμε γενικά την ροή του "Γρίφομποτ". Εδώ θα προσπαθήσουμε να περιγράψουμε αναλυτικά με την σειρά που πραγματοποιούνται οι διαδικασίες ροής στο "Γρίφομποτ".

Αρχικά ο μαθητής μπορεί να επιλέξει ανάμεσα σε ελεύθερο quiz και σε ιδιωτικό. Εάν επιλέξει το ιδιωτικό quiz θα πρέπει να συμπληρώσει το Pin του Quiz και ένα όνομα χρήστη. Εάν επιλέξει Free quiz τότε θα εμφανίσει μία λίστα των διαθέσιμων quiz χρησιμοποιώντας το endpoints "Λήψη λίστας των public quiz". Όταν επιλέξει το quiz που επιθυμεί να παίξει ο μαθητής θα συμπληρώσει το όνομα χρήστη. Στη συνέχεια είτε ο μαθητής επέλεξε από την λίστα ένα quiz είτε από την αρχή της εφαρμογής δηλώνοντας ένα PIN γίνεται κλήση του endpoints "Λήψη του ID ενός Quiz από ένα PIN" επειδή απαιτείται το id του quiz για να ληφθούν οι πληροφορίες για το συγκεκριμένο quiz. Εφόσον ληφθεί το id του quiz πραγματοποιείται η κλήση του endpoints "Λήψη των στοιχείων ενός Quiz από το ID του" για να λάβει τα στοιχεία του quiz με το id που λήφθηκε προηγουμένως. Μία σημαντική παραμετρος που επιστρέφεται μαζί με τα στοιχεία του quiz είναι το uuid, η οποία είναι ένας μοναδικός αριθμός καταγραφής score για το quiz.

Για να μπορέσει να παίξει ο μαθητής το παιχνίδι πρέπει πρώτα να δημιουργηθεί το ταμπλό του παιχνιδιού. Τα στοιχεία για την δημιουργία του ταμπλό έρχονται από την κλήση του προηγούμενου endpoints. Τα στοιχεία που απαιτούνται για την δημιουργία του ταμπλό είναι το μέγεθος του, τα στοιχεία για κάθε κουτάκι του ταμπλό δηλαδή αν είναι κενό, αν περιέχει βόμβα, αν περιέχει ερώτηση ή τον χαρακτήρα και τέλος τα στοιχεία των ερωτήσεων. Το ταμπλό πάντα είναι τετράγωνο και αριθμός των τετραγώνων κουτιών που θα περιέχει είναι η παράμετρος size στο τετράγωνο. Δηλαδή αν η παράμετρος size είναι 6 τότε το μέγεθος θα είναι $6 \times 6 = 36$. Άρα 36 κουτάκια. Το κάθε ένα από αυτά τα κουτάκια περιέχει πληροφορίες όπως αναφέραμε παραπάνω. Αυτές οι πληροφορίες δίνονται μέσα στην παράμετρο code που περιέχει για κάθε ένα ξεχωριστά τις εξής πληροφορίες:

- εαν το τετράγωνο περιέχει βόμβα
- εαν το τετράγωνο περιέχει το πιόνι(σκύλο)
- εαν το τετράγωνο είναι απαντημένο
- εαν το τετράγωνο περιέχει ερώτηση
- τον αριθμό της ερώτησης

Η φορά με την οποία δημιουργούνται τα τετράγωνα είναι από τα αριστερά προς τα δεξιά. Για παραδειγμα εαν είχαμε ένα ταμπλό 6x6 τότε η φορά θα ήταν όπως φαίνεται στην παρακάτω εικόνα.

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Εικόνα 1.2. Αρίθμηση του ταμπλό του παιχνιδιού

Αφου δημιουργήθηκε το ταμπλό μπορεί ο μαθητής να ξεκινήσει το παιχνίδι. Ο μαθητής έχει ένα πιόνι "σκύλο" που πρέπει να τον μετακινεί μετά απο κάθε σωστή ερώτηση στο συγκεκριμένο κουτάκι ώστε να βαθμολογηθεί. Ο μαθητής θα πρέπει να επιλέξει ένα απο τα κουτάκια που περιέχουν ερωτήσεις. Αυτα τα κουτάκια έχουν το εικονίδιο "κόκκαλο". Μόλις το επιλέξει του εμφανίζεται η ερώτηση και πρέπει να επιλέξει την απάντηση. /newline Όταν επιλέξει την απάντηση εάν είναι λανθασμένη του εμφανίζεται ένα μήνυμα και ακούγεται ένας ήχος. Τα κουμπιά ελέγχου (βελάκια) παραμένουν απενεργοποιημένα αφου δεν επιτρέπεται να μετακινηθεί το πιόνι./newline Εαν η απάντηση είναι σωστή τότε το πιόνι μπορεί να μετακινηθει και ενεργοποιούνται τα κουμπιά ελέγχου. Ο μαθητής πρέπει να σχηματίσει την διαδρομή με τα βελάκια και στην συνέχεια να πατήσει το κουμπί "Ξεκίνα" για να μετακινηθεί το πιόνι./newline Εάν σχεδιάσει λάθος την διαδρομή και χτυπήσει στις άκρες του ταμπλό ή σε κουτάκι με βόμβα τότε εμφανίζεται αντίστοιχο μήνυμα και ήχος με αποτέλεσμα ο μαθητής να μην κερδίζει βαθμούς. Εαν φτάσει στο σωστό κουτάκι τότε υπολογίζονται οι βαθμοί του διαιρώντας το 100 με τις κινήσεις που χρειάστηκε να πραγματοποιήσει το πιόνι ωστε να φτάσει στο σωστό κουτάκι. Σε αυτό το σημείο καλείται το endpoint "Καταγραφή score ενός χρήστη σύμφωνα με το ID". Με αυτόν τον τρόπο συνεχίζεται το παιχνίδι μεχρι να ολοκληρωθεί. Όταν ολοκληρωθεί μπορεί ο μαθητής να δει την λίστα των αποτελεσμάτων του quiz. Για να ληφθεί η λίστα των αποτελεσμάτων καλείται το endpoints "Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του".

1.5 Οδηγός ”Γρίφομποτ”

Τα quiz τα δημιουργούν οι εκπαιδευτικοί σύμφωνα με τις ανάγκες του μαθητός τους. Τα quiz αυτά μπορούν να είναι είτε ελεύθερα προσβάσιμα απ’ όλους (δημόσια) είτε να απαιτείτε η χρήση ενός PIN (ιδιωτικά). Πρόσβαση στα ιδιωτικά quiz μπορούν να έχουν μόνο όσοι διαθέτουν το PIN.

Το PIN το παρέχει ο εκπαιδευτικός στους μαθητές του, αφού έχει δημιουργήσει το quiz, ώστε μόνο αυτοί να έχουν πρόσβαση σε αυτό.

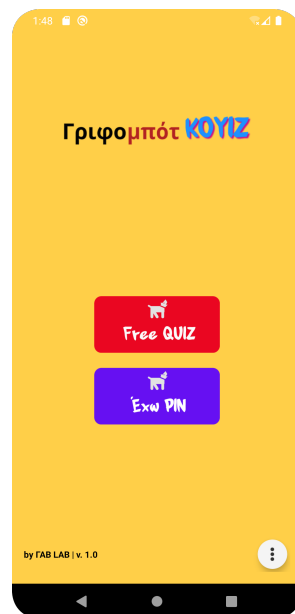
1.5.1 Αρχική Οθόνη

Όταν κάποιος ανοίγει την εφαρμογή στην αρχική οθόνη εμφανίζονται δύο κουμπιά.

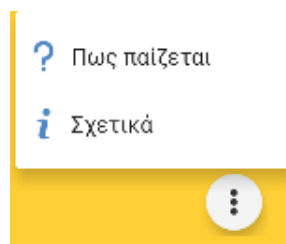
- Free Quiz
- Έχω Pin

Ετσι αρχικά ο παίκτης μπαίνοντας στην εφαρμογή θα μπορεί να επιλέξει είτε να δει ”Free Quiz” είτε να βάλει ένα ”PIN” το οποίο του το παρέίχε ο εκπαιδευτικός απο κάποιο ιδιωτικό quiz(Εικόνα 1.3).

Επιπλέον κάτω δεξιά υπάρχει ένα popup μενού με τις οδηγίες του παιχνιδιού (Εικόνα 1.4).



Εικόνα 1.3. Αρχική Οθόνη

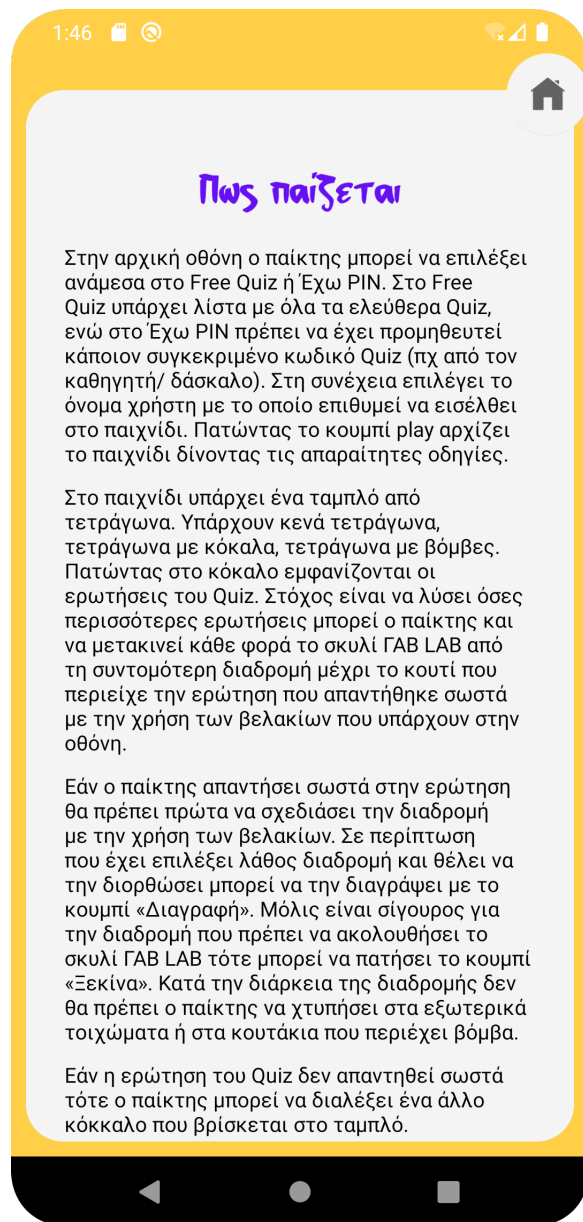


Εικόνα 1.4. Menu Αρχικής Οθόνης

1.5.2 Οθόνη "Πώς παίζεται"

Σε αυτήν την οθόνη οδηγείται ο μαθητής όταν επιλέξει από το μενού της αρχικής οθόνης "Πώς παίζεται" (Εικόνα 1.4)).

