ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
&
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
&
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδίαση, υλοποίηση και αξιολόγηση προγραμματιστικού περιβάλλοντος για την πλατφόρμα κινητών συσκευών App Inventor

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

της

ΤΣΟΛΑΚΗ ΣΤΑΜΑΤΙΑΣ

ΑΕΜ: 200

Επιβλέπων : **ΒΕΡΓΑΔΟΣ Ι. ΔΗΜΗΤΡΙΟΣ**
Επίκουρος Καθηγητής

Καστοριά Μάρτιος - 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
&
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
&
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδίαση, υλοποίηση και αξιολόγηση προγραμματιστικού περιβάλλοντος για την πλατφόρμα κινητών συσκευών App Inventor

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

της

ΤΣΟΛΑΚΗ ΣΤΑΜΑΤΙΑΣ

(ΑΕΜ: 200)

Επιβλέπων : **ΒΕΡΓΑΔΟΣ Ι. ΔΗΜΗΤΡΙΟΣ**
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

....

Καστοριά Μάρτιος – 2023

Copyright © 2023 – ΤΣΟΛΑΚΗ ΣΤΑΜΑΤΙΑ
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.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Ευχαριστώ τον επιβλέποντα καθηγητή μου, κύριο Δημήτριο Ι. Βέργαδο, για την πολύτιμη συνεισφορά του στην εκπόνηση της παρούσας εργασίας.

Θα ήθελα επίσης να εκφράσω την ευγνωμοσύνη μου στον αδελφό μου, τον «θεματοφύλακα» και στήριγμα της ζωής μου.

Περίληψη

Η διπλωματική εργασία που παρουσιάζεται εδώ έχει για θέμα της τον σχεδιασμό και την ανάπτυξη εφαρμογής μέσω του App Inventor. Οι έξυπνες συσκευές κινητής τηλεφωνίας και οι ταμπλέτες καλύπτουν μεγάλο εύρος των αναγκών των ανθρώπων σήμερα. Το γεγονός αυτό έχει αυξήσει τον βαθμό εξοικείωσης των μαθητών με τις συσκευές αυτές. Υπάρχει μεγάλη ποικιλία εφαρμογών που ο σύγχρονος άνθρωπος μπορεί να εγκαταστήσει στις συσκευές του. Εκμεταλλευόμενος το γεγονός αυτό ο εκπαιδευτικός μπορεί να το αξιοποιήσει προκειμένου να αποκτήσει νέα εργαλεία για την βελτίωση του μαθήματος που προσφέρει. Στην εργασία αυτή γίνεται μία ιστορική αναδρομή του App Inventor, παρουσιάζεται η σύγχρονη μορφή της, η υποστήριξή της μέσω του AI Companion. Αναλύονται πτυχές του στην εκπαίδευση και τέλος μία εφαρμογή αξιολόγησης αναπτύσσεται στην παρούσα εργασία. Ένας σκοπός της εργασίας είναι να δείξει την απλότητα της χρήσης της εφαρμογής App Inventor.

Λέξεις Κλειδιά: App Inventor, Designer, Blocks, AI Companion

Abstract

The thesis presented here is about designing and developing through App Inventor. Smart mobile devices and tablets cover a wide range of people's needs today. This fact has increased students' familiarity with these devices. There is a wide variety of applications that the modern people can install on his devices. By taking advantage of this fact, teachers can use it to acquire new tools to improve the course he offers. In this paper, a historical review of App Inventor is given, its modern form is presented, its support through the AI Companion. Aspects of it in education are analyzed and finally an evaluation application developed in this thesis. One purpose of the paper is to demonstrate the simplicity of using App Inventor.

Key Words:App Inventor, Designer, Blocks, AI Companion

Περιεχόμενα

Εισαγωγή 6

1. Εφαρμογή App Inventor 7

1.1 Ιστορικό σημείωμα 7
1.2 Επισκόπηση MIT App Inventor 8
1.3 Γρήγορη επανάληψη και σχεδίαση με χρήση του AI Companion 13

2. Υποστήριξη του App Inventor 15

2.1 Στόχοι σχεδίασης MIT App Inventor 15
2.2 Αφαίρεση στοιχείων για συμπεριφορά πλατφόρμας 15
2.3 Μπλοκ ως Λογική 16
2.4 Μπλοκ ανωτάτου επιπέδου 17
2.5 WYSIWYG What You See Is What You Get 17
2.6 Δημιουργία εξαρτήματος χρόνου σχεδίασης 18
2.7 Φυσική αρίθμηση 18

3. MIT App Inventor στην εκπαίδευση 19

3.1 Μαζικά ανοιχτά διαδικτυακά μαθήματα 19
3.2 Πρόγραμμα MIT Expert Trainers in Educational Mobile Computing 20
3.3 Επεκτάσεις 21
3.4 Ερευνητικά έργα 21
3.5 Ενδυνάμωση μέσω Προγραμματισμού 21

4. Ανάπτυξη εφαρμογής 23

4.1 Αρχική Σελίδα 23
4.2 Σελίδες Ερωτήσεων 25
4.3 Σελίδα αποτελεσμάτων 43

Συμπεράσματα 45

5. Βιβλιογραφία 48

Παράρτημα Κώδικα 49

Λίστα Σχημάτων

Εικόνα 1 Δικτυακός τόπος App Inventor 8
Εικόνα 2 Σύνδεση μέσω Google 9
Εικόνα 3 Designer 10
Εικόνα 4 Blocks 11
Εικόνα 5 Μενού συνδέσεων 12
Εικόνα 6 Έναρξη του emulator 12
Εικόνα 7 AI Companion 12
Εικόνα 8 Emulator 13
Εικόνα 10 AI Companion στο Play Store 14
Εικόνα 11 Designer αρχική 24
Εικόνα 12 Blocks αρχικής οθόνης 25
Εικόνα 13 Designer Εύκολο 1 26
Εικόνα 14 Blocks εύκολο 1 27
Εικόνα 15 Designer Εύκολο 2 29
Εικόνα 16 Blocks εύκολο 2 30
Εικόνα 17 Designer Εύκολο 3 31
Εικόνα 18 Blocks Εύκολο 3 32
Εικόνα 19 Designer Εύκολο 4 33
Εικόνα 20 Blocks εύκολο 4 34
Εικόνα 21 Designer Δύσκολο 1 35
Εικόνα 22 Blocks δύσκολο 1 36
Εικόνα 23 Designer Δύσκολο 2 37
Εικόνα 24 Blocks δύσκολο 2 38
Εικόνα 25 Designer Δύσκολο 3 39
Εικόνα 26 Blocks δύσκολο 3 40
Εικόνα 27 Designer Δύσκολο 4 41
Εικόνα 28 Blocks δύσκολο 4 42
Εικόνα 29 Designer αποτέλεσμα 43
Εικόνα 30 Blocks αποτέλεσμα 44

Εισαγωγή

Το App Inventor είναι ένα εκπαιδευτικό εργαλείο εκμάθησης που παρέχεται από το MIT και επιτρέπει στους χρήστες να δημιουργούν εφαρμογές Android χωρίς καμία γνώση προγραμματισμού. Καθώς το App Inventor κερδίζει δημοτικότητα μεταξύ των εκπαιδευτικών και των μαθητών σε όλο τον κόσμο, γίνεται όλο και πιο σημαντικό να διασφαλίζεται ότι το εργαλείο προσφέρει στους χρήστες του το εύρος και το βάθος της λειτουργικότητας και της ανάπτυξης των εφαρμογών που επιθυμούν. Εν αναμονή του διευρυμένου ρόλου και της επιρροής του App Inventor στα εκπαιδευτικά ιδρύματα παγκοσμίως (γυμνάσια και λύκεια, πρωτίστως), αυτή η διατριβή εστιάζει στην ηλικιακή ομάδα των μαθητών γυμνασίου και λυκείου και σε ένα από τα θέματα που τους ενδιαφέρει περισσότερο, την αξιολόγηση τους. Ο στόχος αυτής της διατριβής είναι να προσδιορίσει τις υπάρχουσες δυνατότητες και περιορισμούς του App Inventor σε σχέση με την ανάπτυξη εφαρμογής και να εφαρμόσει ιδέες (παιδαγωγικής και τεχνολογικής φύσης) που θα βελτιώσουν την ποικιλομορφία, την πολυπλοκότητα, την αισθητική ελκυστικότητα και την απόδοση των εφαρμογών που μπορούν να κατασκευαστούν χρησιμοποιώντας το App Inventor. Είναι γενικά παραδεκτό ότι εάν οι δυνατότητες ανάπτυξης εφαρμογών του App Inventor μπορούν να αυξηθούν, το ποσοστό υιοθέτησης του εργαλείου και η δημοτικότητά του μεταξύ των μαθητών σχολείων θα επηρεαστεί πολύ θετικά. Σε αυτή τη διατριβή περιγράφονται προσωπικές εμπειρίες διδασκαλίας της ανάπτυξης εφαρμογής App Inventor καθώς και τους περιορισμούς στη μεθοδολογία διδασκαλίας και στο σύνολο χαρακτηριστικών του App Inventor που εντόπισε μέσω αυτής της εμπειρίας. Οι κύριες συνεισφορές της εργασίας αυτής είναι η ανάπτυξη μιας εφαρμογής εξέτασης και τα πολύτιμα συμπεράσματα που αποκομίζονται κατά την ανάπτυξή της, καθώς δοκιμάζει σε ποιο βαθμό το δημιουργικό και οι νέες τροποποιήσεις του App Inventor διευκολύνουν την ανάπτυξη εφαρμογών αυτού του τύπου.

1. Εφαρμογή App Inventor

Σε αυτό το κεφάλαιο, περιγράφονται οι στόχοι του MIT App Inventor και πώς επηρέασαν τον σχεδιασμό και την ανάπτυξή μας, από την έναρξη του προγράμματος στην Google το 2008, τη μετάβαση στο MIT, μέχρι σήμερα. Συζητάμε την παιδαγωγική αξία του MIT App Inventor και τη χρήση του ως εργαλείου για τη διδασκαλία και την ενθάρρυνση ανθρώπων όλων των ηλικιών να σκέφτονται και να ενεργούν υπολογιστικά.. Ολοκληρώνουμε συζητώντας τους περιορισμούς και τα οφέλη εργαλείων όπως το App Inventor και προτείνοντας νέες κατευθύνσεις για έρευνα.

1.1 Ιστορικό σημείωμα

Η εφαρμογή έγινε διαθέσιμη, μέσω αιτήματος, στις 12 Ιουλίου 2010 και κυκλοφόρησε δημόσια στις 15 Δεκεμβρίου 2010. Η ομάδα του App Inventor είχε επικεφαλής τον Hal Abelson και τον Mark Friedman. Το δεύτερο εξάμηνο του 2011, η Google κυκλοφόρησε τον πηγαίο κώδικα, τερμάτισε τον διακομιστή της και παρείχε χρηματοδότηση για τη δημιουργία του Κέντρου Εκμάθησης για Κινητά του MIT, με επικεφαλής τον δημιουργό του App Inventor Hal Abelson και τους καθηγητές του MIT Eric Klopfer και Mitchel Resnick. Η έκδοση του MIT κυκλοφόρησε τον Μάρτιο του 2012.

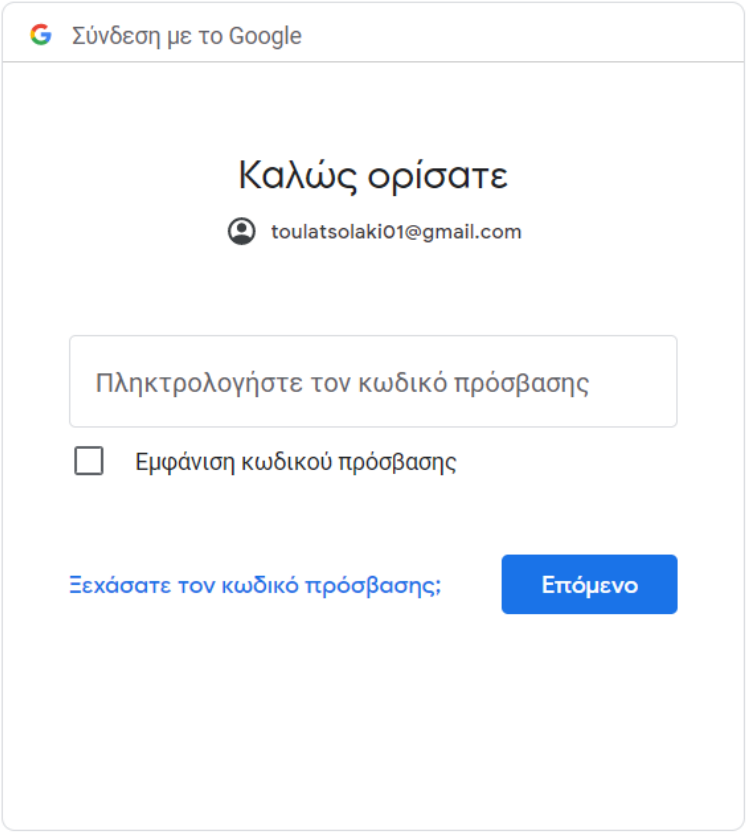
Στις 6 Δεκεμβρίου 2013 το MIT κυκλοφόρησε το App Inventor 2, μετονομάζοντας την αρχική έκδοση σε "App Inventor Classic" Οι κύριες διαφορές είναι: Το πρόγραμμα επεξεργασίας μπλοκ στην αρχική έκδοση εκτελούσε μια ξεχωριστή διαδικασία Java, χρησιμοποιώντας τη βιβλιοθήκη Java Open Blocks για τη δημιουργία γλωσσών προγραμματισμού οπτικών μπλοκ και προγραμματισμού. Το Open Blocks διανέμεται από το Scheller Teacher Education Program (STEP) του MIT και προέρχεται από την έρευνα μεταπτυχιακής διατριβής από τον Ricarose Roque. Ο καθηγητής Eric Klopfer και ο Daniel Wendel του προγράμματος Scheller υποστήριξαν τη διανομή των Open Blocks με άδεια MIT.

Ο οπτικός προγραμματισμός Open Blocks συνδέεται στενά με το StarLogo TNG, ένα έργο του STEP, και το Scratch, ένα έργο της Ομάδας Lifelong Kindergarten του MIT Media Lab με επικεφαλής τον Mitchel Resnick. Το App Inventor 2 αντικατέστησε το Open Blocks με το Blockly, ένα πρόγραμμα επεξεργασίας μπλοκ που εκτελείται σε ένα πρόγραμμα περιήγησης ιστού. Η εφαρμογή MIT AI2 Companion επιτρέπει τον εντοπισμό σφαλμάτων σε πραγματικό χρόνο σε συνδεδεμένες συσκευές μέσω Wi-Fi ή Universal Serial Bus (USB). Επιπλέον, ο χρήστης μπορεί να χρησιμοποιήσει έναν εξομοιωτή "σε υπολογιστή" διαθέσιμο για Windows, MacOS και Linux.

1.2 Επισκόπηση MIT App Inventor

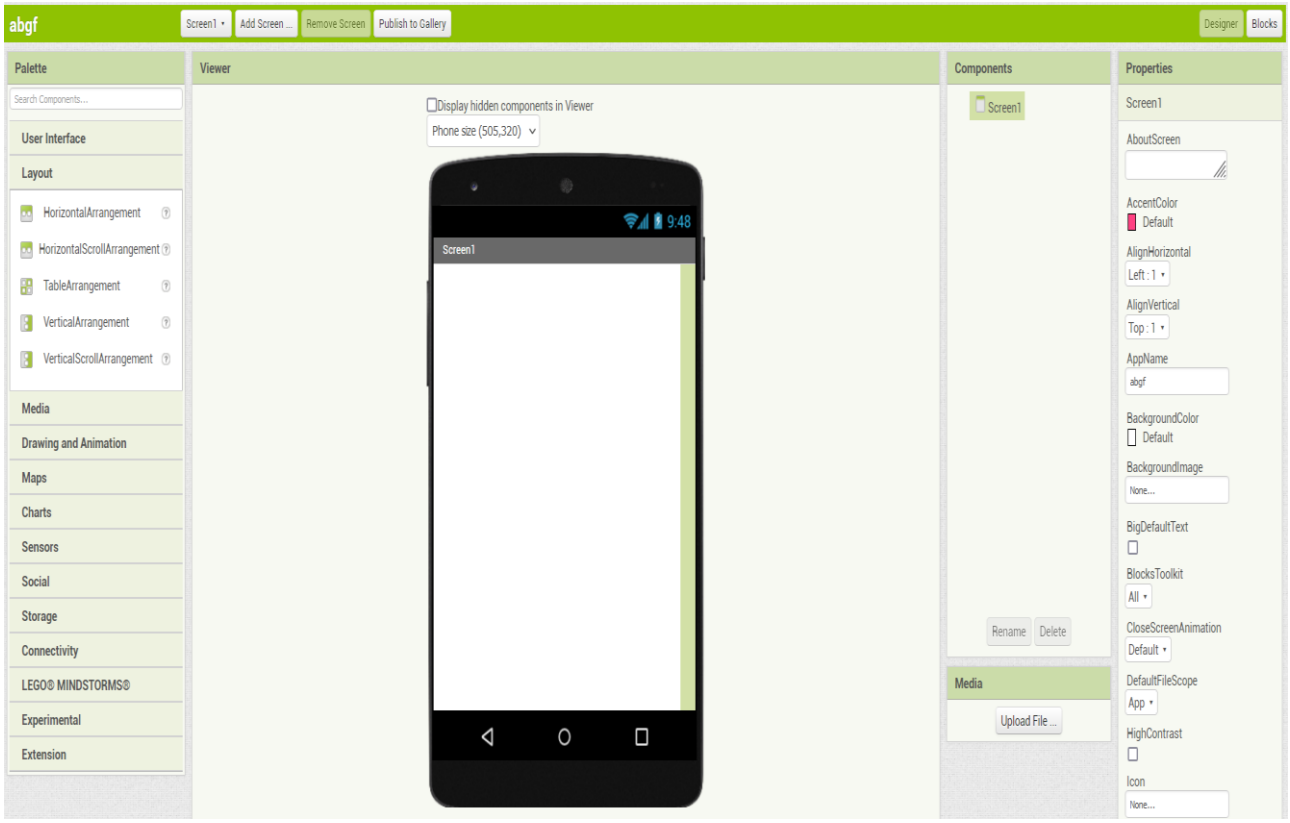
Η δημιουργία εφαρμογών μπορεί να ξεκινήσει με ένα απλό κλικ στο Create Apps! (βλ. Εικ. 1) μετά ο χρήστης πρέπει να κάνει σύνδεση με ένα λογαριασμό της google, αν διαθέτει αλλιώς θα πρέπει να δημιουργήσει κάποιον αν θέλει να συνεχίσει. Με την επιτυχή σύνδεση γίνεται ανακατεύθυνση στην κεντρική σελίδα του App Invertor.

