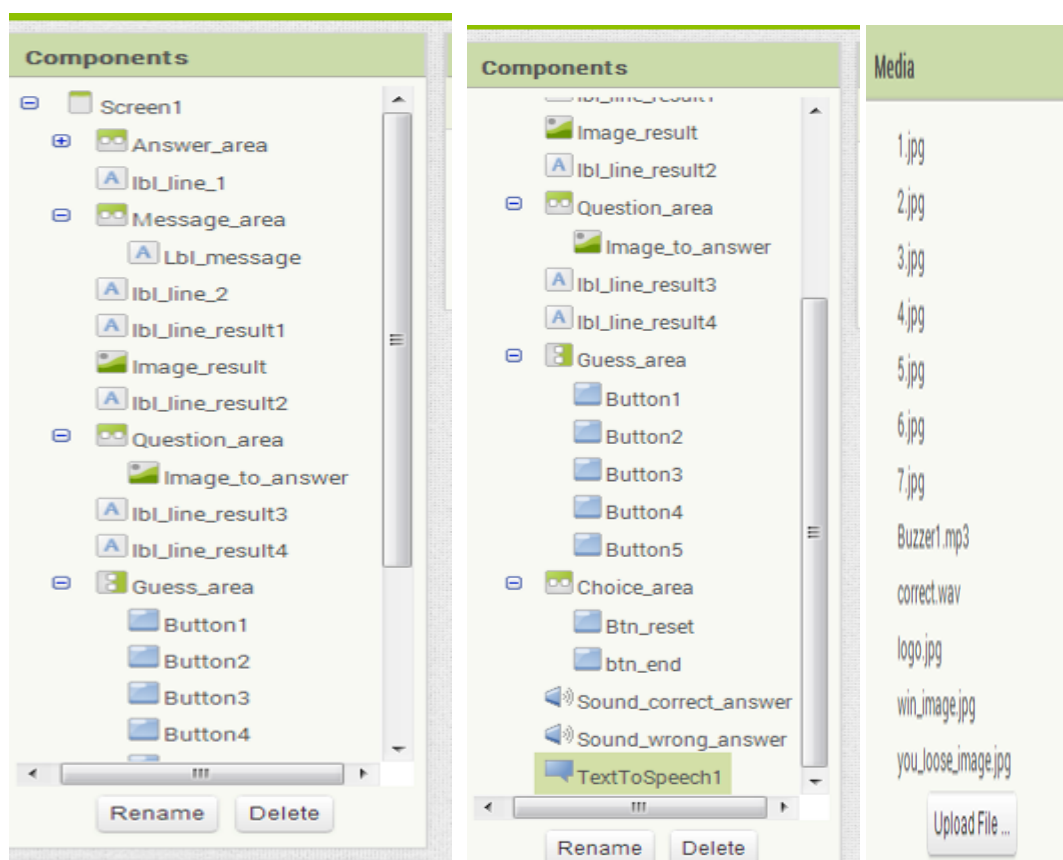


Οπτικό κομμάτι



Στο πίνακα που ακολουθεί φαίνονται τα αντικείμενα μαζί με τις επιθυμητές ιδιότητές τους.

Περιοχή σχεδίασης	Τύπος συστατικού	Ομάδα παλέτας	Όνομα συστατικού	Μεταβολή ιδιοτήτων
	Screen1	-	Screen1	Icon: logo.jpg Title: Βρες την σωστή απάντηση!
A	Horizontal Arrangement	Layout	Answer_area	default
A	Table Arrangement	Layout	Table Arrangement1	Columns: 4 Rows:1
A	Label	User interface	Label1	Text: Σωστές απαντήσεις:
A	Label	User interface	Right_answer	Text: 0 Textcolor: blue
A	Label	User interface	Label2	Text: Λάθος απαντήσεις:
A	Label	User interface	Wrong_answer	Text: 0 Textcolor: red
B	Label	User interface	Lbl_line_1	BackgroundColor: blue
B	Horizontal Arrangement	Layout	Message_area	default
B	Label	User interface	Lbl_message	Text: Προσπάθησε να βρεις την σωστή απάντηση. Μόνο μια απάντηση είναι σωστή.
B	Label	User interface	Lbl_line_2	BackgroundColor: blue
C	Label	User interface	lbl_line_result1	BackgroundColor: none
C	image	User interface	Image_result	Visible: Hidden
C	Label	User interface	lbl_line_result2	BackgroundColor: none
D	Horizontal Arrangement	Layout	Question_area	default
D	image	User interface	Image_to_answer	default
D	Label	User interface	lbl_line_result3	BackgroundColor: none
D	Label	User interface	lbl_line_result4	BackgroundColor: none
E	Vertical Arrangement	Layout	Guess_area	default
E	Button	Layout	Button1	Text: ?
E	Button	Layout	Button2	Text: ?
E	Button	Layout	Button3	Text: ?
E	Button	Layout	Button4	Text: ?
E	Button	Layout	Button5	Text: ?
F	Horizontal Arrangement	Layout	Choice_area	default
F	Button	Layout	Btn_reset	Text: Ξεκίνα ξανά!
F	Button	Layout	Btn_end	Text: Τέλος