Μπορεί να επιστρέψει πίσω στην αρχική οθόνη εάν πατήσει το εικονίδιο με το σπιτάκι.



Εικόνα 1.5. Πως παίζεται

1.5.3 Οθόνη "Έχω PIN"

Σε αυτήν την οθόνη οδηγείται ο μαθητής όταν επιλέξει από την αρχική οθόνη το κουμπί "Έχω PIN" (Εικόνα 1.3)).

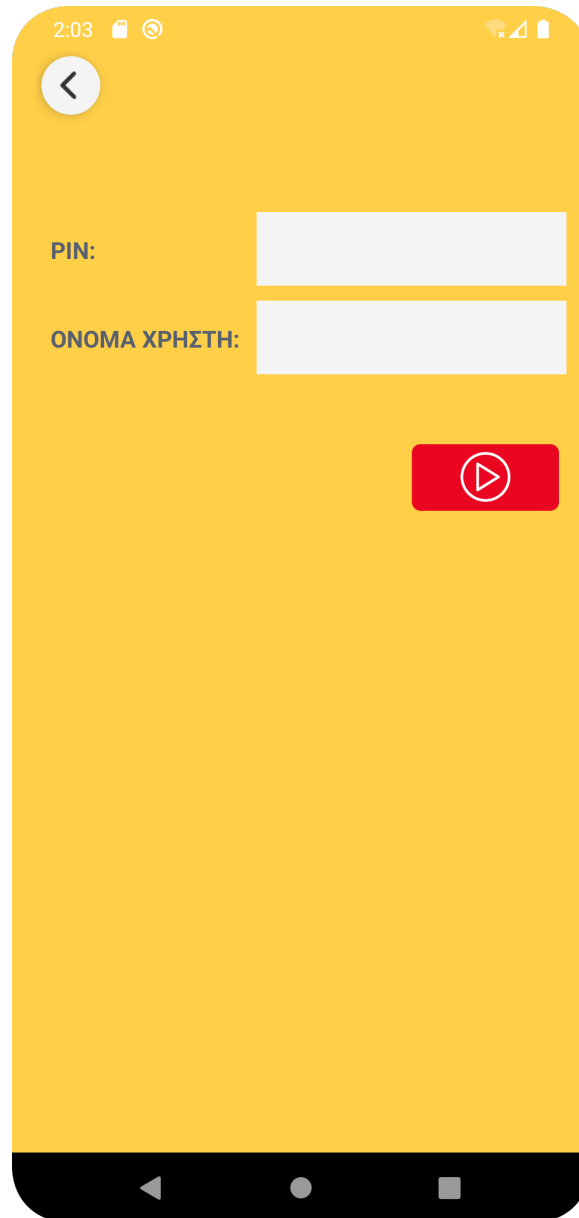
Ο μαθητής θα πρέπει να συμπληρώσει το PIN που του έχει δώσει ο εκπαιδευτικός στο κουτάκι "PIN".

Επίσης στο πεδίο "Όνομα Χρήστη" θα πρέπει να συμπληρώσει ο μαθητής το όνομα που επι-

θυμεί να έχει στο παιχνίδι. Την πρώτη φορά που εισέρχεται το πεδίο είναι κενό, αλλά απο την επόμενη φορά έρχετε προσυμπληρωμενο με το τελευταίο όνομα χρήστη που επέλεξε. Το όνομα χρήστη αποθηκεύεται στην κινητή συσκευή. Μπορεί να το αλλάξει οποιαδήποτε στιγμή θελήσει.

Για να προχωρήσει στο παιχνίδι θα πρέπει να πατήσει το κόκκινο κουμπί δεξιά κατω απο το πεδίο "Όνομα Χρήστη".

Μπορεί να επιστρέψει πίσω στην αρχική οθόνη εαν πατήσει το εικονίδιο με το βελάκι πάνω αριστερά. (Εικόνα 1.6)



Εικόνα 1.6. Έχω PIN

1.5.4 Οθόνη "Λίστα Quiz"

Σε αυτήν την οθόνη οδηγείται ο μαθητής όταν επιλέξει απο την αρχική οθόνη το κουμπί "Free Quiz" (Εικόνα 1.3).

Εδώ ο μαθητής βλέπει μία λίστα των διαθέσιμων Quiz. Για κάθε quiz προβάλλονται τα εξής στοιχεία:

1. Τάξη
2. Μάθημα
3. Τίτλος Quiz
4. Περιγραφή
5. Δύο κουμπιά

Το κουμπί αριστερά οδηγεί στην οθόνη όπου δηλώνεται το όνομα χρήστη και το κουμπί δεξιά οδηγεί στην οθόνη των στατιστικών στοιχείων του παιχνιδιού. Μπορεί να επιστρέψει πίσω στην αρχική οθόνη εάν πατήσει το εικονίδιο με το βελάκι πάνω αριστερά.



Εικόνα 1.7. Λίστα Quiz

1.5.5 Οθόνη "Όνομα Χρήστη"

Σε αυτήν την οθόνη οδηγείται ο μαθητής όταν επιλέξει από την οθόνη "Λίστα Quiz" το κόκκινο κουμπί αριστερά (Εικόνα 1.7). Στο πεδίο "Όνομα Χρήστη" θα πρέπει να συμπληρώσει ο μαθητής το όνομα που επιθυμεί να έχει στο παιχνίδι. Την πρώτη φορά που εισέρχεται το πεδίο είναι κενό, αλλά από την επόμενη φορά έρχετε προσυμπληρωμένο με το τελευταίο όνομα χρήστη που επέλεξε. Το όνομα χρήστη αποθηκεύεται στην κινητή συσκευή. Μπορεί να το αλλάξει οποιαδήποτε στιγμή θελήσει.

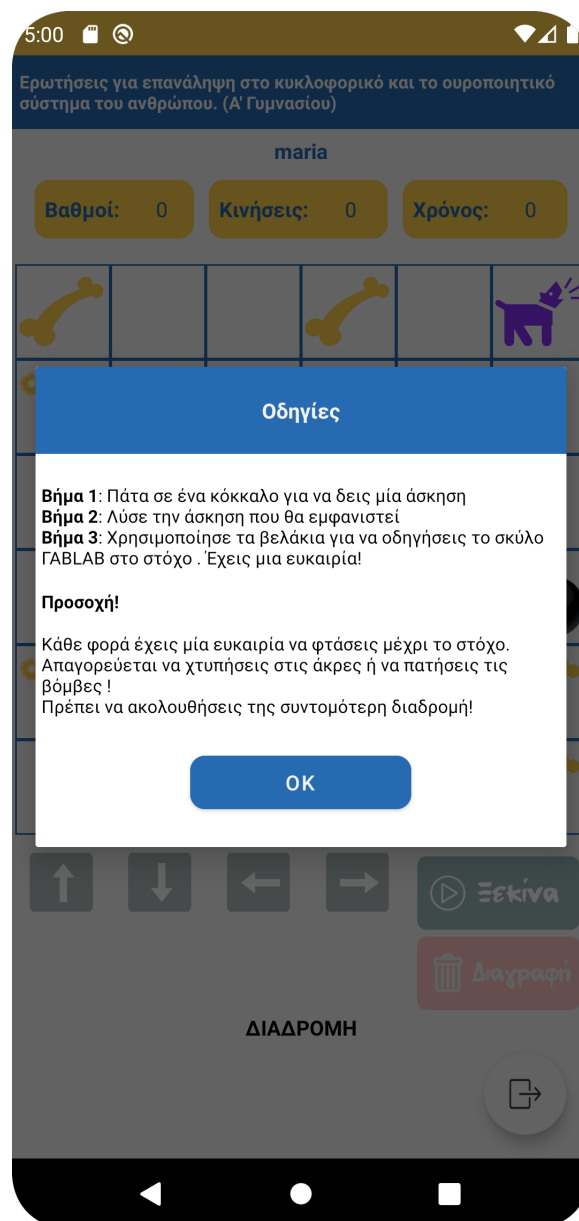


Εικόνα 1.8. Δήλωση ονόματος χρήστη

1.5.6 Οθόνη "Παιχνιδιού"

Σε αυτήν την οθόνη οδηγείται ο μαθητής όταν πατήσει στην οθόνη "Όνομα Χρήστη" το κόκκινο κουμπί δεξιά (Εικόνα 1.8). Αυτή η οθόνη είναι το ταμπλό του παιχνιδιού. Αρχικά εμφανίζει οδηγίες που πρέπει να ακολουθήσει ο παίκτης. Πατώντας το κουμπί "OK" μπορεί να ξεκινήσει το

παιχνίδι.



Εικόνα 1.9. Ταμπλό παιχνιδιού-Οδηγίες

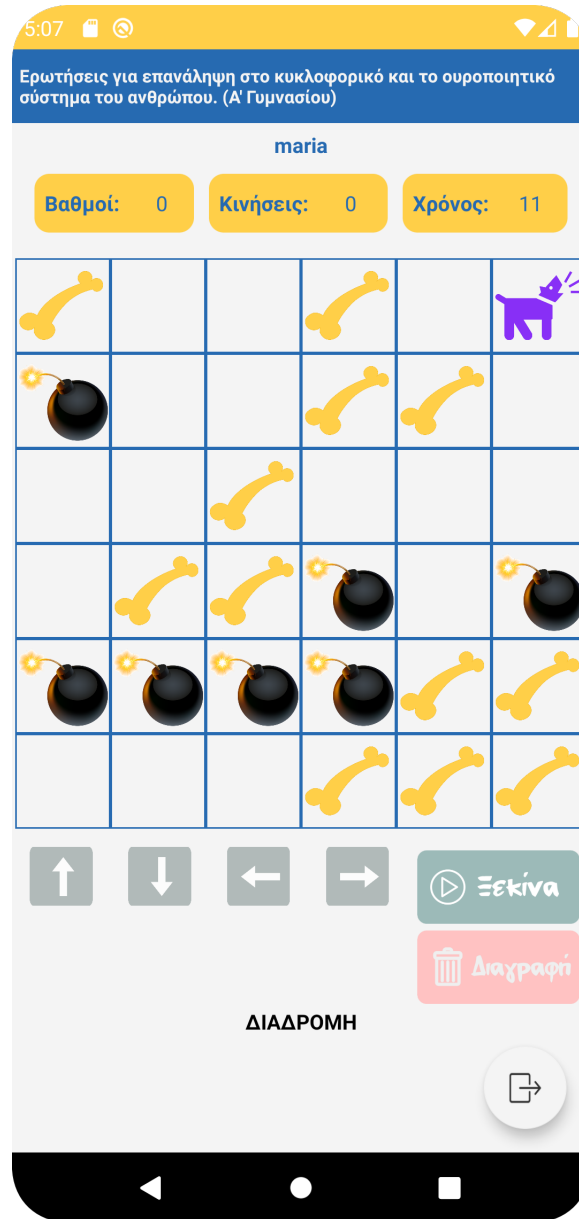
Περιγραφή Στοιχείων της οθόνης

Στην οθόνη του παιχνιδιού πανω πάνω εμφανίζεται ο τίτλος του Quiz. Αμέσως απο κάτω το όνομα του χρήστη. Απο κάτω εμφανίζονται τρία κίτρινα πλαίσια στα οποία θα εμφανίζονται η βαθμοί που κερδίζονται, οι κινήσεις που πραγματοποιεί ο μαθητής μεχρι την ερώτηση και ο χρόνος τον οποίο βρίσκονται μέσα στο συγκεκριμένο Quiz.