Εικόνα 1 Δικτυακός τόπος App Inventor



Εικόνα 2 Σύνδεση μέσω Google

Η διεπαφή χρήστη του MIT App Inventor περιλαμβάνει δύο κύριους συντάκτες: αυτόν της σχεδίασης - Designer και αυτόν των μπλοκ - Blocks. Σκεπτόμενοι τα κουμπιά Designer και Blocks σαν καρτέλες - τα χρησιμοποιούμε για να μετακινούμαστε μεταξύ των δύο περιοχών. Ο συντάκτης σχεδίασης – Designer (βλ. Εικ. 3), είναι μια διεπαφή μεταφοράς και απόθεσης για τη διάταξη των στοιχείων της διεπαφής χρήστη (UI) της εφαρμογής. Έτσι ο χρήστης μπορεί να αποθέτει τα στοιχεία που του είναι διαθέσιμα από λίστες στα αριστερά της οθόνης του σχεδιάζοντας απ' ευθείας την εφαρμογή του. Υπάρχουν δομικά στοιχεία, στοιχεία διαμόρφωσης επιπέδων,

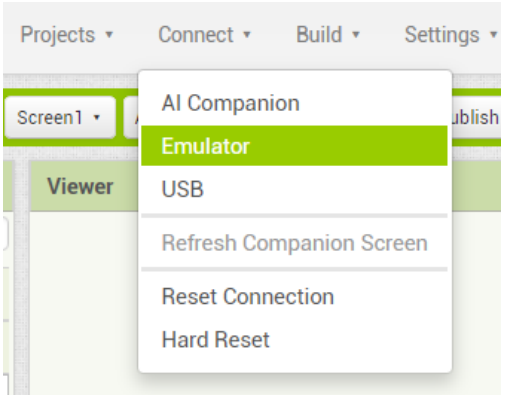


Εικόνα 3 Designer

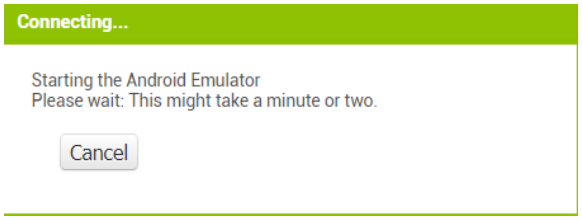
Ο επεξεργαστής μπλοκ - Blocks (βλ. Εικ. 4) είναι ένα περιβάλλον στο οποίο οι δημιουργοί εφαρμογών μπορούν να σχεδιάσουν οπτικά τη λογική των εφαρμογών τους χρησιμοποιώντας χρωματικά κωδικοποιημένα μπλοκ που κουμπώνουν μεταξύ τους σαν κομμάτια παζλ για να περιγράψουν το πρόγραμμα. Το Blocks Editor είναι εκείνο το μέρος όπου προγραμματίζουμε τη συμπεριφορά της εφαρμογής μας. Υπάρχουν πολλά ενσωματωμένα μπλοκ που χειρίζονται καταστάσεις όπως μαθηματικά, λογική και κείμενο. Τα μπλοκ ταιριάζουν με καθένα από τα στοιχεία διαφορετικών τμημάτων στην εφαρμογή μας. Για να χρησιμοποιήσουμε τα μπλοκ για ένα συγκεκριμένο στοιχείο, πρώτα πρέπει να προσθέσουμε αυτό το στοιχείο στην εφαρμογή μας μέσω του Designer. Χρησιμοποιούν δε μια πολύ χρήσιμη λειτουργία αντιγραφής στο πρόχειρο. Ολόκληρες ενότητες μπλοκ μπορούν να τοποθετηθούν στο εικονίδιο του σακιδίου και να χρησιμοποιηθούν ως πρόχειρο στην ίδια ή και σε άλλες εφαρμογές του χρήστη.

Εικόνα 4 Blocks

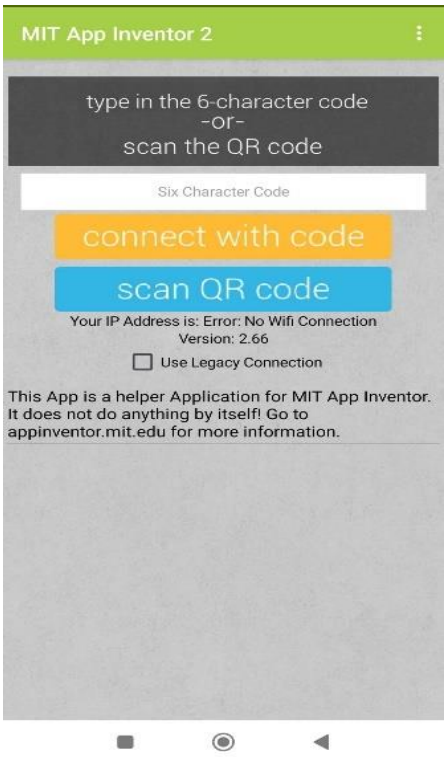
Για να βοηθήσει στην ανάπτυξη και τη δοκιμή, το App Inventor παρέχει μια εφαρμογή για κινητά που ονομάζεται App Inventor Companion ή απλώς AI Companion (βλ. Εικ. 7) την οποία οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να δοκιμάσουν και να προσαρμόσουν τη συμπεριφορά των εφαρμογών τους σε πραγματικό χρόνο. Με αυτόν τον τρόπο, ο καθένας μπορεί να δημιουργήσει γρήγορα μια εφαρμογή για κινητά και να αρχίσει αμέσως να δοκιμάζει και να επαναλαμβάνει. Ενναλακτική επιλογή αποτελεί ο emulator στερείται όμως ταχύτητα στην σύνδεση, θα πρέπει να έχει εγκατασταθεί η εφαρμογή στον υπολογιστή και παίρνει αρκετό χρόνο να φορτωθεί (βλ. Εικ. 6) και αμεσότητα σε σχέση με το AI Companion. Από την άλλη δεν απαιτεί τη χρήση κινητού που είναι περιοριστική αφού κάθε χρήστης θα χρειαζόταν πλειάδα συσκευών για να δει ότι όλα λειτουργούν κανονικά. Επίσης υπάρχει και δυνατότητα ενσύρματης σύνδεσης μέσω USB.



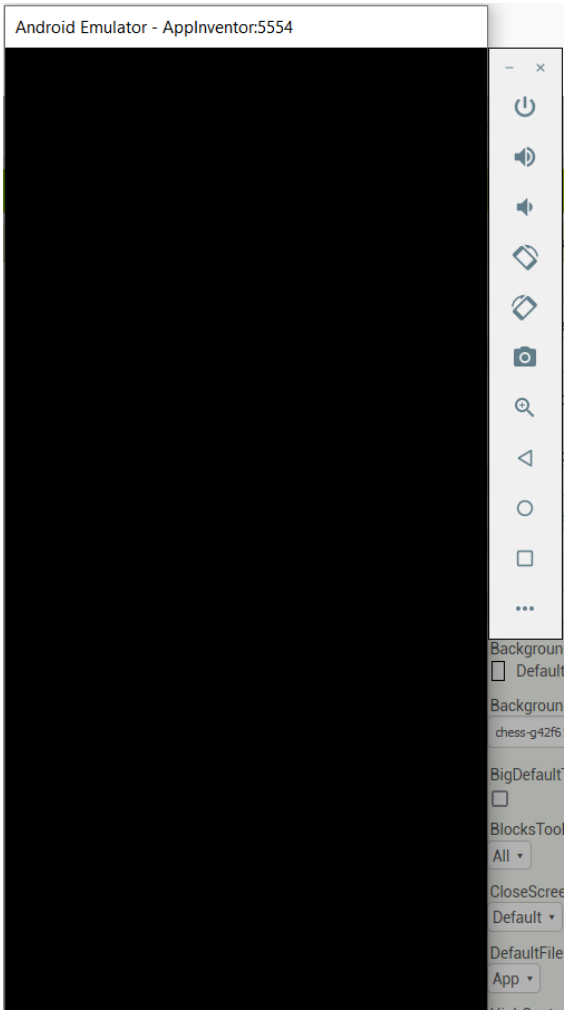
Εικόνα 5 Μενού συνδέσεων



Εικόνα 6 Έναρξη του emulator



Εικόνα 7 AI Companion

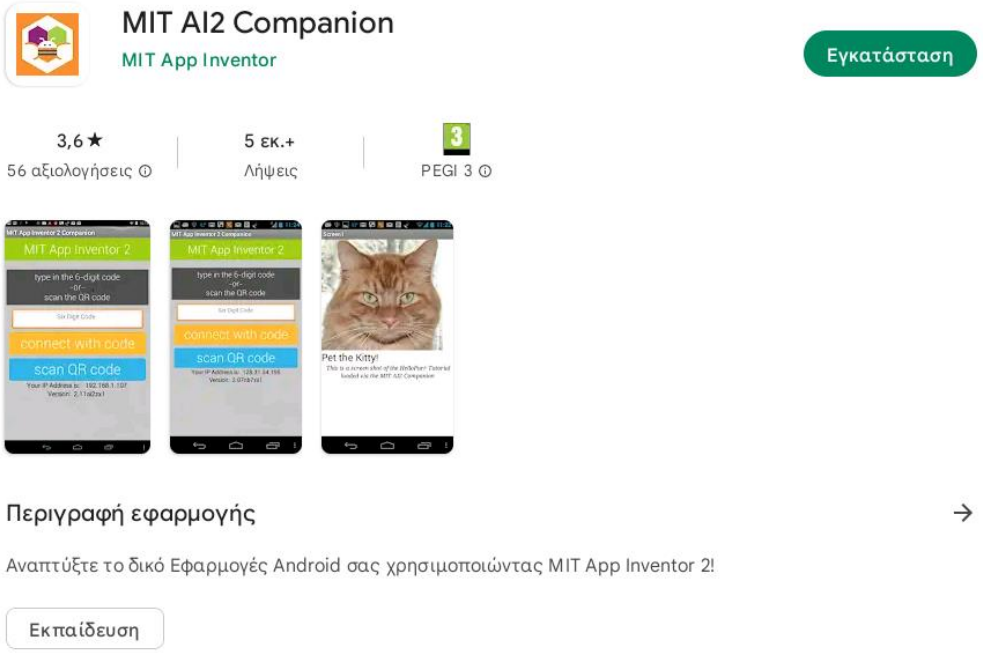


Εικόνα 8 Emulator

1.3 Γρήγορη επανάληψη και σχεδίαση με χρήση του AI Companion

Ένα βασικό χαρακτηριστικό του MIT App Inventor είναι το ζωντανό περιβάλλον ανάπτυξης για εφαρμογές για κινητές συσκευές. Το App Inventor το παρέχει αυτό μέσω μιας συνοδευτικής εφαρμογής εγκατεστημένης στην κινητή συσκευή του χρήστη μέσω του Play Store. Η διεπαφή ιστού App Inventor στέλνει κώδικα στη συνοδευτική

εφαρμογή, η οποία ερμηνεύει τον κώδικα και εμφανίζει την εφαρμογή σε πραγματικό χρόνο στον προγραμματιστή (βλ. Εικ. 6).



Εικόνα 9 AI Companion στο Play Store

Με αυτόν τον τρόπο, ο χρήστης μπορεί να αλλάξει τη διεπαφή και τη συμπεριφορά της εφαρμογής σε πραγματικό χρόνο. Ο παραδοσιακός κύκλος κατασκευής για μια εφαρμογή Android περιλαμβάνει τη σύνταξη κώδικα σε ένα πρόγραμμα επεξεργασίας κειμένου ή ένα ενσωματωμένο περιβάλλον ανάπτυξης και η αναδόμηση της εφαρμογής για δοκιμή μπορεί συχνά να διαρκέσει λίγα λεπτά, ενώ η αλλαγή στο περιβάλλον ζωντανής ανάπτυξης του AI Companion συνήθως ισχύει σε 1 με 2 δευτερόλεπτα. Η γρήγορη εμφάνιση αλλαγών που αντικατοπτρίζονται στην εφαρμογή σημαίνει ότι οι χρήστες μπορούν να σχεδιάσουν, να κατασκευάσουν, να εξερευνήσουν, ακόμη και να κάνουν λάθη κατά την εξερεύνηση, επειδή το κόστος χρόνου αυτών των λαθών είναι σχετικά μικρό.

2. Υποστήριξη του App Inventor

2.1 Στόχοι σχεδίασης MIT App Inventor

Στο σχεδιασμό του MIT App Inventor, η εισαγωγή της ανάπτυξης εφαρμογών για κινητά σε εκπαιδευτικά πλαίσια ήταν κεντρικός στόχος. Πριν από την κυκλοφορία του, τα περισσότερα περιβάλλοντα ανάπτυξης εφαρμογών για κινητές συσκευές ήταν αδέξια, προσβάσιμα μόνο με εξειδίκευση σε επίπεδο συστημάτων ή ενσωματωμένο προγραμματισμό ή και στα δύο. Ακόμη και με το λειτουργικό σύστημα Android της Google και τη γλώσσα προγραμματισμού Java, ο σχεδιασμός της διεπαφής χρήστη ήταν μια πολύπλοκη εργασία. Επιπλέον, η χρήση της πλατφόρμας απαιτούσε εξοικείωση με τη σύνταξη και τη σημασιολογία της Java και τη δυνατότητα εντοπισμού σφαλμάτων μεταγλώττισης Java (π.χ. μεταβλητές με ορθογραφικά λάθη ή λανθασμένα ερωτηματικά) για επιτυχία. Αυτές οι προκλήσεις παρουσίασαν εμπόδια στην είσοδο για άτομα που δεν γνωρίζουν την επιστήμη των υπολογιστών, το δημογραφικό στόχο του App Inventor. Συζητάμε τους στόχους σχεδιασμού για το έργο App Inventor, συγκεκριμένα, τη χρήση στοιχείων για την αφαίρεση μέρους της πολυπλοκότητας της συμπεριφοράς της πλατφόρμας και τη χρήση μπλοκ για την εξάλειψη της πολυπλοκότητας της υποκείμενης γλώσσας προγραμματισμού. Αυτοί οι στόχοι μπορούν να εξηγηθούν περαιτέρω ως ευθυγράμμιση της οπτικής γλώσσας με τα νοητικά μοντέλα νέων προγραμματιστών και επιτρέποντας την εξερεύνηση μέσω γρήγορου, επαναληπτικού σχεδιασμού.