Επίσης θα εισάγουμε και τρία άλλα συστατικά στοιχεία, τα οποία απλά θα τα αφήσουμε σε όποιο σημείο θέλουμε, διότι δε θα εμφανιστούν πουθενά στην οθόνη μας, αλλά είναι απαραίτητα για τη ορθή λειτουργία της εφαρμογής μας.

Τύπος συστατικού	Ομάδα παλέτας	Όνομα συστατικού	Μεταβολή ιδιοτήτων
Sound	Media	Sound_correct_answer	MinimumInterval: 500 Source: correct.wav
Sound	Media	Sound_wrong_answer	MinimumInterval: 500 Source: Buzzer1.mp3
TextToSpeech	Media	TextToSpeech1	default

Αν οποιαδήποτε στιγμή θέλουμε να βλέπουμε τα κρυφά αντικείμενα, αρκεί να τσεκάρουμε την επιλογή 'Display hidden components in viewer' στο πάνω μέρος της οθόνης.

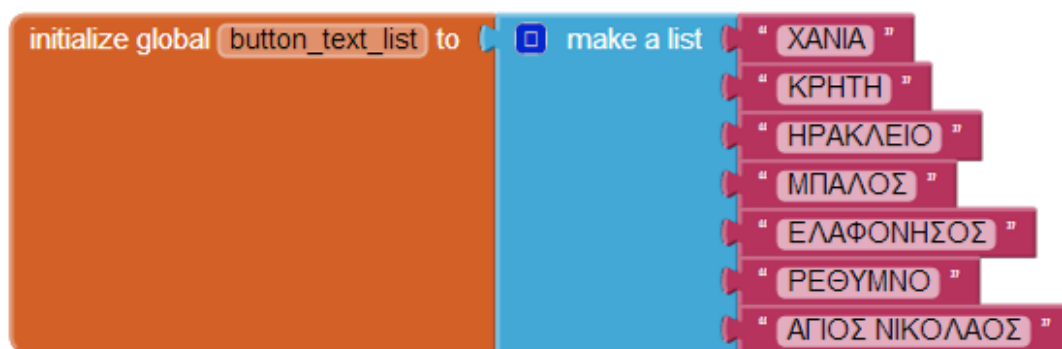
Τμήμα κώδικα

Αρχικά θα δημιουργήσουμε 5 νέες μεταβλητές με τα ονόματα temp1, temp2, temp3, temp4 και temp5, τις οποίες θα αρχικοποιήσουμε με την τιμή 0.



Οι μεταβλητές αυτές θα μας χρησιμεύσουν προκειμένου να αποθηκεύουμε τις προσωρινές τυχαίες τιμές που θα δημιουργεί αυτόματα το πρόγραμμα και τις οποίες θα ελέγχουμε αν μας ικανοποιούν ή όχι.

Με την εντολή που εμφανίζεται παρακάτω, δημιουργούμε μια νέα λίστα με το όνομα button_text_list, στην οποία θα έχουμε καταχωρήσει την σωστή απάντηση για την αντίστοιχη εικόνα. Για παράδειγμα αν η εικόνα 1 είναι ο Κρουσώνας, στην πρώτη θέση της λίστας θα είναι καταχωρημένη η λέξη Κρουσώνας. Αν η δεύτερη φωτογραφία είναι η εικόνα του Ηρακλείου στην δεύτερη θέση της λίστας θα είναι καταχωρημένη η λέξη Ηράκλειο κ.α. Με τον τρόπο αυτό μπορείτε να δημιουργήσετε ένα κουίζ με όσες διαφορετικές ερωτήσεις επιθυμείτε. Ο μόνος περιορισμός είναι η σωστή αντιστοίχιση εικόνας και καταχώρησης του περιεχομένου της εικόνας.



Προκειμένου να κάνουμε την εφαρμογή μας λειτουργική, θα δημιουργήσουμε την διαδικασία με όνομα proceed. Η διαδικασία αυτή είναι ίσως η πιο σημαντική διότι σε αυτή θα γίνεται η εμφάνιση της ερώτησης καθώς και των πιθανών απαντήσεων των χρηστών (1 σωστή και 4 λανθασμένες).