Στο Ταμπλό του παιχνιδιού εμφανίζονται κάποια εικονίδια. Το σκυλί είναι το πiónι που πρεπει να μετακινείται κάθε φορά. Τα κόκκαλα είναι οι ερωτήσεις του Quiz και οι βόμβες είναι κουτάκια στα οποία δεν πρέπει να πατήσει το πiónι.

Η μετακίνηση του πιονιού γίνεται με τα κουμπάκια "βελάκια". Κάθε φορά που παταμε κάποιο βελάκι καταγραφεται στο πίνακα κάτω κάτω με το όνομα "ΔΙΑΔΡΟΜΗ". Εαν ο μαθητής έκανε κάποιο λάθος στις κινήσεις μπορεί να τις διαγράψει απο την λίστα με το ουμπί "Διαγραφή".

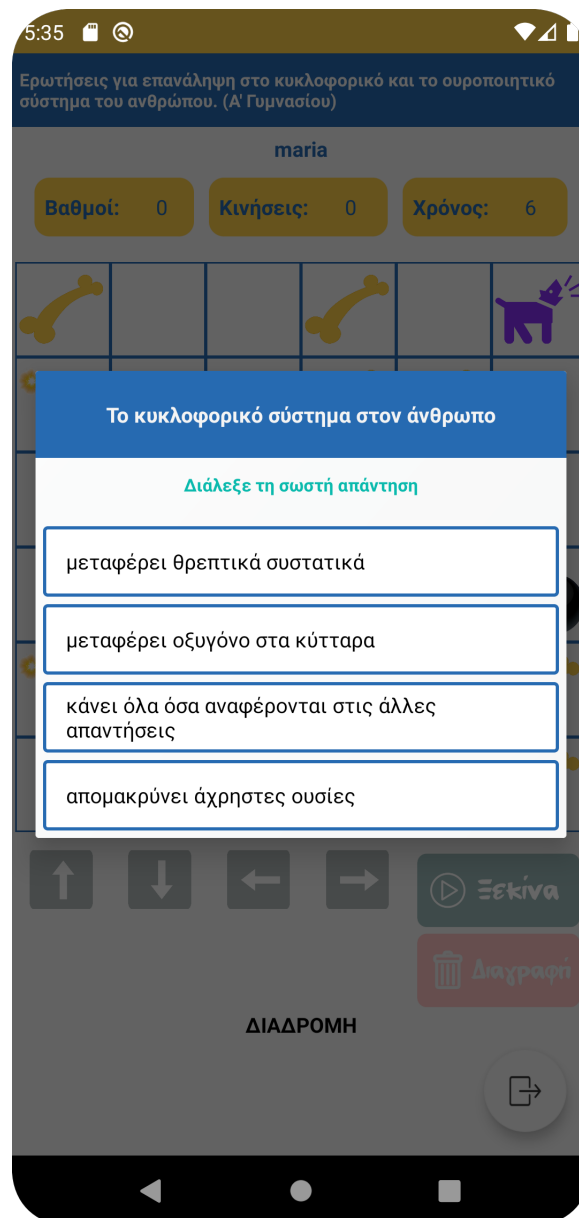
Μόλις ειμαστε σίγουροι για την διαδρομή που θα ακολουθήσει το σκυλί προς το κόκκαλο μπορούμε να πατήσουμε το κουμπί "Ξεκίνα" ώστε να αρχίσει η κίνηση του σκυλιού.



Εικόνα 1.10. Οθόνη παιχνιδιού

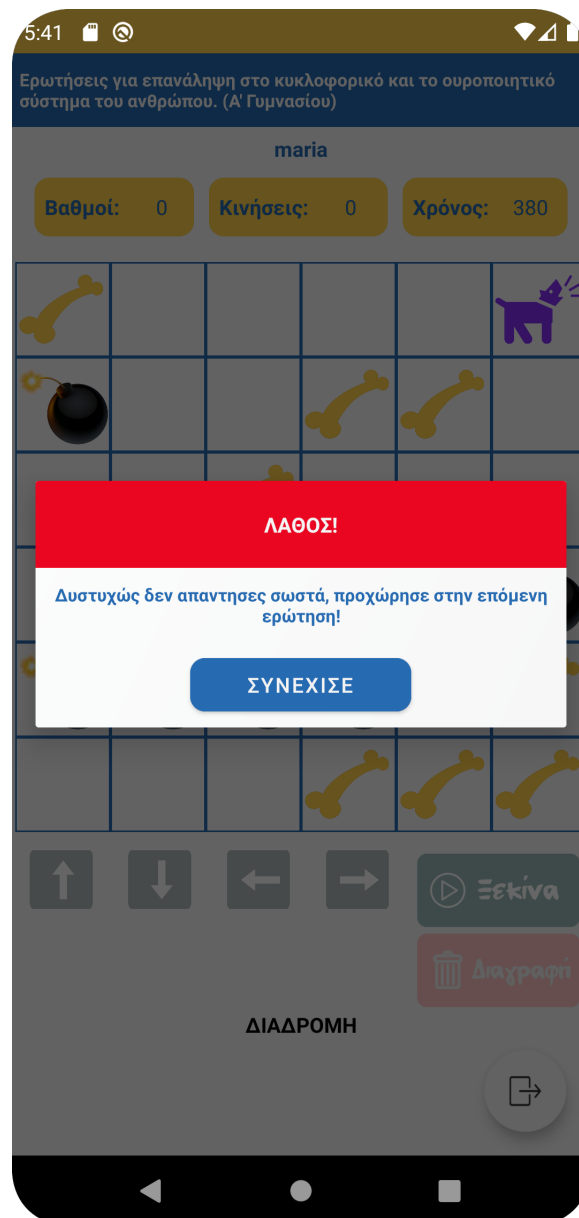
Διαδικασία παιχνιδιού

Καθε κόκκαλο όπως είπαμε αντιπροσωπεύει μία ερώτηση του quiz. Θα πρέπει ο μαθητής να επιλέξει ένα κόκκαλο για να εμφανιστεί η ερώτηση. Μόλις εμφανιστεί θα πρέπει να επιλέξουμε μία απάντηση.



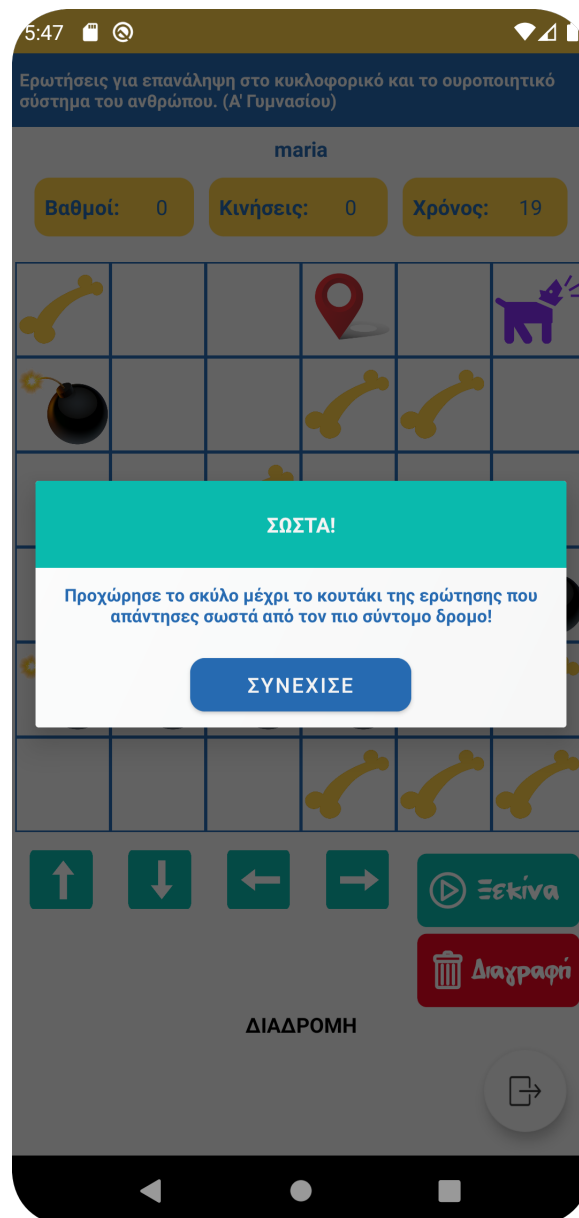
Εικόνα 1.11. Ερώτηση quiz

Εαν η απάντηση είναι λάθος εμφανίζεται το αντίστοιχο μήνυμα. Πατώντας το κουμπί "ΣΥΝΕΧΙΣΕ" παρατήρουμε ότι το κόκκαλο έχει εξαφανιστεί. Επίσης δεν μπορούμε να μετακινήσουμε το σκυλί αφού μετακινείται μόνο όταν η ερώτηση απαντηθεί σωστά.



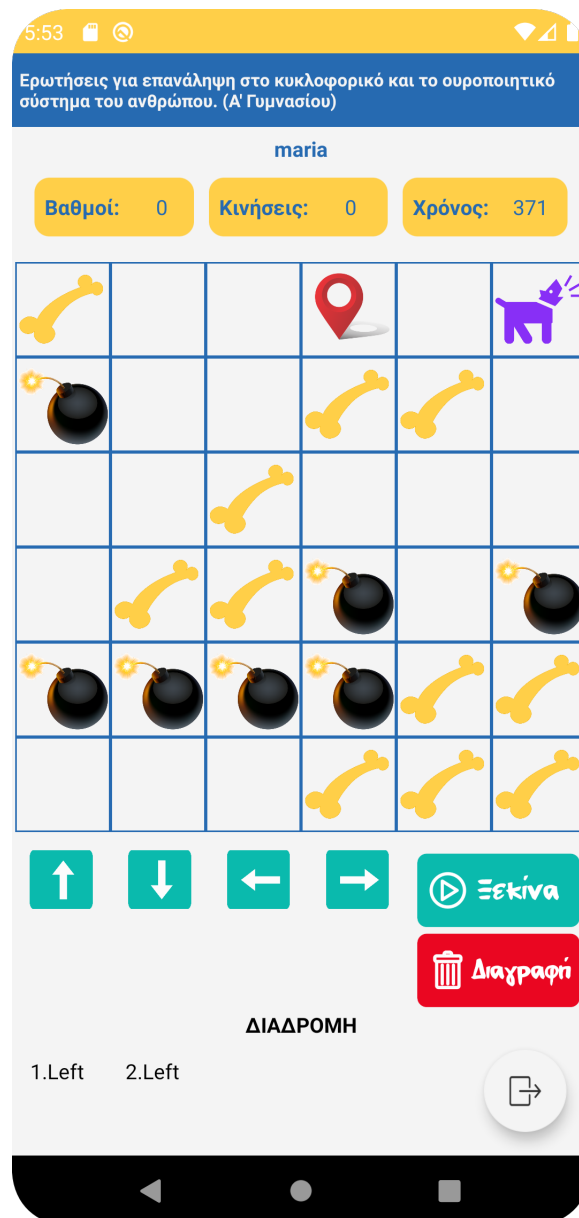
Εικόνα 1.12. Μήνυμα Λάθους

Εαν η απάντηση είναι σωστή εμφανίζεται το αντίστοιχο μήνυμα. Πατώντας το κουμπί "ΣΥΝΕΧΙΣΕ" παρατήρουμε ότι το κόκκαλο έχει αλλάξει εικονίδιο και έχει την μορφή πινέζας, δηλώνοντας ότι το σκυλί μπορεί να μετακινηθεί μέχρι εκείνο το σημείο. Επίσης ενεργοποιούνται και όλα τα στοιχεία ελεγχου για την μετακίνηση του σκυλιού.