2.2 Αφαίρεση στοιχείων για συμπεριφορά πλατφόρμας

Τα στοιχεία είναι βασικές αφαιρέσεις στο MIT App Inventor. Μειώνουν την πολυπλοκότητα της διαχείρισης των αλληλεπιδράσεων με διεπαφές προγραμματισμού εφαρμογών για συγκεκριμένες πλατφόρμες και λεπτομέρειες σχετικά με τη διαχείριση κατάστασης των συσκευών. Αυτό επιτρέπει στο χρήστη να σκεφτεί το πρόβλημα που αντιμετωπίζει και όχι τις μικρολεπτομέρειες που απαιτούνται συνήθως από τους

προγραμματιστές εφαρμογών. Για παράδειγμα, κάποιος που σχεδιάζει να χρησιμοποιήσει το MIT App Inventor για να δημιουργήσει μια εφαρμογή που θα χρησιμοποιεί το παγκόσμιο σύστημα εντοπισμού θέσης (GPS) για την παρακολούθηση της κίνησης, δεν χρειάζεται να ασχολείται με τη διαχείριση του κύκλου ζωής της εφαρμογής, τις κλειδαριές λογισμικού και υλικού GPS ή τη συνδεσιμότητα δικτύου (σε περίπτωση που πέσει η ανίχνευση τοποθεσίας επιστροφή στη θέση που βασίζεται στο δίκτυο). Αντίθετα, ο προγραμματιστής της εφαρμογής προσθέτει ένα στοιχείο αισθητήρα τοποθεσίας που αφαιρεί αυτή την πολυπλοκότητα και παρέχει ένα API για την ενεργοποίηση και την επεξεργασία ενημερώσεων τοποθεσίας. Αυτή η μείωση της πολυπλοκότητας επιτρέπει στους εφευρέτες εφαρμογών να επικεντρωθούν στο πρόβλημα που αντιμετωπίζουν και να επιτύχουν γρήγορα έναν στόχο. Τα στοιχεία αποτελούνται από τρία κύρια στοιχεία: ιδιότητες, μεθόδους και συμβάντα.

Οι ιδιότητες ελέγχουν την κατάσταση του στοιχείου και είναι αναγνώσιμες ή/και εγγράψιμες από τον προγραμματιστή της εφαρμογής. Για παράδειγμα, η ενεργοποιημένη ιδιότητα του αισθητήρα τοποθεσίας περιλαμβάνει τη λειτουργικότητα που απαιτείται για τη διαμόρφωση του δέκτη GPS και τη διαχείριση της κατάστασής του ενώ η εφαρμογή χρησιμοποιείται. Οι μέθοδοι λειτουργούν σε πολλαπλές εισόδους και πιθανώς επιστρέφουν ένα αποτέλεσμα. Τα συμβάντα ανταποκρίνονται σε αλλαγές στην κατάσταση της συσκευής ή της εφαρμογής με βάση εξωτερικούς παράγοντες. Για παράδειγμα, όταν ο χρήστης της εφαρμογής αλλάζει την τοποθεσία του, το γεγονός που άλλαξε τοποθεσία επιτρέπει στη λογική της εφαρμογής να ανταποκριθεί στην αλλαγή.

2.3 Μπλοκ ως Λογική

Στο MIT App Inventor, οι χρήστες κωδικοποιούν τη συμπεριφορά της εφαρμογής χρησιμοποιώντας μια γλώσσα προγραμματισμού που βασίζεται σε μπλοκ. Υπάρχουν δύο τύποι μπλοκ στο App Inventor: ενσωματωμένα μπλοκ και μπλοκ στοιχείων. Η ενσωματωμένη βιβλιοθήκη μπλοκ παρέχει τα βασικά στοιχεία και τις λειτουργίες που είναι γενικά διαθέσιμες σε άλλες γλώσσες προγραμματισμού, όπως Booleans, συμβολοσειρές, αριθμούς, λίστες, μαθηματικούς τελεστές, τελεστές σύγκρισης και τελεστές ροής ελέγχου. Οι προγραμματιστές χρησιμοποιούν μπλοκ στοιχείων

(ιδιότητες, μεθόδους και συμβάντα) για να ανταποκρίνονται σε συμβάντα συστήματος και χρήστη, να αλληλοεπιδρούν με το υλικό της συσκευής και να προσαρμόζουν τις οπτικές και συμπεριφορικές πτυχές των στοιχείων.

2.4 Μπλοκ ανωτάτου επιπέδου

Όλη η λογική του προγράμματος βασίζεται σε τρεις τύπους μπλοκ ανώτατου επιπέδου: καθολικούς ορισμούς μεταβλητών, ορισμούς διαδικασίας και χειριστές συμβάντων στοιχείων. Οι καθολικές μεταβλητές παρέχουν ονομασμένες υποδοχές για την αποθήκευση των καταστάσεων του προγράμματος. Οι διαδικασίες ορίζουν κοινές συμπεριφορές που μπορούν να κληθούν από πολλές θέσεις στον κώδικα. Όταν συμβαίνει ένα συμβάν στη συσκευή, ενεργοποιεί την αντίστοιχη συμπεριφορά εφαρμογής που προδιαγράφεται στο μπλοκ συμβάντων. Το μπλοκ διαχείρισης συμβάντων μπορεί να αναφέρεται σε καθολικές μεταβλητές ή διαδικασίες. Με τον περιορισμό των τύπων μπλοκ ανώτατου επιπέδου, υπάρχουν λιγότερες οντότητες για αιτιολόγηση.

2.5 WYSIWYG What You See Is What You Get

Το πρόγραμμα επεξεργασίας σχεδίασης για το App Inventor επιτρέπει στους προγραμματιστές να δουν πώς θα εμφανίζεται η εφαρμογή στην οθόνη της συσκευής και να προσαρμόσουν τον παράγοντα μορφής της οπτικοποιημένης συσκευής (π.χ. τηλέφωνο ή tablet). Οι προσαρμογές στις ιδιότητες των οπτικών στοιχείων, για παράδειγμα, χρώμα φόντου και μέγεθος, αντικατοπτρίζονται σε πραγματικό χρόνο. Οι εφαρμογές μπορούν επίσης να εκτελεστούν σε λειτουργία ζωντανής ανάπτυξης χρησιμοποιώντας το AI Companion.

Η λειτουργικότητα επιτρέπει στους εφευρέτες εφαρμογών να σύρουν, να αποθέσουν και να επεξεργαστούν στοιχεία καθώς και να ενσωματώσουν δεδομένα που βασίζονται στον ιστό, όπως συστήματα γεωγραφικών πληροφοριών κ.α., για τη δημιουργία εφαρμογών με πλούσιο περιεχόμενο. Ο χρήστης μπορεί να μετακινήσει το

περιεχόμενο εύκολα για να πετύχει εξαιρετικά αποτελέσματα χωρίς να χρειάζεται να παρέχει το μεγαλύτερο μέρος της λογικής για αυτό σε κώδικα.

2.6 Δημιουργία εξαρτήματος χρόνου σχεδίασης

Σε αντίθεση με πολλές γλώσσες προγραμματισμού, το App Inventor περιορίζει τη δημιουργία νέων οντοτήτων σε χρόνο εκτέλεσης. Αυτό παρέχει πολλαπλά οφέλη. Πρώτον, τοποθετώντας ρητά όλα τα στοιχεία στην εφαρμογή, ο χρήστης μπορεί να την οπτικοποιήσει ξεκάθαρα αντί να χρειάζεται να συλλογιστεί πράγματα που δεν θα υπάρξουν στο μέλλον. Δεύτερον, μειώνει τις πιθανότητες οι χρήστες να εισάγουν εξαρτήσεις κυκλικής μνήμης στη διεπαφή χρήστη που θα προκαλούσαν τελικά εξάντληση της μνήμης της εφαρμογής. Αυτό ενθαρρύνει τους εφευρέτες εφαρμογών να σκεφτούν πώς να δομήσουν κατάλληλα τις εφαρμογές τους και να επαναχρησιμοποιήσουν στοιχεία για να αποφύγουν την υπερφόρτωση του συστήματος ή των τελικών χρηστών τους.

2.7 Φυσική αρίθμηση

Το σύστημα αρίθμησης στο App Inventor λαμβάνει μια αρχική τιμή 1, σύμφωνα με τις δεξιότητες μέτρησης που αναπτύσσει ο άνθρωπος. Αυτό γίνεται σε αντίθεση με τις περισσότερες γλώσσες προγραμματισμού, οι οποίες είναι πιο ευθυγραμμισμένες με την αρχιτεκτονική μηχανής και επομένως ξεκινούν από το 0.

3. MIT App Inventor στην εκπαίδευση

Ο πρωταρχικός στόχος του MIT App Inventor είναι να παρέχει σε οποιονδήποτε ενδιαφέρεται να δημιουργήσει εφαρμογές για την επίλυση προβλημάτων με τα απαραίτητα εργαλεία για να το κάνει. Το εκπαιδευτικό υλικό που αναπτύχθηκε από την ομάδα προσανατολίζεται κυρίως σε δασκάλους και μαθητές στο γυμνάσιο και στο λύκειο, αλλά οι εφευρέτες εφαρμογών προέρχονται από όλες τις ηλικίες από όλο τον κόσμο. Μερικά από τα βασικά στοιχεία της εκπαιδευτικής στρατηγικής του MIT App Inventor, συμπεριλαμβάνουν τα μαζικά διαδικτυακά ανοικτά μαθήματα (MOOCs) που επικεντρώνονται στο MIT App Inventor, το πρόγραμμα Master Trainer (MT), τη λειτουργικότητα επεκτάσεων του App Inventor που επιτρέπει ενσωμάτωση νέου υλικού για εκπαίδευση και ερευνητικά έργα που έχουν αξιοποιήσει το App Inventor ως πλατφόρμα για την ενεργοποίηση υπολογιστών για συγκεκριμένο τομέα.

3.1 Μαζικά ανοιχτά διαδικτυακά μαθήματα

Η επιθυμία για μάθηση υπολογιστικής σκέψης έχει οδηγήσει στον πολλαπλασιασμό του διαδικτυακού εκπαιδευτικού υλικού στο οποίο ο καθένας μπορεί να έχει πρόσβαση για να αυξήσει τις γνώσεις και την κατανόησή του. Καθώς συνεχίζουμε να ενσωματώνουμε την τεχνολογία της πληροφορίας στην καθημερινή μας ζωή, τις κινητές συσκευές και άλλες νέες τεχνολογίες, μπορούμε να παρατηρήσουμε ότι η βαθύτερη κατανόηση των υπολογιστών είναι απαραίτητη για να είσαι αποτελεσματικό μέλος της κοινωνίας και όσοι μαθαίνουν υπολογιστική σκέψη θα έχουν πλεονέκτημα οικονομία μας που βασίζεται στη γνώση.

Πολλά τεράστια ανοιχτά διαδικτυακά μαθήματα έχουν αναπτυχθεί εξ ολοκλήρου ή εν μέρει χρησιμοποιώντας το App Inventor. Για παράδειγμα, ένα μάθημα App Inventor EdX ενσωματώνεται στενά με το μάθημα AP CS Principles και ενσωματώνει πολλά στοιχεία υπολογιστικής σκέψης. Επομένως, οι μαθητές μπορούν να δημιουργήσουν τις δικές τους εφαρμογές για κινητά και να μάθουν βασικές ικανότητες που σχετίζονται με τον υπολογισμό.

3.2 Πρόγραμμα MIT Expert Trainers in Educational Mobile Computing

Το MIT παρέχει ειδικές οδηγίες στους εκπαιδευτικούς μέσω του προγράμματος MIT Expert Trainers in Educational Mobile Computing. Ένα πρωτότυπο του προγράμματος Master Trainers ξεκίνησε κατά τη διάρκεια μιας συνεργασίας με το Verizon App Challenge το 2012. Προσλήφθηκαν ειδικευμένοι εκπαιδευτές App Inventor και έλαβαν μια μικρή ειδική εκπαίδευση για να βοηθήσουν μέντορες και να εκπαιδεύσουν ομάδες που στη συνέχεια κέρδισαν το App Challenge. Το τρέχον πρόγραμμα Master Trainers σχεδιάστηκε το 2015, από το MIT για να « προσφέρει ένα Πρόγραμμα Ειδικών Εκπαιδευτών για την πιστοποίηση εκπαιδευτών K-12 και άλλων επαγγελματιών στην εκπαίδευση φορητών υπολογιστών χρησιμοποιώντας την πλατφόρμα MIT App Inventor. Αυτό το πρόγραμμα επαγγελματικής ανάπτυξης θα συνεχίσει να αναπτύσσει μια παγκόσμια κοινότητα ειδικών στην ανάπτυξη εφαρμογών για κινητά που είναι διαθέσιμοι για να καθοδηγήσουν άλλους στην εξερεύνηση της δημιουργίας εφαρμογών που έχουν σημασία για κινητά. Αυτοί οι ειδικοί θα καθοδηγήσουν εργαστήρια σε τάξεις, κοινότητες και προγράμματα μετά το σχολείο για να ενδυναμώσουν τους δασκάλους και τους μαθητές καθώς γίνονται σχεδιαστές εφαρμογών για κινητά, παρέχοντας έτσι μια διαδρομή στην επιστήμη των υπολογιστών, την ανάπτυξη λογισμικού και άλλους κλάδους σχετικούς με τον σημερινό ψηφιακό κόσμο.» [1].

Για να γίνει κάποιος Master Trainer, πρέπει να επιδείξει επάρκεια στο App Inventor, για παράδειγμα, μέσω του App Inventor EdX MOOC. Το MOOC είναι εξαιρετικά ενσωματωμένο με έννοιες υπολογιστικής σκέψης, δίνοντας στους υποψηφίους ένα ισχυρό θεμέλιο στις έννοιες και τις πρακτικές που σχετίζονται με την υπολογιστική σκέψη. Στη συνέχεια, οι επίδοξοι Master Trainers ολοκληρώνουν ένα διαδικτυακό μάθημα ανάγνωσης διάρκειας 10 εβδομάδων που καλύπτει θέματα όπως η αποστολή και η φιλοσοφία του App Inventor, η παιδαγωγική διδασκαλίας σε παιδιά και ενήλικες, ο δομισμός και η σχεδιαστική σκέψη. Τέλος, υπάρχει ένα επιτόπιο 3ήμερο εργαστήριο στο MIT, όπου οι συμμετέχοντες εξετάζουν τις δυνατότητες του App Inventor και μαθαίνουν να χρησιμοποιούν το App Inventor σε μια τάξη για να ενισχύσουν τη δημιουργικότητα, τη συνεργασία και την επίλυση προβλημάτων.

3.3 Επεκτάσεις

Οποιοσδήποτε έχει εμπειρία προγραμματισμού Java και Android μπορεί να γράψει τα δικά του στοιχεία για το App Inventor χρησιμοποιώντας τον μηχανισμό επέκτασης. Για παράδειγμα, το MIT δημοσίευσε πρόσφατα μια σουίτα επεκτάσεων που σχετίζονται με το Internet of things (IOT) για διασύνδεση με Lego Mindstorms. Χρησιμοποιώντας αυτές τις επεκτάσεις, οι εκπαιδευτικοί μπορούν να συγκεντρώσουν προσαρμοσμένα προγράμματα σπουδών για να αξιοποιήσουν αυτές τις τεχνολογίες στην τάξη και να ενθαρρύνουν τους μαθητές τους να εξερευνήσουν τη διεπαφή μεταξύ του κόσμου του λογισμικού και του κόσμου του υλικού. Προβλέπουμε την ανάπτυξη επεκτάσεων που σχετίζονται με τεχνολογίες τεχνητής νοημοσύνης, συμπεριλαμβανομένης της βαθιάς εκμάθησης, υποστήριξης συσκευών για αναγνώριση εικόνας, ανάλυσης συναισθημάτων, επεξεργασίας φυσικής γλώσσας και πολλά άλλα. Στην ιδανική περίπτωση, αυτές οι πολύπλοκες τεχνολογίες θα μπορούσαν να αξιοποιηθούν από οποιονδήποτε θέλει να λύσει ένα πρόβλημα με το smartphone ως πλατφόρμα.

3.4 Ερευνητικά έργα

Εκτός από τις παιδαγωγικές του εφαρμογές, το App Inventor προσφέρει εξαιρετικές ευκαιρίες για έρευνα στην εκπαίδευση και σε άλλους τομείς. Η πρώιμη εργασία επικεντρώθηκε στην κατανόηση του τρόπου με τον οποίο να ονομάζονται κατάλληλα συστατικά για εκπαιδευτική χρήση. Η χρηστικότητα σε συγκεκριμένα πλαίσια, όπως οι ανθρωπιστικές ανάγκες και τα εκπαιδευτικά περιβάλλοντα, είναι επίσης ένας τομέας ενδιαφέροντος. Πιο πρόσφατα, το App Inventor έχει χρησιμοποιηθεί ως μηχανισμός συλλογής και οπτικοποίησης δεδομένων. Αυτήν τη στιγμή διερευνούμε την επέκταση των δυνατοτήτων του App Inventor για να συμπεριλάβουμε τη συνεργασία σε πραγματικό χρόνο μεταξύ των μαθητών, η οποία θα πρέπει να αποφέρει πρόσθετες εκπαιδευτικές ευκαιρίες.

3.5 Ενδυνάμωση μέσω Προγραμματισμού

Τοποθετώντας την έξοδο του μαθητικού προγραμματισμού σε κινητές συσκευές, το App Inventor επιτρέπει στους μαθητές να μετακινήσουν την εργασία τους από τα παραδοσιακά εργαστήρια υπολογιστών και στην καθημερινή τους ζωή και τις κοινότητές τους. Αυτή η μετάβαση έχει ισχυρές συνέπειες για το τι δημιουργούν οι μαθητές και πώς οραματίζονται τον εαυτό τους ως ψηφιακοί δημιουργοί. Επιτρέπει στους μαθητές να μετατοπίσουν την αίσθηση του εαυτού τους από άτομα που «ξέρουν να κωδικοποιούν» σε μέλη μιας κοινότητας που έχουν τη δυνατότητα να έχουν πραγματικό αντίκτυπο στη ζωή τους και στις ζωές των άλλων.