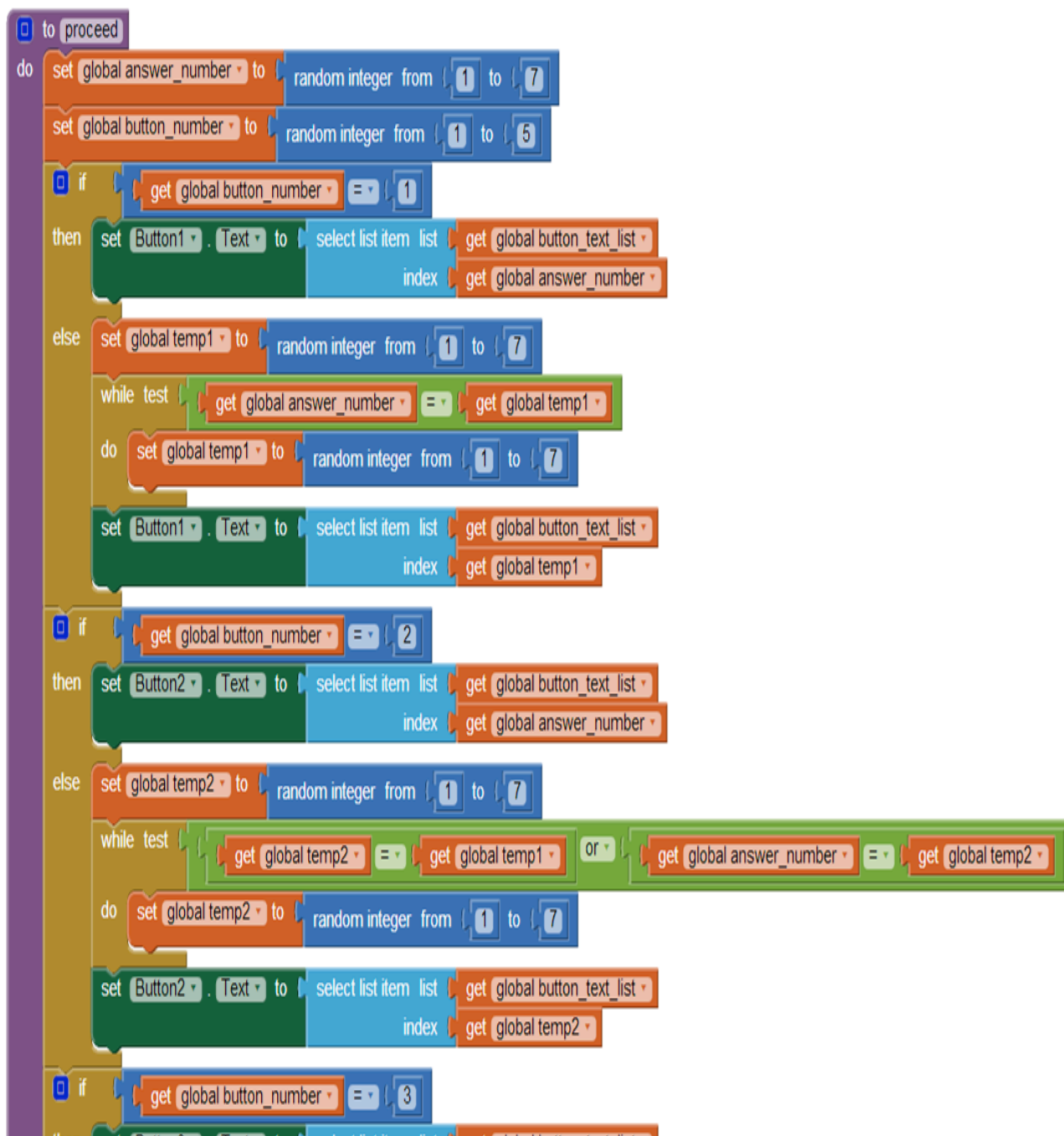
Αρχικά η μεταβλητή `answer_number` λαμβάνει μια τυχαία τιμή μεταξύ 1 έως 7 για να εμφανίσει μια εικόνα που αντιστοιχεί σε μια ερώτηση. Αν θέλετε η εφαρμογή να έχει 50 ερωτήσεις αρκεί να έχετε φορτώσει την εφαρμογή με 50 εικόνες και τις αντίστοιχες ερωτήσεις και απλά να αλλάξετε το εύρος των αριθμών από το 1..50. Στην μεταβλητή `button_number` θέτουμε μια τυχαία τιμή μεταξύ 1 έως 5 καθώς έχουμε 5 κουμπιά από τα οποία ο χρήστης καλείται να επιλέξει τη σωστή απάντηση. Στη συνέχεια ακολουθεί μια συνθήκη `if..else`. Τι ελέγχουμε σε αυτή την συνθήκη; Εάν η μεταβλητή `button_number` έχει την τιμή 1 σημαίνει ότι η σωστή απάντηση είναι στο κουμπί 1. Επομένως θα πρέπει το κουμπί ως κείμενο να εμφανίζει την σωστή απάντηση την οποία την επιλέγει από την λίστα με όνομα `button_text_list`. (Την λίστα αυτή θα την γεμίσουμε εμείς με τις τιμές που επιθυμούμε). Σε περίπτωση που η μεταβλητή `button_number` δεν έχει την τιμή 1 αυτό σημαίνει ότι η σωστή απάντηση δεν είναι στο κουμπί 1. Επομένως πρέπει να πάρει μια τιμή διαφορετική από την σωστή απάντηση προκειμένου να μπερδεύει στον χρήστη. Ωστόσο θα πρέπει να ελέγχουμε μη τυχόν και η απάντηση που θα εμφανίσει το κουμπί είναι η ίδια με την σωστή απάντηση. Ως εκ τούτου λέμε στο πρόγραμμα μας να ελέγχει και όσο η τυχαία τιμή που παίρνει το κουμπί1 ισούται με την τιμή της σωστής απάντησης να μην την δέχεται το πρόγραμμα, αλλά να προσπαθεί να του δίνει μια νέα τιμή στο διάστημα που έχουμε ορίσει εμείς [1..7].