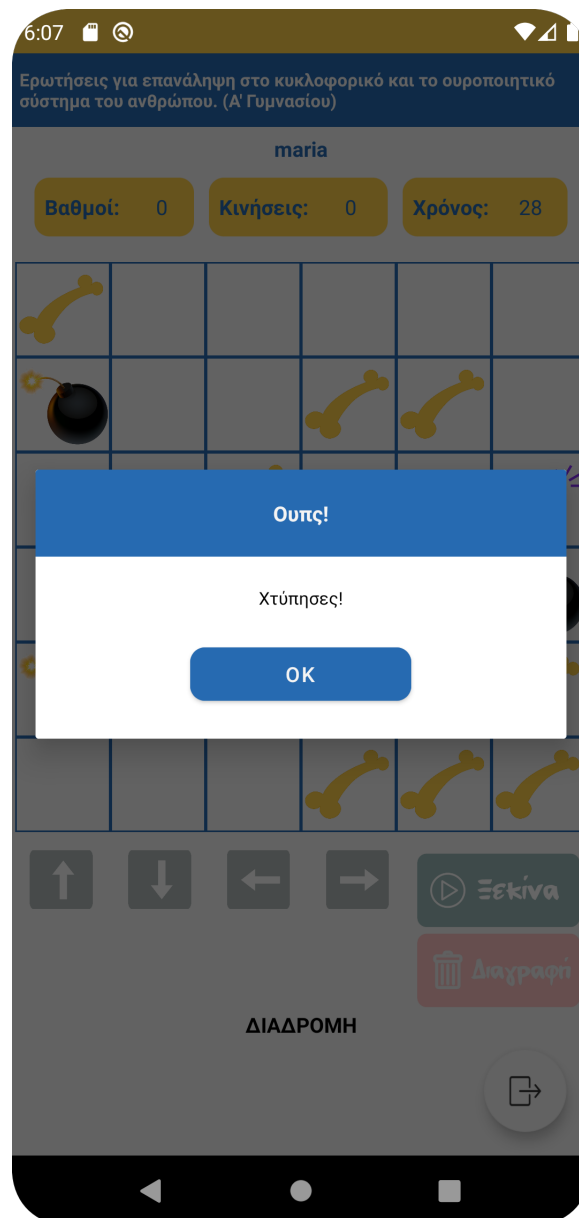
Εικόνα 1.13. Μήνυμα Σωστού

Με τα βελάκια σχηματίζουμε πρώτα την διαδρομή που θα ακολουθήσει το σκυλί προσέχοντας να μην χτυπήσει σε βόμβα ή στις άκρες του ταμπλό.



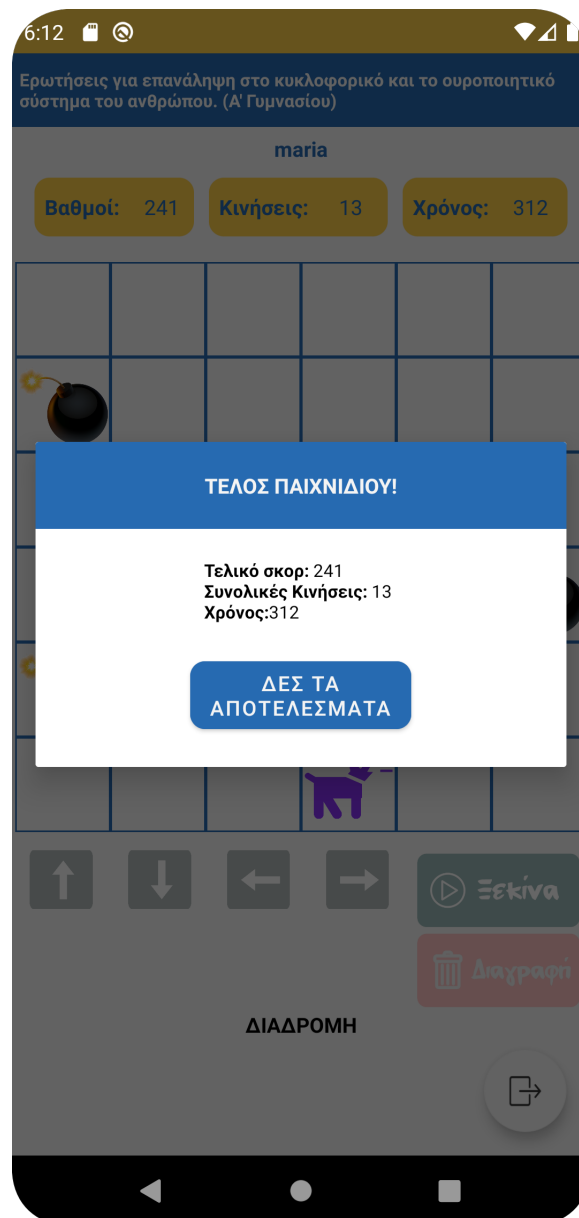
Εικόνα 1.14. Σχεδίαση διαδρομής

Στη συνέχεια μπορεί ο μαθητής να πατήσει το κουμπί "Ξεκίνα". Μόλις φτάσει στο σωτό κουτάκι εμφανίζεται ένα μήνυμα και ακούγεται και ένας συντομος ήχος επιτευξης του στόχου. Στο μήνυμα αυτό αναγράφονται οι κινήσεις που πραγματοποίησε ο μαθητής και οι πόντοι που κέρδισε. Τώρα είναι έτοιμος ο μαθητής να συνεχίσει στην επόμενη ερώτηση. Σε περίπτωση που ο μαθητής κάνει λάθος στην διαδρομή και χτυπήσει στο τοίχωμα του ταμπλό ή σε βόμβα τότε εμφανίζεται ένα μήνυμα λάθους και ο αντίστοιχος ήχος.



Εικόνα 1.15. Μήνυμα λάθους διαδρομής

Το παιχνίδι ολοκληρώνεται μόλις απαντηθούν όλες οι ερωτήσεις. Τότε εμφανίζεται ένα μήνυμα με το οποίο ενημερώνει τον μαθητή για το τελική βαθμολογία, τις συνολικές κινήσεις και τον χρόνο που χρειάστηκε για να ολοκληρώσει το quiz. Ακόμα έχει ένα κυμπί το οποίο οδηγεί στην λίστα των αποτελέσμων.



Εικόνα 1.16. Μήνυμα τέλους παιχνιδιού

1.5.7 Οθόνη "Λίστα αποτελεσμάτων"

Η οθόνη "Λίστα αποτελεσμάτων" εμφανίζεται είτε όταν ολοκληρωθεί το παιχνίδι και πατήσει ο μαθητής το κουμπί "ΔΕΣ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ" (Εικόνα 1.16) είτε από την οθόνη "Λίστα Quiz" πατώντας το μπλέ κουμπί (Εικόνα 1.7).

A/A	Username	Βαθμοί	Κινήσεις	Απαντήσεις	Χρόνος
1	βββ	899	19	11	355
2	αγγελοπουλος	891	21	12	228
3	Κυριακή Β.	891	21	12	445
4	Spyrogiorgos Papos	869	25	12	516
5	Πωργος Ψ.	857	25	33	734
6	Γιαννης Ψ.	857	25	21	1774
7	Good morning	853	19	11	356
8	zen	826	27	11	1634
9	Μανόλης	824	22	11	198
10	D 😊	824	22	11	504
11	Ιάσοντας	820	22	11	898
12	Hee	811	25	12	370
13	Μαρίνα	811	25	12	616
14	Νίκος Σιαμπάνος	811	25	12	938
15	iolii	806	23	11	974
16	καποιος	803	17	10	340
17	gsm	799	18	10	
18	οκλφηνδεθιγ				

Εικόνα 1.17. Οθόνη Λίστα Αποτελεσμάτων

Κεφάλαιο 2

Χρήση Τεχνολογιών & Τεχνολογικών Εργαλείων

2.1 Εισαγωγή

Για να υλοποιηθεί μία εφαρμογή για κινητές συσκευές χρειάζεται να καθοριστούν οι βασικές λειτουργίες της. Βάση αυτών των λειτουργιών επιλέγονται τα κατάλληλα τεχνολογικά εργαλεία προκειμένου να προχωρήσει κάποιος στην υλοποίηση της. Ειδικότερα για την υλοποίηση του Εκπαιδευτικού παιχνιδιού "Γρίφομποτ" επιλέχθηκε να αναπτυχθεί με το Android Studio και την γλώσσα προγραμματισμού JAVA.

Για την δημιουργία των layout το Android studio χρησιμοποιεί τη γλώσσα XML (eXtensible Markup Language) ώστε να περιγράψει τη διάταξη των στοιχείων UI, όπως κουμπιά, κείμενα, εικόνες και πεδία εισόδου.

Επίσης χρησιμοποιήθηκε το JavaScript Object Notation (JSON) αφού η μορφή των δεδομένων κατά την επικοινωνία της mobile εφαρμογής με τον διακομιστή έρχεται σε μορφή JSON. Για την επικοινωνία της εφαρμογής με την βάση δεδομένων χρειάστηκαν οι τεχνολογίες HTTP protocol και API (Application Programming Interface).

Τέλος, για την δημιουργία των κατάλληλων εικονιδίων χρησιμοποιήθηκαν τα προγράμματα Adobe Photoshop και Adobe Illustrator.

Στις επόμενες ενότητες παρουσιάζονται οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν.

2.2 Android studio

Android Studio



Το Android Studio αναπτύχθηκε από την Google και είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Έχει σχεδιαστεί για την ανάπτυξη εφαρμογών για συσκευές που διαθέτουν λειτουργ-

γικό σύστημα Android. Είναι διαθέσιμο για εγκατάσταση για τα λειτουργικά συστήματα Windows, MacOS, Linux. Το Android Studio βασίζεται στο IntelliJ IDEA, ένα ισχυρό ολοκληρωμένο περιβάλλον ανάπτυξης για την ανάπτυξη λογισμικού σε Java. Το Android Studio διαθέτει επιπλέον εργαλεία και λειτουργίες τα οποία είναι χρήσιμα στην ανάπτυξη εφαρμογών Android. Το Android Studio δίνει την δυνατότητα προγραμματισμού και σε άλλες γλώσσες όπως C++, kotlin.