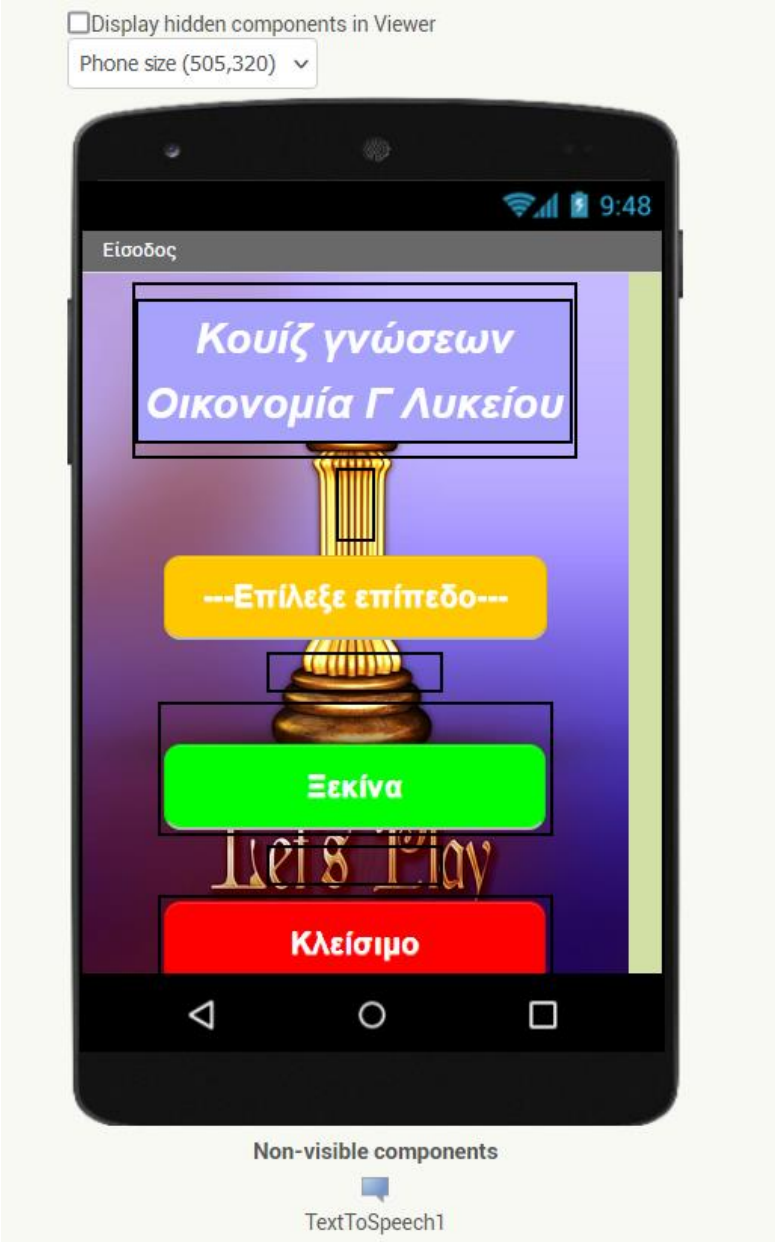
4. Ανάπτυξη εφαρμογής

Στο μέρος της πρακτικής στην εργασία αυτή δημιουργήθηκε μια εφαρμογή με την χρήση του App Inventor. Πρόκειται για μία εφαρμογή εξέτασης γνώσεων, με κοΰΐζ ερωτήσεων πολλαπλών επιλογών, των μαθητών στο μάθημα της Οικονομίας για τη Γ Λυκείου. Η εφαρμογή μπορεί εύκολα να μετατραπεί για να απευθύνεται για εξέταση και άλλων αντικειμένων ή του ίδιου με διαφορετικές ερωτήσεις. Είναι μια από τις βασικές δυνατότητες που δίνει για μελλοντικές επεκτάσεις και τροποποιήσεις. Η εφαρμογή αποτελείται από δέκα σελίδες, βάση των δυνατοτήτων που παρέχει το διαδικτυακό App Inventor. Η αρχική σελίδα δίνει τη δυνατότητα επιλογής επιπέδου ανάμεσα σε εύκολο και δύσκολο, υπάρχουν τέσσερις σελίδες με ερωτήσεις πολλαπλής επιλογής σε κάθε επίπεδο και τέλος μία σελίδα αποτελεσμάτων – ανατροφοδότησης. Με τη χρήση Text to speech γίνεται ανάγνωση των κειμένων των ερωτήσεων και των δυνατών απαντήσεων. Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά η εφαρμογή στο Designer και γίνεται σχολιασμός του κώδικά της όπως εμφανίζεται στο Blocks Editor. Έχει επιλεχθεί να χρησιμοποιηθούν ευμεγέθη κουμπιά σε όλη την εφαρμογή που διευκολύνουν το χρήστη. Η εφαρμογή έχει σχεδιαστεί για κινητό τηλέφωνο όμως δεν αποκλείεται η χρήση της από συσκευές τύπου ταμπλέτας.

4.1 Αρχική Σελίδα

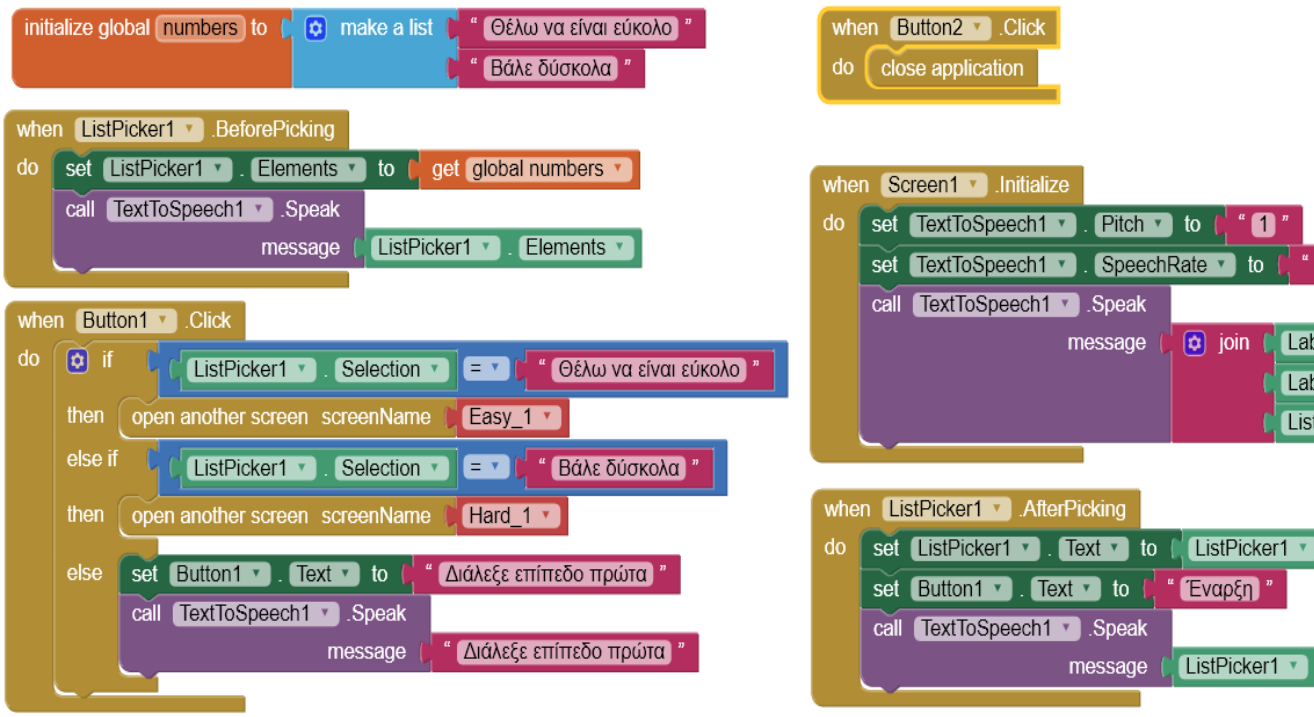
Η αρχική οθόνη της εφαρμογής έχει απλή εμφάνιση (βλ. Εικ. 5), φιλική στο χρήστη και υποστηρίζει φωνητική καθοδήγηση μέσω της δυνατότητας Text to speech. Από αυτή γίνεται η επιλογή επιπέδου δυσκολίας των ερωτήσεων με μορφή λίστας. Ο χρήστης μπορεί να επιλέξει ανάμεσα σε δύο επίπεδα, εύκολο και δύσκολο μέσω λίστας επιλογών. Αν ο χρήστης δεν επιλέξει επίπεδο δεν μπορεί να προχωρήσει παρακάτω και εμφανίζεται το μήνυμα «επίλεξε επίπεδο πρώτα», ενημερώνεται δε με το αντίστοιχο ηχητικό μήνυμα. Αφού επιλέξει το επίπεδο μπορεί να προχωρήσει στις ερωτήσεις πατώντας το κουμπί «Ξεκίνα». Στην αρχική σελίδα υπάρχει και κουμπί το οποίο κλείνει

την εφαρμογή, αν και αυτό δεν είναι απαραίτητο αφού μπορούν να χρησιμοποιηθούν τα χειριστήρια του Android, τοποθετήθηκε όμως για να αυξήσει την ευκολία χρήσης.



Εικόνα 10 Designer αρχική

Στο επίπεδο των blocks έχουν ρυθμιστεί οι λειτουργίες των κουμπιών έναρξης, κλεισίματος της εφαρμογής και της λίστας επιλογών επιπέδου. Επίσης έχει ρυθμιστεί η λειτουργία φωνητικής καθοδήγησης (βλ. Εικ. 12).



Εικόνα 11 Blocks αρχικής οθόνης

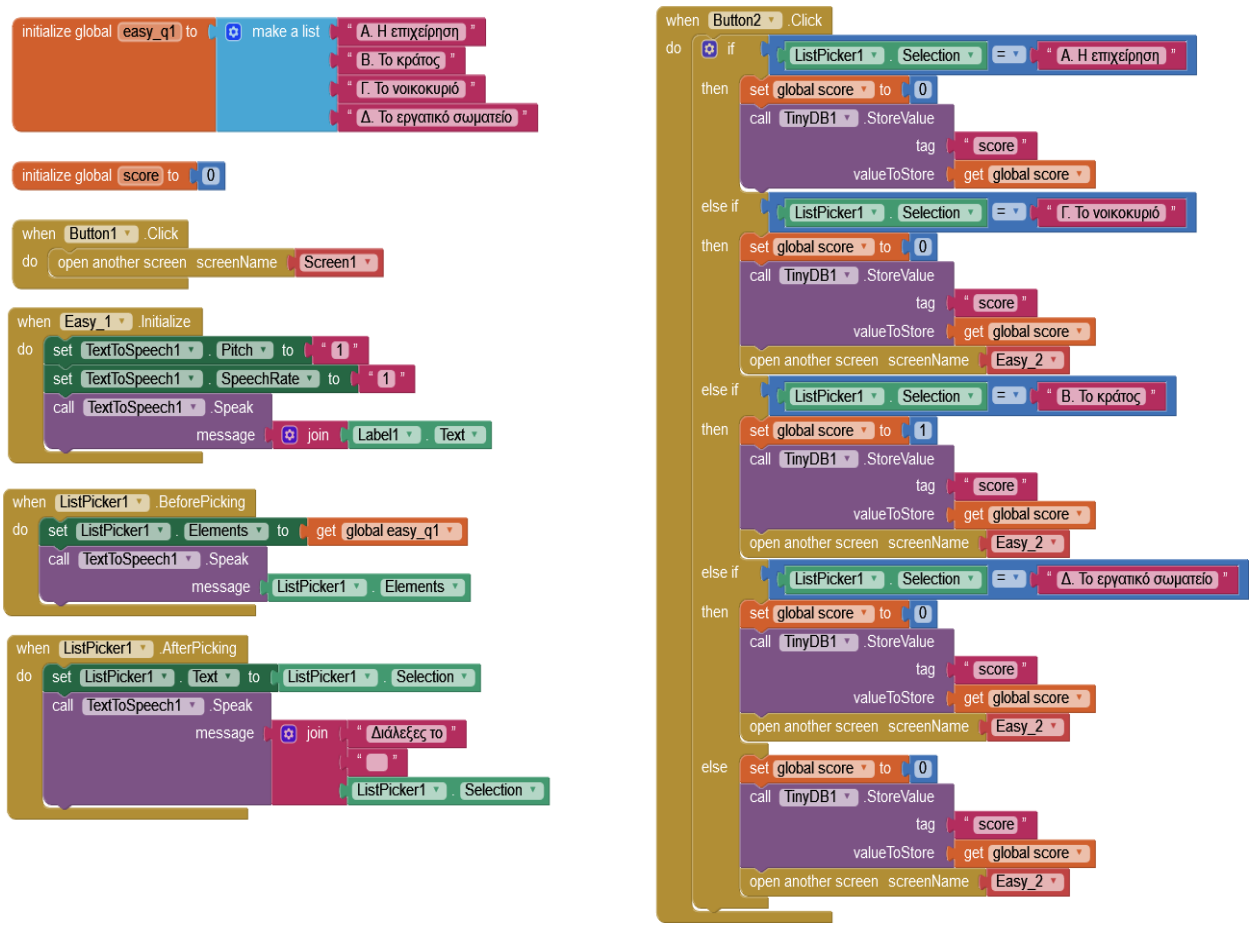
4.2 Σελίδες Ερωτήσεων

Η σελίδα της πρώτης ερώτησης του εύκολου επιπέδου έχει την ερώτηση σε γραφικό φόντο που δεν επηρεάζει την ορατότητα. Κάτω από την ερώτηση υπάρχει το κουμπί ενεργοποίησης της λίστας επιλογών για τις απαντήσεις. Ο χρήστης μπορεί να επιλέξει μία από τις τέσσερις δυνατές επιλογές. Οι επιλογές διαβάζονται στο χρήστη μέσω του Text to speech και η επιλογή που κάνει επιβεβαιώνεται τόσο οπτικά όσο και φωνητικά.

Στο κάτω μέρος της οθόνης υπάρχει κουμπί με το οποίο ο χρήστης δύναται να επιστρέψει στην αρχική οθόνη και κουμπί με το οποίο επιβεβαιώνει την επιλογή του και προχωρά το τεστ στην επόμενη ερώτηση. Μη ορατό στοιχείο είναι μια μικρή βάση δεδομένων που έχει προστεθεί και η οποία αποτελεί λειτουργικό στοιχείο της εφαρμογής για την καταμέτρηση του σκορ επιτυχιών του χρήστη.

Εικόνα 12 Designer Εύκολο 1

Στο επίπεδο των blocks έχουμε τον ορισμό της λειτουργίας των κουμπιών, της λίστας και της αρχικοποίησης της global score. Το κουμπί αρχική (Button1) επιστρέφει στην αρχική οθόνη. Το κουμπί επόμενο (Button2) οδηγεί στην επόμενη σελίδα ανάλογα με το αν η απάντηση είναι σωστή ή λάθος προσθέτει στη global score ένα για το σωστό και μηδέν για το λάθος. Αν ο χρήστης δεν επιλέξει κάτι μπορεί να προχωρήσει στην επόμενη ερώτηση χωρίς να προστεθεί βαθμός. Το Text to speech κάνει ανάγνωση της ερώτησης, λίστας επιλογών και της επιλογής απάντησης από το χρήστη.