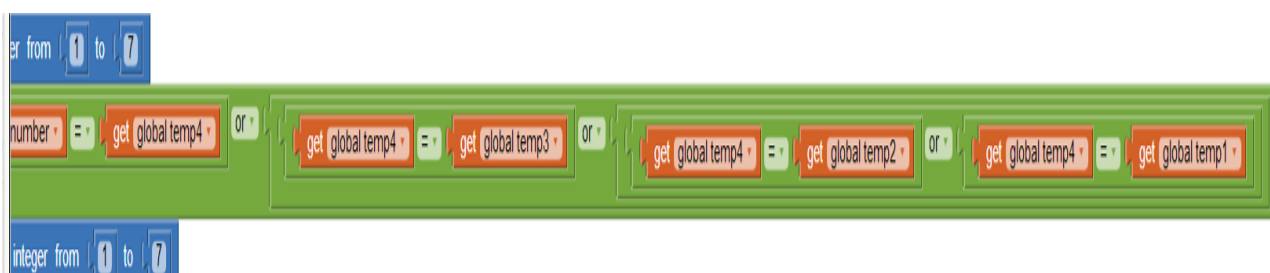
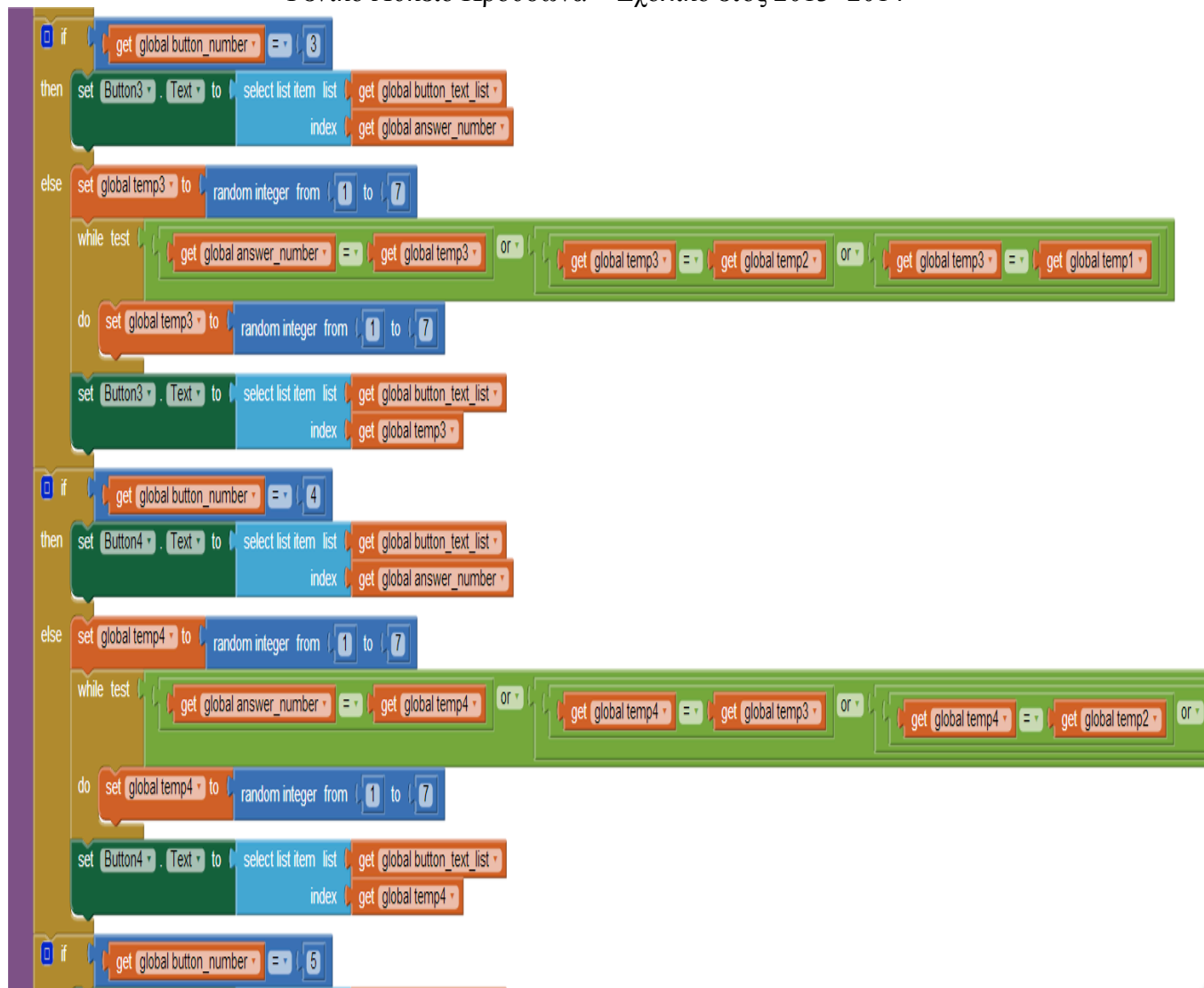
Με παρόμοιο τρόπο γίνεται ο έλεγχος και στις τιμές που παίρνουν και τα υπόλοιπα κουμπιά. Ωστόσο κάθε φορά η συνθήκη ελέγχου θα γίνεται όλο και πιο πολύπλοκη (αλλά όχι δύσκολη στην κατανόηση της) διότι δεν αρκεί να ελέγχουμε αν η τυχαία τιμή που παίρνει το κουμπί είναι η σωστή απάντηση (το οποίο δεν το θέλουμε διότι την σωστή απάντηση θα την έχεις πάρει κάπιο άλλο κουμπί, οπότε θα έχουμε 2 σωστές απαντήσεις στην εφαρμογή, το οποίο δεν είναι σωστό) αλλά δεν θα θέλουμε να εμφανίζονται και 2 όμοιες ή περισσότερες λανθασμένες απαντήσεις. Δηλαδή φανταστείτε να είχαμε μια εικόνα που εμφανίσει τον Κρουσώνα και η λίστα των πιθανών απαντήσεων στα κουμπιά να ήταν της μορφής:

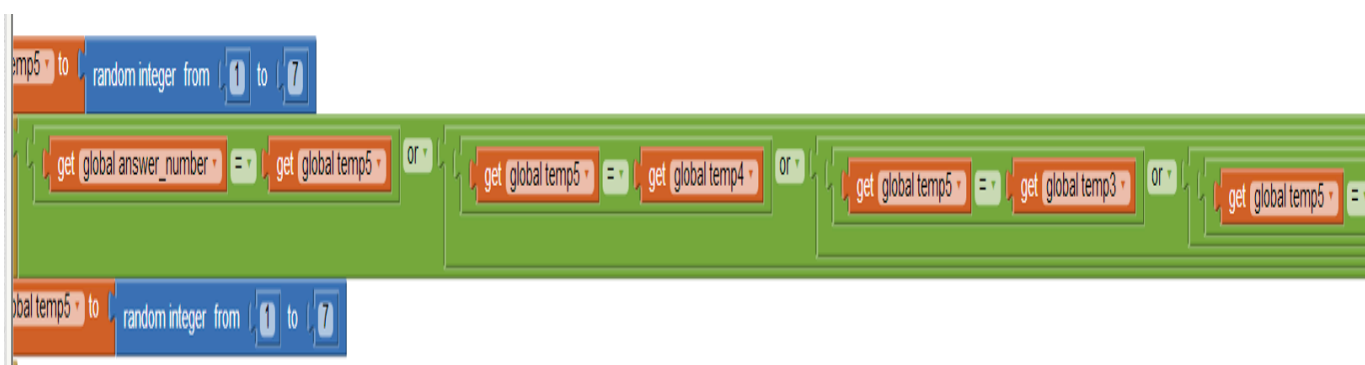
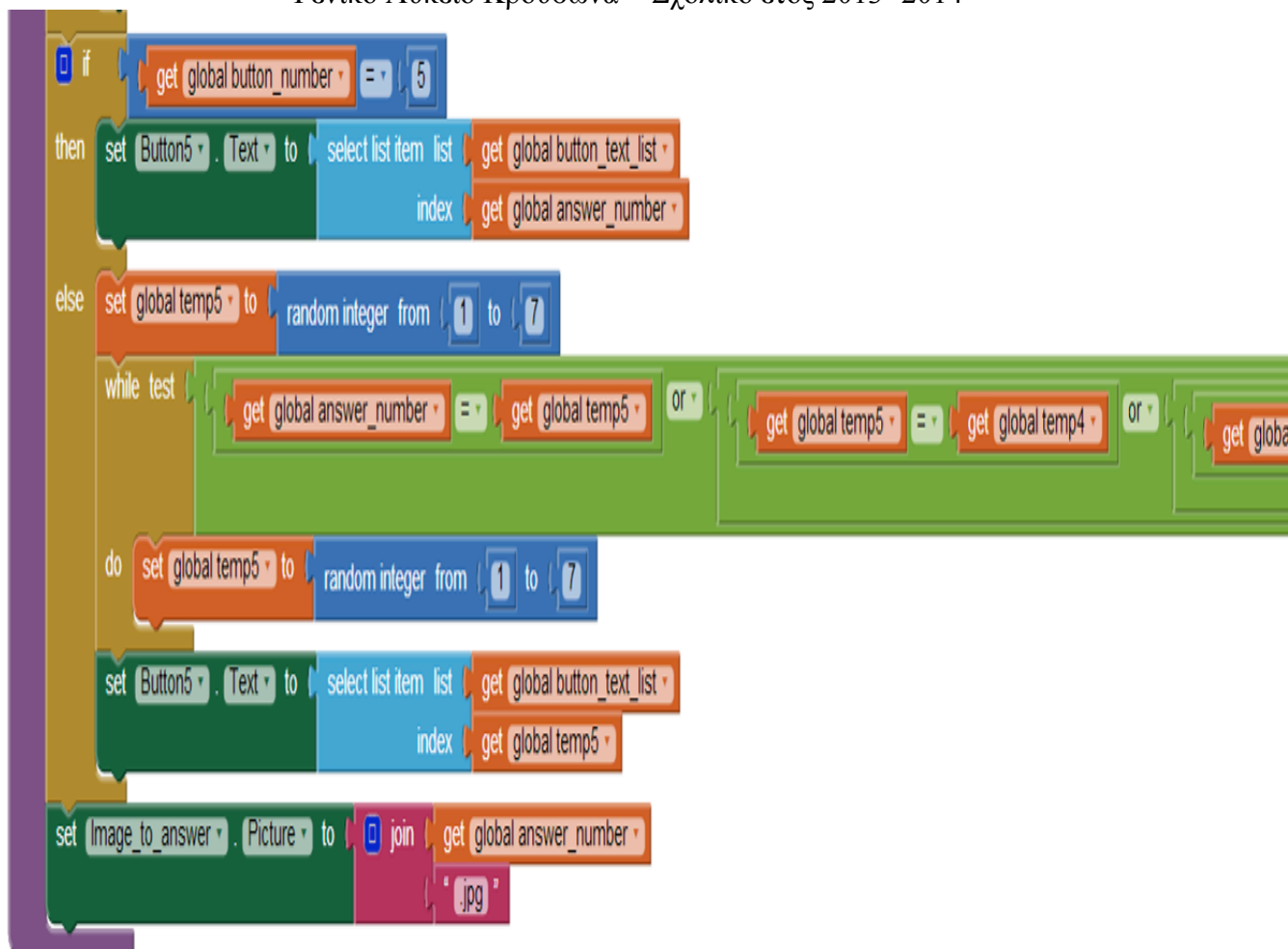
Κουμπί 1	Κρουσώνας
Κουμπί 2	Σητεία
Κουμπί 3	Ρέθυμνο
Κουμπί 4	Σητεία
Κουμπί 5	Ρέθυμνο