Με το Android Studio δίνει την δυνατότητα να σχεδιάσουμε την διεπαφή χρήστη (UI Designer). Παρέχει ένα γραφικό περιβάλλον για την σχεδίαση και προβάλεται άμεσα το αποτέλεσμα. Τα αρχεία αυτά χρησιμοποιούν την γλώσσα XML (eXtensible Markup Language). Επιτρέπει τον έλεγχο και την εντοπισμό σφαλμάτων στην εφαρμογή mobile κατά τη διάρκεια της εκτέλεσης.

2.3 JAVA

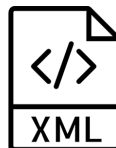


Η JAVA είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου. Αναπτύχθηκε στα μέσα τις δεκαετίας του 1990 από την Sun Microsystems, η οποία μεταγενέστερα αποκτήθηκε από την Oracle Corporation. Το συντακτικό της JAVA είναι παρόμοιο με της γλώσσας c και C++ γιατί ξεκίνησε ως μία απλοποίηση τους.

Η Java χρησιμοποιείται για ένα ευρύ φάσμα εφαρμογών όπως εφαρμογές για κινητές συσκευές, web, Desktop κα. Βασικό χαρακτηριστικό της είναι η φορητότητα του κώδικα δηλαδή ο κώδικας γράφεται μία φορά και μπορεί να εκτελεστεί σε διαφορες πλατφόρμες. Αυτό το πετυχαίνει με την βοήθεια της Java Virtual Machine.

Οι προγραμματιστές μπορούν με την JAVA να αναπτύξουν εύκολα και αποτελεσματικά εφαρμογές αφού υπάρχει ένα ευρύ φάσμα βιβλιοθηκών. Οι βιβλιοθήκες αυτές ενσωματώνουν λειτουργίες για την διαχείριση του δικτύου, ασφάλεια, βάσεις δεδομένων κα.

2.4 XML(Extensible Markup Language)



Μία από τις πιο δημοφιλής γλώσσες markup οι οποία χρησιμοποιείται ευρέως είναι η XML (Extensible Markup Language). Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και την χρησιμότητα στο διαδίκτυο. Παρέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων, ώστε αυτά να είναι αναγνώσιμα και από τον άνθρωπο και από μηχανές. Χρησιμοποιείτε συχνά για την ανταλλαγή δεδομένων και τον καθορισμό της δομής τους.

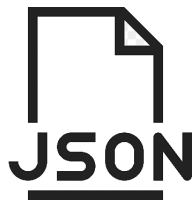
Τα αρχεία XML ακολουθούν αυστηρούς κανόνες, το "σχήμα XML" (XML Schema), που προσδιορίζει τη δομή και τις περιορισμένες επιτρεπτές τιμές των δεδομένων που περιέχονται μέσα στο αρχείο XML. Αυτό επιτρέπει την αποτελεσματική ανταλλαγή και ανάγνωση δεδομένων από διάφορες εφαρμογές. Εάν το αρχείο δεν συμμορφώνεται με αυτούς τους κανόνες, η XML σταματά να λειτουργεί. Στην XML οι ετικέτες ορίζουν τη δομή και τη σημασία των δεδομένων.

Οποιοδήποτε πρόγραμμα έχει σχεδιαστεί για να χρησιμοποιεί την XML μπορεί να διαβάσει και να επεξεργαστεί τα δεδομένα XML, ανεξάρτητα από το υλικό ή το λειτουργικό σύστημα.

```
<email>
  <to>name</to>
  <from>name2</from>
</email>
```

Εικόνα 2.1. Παράδειγμα ενός αρχείου XML

2.5 JSON (JavaScript Object Notation)



Το JSON είναι ένα πρότυπο ανταλλαγής δεδομένων και απαιτεί λιγότερη μορφοποίηση. Τα αρχεία JSON αποθηκεύονται με την επέκταση .json. Το JSON βασίστηκε στο Standard ECMA-262 3rd Edition—December 1999, το οποίο είναι υποσύνολο της JavaScript.

Τα δεδομένα JSON γράφονται σε ζεύγη κλειδί/τιμή. Το κλειδί και η τιμή χωρίζονται με άνω και κάτω τελεία. Διαφορετικά ζεύγη κλειδιών/τιμών χωρίζονται με κόμμα. Το κλειδί είναι μια συμβολοσειρά που περιβάλλεται από διπλά εισαγωγικά για παράδειγμα "label". Οι τιμές των δεδομένων μπορούν να είναι:

- String
- Boolean
- Πίνακας
- Αντικείμενο

```
{  
  "label1" : "τιμή1"  
  "label2" : true  
  "label3" : 12  
}
```

Εικόνα 2.2. Παράδειγμα ενός αρχείου JSON

2.6 Adobe Photoshop



Το Adobe Photoshop είναι ένα πρόγραμμα επεξεργασίας γραφικών ράστερ που αναπτύχθηκε και δημοσιεύτηκε από την Adobe Inc.. Είναι διαθέσιμο για εγκατάσταση για τα λειτουργικά συστήματα Windows και macOS. Δημιουργήθηκε αρχικά το 1987 από τους Thomas και John Knoll. Είναι ένα από τα πιο δημοφιλή προγράμματα το οποίο το χρησιμοποιούν επαγγελματίες φωτογράφους, σχεδιαστές γραφικών για την επεξεργασία εικόνων και γραφικών.

Το Photoshop μπορεί να επεξεργαστεί και να συνθέσει εικόνες ράστερ και δίνει την δυνατότητα αλλαγής των χρωμάτων, την προσθήκη εφέ, την διόρθωση φωτογραφικών ελαττωμάτων, την αφαίρεση ανεπιθύμητων στοιχείων και πολλά άλλα. Οι μορφές αρχείων που χρησιμοποιεί το Photoshop για να υποστηρίξει αυτές τις δυνατότητες είναι PSD και PSB. Επίσης επιτρέπει την δημιουργία προσαρμοσμένων γραφικών, τον σχεδιασμό λογοτύπου, αφισών κα.

2.7 Adobe Illustrator



Το Adobe Illustrator είναι ένα πρόγραμμα επεξεργασίας και σχεδίασης διανυσματικών γραφικών. Αναπτύχθηκε και διατέθηκε στην αγορά από την Adobe Inc. Σχεδιάστηκε αρχικά για το Apple Macintosh. Η ανάπτυξη Adobe Illustrator ξεκίνησε το 1985. Χρησιμοποιείται από επαγγελματίες σε διάφορους τομείς για την σχεδίαση διανυσματικών γραφικών.

Το Illustrator επιτρέπει την δημιουργία σχημάτων, γραμμών, την προσθήκη γραμμμάτων, την αλλαγή χρώματος, προσθήκη διαφόρων εφέ. Το Illustrator αποθηκεύει τα αρχεία σε μορφή AI (Adobe Illustrator), EPS (Encapsulated PostScript), PDF (Portable Document Format) και SVG (Scalable Vector Graphics) για να υποστηρίξει αυτές τις δυνατότητες.

2.8 HTTP Protocol(Πρωτόκολλο μεταφοράς υπερκειμένου)



Το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) είναι πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο μεταφοράς δεδομένων στον Παγκόσμιο Ιστό μεταξύ ενός διακομιστή (server) και έναν πελάτη (client). Κάθε αλληλεπίδραση μεταξύ του πελάτη και του διακομιστή ονομάζεται μήνυμα. Τα μηνύματα HTTP είναι αιτήματα ή απαντήσεις. Οι συσκευές-πελάτες υποβάλλουν αιτήματα HTTP σε διακομιστές, οι οποίοι απαντούν στέλνοντας απαντήσεις HTTP πίσω στους πελάτες. Έτσι το HTTP (Πρωτόκολλο μεταφοράς υπερκειμένου) καθορίζει μια συλλογή μεθόδων αιτήματος για να καθορίσει ποια ενέργεια πρέπει να εκτελεστεί σε έναν συγκεκριμένο πόρο. Οι πιο χρησιμοποιούμενες μέθοδοι αιτήματος HTTP είναι GET, POST, PUT, PATCH και DELETE. Αυτές είναι ισοδύναμες με τις λειτουργίες CRUD (δημιουργία, ανάγνωση, ενημέρωση και διαγραφή).

2.9 API (Application Programming Interface)



Μια διεπαφή προγραμματισμού εφαρμογών (API) είναι ένας τρόπος ανταλλαγής δεδομένων μεταξύ δύο ή περισσότερων προγράμματος υπολογιστών. Αναφέρεται σε ένα σύνολο προκαθορισμένων κανόνων, προδιαγραφών και λειτουργιών που επιτρέπουν την αλληλεπίδραση μεταξύ διαφορετικών λογισμικών εφαρμογών. Ένα API παρέχει όλες εκείνες τις λειτουργίες που απαιτούνται ώστε μία εφαρμογή να επικοινωνήσει με μία άλλη εφαρμογή.

Τα API μπορούν να υλοποιηθούν με διάφορους τρόπους, και ο καθένας έχει τα προτερήματά του. Μερικοί από αυτούς είναι

- Web Services
- SDKs (Software Development Kits)
- Βιβλιοθήκες λογισμικού

Κεφάλαιο 3

Παρουσίαση παιχνιδιού "Γρίφομποτ"

Σε αυτήν την ενότητα θα παρουσιαστεί η υλοποίηση του παιχνιδιού "Γρίφομποτ". Το "Γρίφομποτ" δημιουργήθηκε με το Android Studio. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η JAVA. Το "Γρίφομποτ" είναι μια ένα εκπαιδευτικό παιχνίδι για συσκευές που χρησιμοποιούν λειτουργικό Android. Η ελάχιστη έκδοση που μπορούν να έχουν οι κινητές συσκευές android για να το εγκαταστήσουν είναι το Marshmallow 6.0 .

Αρχικά χρειάστηκε να ρυθμιστούν τα permission που θα χρειαστεί η εφαρμογή για να λειτουργήσει. Αυτά τα permission είναι η χρήση του διαδικτύου, κατάσταση δικτύου, κατάσταση ασύρματου δικτύου και άδεια για εμφάνιση μηνυμάτων ή ειδοποιήσεων.

```
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

3.1 Κλάσεις MainActivity και MainFragment

Η κλάση "MainActivity" είναι η πρώτη που εκτελείται όταν ξεκινά η εφαρμογή και παρέχει την κεντρική διεπαφή χρήστη και την πλοήγηση σε άλλες κλάσεις. Η MainActivity περιλαμβάνει τη δημιουργία και την εμφάνιση των απαραίτητων πόρων και στοιχείων διεπαφής για την αρχική οθόνη της εφαρμογής.