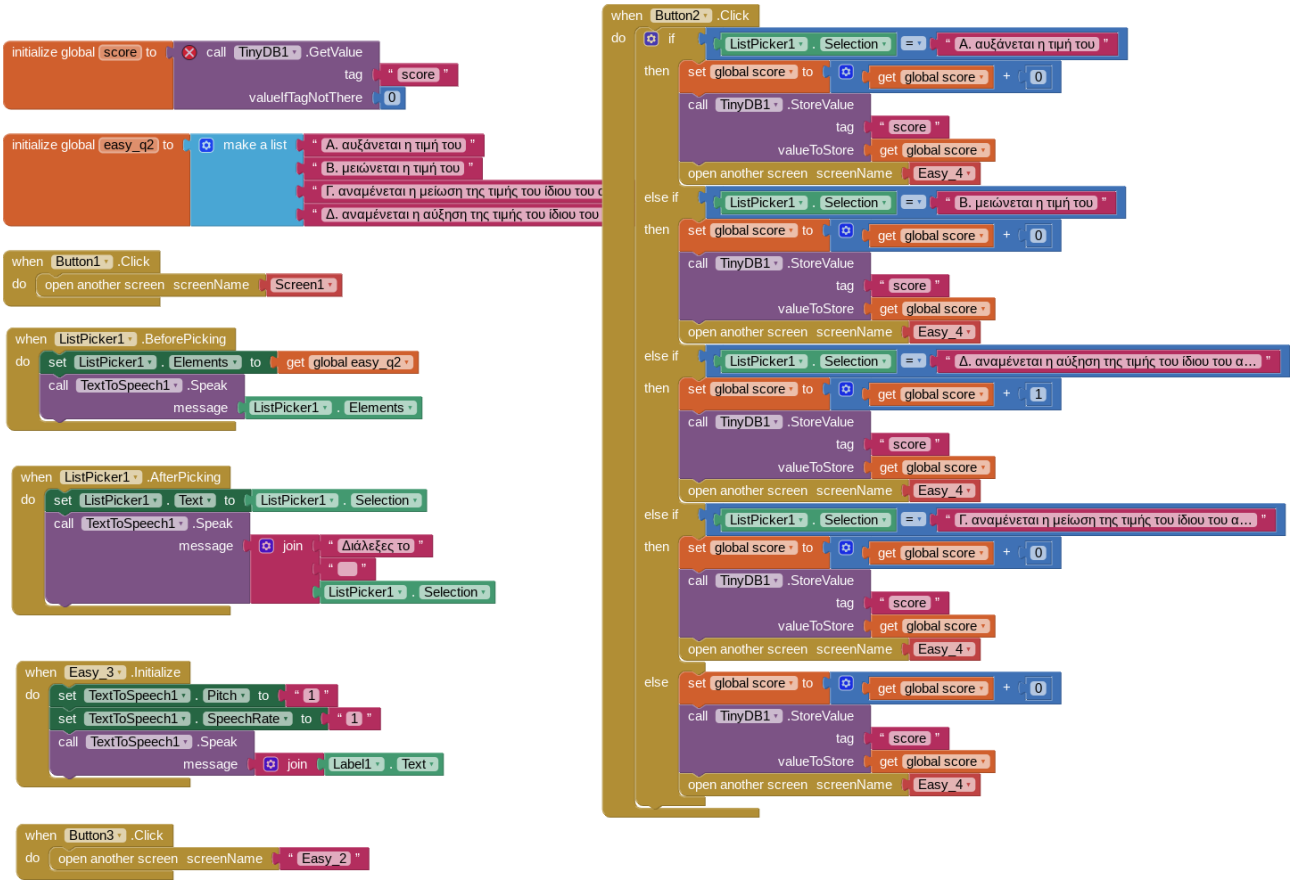


Εικόνα 13 Blocks εύκολο 1

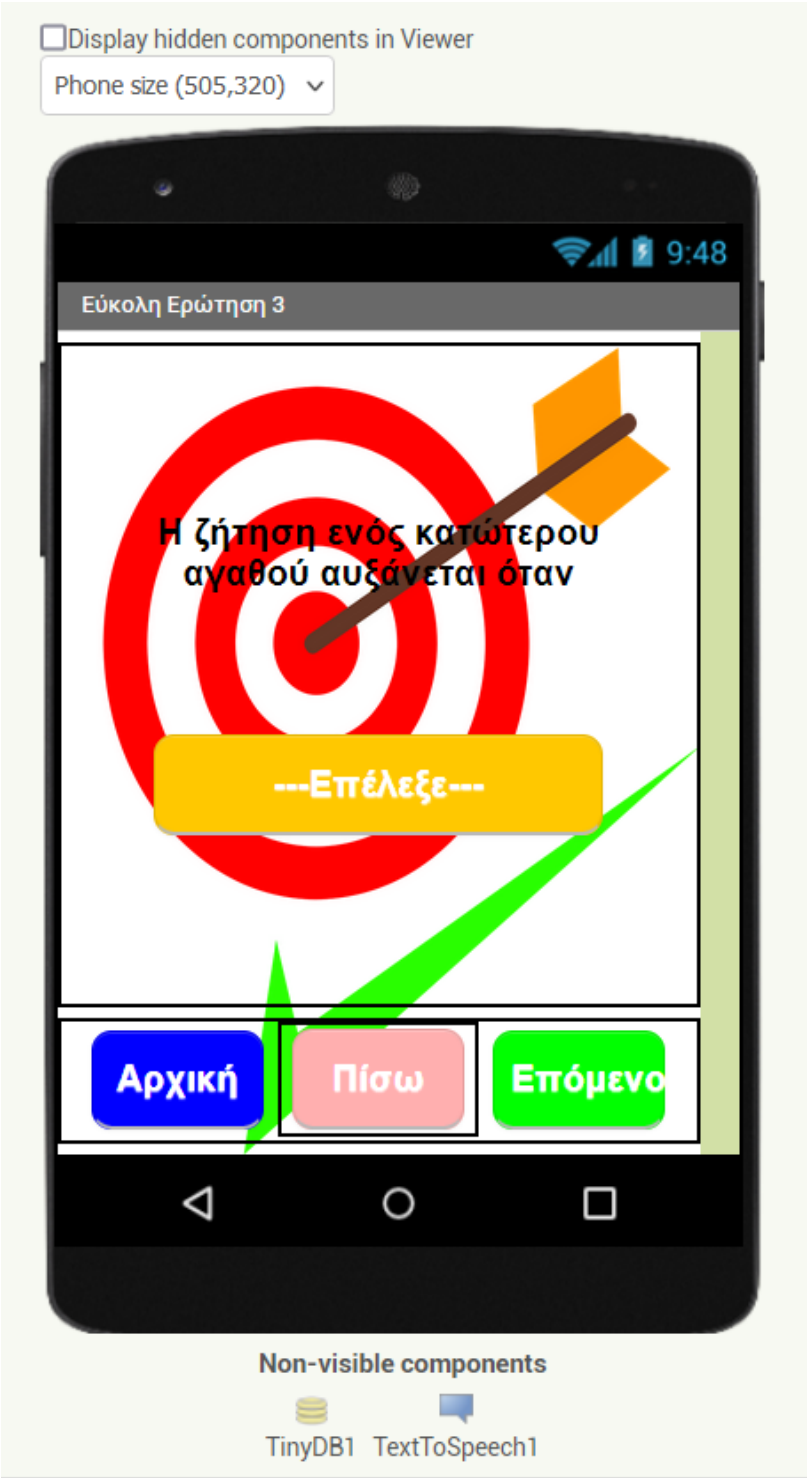
Οι υπόλοιπες τρεις σελίδες των εύκολων ερωτήσεων δεν διαφοροποιούνται ιδιαίτερα από την πρώτη ερώτηση. Οι λειτουργίες διατηρήθηκαν ίδιες κάτι το οποίο είναι θεμιτό αφού δεν δημιουργεί σύγχυση στο χρήστη. Η φωνητική λειτουργία είναι η ίδια σε όλες της σελίδες. Η μοναδική αλλαγή που υπάρχει αφορά στην προσθήκη κουμπιού «Πίσω», σε όλες τις οθόνες, με το οποίο ο χρήστης επιστρέφει στην

προηγούμενη σελίδα για να επαναλάβει την προηγούμενη ερώτηση. Στην τελική ερώτηση το κουμπί «Επόμενο» μετατρέπεται σε «Τέλος» και οδηγεί στη σελίδα αποτελεσμάτων. Η score λαμβάνει τιμή από την προηγούμενη σελίδα καλώντας την βάση δεομένων TinyDB1 και προσθέτει την τιμή ένα αν η επιλογή είναι σωστή και μηδέν αν η επιλογή είναι μία από τις λάθος απαντήσεις.