Επομένως πρέπει να ελέγχουμε η τυχαία λάθος τιμή που παίρνει ένα κουμπί, να μην είναι όμοια με την τυχαία λάθος τιμή που έχει πάρει ήδη ένα άλλο κουμπί.

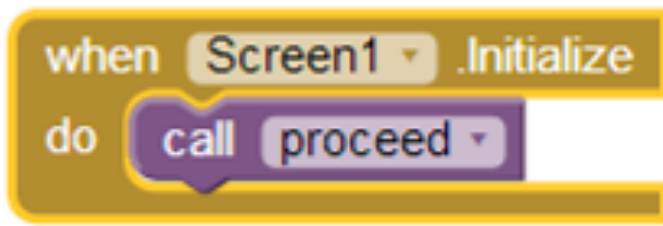
Για το 2ο κουμπί ελέγχουμε την τιμή του κουμπιου2 να μην είναι όμοια με την τιμή του κουμπιού 1 αλλά και με την σωστή απάντηση. Αν ένα από τα 2 παραπάνω ισχύει, απλά δεν δεχόμαστε την τιμή και ζητάμε μια νέα τιμή, μέχρι να μην ισχύει και οι 2 συνθήκες που περιγράψαμε παραπάνω.



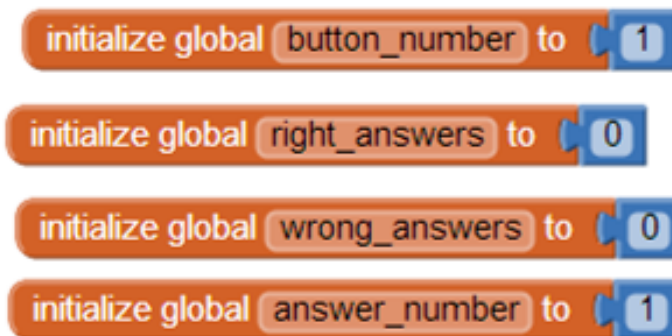




Με την εντολή αυτή , όταν ξεκινάει η εφαρμογή, απλά καλούμε την διαδικασία με ονομασία proceed, προκειμένου να ξεκινήσει η δημιουργία της σωστής και των λανθασμένων απαντήσεων.



Με τις εντολές που ακολουθούν, δημιουργούμε και αρχικοποιούμε 4 διαφορετικές μεταβλητές, οι οποίες μας χρειάζονται για την ορθή λειτουργία της εφαρμογής όπως θα εξηγήσουμε τα επόμενα βήματα.