```

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setSupportActionBar(binding.toolbar);
        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
        appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_help) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
        return NavigationUI.navigateUp(navController, appBarConfiguration)
            || super.onSupportNavigateUp();
    }

    public boolean isOnline() {
        ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        return cm.getActiveNetworkInfo() != null && cm.getActiveNetworkInfo().isConnectedOrConnecting();
    }
}

```

Εικόνα 3.1. MainActivity.java

Η "MainFragment" αναφέρεται στον κύριο fragment μιας εφαρμογής Android. Ένα fragment αποτελεί ένα ανεξάρτητο κομμάτι της διεπαφής χρήστη και μπορεί να ενσωματωθεί σε μια δραστηριότητα. Η MainFragment συνήθως περιλαμβάνει την κύρια λειτουργικότητα της οθόνης της εφαρμογής και μπορεί να περιέχει sub-fragments για την οργάνωση και την παρουσίαση των διαφόρων τμημάτων της διεπαφής.

```

public class MainFragment extends Fragment {
    private FragmentMainBinding binding;
    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {
        binding = FragmentMainBinding.inflate(inflater, container, attachToParent: false);
        return binding.getRoot();
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        binding.btnwithoutpin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                NavHostFragment.findNavController(MainFragment.this)
                    .navigate(R.id.action_MainFragment_to_ListQuizFragment);
            }
        });
        binding.btnwithpin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                NavHostFragment.findNavController(MainFragment.this)
                    .navigate(R.id.action_MainFragment_to_SetPinFragment);
            }
        });
        binding.tv1.setText("by ΓΑΒ LAB | v. " + BuildConfig.VERSION_NAME);
    }
}

```

Εικόνα 3.2. mainfragment.java

```

binding.fabclose.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        PopupMenu popupMenu = new PopupMenu(getContext(), binding.fabclose);
        popupMenu.getMenuInflater().inflate(R.menu.popup_menu, popupMenu.getMenu());
        popupMenu.setForceShowIcon(true);
        popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {

            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                switch(menuItem.getItemId()) {
                    case R.id.menuhelp:

                        NavHostFragment.findNavController ( fragment: MainFragment.this)
                            .navigate (R.id.action_MainFragment_to_HowToFragment);
                        break;
                    case R.id.menuabout:
                        NavHostFragment.findNavController ( fragment: MainFragment.this)
                            .navigate (R.id.action_MainFragment_to_AboutFragment);
                        break;
                    default:
                        break;
                }
                return true;
            }
        });
        popupMenu.show();
    }
});

@Override
public void onDestroyView() {
    super.onDestroyView ();
    binding = null;
}

@Override
public void onResume() {
    super.onResume();
    ActionBar supportActionBar = ((AppCompatActivity) requireActivity()).getSupportActionBar();
    if (supportActionBar != null)
        supportActionBar.hide();
}
}

```

Εικόνα 3.3. mainfragment.java

3.2 Κλάση ListQuizItemFragment

Η κλάση ListQuizItemFragment εμφανίζει την λίστα με τα διαθέσιμα ελεύθερα quiz. Ο κώδικας που κάνει κλήση το endpoint για να λάβει σαν απάντηση τα διαθέσιμα quiz φαίνεται στην εικόνα 3.4.

```

private class LoadQuiz extends AsyncTask<String,String, List<QuizModel>> {
    HttpURLConnection httpURLConnection = null;
    BufferedReader bufferedReader = null;
    String FullJsonData;

    protected List<QuizModel> doInBackground(String... strings) {
        final String REQUEST_METHOD = "GET";
        try {
            URL url = new URL( spec: "http://grifobot.gr/api/quiz/public");
            httpURLConnection = (HttpURLConnection) url.openConnection();

            httpURLConnection.setRequestMethod(REQUEST_METHOD);
            httpURLConnection.connect();
            InputStream inputStream = httpURLConnection.getInputStream();
            bufferedReader = new BufferedReader(new InputStreamReader (inputStream));
            StringBuffer stringBuffer= new StringBuffer();
            String line="";
            while ((line= bufferedReader.readLine())!= null){
                stringBuffer.append(line);
            }
            FullJsonData = stringBuffer.toString();
            List<QuizModel> jsonModelList = new ArrayList<> ();

            JSONObject jsonStartingObject = new JSONObject(FullJsonData);
            JSONArray jsonQuizArray = jsonStartingObject.getJSONArray( name: "response");
            for(int i=0; i<jsonQuizArray.length(); i++) {
                JSONObject jsonUnderArrayObject = jsonQuizArray.getJSONObject(i);
                QuizModel jsonModel = new QuizModel ();
                jsonModel.setId(jsonUnderArrayObject.getInt ( name: "id"));
                jsonModel.setTitle (jsonUnderArrayObject.getString( name: "name"));
                jsonModel.setDescription (jsonUnderArrayObject.getString( name: "description"));
                jsonModel.setSize (jsonUnderArrayObject.getInt ( name: "size"));
                jsonModel.setLevel (jsonUnderArrayObject.getString( name: "level"));
                jsonModel.setExercises(jsonUnderArrayObject.getInt( name: "exercises"));
                jsonModel.setPin(jsonUnderArrayObject.getString( name: "pin"));
                jsonModelList.add(jsonModel);
            }
            return jsonModelList;
        } catch (IOException e) {
            e.printStackTrace();
        } catch (JSONException e) {
            e.printStackTrace();
        } finally{
            httpURLConnection.disconnect();
            try {
                bufferedReader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
}

```

Εικόνα 3.4. ListQuizItemFragment.java : Κλήση endPoint "Λήψη λίστας των public quiz"

3.3 Κλάση SetPinFragment

Η κλάση SetPinFragmentet είναι η κλάση η οποία καλείτε όταν ένας μαθητής διαλέξει στην αρχική οθόνη "Έχω PIN". Μόλις συμπληρώσει το pin, το όνομα χρήστη και πατηθεί το κουμπί επόμενο τότε θα γίνει κλήση του endPoint "Λήψη του ID ενός Quiz από ένα PIN". Ο κώδικας που κάνει κλήση το endpoint για να λάβει σαν απάντηση το id quiz φαίνεται στην εικόνα 3.5 και 3.6.

```

private void LoadID(String qpin, View view) {
    RequestQueue queue = Volley.newRequestQueue (getContext ( ));

    StringRequest sr = new StringRequest (Request.Method.POST, url: "http://grifobot.gr/api/quiz/id/pin",
    new Response.Listener<String> ( ) {
        @Override
        public void onResponse(String response) {

            try {
                onGameModel = new OnGameModel ();
                List<GameModel> jsonModelListGame = new ArrayList<> ( );
                JSONObject jsonQuizObject= new JSONObject (response);
                JSONObject jsonQuizObject2= jsonQuizObject.getJSONObject ("response");
                txtid = Integer.valueOf (jsonQuizObject2.getString( name: "id"));
                if (txtid>0){
                    Bundle bundle = new Bundle();
                    bundle.putInt ("id",txtid);
                    bundle.putString ("pin",txtpin);
                    bundle.putString ("name",txtName.getText ().toString ());
                    SaveUserName("username", txtName.getText ().toString ());
                    NavHostFragment.findNavController ( fragment: SetPinFragment.this)
                        .navigate (R.id.action_SetPinFragment_to_GameFragment, bundle);
                }
            } catch (Throwable t) {
                Log.e ( tag: "Grifobot", msg: "Could not parse malformed JSON: \"" + response + "\"");
                new AlertDialog.Builder(getContext ())
                    .setTitle("")
                    .setMessage("Κατ' όσον ήμεις σωστό. Δοκιμάστε ξανά.")
                    .setCancelable(false)
                    .setPositiveButton( text: "ok", new DialogInterface.OnClickListener () {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.cancel ();
                        }
                    })
                    .show();
            }
        }
    }
}
}

```

Εικόνα 3.5. SetPinFragment.java : Κλήση endPoint "Λήψη του ID ενός Quiz από ένα PIN" 1

```

}, new Response.ErrorListener ( ) {
    @Override
    public void onErrorResponse(VolleyError error) {
        new AlertDialog.Builder(getContext ())
            .setTitle("")
            .setMessage("Δεν βρέθηκε το αντίστοιχο quiz")
            .setCancelable(false)
            .setPositiveButton( text: "ok", new DialogInterface.OnClickListener () {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.cancel ();
                }
            })
            .show();
    }
}) {
    @Override
    protected Map<String, String> getParams() {
        Map<String, String> params = new HashMap<> ( );
        params.put ( k: "pin", qpin);
        return params;
    }
    @Override
    public Map<String, String> getHeaders() throws AuthFailureError {
        Map<String, String> params = new HashMap<> ( );
        params.put ( k: "Content-Type", v: "application/x-www-form-urlencoded");
        return params;
    }
};
queue.add (sr);
}

```

Εικόνα 3.6. ListQuizItemFragment.java : Κλήση endPoint "Λήψη του ID ενός Quiz από ένα PIN" 2

3.4 Κλάση ResultFragment

Η κλάση ResultFragment εμφανίζει την λίστα με τα αποτελέσματα ενός quiz. Ο κώδικας που κάνει κλήση το endpoint για να λάβει σαν απάντηση τα αποτελέσματα ενός quiz φαίνεται στην εικόνα 3.7.

```

private void LoadResult(Integer qid){
    RequestQueue queue = Volley.newRequestQueue(getApplicationContext());
    StringRequest sr = new StringRequest(Request.Method.POST, url: "http://grifobot.gr/api/quiz/scores",
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONObject jsonQuizArray = new JSONObject(response);
                    List<ResultModel> jsonModelList = new ArrayList<> ();
                    JSONArray responseArray = (JSONArray) jsonQuizArray.get("response");
                    for(int i=0; i<responseArray.length(); i++) {

                        JSONObject jsonUnderArrayObject = responseArray.getJSONObject(i);
                        ResultModel jsonModel = new ResultModel ();
                        jsonModel.setUsername (jsonUnderArrayObject.getString ( name: "username"));
                        jsonModel.setQuestions (jsonUnderArrayObject.getInt ( name: "questions"));
                        jsonModel.setMovements (jsonUnderArrayObject.getInt ( name: "movements"));
                        jsonModel.setScore (jsonUnderArrayObject.getInt ( name: "score"));
                        jsonModel.setSeconds (jsonUnderArrayObject.getInt ( name: "seconds"));
                        jsonModelList.add (jsonModel);
                    }
                    List<ResultModel> jsonModels = jsonModelList;
                    ResultQuizListAdapter = new ResultQuizListAdapter (getActivity (), jsonModels, username);
                    recyclerView.setAdapter(ResultQuizListAdapter);
                    progressBar.setVisibility(View.GONE);
                } catch (Throwable t) {
                    Log.e( tag: "GrifoBot", msg: "Could not parse malformed JSON: \"" + response + "\"");
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
            }
        }) {
            @Override
            protected Map<String,String> getParams(){
                Map<String,String> params = new HashMap<> ();
                params.put( k: "id", String.valueOf (qid));
                return params;
            }
            @Override
            public Map<String, String> getHeaders() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
                params.put( k: "Content-Type", v: "application/x-www-form-urlencoded");
                return params;
            }
        };
    queue.add(sr);
}

```

Εικόνα 3.7. ResultFragment.java : Κλήση endPoint “Προβολή αποτελεσμάτων ενός Quiz σύμφωνα με το ID του”

3.5 Κλάση GameBoardFragment

Η κλάση GameBoardFragment είναι η κλάση που διαχειρίζεται το παιχνίδι.

Λήψη στοιχείων ενός quiz

Ο κώδικας που κάνει κλήση το endpoint για να λάβει σαν απάντηση τα στοιχεία ενός quiz φαίνεται στις εικόνες 3.8 , 3.9, 3.10, 3.11 και 3.12.