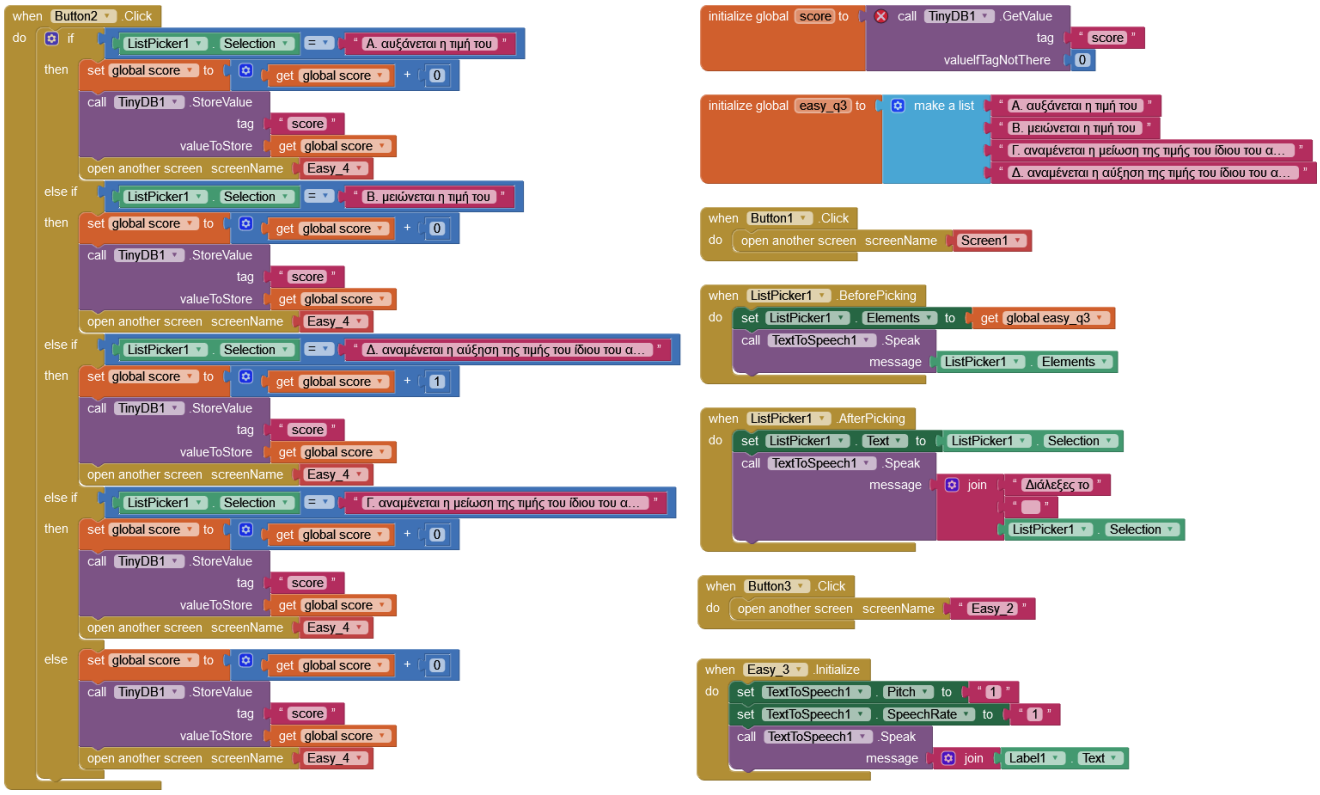
Εικόνα 14 Designer Εύκολο 2



Εικόνα 15 Blocks εύκολο 2

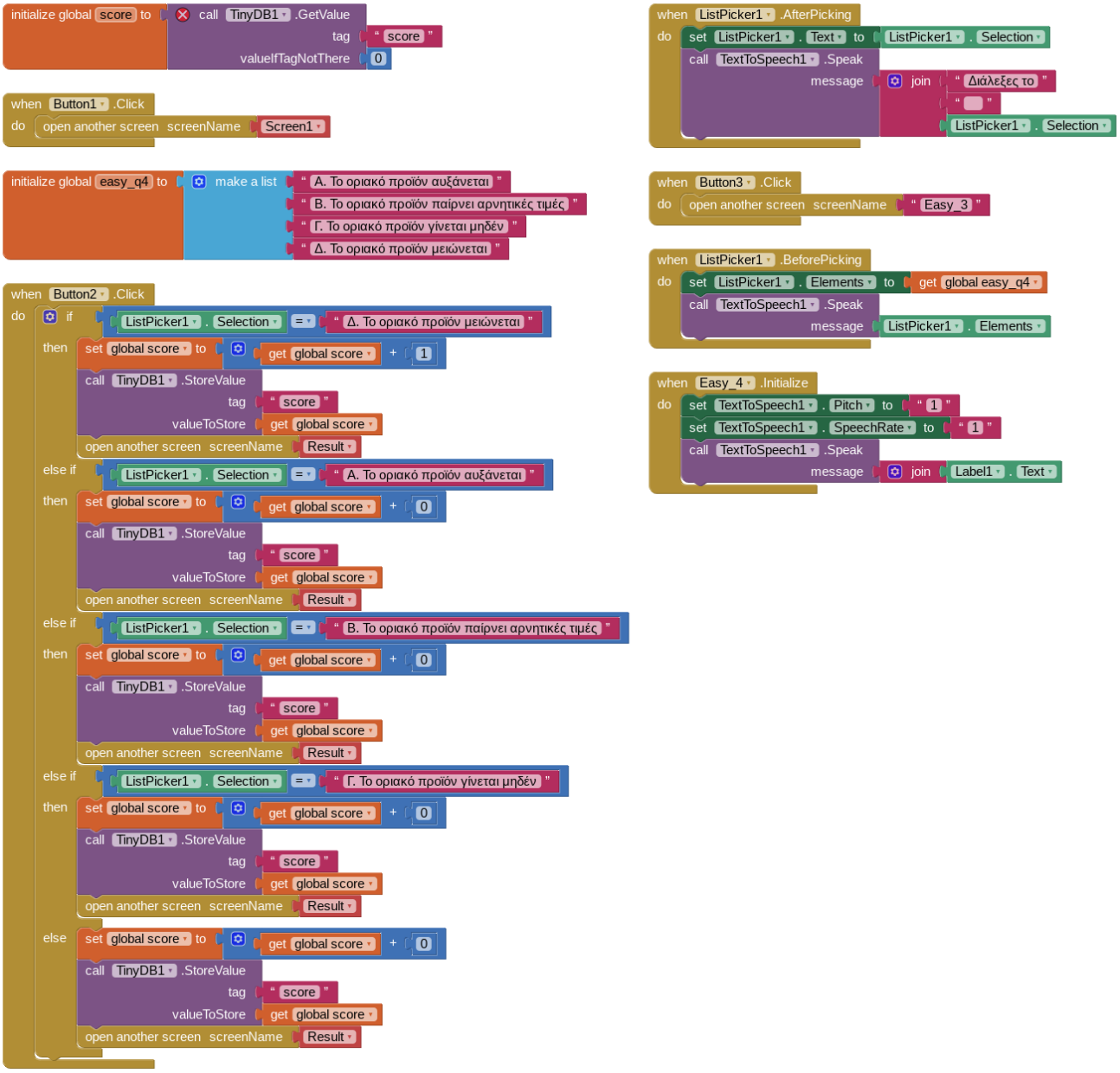


Εικόνα 16 Designer Εύκολο 3



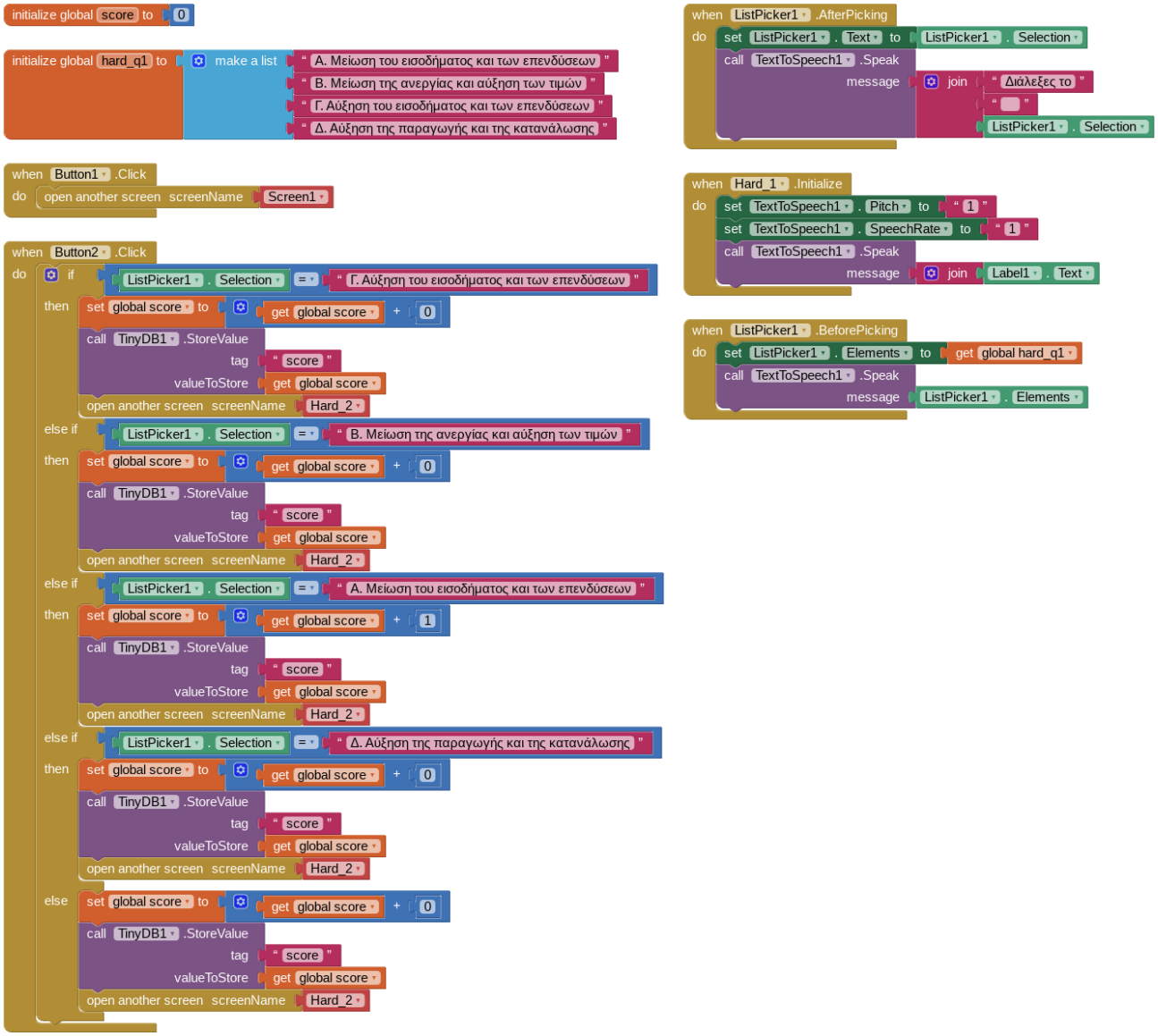
Εικόνα 17 Blocks Εύκολο 3

Εικόνα 18 Designer Εύκολο 4



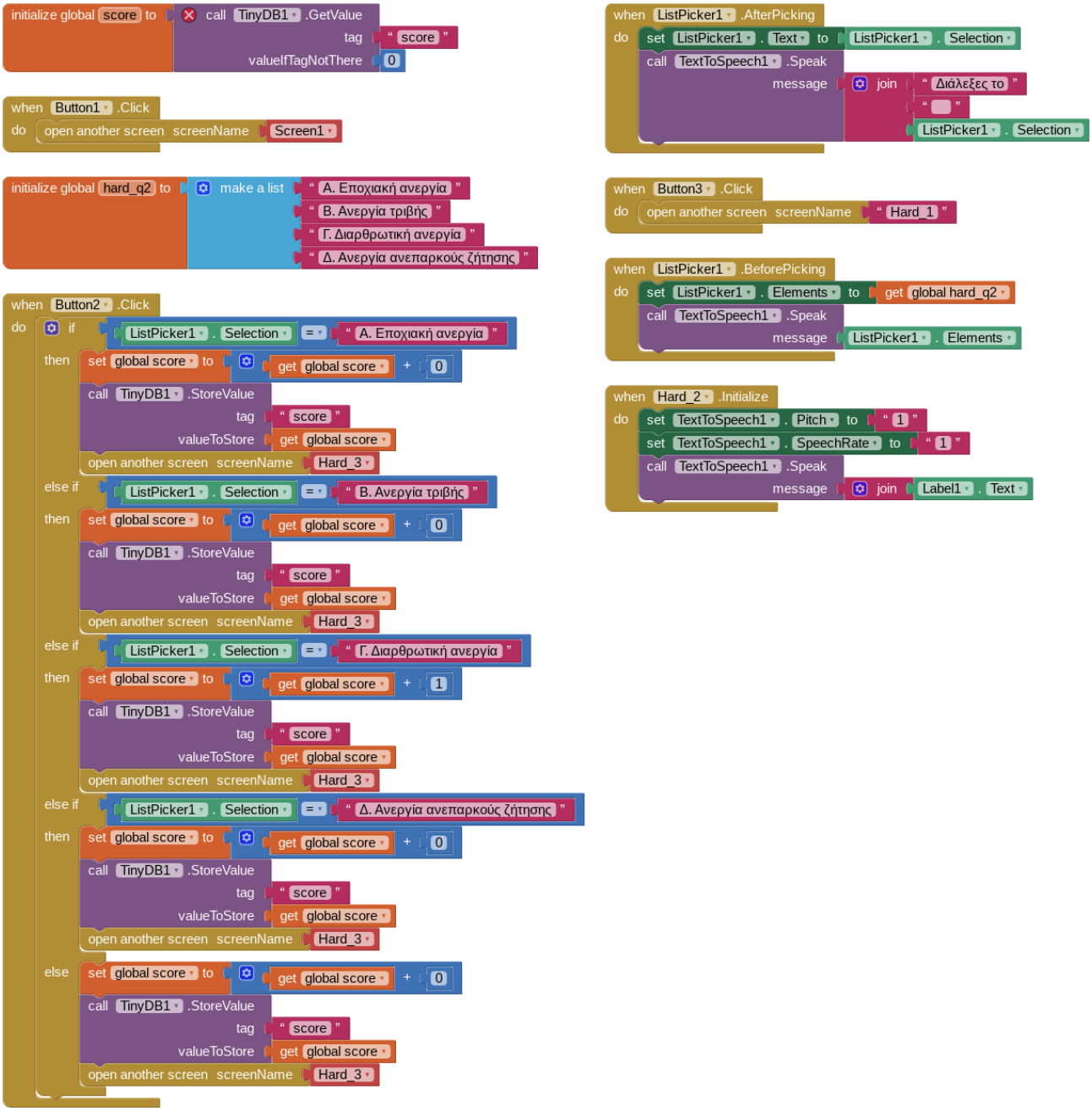
Εικόνα 19 Blocks εύκολο 4

Εικόνα 20 Designer Δύσκολο 1



Εικόνα 21 Blocks δύσκολο 1

Εικόνα 22 Designer Δύσκολο 2

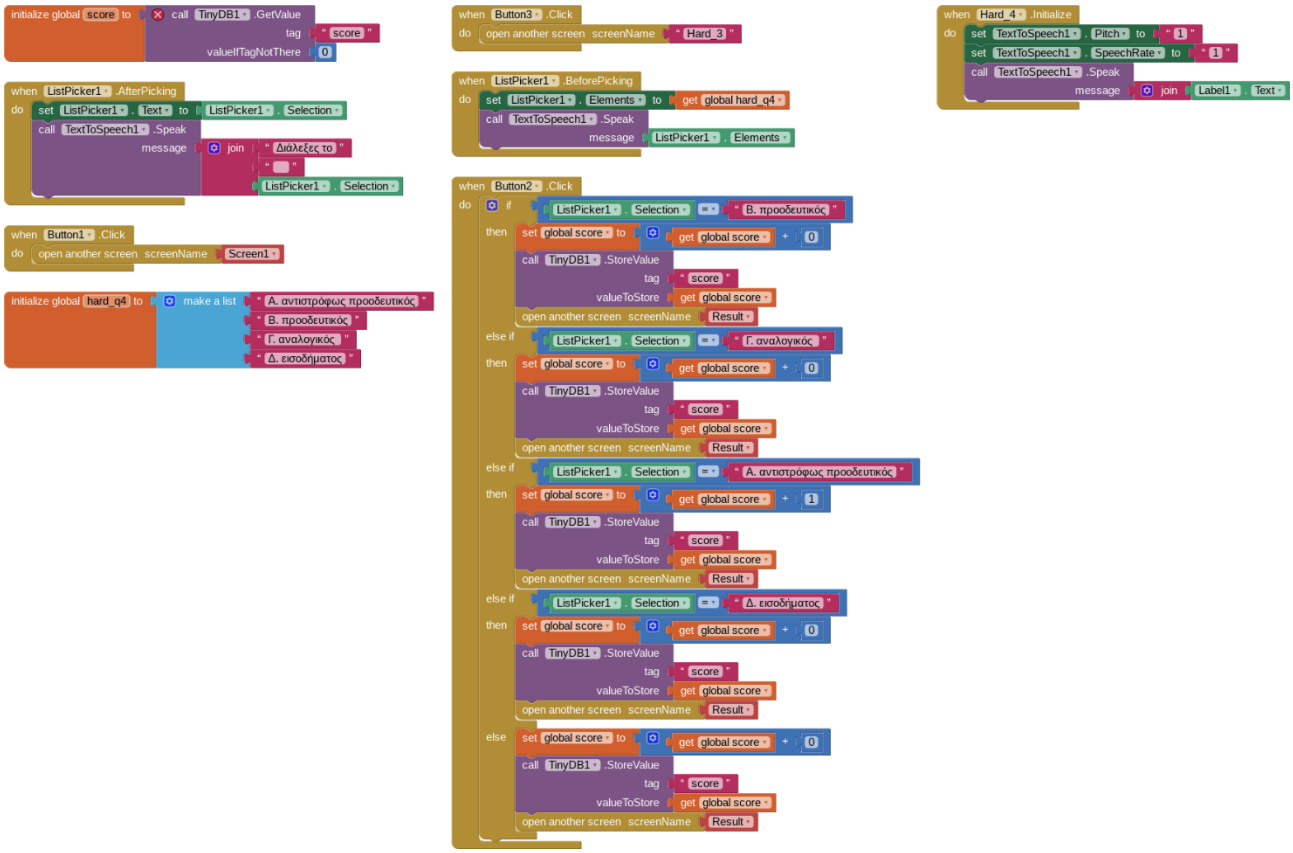


Εικόνα 23 Blocks δύσκολο 2

Εικόνα 24 Designer Δύσκολο 3

Εικόνα 25 Blocks δύσκολο 3

Εικόνα 26 Designer Δύσκολο 4



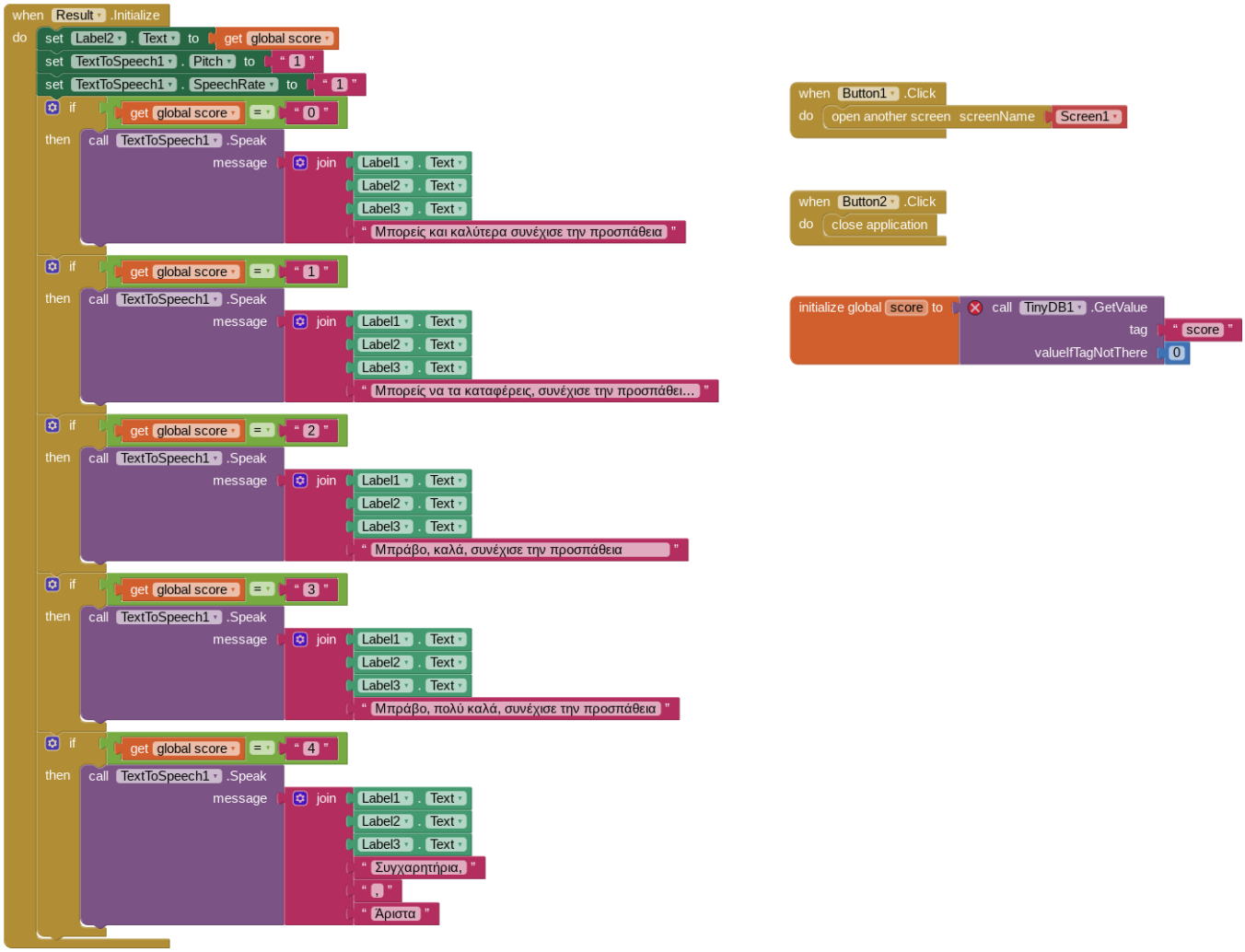
Εικόνα 27 Blocks δύσκολο 4

4.3 Σελίδα αποτελεσμάτων

Στη σελίδα των αποτελεσμάτων παρουσιάζεται το σκορ στη μορφή «Αποτέλεσμα Χ στα 4» ανάλογα με το σκορ και ταυτόχρονα ακούγεται ένα θετικό μήνυμα ενθάρρυνσης για το χρήστη (βλ. Εικ. 30). Θετική ενθάρρυνση έχει και η εικόνα φόντου που έχει επιλεγεί (βλ. Εικ. 29). Η μεταβλητή σκορ συλλέγει την πληροφορία της από την βάση δεδομένων που χρησιμοποιήθηκε στις προηγούμενες οθόνες και αποδίδει την τιμή αυτή σε ένα πλαίσιο κειμένου προκειμένου να δημιουργήσει πρόταση με τα υπόλοιπα σταθερά κείμενα

Εικόνα 28 Designer αποτέλεσμα

Το κουμπί Επανέναρξη στο κάτω μέρος της οθόνης του χρήστη μπορεί να χρησιμοποιηθεί για να επιστρέψει το χρήστη στην αρχική οθόνη προκειμένου να ξεκινήσει εκ νέου το τεστ στο ίδιο ή σε διαφορετικό επίπεδο. Το κουμπί «Έξοδος» τερματίζει την λειτουργία της εφαρμογής. Στα μπλοκ διακρίνουμε τις λειτουργίες των δύο αυτών κουμπιών καθώς επίσης την λειτουργία της σελίδας με τον έλεγχο της τιμής της global score και τα φωνητικά μηνύματα που ακούγονται κατά περίπτωση.



Εικόνα 29 Blocks αποτέλεσμα

Συμπεράσματα

Τεχνολογία και εκπαίδευση είναι δύο συνδεδεμένοι τομείς με ταχεία ανάπτυξη. Η σύγχρονη εκπαιδευτική διαδικασία απαιτεί την προσθήκη νέων εργασιών και στυλ στη διδασκαλία τόσο στην πρόταση όσο και στην πράξη. Σε αυτήν την εργασία, παρουσιάστηκε μια περίπτωση στη χρήση του App Inventor με στόχο να κάνει την αξιολόγηση των μαθητών πιο εύκολη αλλά και πιο ευχάριστη. Η ευκολία χρήσης του App Inventor το καθιστά σαγηνευτικό για όλους, καθώς προσφέρει ποικιλία δυνατοτήτων για αποτελεσματική χρήση. Οι γραφικές γλώσσες επιτρέπουν στους χρήστες να ασχοληθούν με την δημιουργία εφαρμογών ακόμα και αν δεν γνωρίζουν προγραμματισμό ενθαρρύνοντάς τους να ασχοληθούν με το σχεδιασμό εφαρμογών για κινητές συσκευές. Η εκπαιδευτική αξία της σχεδίασης εφαρμογών, η δυνατότητα και πρακτικότητα του σχεδιασμού διαδικτυακών εφαρμογών διδασκαλίας/γραμματισμού μπορεί να ενθαρρύνει τους εκπαιδευτές να εξερευνήσουν και να πειραματιστούν με τις δυνατότητες ενσωμάτωσης αυτών των διαφορετικών συνθηκών γραμματισμού. Δημιουργώντας καινοτόμες εφαρμογές, προσφέρουμε πιο ολοκληρωμένη προσέγγιση στα μαθήματα. Το εργαλείο App Inventor βοηθά στην εκπαιδευτική διαδικασία και στην ανάπτυξη δράσης καθώς και στην εξοικείωση με τις σύγχρονες τεχνολογίες. Η ανάπτυξη και η εκτέλεση μιας εφαρμογής για Android είναι ιδιαίτερα απαιτητική, όμως γίνονται συνεχώς προσπάθειες για την παραγωγή εργαλείων για τη διευκόλυνση των χρηστών. Η εφαρμογή App Inventor είναι εύκολο να χρησιμοποιηθεί στην εκπαίδευση για να κάνει την εκμάθηση πιο ελκυστική και εύκολη στους μαθητές. Από μικρή ηλικία οι μαθητές είναι εξοικειωμένοι με τη χρήση κινητών τηλεφώνων και γενικότερα ηλεκτρονικών συσκευών, με την αξιοποίηση της υπάρχουσας εμπειρίας τους μπορούν να ανακαλύπτουν τη γνώση από μόνοι τους. Το App Inventor είναι ένας αξιόπιστος και γρήγορος τρόπος ώστε ο εκπαιδευτικός να δημιουργήσει ποικιλία εφαρμογών για κινητά, tablet κλπ.