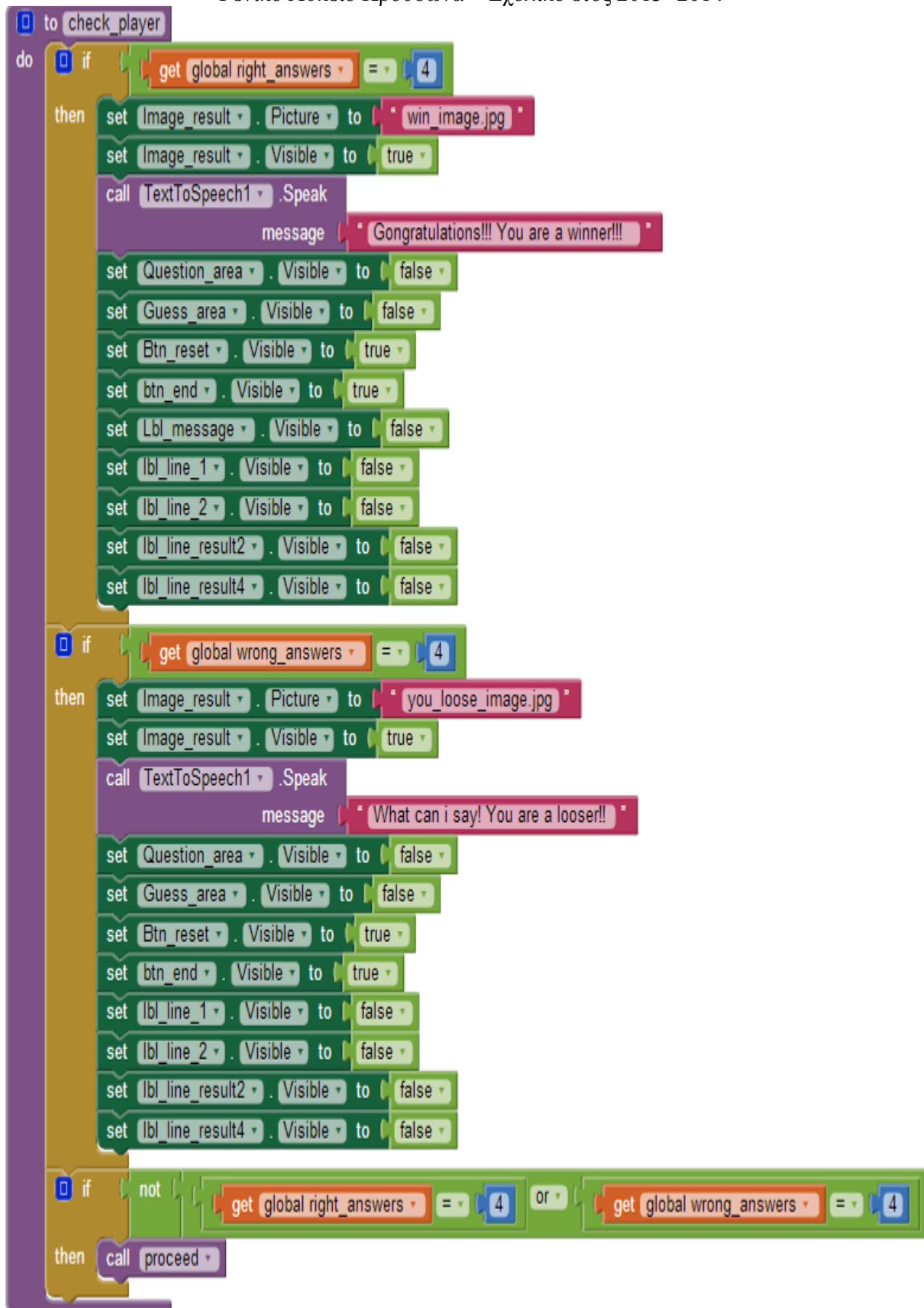


Η διαδικασία με όνομα check_player, όπως λέει και το όνομα της ελέγχει αν ο παίκτης έχει απαντήσει σωστά ή λάθος στις ερωτήσεις του παιχνιδιού. Σε αυτό το παράδειγμα (για να τελειώσει γρήγορα το παιχνίδι, έχουμε ορίσει το όριο στις 4 σωστές ή λανθασμένες απαντήσεις, αλλά απλά αλλάζοντας το νούμερο να θέτατε το δικό σας όριο π.χ. αν το κάνατε 10, θα σήμαινε ότι το παιχνίδι πρέπει να τελειώσει στις 10 σωστές ή λανθασμένες απαντήσεις).

Αρχικά ελέγχουμε αν οι σωστές απαντήσεις (right_answers) έχουν φτάσει το νούμερο 4 (δηλαδή ο χρήστης κέρδισε). Αν η παραπάνω υπόθεση μας ισχύει τότε η εικόνα image_result θα εμφανίζει την εικόνα με όνομα win_image, ενώ θα ακουστεί και το μήνυμα 'Συγχαρητήρια! Είσαι νικητής'. Στην συνέχεια εφόσον το παιχνίδι τελείωσε δεν έχει νόημα να φαίνονται ούτε η εικόνα του κουίζ ούτε οι πιθανές απαντήσεις (δηλαδή οι περιοχές Question_area και Guess_area) οπότε τις κρύβουμε