```

//*****LOAD GAME DATA***//
private void LoadGame(Integer gameId) {
    RequestQueue queue = Volley.newRequestQueue (getContext ( ));

    StringRequest sr = new StringRequest (Request.Method.POST, url: "http://grifobot.gr/api/quiz/play/game",
    new Response.Listener<String> ( ) {

        @Override
        public void onResponse(String response) {

            try {
                ongameModel = new OnGameModel();
                gameModel = new GameModel ( );
                List<GameModel> jsonModelListGame = new ArrayList<> ( );
                List<GameBoardModel> jsonModelListBoardGame = new ArrayList<> ( );
                List<GameQuestionsModel> jsonModelListGameQuestion = new ArrayList<> ( );
                JSONObject jsonQuizArray = new JSONObject (response);

                JSONArray responseArray = (JSONArray) jsonQuizArray.get ( "response" );

                for (int i = 0; i < responseArray.length ( ); i++) {

                    JSONObject jsonUnderArrayObject = responseArray.getJSONObject (i);
                    //GameModel
                    GameModel jsonModel = new GameModel ( );
                    gameModel.setId (jsonUnderArrayObject.getInt ( name: "id"));
                    gameModel.setTitle (jsonUnderArrayObject.getString ( name: "name"));
                    gameModel.setDescription (jsonUnderArrayObject.getString ( name: "description"));
                    gameModel.setSize (jsonUnderArrayObject.getInt ( name: "size"));
                    gameModel.setLevel (jsonUnderArrayObject.getString ( name: "level"));
                    gameModel.setExercises (jsonUnderArrayObject.getInt ( name: "exercises"));
                    gameModel.setPin (jsonUnderArrayObject.getString ( name: "pin"));
                    gameModel.setUuid (jsonUnderArrayObject.getString ( name: "uuid"));
                    jsonModelListGame.add (gameModel);
                }
            }
        }
    });
}

```

Εικόνα 3.8. GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-1

```

//GameBoardModel --"code"
JSONObject responseArray1 = jsonUnderArrayObject.getJSONObject ("code");
for (int j = 0; j < responseArray1.length ( ); j++) {

    Integer position = j + 1;
    JSONObject jsonUnderArrayObject1 = responseArray1.getJSONObject (position.toString ( ));
    GameBoardModel gameboardmodel = new GameBoardModel ( );
    gameboardmodel.setId (jsonUnderArrayObject1.getInt ( name: "id"));
    gameboardmodel.setBlocked (jsonUnderArrayObject1.getString ( name: "blocked"));
    gameboardmodel.setPlayer (jsonUnderArrayObject1.getString ( name: "player"));
    gameboardmodel.setSolved (jsonUnderArrayObject1.getString ( name: "solved"));
    gameboardmodel.setMath (jsonUnderArrayObject1.getString ( name: "math"));
    gameboardmodel.setMathEq (jsonUnderArrayObject1.getString ( name: "mathEq"));
    if ((jsonUnderArrayObject1.getString ( name: "player"))=="true"){
        ongameModel.setPosavatar (String.valueOf (j));
        ongameModel.setStartPos (String.valueOf (j));
    }
    jsonModelListBoardGame.add (gameboardmodel);
}
}

```

Εικόνα 3.9. GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-2

```

//GameQuestionModel --"maths"
JSONArray responseArray2 = (JSONArray) jsonUnderArrayObject.get ("maths");
for (int j = 0; j < responseArray2.length ( ); j++) {
    JSONObject jsonUnderArrayObject2 = responseArray2.getJSONObject (j);
    GameQuestionsModel gamequestionmodel = new GameQuestionsModel ( );
    gamequestionmodel.setId (jsonUnderArrayObject2.getInt ( name: "id"));
    gamequestionmodel.setQuestion (jsonUnderArrayObject2.getString ( name: "question"));
    gamequestionmodel.setCorrect (jsonUnderArrayObject2.getString ( name: "correct"));
    gamequestionmodel.setAlt1 (jsonUnderArrayObject2.getString ( name: "alt1"));
    gamequestionmodel.setAlt2 (jsonUnderArrayObject2.getString ( name: "alt2"));
    gamequestionmodel.setAlt3 (jsonUnderArrayObject2.getString ( name: "alt3"));
    gamequestionmodel.setAlt4 (jsonUnderArrayObject2.getString ( name: "alt4"));
    jsonModelListGameQuestion.add (gamequestionmodel);
}
}

```

Εικόνα 3.10. GameBoardFragment.java : Κλήση endPoint "Λήψη των στοιχείων ενός Quiz από το ID του"-3

```

List<GameQuestionsModel> jsonModelsQuestion = jsonModelListGameQuestion;
List<GameModel> jsonModelsList= jsonModelListGame;
gameboardmodel= jsonModelListBoardGame;

//ΥΠΟΛΟΓΙΣΤΕ ΤΙΣ ΣΤΗΛΕΣ ΤΟΥ ΠΛΕΓΜΑΤΟΣ
int numberOfColumns = (int) Math.sqrt (gameboardmodel.size ( ));
//ΕΝΗΜΕΡΩΣΕ ΜΕ ΤΙΣ ΣΤΗΛΕΣ ΤΟΥ ΠΛΕΓΜΑΤΟΣ
ongamemodel.setColumngrid(String.valueOf (numberOfColumns));
//ΑΡΧΙΚΟΠΟΙΗΣΕ ΤΟ ΠΙΝΑΚΑ
recyclerView.setLayoutManager (new GridLayoutManager (getActivity ( ), numberOfColumns));
gmadapter=new GameBoardAdapter (gameboardmodel,jsonModelsQuestion, jsonModelsList);
recyclerView.setAdapter (gmadapter);
//ΕΝΗΜΕΡΩΣΕ ΜΕ ΤΟ ΣΥΝΟΛΟ ΤΩΝ ΕΡΩΤΗΣΕΩΝ
ongamemodel.setExercises (String.valueOf (jsonModelsList.get(0).getExercises ()));
ongameclass.LENDOFFGAME.setValue(ongamemodel.getExercises ());
//Log.d (" ongamemodel.setExercises", ongamemodel.getExercises());
ongamemodel.setTotalMoves(0);
txtTotalScore.setText (String.valueOf (ongamemodel.getTotalScores()));
txtTotalMoves.setText (String.valueOf (ongamemodel.getTotalMoves()));
// Setting up the events that will occur on each Main-List item click
gmadapter.setWhenClickListener (new GameBoardAdapter.OnItemsClickListener () {
    @Override
    public void onItemClick(Integer Questionposition, String IDQuestion, LinearLayout relativeLayout) {
        //Εάν δεν έχει πατηθεί άλλη ερώτηση
        if (ongamemodel.getPosquestion()<0){
            if (PathArray.size ()!=0){
                PathArray.clear ();
                gmpathadapter.UpdateData(PathArray);
            }
            //ΕΝΗΜΕΡΩΣΕ ΜΕ ΤΗΝ ΘΕΣΗ ΤΗΣ ΕΡΩΤΗΣΗΣ
            ongamemodel.setPosquestion (Questionposition);
            //ΑΝΑΖΗΤΗΣΕ ΤΗΝ ΕΡΩΤΗΣΗ
            List<GameQuestionsModel> searchQuestion = filterQuestion(IDQuestion,jsonModelListGameQuestion);
            SetPopupDialog(Questionposition,searchQuestion,relativeLayout);
        }
    }
});

```

Εικόνα 3.11. GameBoardFragment.java : Κλήση endPoint ”Λήψη των στοιχείων ενός Quiz από το ID του”-4

```

        btnStart.setOnClickListener(btnStartMoveListener);
        progressBar.setVisibility(View.GONE); //to show
    } catch (Throwable t) {
        Log.e ( tag: "Grifobot", msg: "Could not parse malformed JSON: \"" + response + "\"");
    }

    }, new Response.ErrorListener () {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<> ( );
            params.put (k: "id", String.valueOf (qid));
            return params;
        }
        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> params = new HashMap<> ( );
            params.put (k: "Content-Type", v: "application/x-www-form-urlencoded");
            return params;
        }
    };
    queue.add (sr);
}

```

Εικόνα 3.12. GameBoardFragment.java : Κλήση endPoint ”Λήψη των στοιχείων ενός Quiz από το ID του”-5

Αναζήτηση Ερώτησης

Η αναζήτηση της ερώτησης όταν καποιος μαθητής επιλέγει ένα κουτάκι με ερώτηση γίνεται με τον κώδικα που φαίνεται στην εικόνα [3.13](#)

```

//*****FIND SPECIFIC QUESTION*****
// FIND SPECIFIC QUESTION
//*****
public List<GameQuestionsModel> filterQuestion( String questionid, List<GameQuestionsModel> listdataQuestion) {
    List<GameQuestionsModel> searchQuestion = new ArrayList<> ( );
    for (int i = 0; i < listdataQuestion.size ( ); i++) {
        if (listdataQuestion.get (i).getId ( ).equals (Integer.valueOf (questionid))) {
            searchQuestion.add (listdataQuestion.get (i));
        }
    }
    return searchQuestion;
}
}

```

Εικόνα 3.13. GameBoardFragment.java : Αναζήτηση Ερώτησης

Αποθήκευση αποτελέσματος

Ο κώδικας που κάνει κλήση το endpoint για να αποθηκεύσει το αποτέλεσμα φαίνεται στην εικόνα 3.14.