Στα πλεονεκτήματα του συγκαταλέγονται

- Ευκολία χρήσης
- Πολλαπλές δυνατότητες

- Δωρεάν χρήση
- Υποστήριξη από Google και MIT
- Μπορεί να κάνει πολλές ενέργειες ταυτόχρονα
- Ευνοεί την ανάπτυξη κριτικής σκέψης
- Δύναται να επικαλεστεί και άλλες εφαρμογές.
- Με τη χρήση των μπλοκ γίνεται προσιτό σε άτομα που δεν γνωρίζουν από προγραμματισμό.
- Εισάγει έννοιες προγραμματισμού υπολογιστών κατά τη δημιουργία μιας εφαρμογής.
- Εκπαιδευτικά βίντεο για αρχάριους.
- Οι χρήστες μπορούν να προκαλέσουν τον εαυτό τους με ήδη δημιουργημένα σενάρια ή να δοκιμάσουν νέα έργα για να δουλέψουν από την αρχή.
- Οι εφαρμογές που δημιουργούνται από τους χρήστες παρουσιάζονται στον ιστότοπο.
- Παρέχονται σχέδια μαθημάτων για εκπαιδευτικούς που ενσωματώνουν τη χρήση της εφαρμογής.
- Φόρουμ κοινότητας για απαντήσεις σε ερωτήσεις και βοήθεια σε χρήστες που αντιμετωπίσουν προβλήματα

Στα μειονεκτήματα περιλαμβάνονται:

- Οι έτοιμες εντολές δεν επαρκούν για σύνθετες λειτουργίες
- Η ονομασία των σελίδων αποτελεί πρόκληση
- Η σύνδεση στο διαδίκτυο είναι απαραίτητη
- Δεν υποστηρίζει ελληνικά
- Ο Companion παρουσιάζει προβλήματα ανάλογα τις εκδόσεις του κινητού.
- Υπάρχει περιορισμός στο πλήθος των σελίδων που μπορεί να υποστηρίξει
- Οι διαφοροποιήσεις μεταξύ κινητών συσκευών σε υλικό και λογισμικό δημιουργούν προβλήματα στο σχεδιασμό.
- Η διαδικασία δημιουργίας μιας εφαρμογής μπορεί να προκαλεί σύγχυση για ορισμένους.
- Οι εφαρμογές περιορίζονται στις συσκευές Android

- Απαιτεί σύνδεση με λογαριασμό Google.

Η εφαρμογή μπορεί να αλλάξει σε σχετικά μικρό χρόνο από τον εκπαιδευτικό τα κείμενα της και να χρησιμοποιηθεί με νέες ερωτήσεις για το ίδιο μάθημα ή διαφορετικά μαθήματα. Μία μελλοντική επέκταση της εργασίας αυτής θα μπορούσε να είναι χρήση διαφορετικών τύπων ερωτήσεων με στοιχεία όπως μεταφοράς και απόθεσης ή αντιστοιχίσεων με την προϋπόθεση πως δε θα αυξηθεί ο βαθμός δυσκολίας για στην χρήση από τους μαθητές αλλά και από τον εκπαιδευτικό που θα επεμβαίνει για να δημιουργεί εύκολα νέες εφαρμογές.

5. Βιβλιογραφία

[1] M. I. o. Technology, «MIT Expert Trainer,» MIT, 2012-2022. [Ηλεκτρονικό]. Available: http://appinventor.mit.edu/explore/mit-expert-trainer-course. [Πρόσβαση 1 Φεβρουάριος 2023].

[2] «pixabay,» [Ηλεκτρονικό]. Available: https://pixabay.com/el/illustrations/%cf%83%ce%ba%ce%ac%ce%ba%ce%b9-%ce%b2%ce%b1%cf%83%ce%af%ce%bb%ce%b9%cf%83%cf%83%ce%b1-%cf%83%cf%84%cf%81%ce%b1%cf%84%ce%b7%ce%b3%ce%b9%ce%ba%ce%ae-454475/. [Πρόσβαση 1 2 2023].

[3] «pixabay,» [Ηλεκτρονικό]. Available: https://pixabay.com/el/illustrations/%cf%83%ce%ba%ce%ac%ce%ba%ce%b9-%ce%b2%ce%b1%cf%83%ce%af%ce%bb%ce%b9%cf%83%cf%83%ce%b1-%cf%83%cf%84%cf%81%ce%b1%cf%84%ce%b7%ce%b3%ce%b9%ce%ba%ce%ae-454475/. [Πρόσβαση 1 2 2023].

[4] «pixabay,» [Ηλεκτρονικό]. Available: https://pixabay.com/el/vectors/%cf%83%cf%84%cf%8c%cf%87%ce%bf%cf%82-%ce%b2%ce%ad%ce%bb%ce%bf%cf%82-%cf%86%ce%ad%cf%81%ce%bd%cf%89-%cf%83%ce%b5-%cf%80%ce%ad%cf%81%ce%b1%cf%82-2704455/. [Πρόσβαση 1 2 2023].

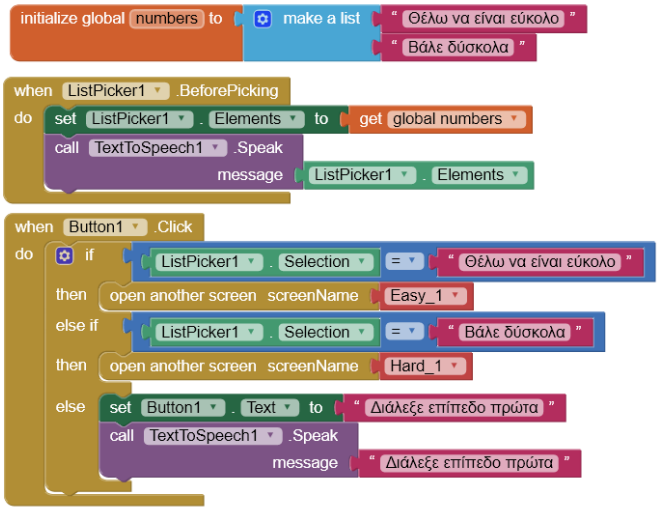
[5] «pixabay,» [Ηλεκτρονικό]. Available: https://pixabay.com/el/illustrations/%cf%83%ce%b9%ce%bb%ce%bf%cf%85%ce%ad%cf%84%ce%b1-%ce%ac%ce%bd%ce%b4%cf%81%ce%b1%cf%82-%cf%80%ce%b7%ce%b4%ce%ac%cf%89-%ce%ba%ce%af%ce%bd%ce%b7%cf%84%cf%81%ce%bf-936719/. [Πρόσβαση 1 2 2023].

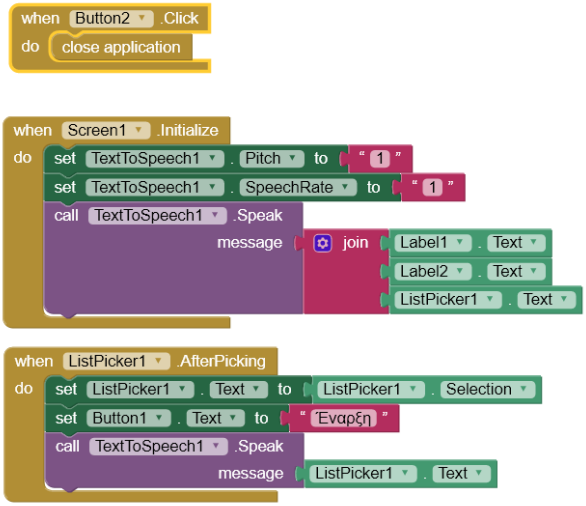
[6] H. A. E. S. L. L. David Wolber, App inventor : create your own Android apps, Sebastopol: O'Reilly, 2011.

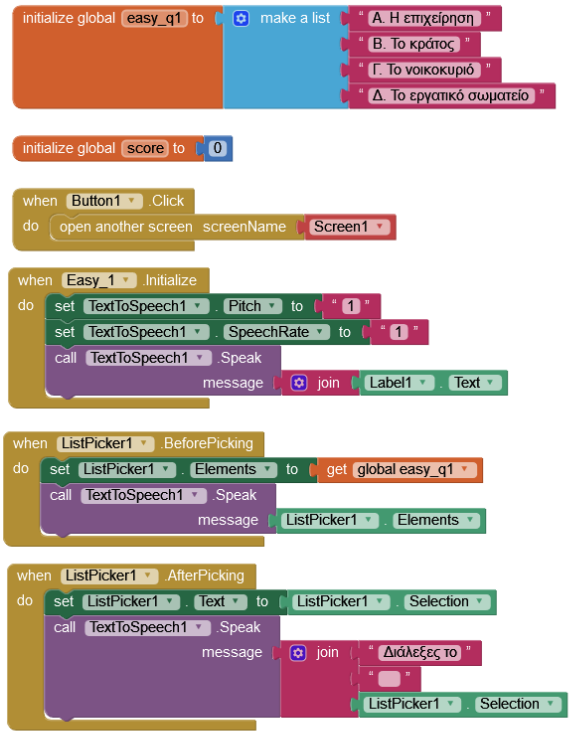
[7] «MIT App Inventor,» MIT, [Ηλεκτρονικό]. Available: https://appinventor.mit.edu/. [Πρόσβαση 1 2 2023].

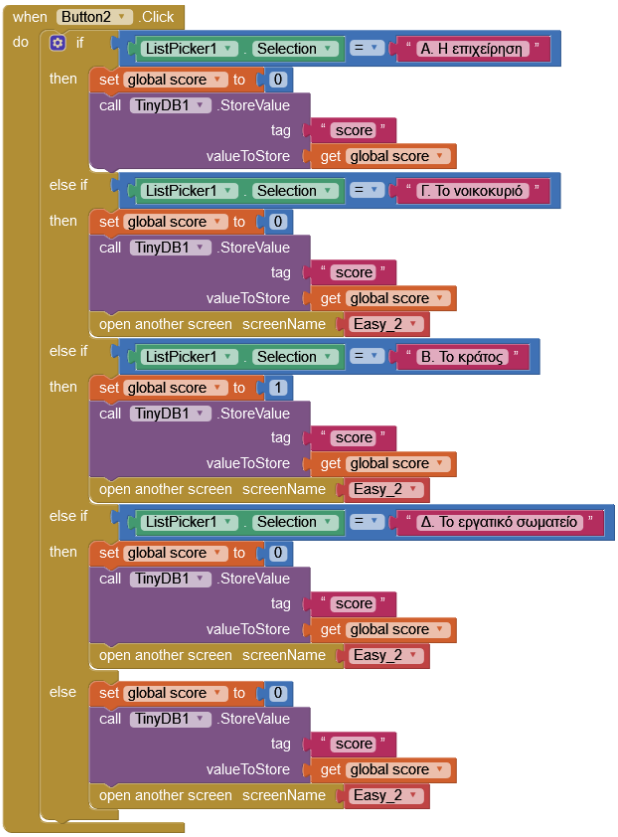
[8] K. R. Felicia Kamriani, App Inventor 2 Essentials, Packt, 2016.

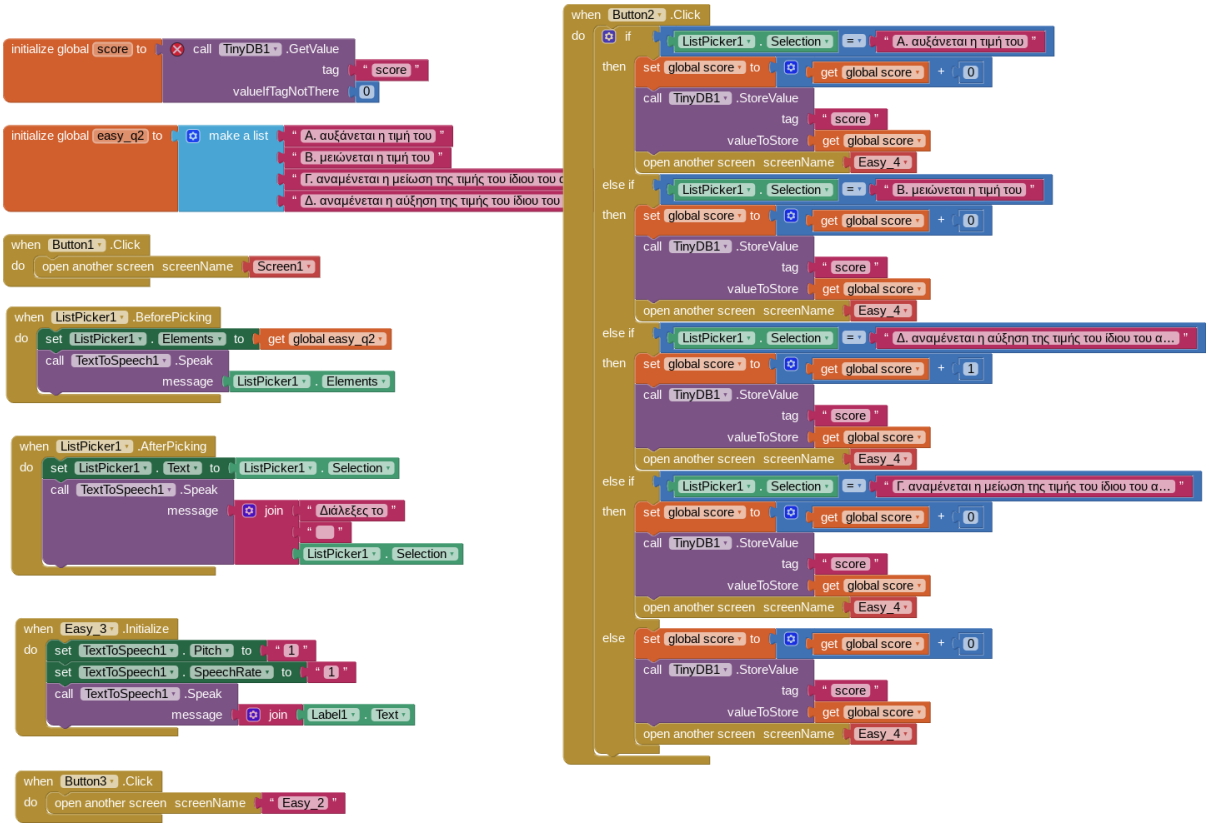
Παράρτημα Κώδικα

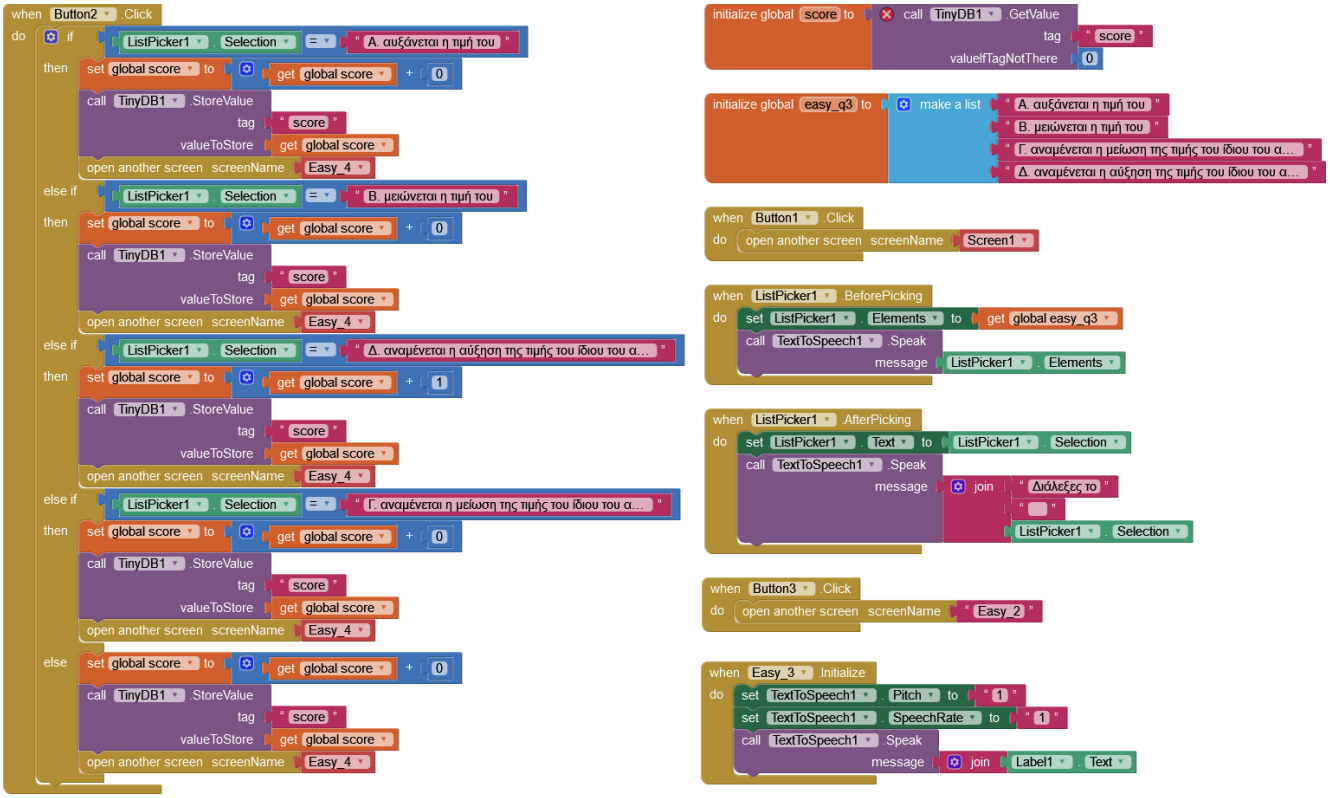









initialize global score to
call TinyDB1 .GetValue
tag
" score "
valueIfTagNotThere
0
initialize global easy_q2 to
make a list
" Α. αυξάνεται η τιμή του "
" Β. μειώνεται η τιμή του "
" Γ. αναμένεται η μείωση της τιμής του ίδιου του α
" Δ. αναμένεται η αύξηση της τιμής του ίδιου του
when Button1 .Click
do open another screen screenName Screen1
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global easy_q2
call TextToSpeech1 .Speak
message ListPicker1 . Elements
when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
call TextToSpeech1 .Speak
message join " Διάλεξες το "
" "
ListPicker1 . Selection
when Easy_3 .Initialize
do set TextToSpeech1 . Pitch to " 1 "
set TextToSpeech1 . SpeechRate to " 1 "
call TextToSpeech1 .Speak
message join Label1 . Text
when Button3 .Click
do open another screen screenName " Easy_2 "
when Button2 .Click
do if ListPicker1 . Selection = " Α. αυξάνεται η τιμή του "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Β. μειώνεται η τιμή του "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Δ. αναμένεται η αύξηση της τιμής του ίδιου του α... "
then set global score to get global score + 1
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Γ. αναμένεται η μείωση της τιμής του ίδιου του α... "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4


when Button2 .Click
do if ListPicker1 . Selection = " Α. αυξάνεται η τιμή του "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Β. μειώνεται η τιμή του "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Δ. αναμένεται η αύξηση της τιμής του ίδιου του α... "
then set global score to get global score + 1
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else if ListPicker1 . Selection = " Γ. αναμένεται η μείωση της τιμής του ίδιου του α... "
then set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
else set global score to get global score + 0
call TinyDB1 .StoreValue
tag " score "
valueToStore get global score
open another screen screenName Easy_4
initialize global score to
call TinyDB1 .GetValue
tag
" score "
valueIfTagNotThere
0
initialize global easy_q3 to
make a list
" Α. αυξάνεται η τιμή του "
" Β. μειώνεται η τιμή του "
" Γ. αναμένεται η μείωση της τιμής του ίδιου του α... "
" Δ. αναμένεται η αύξηση της τιμής του ίδιου του α... "
when Button1 .Click
do open another screen screenName Screen1
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global easy_q3
call TextToSpeech1 .Speak
message ListPicker1 . Elements
when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
call TextToSpeech1 .Speak
message join " Διάλεξες το "
" "
ListPicker1 . Selection
when Button3 .Click
do open another screen screenName " Easy_2 "
when Easy_3 .Initialize
do set TextToSpeech1 . Pitch to " 1 "
set TextToSpeech1 . SpeechRate to " 1 "
call TextToSpeech1 .Speak
message join Label1 . Text

```
initialize global score to
    call TinyDB1 .GetValue
        tag "score"
        valueIfTagNotThere 0

when Button1 .Click
do open another screen screenName Screen1

initialize global easy_q4 to
    make a list
        "Α. Το οριακό προϊόν αυξάνεται"
        "Β. Το οριακό προϊόν παίρνει αρνητικές τιμές"
        "Γ. Το οριακό προϊόν γίνεται μηδέν"
        "Δ. Το οριακό προϊόν μειώνεται"

when Button2 .Click
do if ( ListPicker1 . Selection = "Δ. Το οριακό προϊόν μειώνεται" )
   then set global score to ( get global score + 1 )
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Result
   else if ( ListPicker1 . Selection = "Α. Το οριακό προϊόν αυξάνεται" )
   then set global score to ( get global score + 0 )
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Result
   else if ( ListPicker1 . Selection = "Β. Το οριακό προϊόν παίρνει αρνητικές τιμές" )
   then set global score to ( get global score + 0 )
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Result
   else if ( ListPicker1 . Selection = "Γ. Το οριακό προϊόν γίνεται μηδέν" )
   then set global score to ( get global score + 0 )
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Result
   else set global score to ( get global score + 0 )
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Result

when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
   call TextToSpeech1 .Speak
       message join "Διάλεξες το " " " ListPicker1 . Selection

when Button3 .Click
do open another screen screenName "Easy_3"

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global easy_q4
   call TextToSpeech1 .Speak
       message ListPicker1 . Elements

when Easy_4 .Initialize
do set TextToSpeech1 . Pitch to "1"
   set TextToSpeech1 . SpeechRate to "1"
   call TextToSpeech1 .Speak
       message join Label1 . Text
```

```
initialize global score to 0

initialize global hard_q1 to make a list
    "Α. Μείωση του εισοδήματος και των επενδύσεων"
    "Β. Μείωση της ανεργίας και αύξηση των τιμών"
    "Γ. Αύξηση του εισοδήματος και των επενδύσεων"
    "Δ. Αύξηση της παραγωγής και της κατανάλωσης"

when Button1 .Click
do open another screen screenName Screen1

when Button2 .Click
do if ListPicker1 . Selection = "Γ. Αύξηση του εισοδήματος και των επενδύσεων"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_2
   else if ListPicker1 . Selection = "Β. Μείωση της ανεργίας και αύξηση των τιμών"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_2
   else if ListPicker1 . Selection = "Α. Μείωση του εισοδήματος και των επενδύσεων"
   then set global score to get global score + 1
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_2
   else if ListPicker1 . Selection = "Δ. Αύξηση της παραγωγής και της κατανάλωσης"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_2
   else set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_2

when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
   call TextToSpeech1 .Speak
       message join "Διάλεξες το" " " ListPicker1 . Selection

when Hard_1 .Initialize
do set TextToSpeech1 . Pitch to "1"
   set TextToSpeech1 . SpeechRate to "1"
   call TextToSpeech1 .Speak
       message join Label1 . Text

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global hard_q1
   call TextToSpeech1 .Speak
       message ListPicker1 . Elements
```

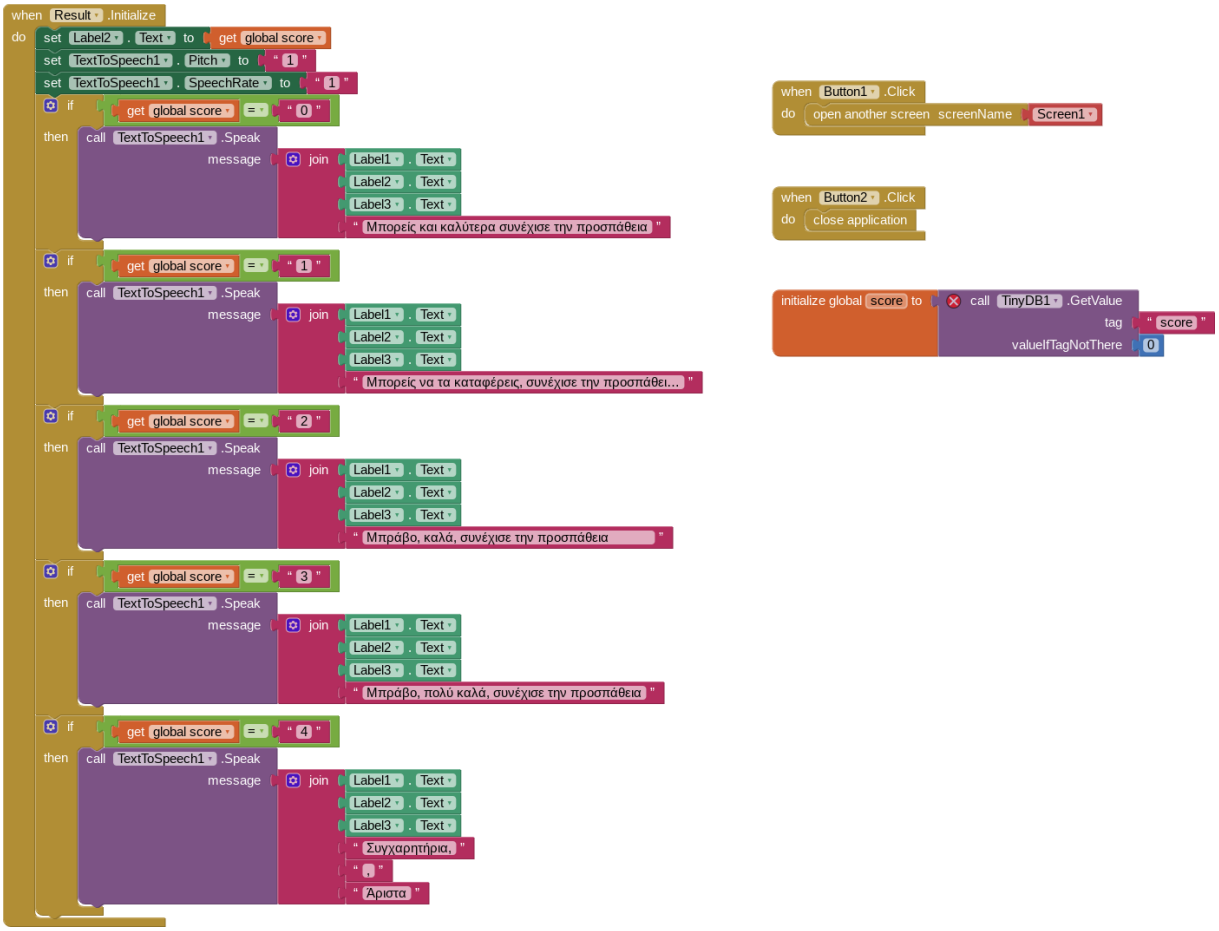
```
initialize global score to
    call TinyDB1 .GetValue
        tag "score"
        valueIfTagNotThere 0

when Button1 .Click
do open another screen screenName Screen1

initialize global hard_q2 to
    make a list
        "Α. Εποχιακή ανεργία"
        "Β. Ανεργία τριβής"
        "Γ. Διαρθρωτική ανεργία"
        "Δ. Ανεργία ανεπαρκούς ζήτησης"

when Button2 .Click
do if
    ListPicker1 . Selection = "Α. Εποχιακή ανεργία"
    then set global score to
        get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_3
    else if
    ListPicker1 . Selection = "Β. Ανεργία τριβής"
    then set global score to
        get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_3
    else if
    ListPicker1 . Selection = "Γ. Διαρθρωτική ανεργία"
    then set global score to
        get global score + 1
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_3
    else if
    ListPicker1 . Selection = "Δ. Ανεργία ανεπαρκούς ζήτησης"
    then set global score to
        get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_3
    else set global score to
        get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_3
```

```
when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
   call TextToSpeech1 .Speak
       message join "Διάλεξες το"
                    " "
                    ListPicker1 . Selection

when Button3 .Click
do open another screen screenName "Hard_1"

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global hard_q2
   call TextToSpeech1 .Speak
       message ListPicker1 . Elements

when Hard_2 .Initialize
do set TextToSpeech1 . Pitch to "1"
   set TextToSpeech1 . SpeechRate to "1"
   call TextToSpeech1 .Speak
       message join Label1 . Text
```

```
initialize global score to
    call TinyDB1 .GetValue
        tag "score"
        valueIfTagNotThere 0

when Button1 .Click
do open another screen screenName Screen1

initialize global hard_q3 to
    make a list
        "Α. Οι υποτροφίες"
        "Β. Οι επενδύσεις για δημόσια έργα"
        "Γ. Οι μισθοί των δημοσίων υπαλλήλων"
        "Δ. Η γραφική ύλη"

when Button2 .Click
do if ListPicker1 . Selection = "Α. Οι υποτροφίες"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_4
   else if ListPicker1 . Selection = "Γ. Οι μισθοί των δημοσίων υπαλλήλων"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_4
   else if ListPicker1 . Selection = "Β. Οι επενδύσεις για δημόσια έργα"
   then set global score to get global score + 1
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_4
   else if ListPicker1 . Selection = "Δ. Η γραφική ύλη"
   then set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_4
   else set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag "score"
            valueToStore get global score
        open another screen screenName Hard_4
```

```
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to get global hard_q3
   call TextToSpeech1 .Speak
       message ListPicker1 . Elements

when ListPicker1 .AfterPicking
do set ListPicker1 . Text to ListPicker1 . Selection
   call TextToSpeech1 .Speak
       message join "Διάλεξες το "
                   " "
                   ListPicker1 . Selection

when Button3 .Click
do open another screen screenName "Hard_2"

when Hard_3 .Initialize
do set TextToSpeech1 . Pitch to "1"
   set TextToSpeech1 . SpeechRate to "1"
   call TextToSpeech1 .Speak
       message join Label1 . Text
```

```
initialize global score to
    call TinyDB1 .GetValue
        tag " score "
        valueIfTagNotThere 0

when ListPicker1 .AfterPicking
do
    set ListPicker1 . Text to ListPicker1 . Selection
    call TextToSpeech1 .Speak
        message join " Διάλεξες το "
                    " "
                    ListPicker1 . Selection

when Button1 .Click
do
    open another screen screenName Screen1

initialize global hard_q4 to
    make a list " Α. αντιστρόφως προοδευτικός "
                " Β. προοδευτικός "
                " Γ. αναλογικός "
                " Δ. εισοδήματος "

when Button3 .Click
do
    open another screen screenName " Hard_3 "

when ListPicker1 .BeforePicking
do
    set ListPicker1 . Elements to get global hard_q4
    call TextToSpeech1 .Speak
        message ListPicker1 . Elements

when Button2 .Click
do
    if ListPicker1 . Selection = " Β. προοδευτικός "
    then
        set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag " score "
            valueToStore get global score
        open another screen screenName Result
    else if ListPicker1 . Selection = " Γ. αναλογικός "
    then
        set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag " score "
            valueToStore get global score
        open another screen screenName Result
    else if ListPicker1 . Selection = " Α. αντιστρόφως προοδευτικός "
    then
        set global score to get global score + 1
        call TinyDB1 .StoreValue
            tag " score "
            valueToStore get global score
        open another screen screenName Result
    else if ListPicker1 . Selection = " Δ. εισοδήματος "
    then
        set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag " score "
            valueToStore get global score
        open another screen screenName Result
    else
        set global score to get global score + 0
        call TinyDB1 .StoreValue
            tag " score "
            valueToStore get global score
        open another screen screenName Result

when Hard_4 .Initialize
do
    set TextToSpeech1 . Pitch to " 1 "
    set TextToSpeech1 . SpeechRate to " 1 "
    call TextToSpeech1 .Speak
        message join Label1 . Text
```


when Result .Initialize
do
set Label2 . Text to get global score
set TextToSpeech1 . Pitch to " 1 "
set TextToSpeech1 . SpeechRate to " 1 "
if get global score = " 0 "
then call TextToSpeech1 .Speak
message join Label1 . Text
Label2 . Text
Label3 . Text
" Μπορείς και καλύτερα συνέχισε την προσπάθεια "
if get global score = " 1 "
then call TextToSpeech1 .Speak
message join Label1 . Text
Label2 . Text
Label3 . Text
" Μπορείς να τα καταφέρεις, συνέχισε την προσπάθει... "
if get global score = " 2 "
then call TextToSpeech1 .Speak
message join Label1 . Text
Label2 . Text
Label3 . Text
" Μπράβο, καλά, συνέχισε την προσπάθεια "
if get global score = " 3 "
then call TextToSpeech1 .Speak
message join Label1 . Text
Label2 . Text
Label3 . Text
" Μπράβο, πολύ καλά, συνέχισε την προσπάθεια "
if get global score = " 4 "
then call TextToSpeech1 .Speak
message join Label1 . Text
Label2 . Text
Label3 . Text
" Συγχαρητήρια, "
" , "
" Αριστα "
when Button1 .Click
do open another screen screenName Screen1
when Button2 .Click
do close application
initialize global score to call TinyDB1 .GetValue
tag " score "
valueIfTagNotThere 0