```

//*****Save GAME DATA -EndPoint****/
private void SaveGameResult(Integer qid, Integer QuestionMoves, Integer QuestionScore) {
    RequestQueue queue = Volley.newRequestQueue (getContext ( ));

    StringRequest sr = new StringRequest (Request.Method.POST, url: "http://grifobot.gr/api/quiz/record/score",
        new Response.Listener<String> ( ) {
            @Override
            public void onResponse(String response) {
                try {
                    Log.e ( "tag: "Grifobot", response);
                } catch (Throwable t) {
                    Log.e ( "tag: "Grifobot", msg: "CouId not parse malformed JSON: \" + response + "\"");
                }
            }
        }, new Response.ErrorListener ( ) {
            @Override
            public void onErrorResponse(VolleyError error) {
            }
        }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<> ( );
            params.put ( k: "id", String.valueOf (qid));
            params.put ( k: "username", String.valueOf (username));
            params.put ( k: "score", String.valueOf ( QuestionScore));
            params.put ( k: "movements", String.valueOf (QuestionMoves));
            params.put ( k: "uuid", String.valueOf (gameModel.getUuid ( ));
            return params;
        }
        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> params = new HashMap<> ( );
            params.put ( k: "Content-Type", v: "application/x-www-form-urlencoded");
            return params;
        }
    };
    queue.add (sr);
}

```

Εικόνα 3.14. GameBoardFragment.java : Κλήση endPoint "Καταγραφή score ενός χρήστη σύμφωνα με το ID"

Κίνηση πιονιού(σκύλος)

Η κίνηση του πιονιού πραγματοποιείτε με τον κώδικα που φαίνεται στην εικόνα 3.15

```

switch(btnn)
{
    case "Up"://RAND
        step=curposAvatar-colnum;
        if (step<0|| step >colnum*colnum||ongameclass.isBlocked (step, gameboardmodel)==true){
            blog=true;
        }
        break;
    case "Down"://KATO
        step =curposAvatar+colnum;
        if (step <0|| step>colnum*colnum||ongameclass.isBlocked (step, gameboardmodel)==true){
            blog=true;
        }
        break;
    case "Right"://ΔΕΞΙΑ
        step =curposAvatar+1;
        if (step<0|| step >colnum*colnum||(cell%colnum==0) ||ongameclass.isBlocked (step, gameboardmodel)==true){
            blog=true;
        }
        break;
    case "Left": //ΑΡΙΣΤΕΡΑ
        step =curposAvatar-1;
        if (step <0|| step >colnum*colnum||(cell%colnum==1) ||ongameclass.isBlocked (step, gameboardmodel)==true){
            blog=true;
        }
        break;
}

```

Εικόνα 3.15. GameBoardFragment.java : κίνηση του πιονιού

3.6 Κλάση HowToFragment

Η κλάση HowToFragment είναι η κλάση με την οποία εμφανίζονται οι οδηγίες στους μαθητές όταν επιλέξουν να τις εμφανίσουν μέσα από το μενού στην αρχική οθόνη.

```

public class HowToFragment extends Fragment {
    private FragmentHowToBinding binding;
    private FloatingActionButton fabicon;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        binding = FragmentHowToBinding.inflate (inflater, container, attachToParent: false);

        fabicon = (FloatingActionButton)binding.fabclose;

        fabicon.setOnClickListener (new View.OnClickListener () {
            @Override
            public void onClick(View view) {
                getActivity ().onBackPressed();
            }
        });

        // Inflate the layout for this fragment
        return binding.getRoot ( );
    }

    @Override
    public void onResume() {
        super.onResume();
        ActionBar supportActionBar = ((AppCompatActivity) requireActivity()).getSupportActionBar();
        if (supportActionBar != null)
            supportActionBar.hide();
    }
}

```

Εικόνα 3.16. HowToFragment.java

3.7 Κλάση OnGameClass

Η κλάση OnGameClass είναι η κλάση η οποία περιέχει βοηθητικές συναρτήσεις για το παιχνίδι. Μερικές απο αυτές είναι η συνάρτηση για την εμφάνιση των μηνυμάτων, η συνάρτηση για το ξεκίνημα του χρόνου στην αρχή του παιχνιδιού και το σταμάτημα στο τέλος του παιχνιδιού, η συνάρτηση για την ενημέρωση του ταμπλό με την κίνηση του πιονιού (σκύλος) κα. Μερικές απο τις συναρτήσεις φαίνονται στις παρακάτω εικόνες:

```

public void wrongDialog(Context mContext, Fragment fragment, String qusername, OnGameModel ongameModel, GameModel gameModel){
    final Dialog dialognew = new Dialog(mContext);
    dialognew.requestWindowFeature(Window.FEATURE_NO_TITLE);
    dialognew.setCancelable(false);
    dialognew setContentView(R.layout.modal_message_wrong);

    Button dialogButton = (Button) dialognew.findViewById(R.id.btn_yes);
    dialogButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dialognew.dismiss();
            if (ENDGAME==Boolean.TRUE){
                OnGameClass.EndGameDialog(mContext, fragment, ongameModel.getTotalScores (),
                    ongameModel.getTotalMoves (), ongameModel.getTime (), gameModel.getUuid (), qusername, gameModel.getId ());
            }
        }
    });

    dialognew.show();
    dialognew.getWindow().
        setLayout(WindowManager.LayoutParams.MATCH_PARENT, WindowManager.LayoutParams.WRAP_CONTENT);
}

```

Εικόνα 3.17. OnGameClass.java - Συνάρτηση για την εμφάνιση μηνυμάτων

```

public static void HitPathDialog(Context mContext, Fragment fragment, String qusername, OnGameModel ongameModel, GameModel gameModel){
    final Dialog dialognew = new Dialog(mContext);
    dialognew.requestWindowFeature(Window.FEATURE_NO_TITLE);
    dialognew.setTitle ("!");
    String odigies = "Χτύπησε!";
    dialognew.setCancelable(false);
    dialognew setContentView(R.layout.modal_message_first);
    TextView dialogTitleText= (TextView) dialognew.findViewById(R.id.txttitledialog);
    TextView dialogText= (TextView) dialognew.findViewById(R.id.txtdialog);
    dialogTitleText.setText ("Ουμ!");
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
        dialogText.setText(Html.fromHtml(odigies, Html.FROM_HTML_MODE_COMPACT));
    } else {
        dialogText.setText(Html.fromHtml(odigies));
    }

    Button dialogButton = (Button) dialognew.findViewById(R.id.btn_ok);
    dialogButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dialognew.dismiss();
            WrongPathDialog (mContext, fragment, qusername, ongameModel, gameModel);
        }
    });

    dialognew.show();
    dialognew.getWindow().
        setLayout(WindowManager.LayoutParams.MATCH_PARENT, WindowManager.LayoutParams.WRAP_CONTENT);
}

```

Εικόνα 3.18. OnGameClass.java

```

public static void onStartTimer(TextView timerView){
    startTime = System.currentTimeMillis();

    handler.post(myRunnable = new Runnable() {
        @Override
        public void run() {
            long diff = System.currentTimeMillis() - startTime;
            timerView.setText(Long.toString( diff / 1000));
            handler.postDelayed( this, delayMillis: 1000);
        }
    });
}

```

Εικόνα 3.19. OnGameClass.java - Συνάρτηση για το ξεκίνημα του χρόνου

```

public Integer scoreCalculate(Integer move){
    Integer score=0;

    score=100/move;
    return score;
}

```

Εικόνα 3.20. OnGameClass.java - Συνάρτηση για τον υπολογισμό της βαθμολογίας

```

//ENABLE-DISABLE BUTTON
public void togglebutton(LinearLayout viewbtttn, LinearLayout viewbtttn2, Boolean cast){
    for ( int i = 0; i < viewbtttn.getChildCount(); i++ ){
        View view1 = viewbtttn.getChildAt(i);
        view1.setEnabled(cast);
    }

    for ( int i = 0; i < viewbtttn2.getChildCount(); i++ ){
        View view2 = viewbtttn2.getChildAt(i);
        view2.setEnabled(cast);
    }
}

```

Εικόνα 3.21. OnGameClass.java - Συνάρτηση για την ενεργοποίηση των κουμπιών

```

public boolean isBLocked(Integer position, List<GameBoardModel> modeldata){
    GameBoardModel myListData = modeldata.get (position);
    Boolean bomb=false;

    if (myListData.getBLocked ()=="true"){
        bomb=true;
    }

    return bomb;
}

//MOVE AVATAR
public void moveAvatar(Integer curposAvatar,Integer newcurposAvatar, List<GameBoardModel> gameboardmodel, GameBoardAdapter gmadapter){
    //*****
    //Ενημέρωσε την παλιά θέση του AVATAR
    //*****
    GameBoardModel ItenQuestion = gameboardmodel.get (curposAvatar);
    ItenQuestion.setPlayer ("false");

    //This will trigger onBindViewHolder method from the adapter.
    gmadapter.notifyItemChanged (curposAvatar);

    //*****
    //Ενημέρωσε την νέα θέση του AVATAR
    //*****
    GameBoardModel ItenQuestion1= gameboardmodel.get (newcurposAvatar);
    ItenQuestion1.setPlayer ("true");

    gmadapter.notifyItemChanged (newcurposAvatar);
}

```

Εικόνα 3.22. OnGameClass.java - Συνάρτησεις για τον έλεγχο αν ένα κουτακι περιέχει "βόμβα" και συνάρτηση για την κίνηση του πιονιού

Κεφάλαιο 4

Συμπεράσματα

Η εκπαίδευση δεν πρέπει να περιορίζεται στις παραδοσιακές μεθόδους διδασκαλίας γιατί η τεχνολογία εξελίσσεται με γρήγορους ρυθμούς. Γι' αυτό εμφανίζονται συνεχώς νέα εργαλεία, τεχνικές και πρακτικές που θα μπορούσαν να φανούν χρήσιμες στην εκπαιδευτική διαδικασία. Κάθε μαθητής έχει έναν διαφορετικό τρόπο μάθησης και αντιλαμβάνεται την πληροφορία με διαφορετικό τρόπο. Τα εκπαιδευτικά παιχνίδια σχεδιάζονται έτσι ώστε να είναι παράλληλα και εκπαιδευτικά αλλά και διασκεδαστικά ώστε να κεντρίσει το ενδιαφέρον των μαθητών. Γι' αυτό και ο συνδυασμός των εκπαιδευτικών παιχνιδιών με τις παραδοσιακές μεθόδους εκπαίδευσης μπορεί να προσφέρει μια πλούσια εκπαιδευτική εμπειρία για τους μαθητές.

Βιβλιογραφία

- [1] *Adobe Photoshop*. URL: https://en.wikipedia.org/wiki/Adobe_Photoshop.
- [2] *Android Studio*. URL: https://en.wikipedia.org/wiki/Android_Studio.
- [3] *API*. URL: <https://en.wikipedia.org/wiki/API>.
- [4] *HTTP - Requests*. URL: https://www.tutorialspoint.com/http/http_requests.htm.
- [5] *HTTP Requests*. URL: <https://www.codecademy.com/article/http-requests>.
- [6] *Illustrator vs. Photoshop: When to use each application*. URL: <https://www.adobe.com/creativecloud/design/illustrator-vs-photoshop.html>.
- [7] *Java (programming language)*. URL: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).
- [8] *What is Adobe Illustrator*. URL: https://en.wikipedia.org/wiki/Adobe_Photoshopagittraining.com/adobe/illustrator/classes/what-is-adobe-illustrator.
- [9] *What is an API*. URL: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>.
- [10] *What is Java technology and why do I need it?* URL: https://www.java.com/en/download/help/whatis_java.html.
- [11] *What is JSON*. URL: https://www.w3schools.com/whatis/whatis_json.aspL.
- [12] *What is XML*. URL: https://www.w3schools.com/xml/xml_whatis.asp.
- [13] *Γριφομπότ*. URL: <http://grifobot.gr/>.