κάνοντας τις μη ορατές (visible=false). Με παρόμοιο τρόπο κάνουμε και το μήνυμα 'Μόνο μια απάντηση είναι σωστή' αόρατο (lbl_message=false). Με παρόμοιο τρόπο κάνουμε και τα υπόλοιπα οπτικά κομμάτια της εφαρμογής μας αόρατα.

Αν η παραπάνω υπόθεση, δεν ισχύει πρέπει να ελέγξουμε ότι οι λανθασμένες απαντήσεις (wrong_answers) έχουν φτάσει το νούμερο 4 (δηλαδή ο χρήστης έχασε). Αν η παραπάνω υπόθεση μας ισχύει τότε η εικόνα image_result θα εμφανίζει την εικόνα με όνομα you_loose_image, ενώ θα ακουστεί και το μήνυμα 'Τι να πω! Έχασες'. Στην συνέχεια εφόσον το παιχνίδι τελείωσε δεν έχει νόημα να φαίνονται ούτε η εικόνα του κουίζ ούτε οι πιθανές απαντήσεις (δηλαδή οι περιοχές Question_area και Guess_area) οπότε τις κρύβουμε κάνοντας τις μη ορατές (visible=false). Με παρόμοιο τρόπο κάνουμε και το μήνυμα 'Μόνο μια απάντηση είναι σωστή' αόρατο (lbl_message=false). Με παρόμοιο τρόπο κάνουμε και τα υπόλοιπα οπτικά κομμάτια της εφαρμογής μας αόρατα.

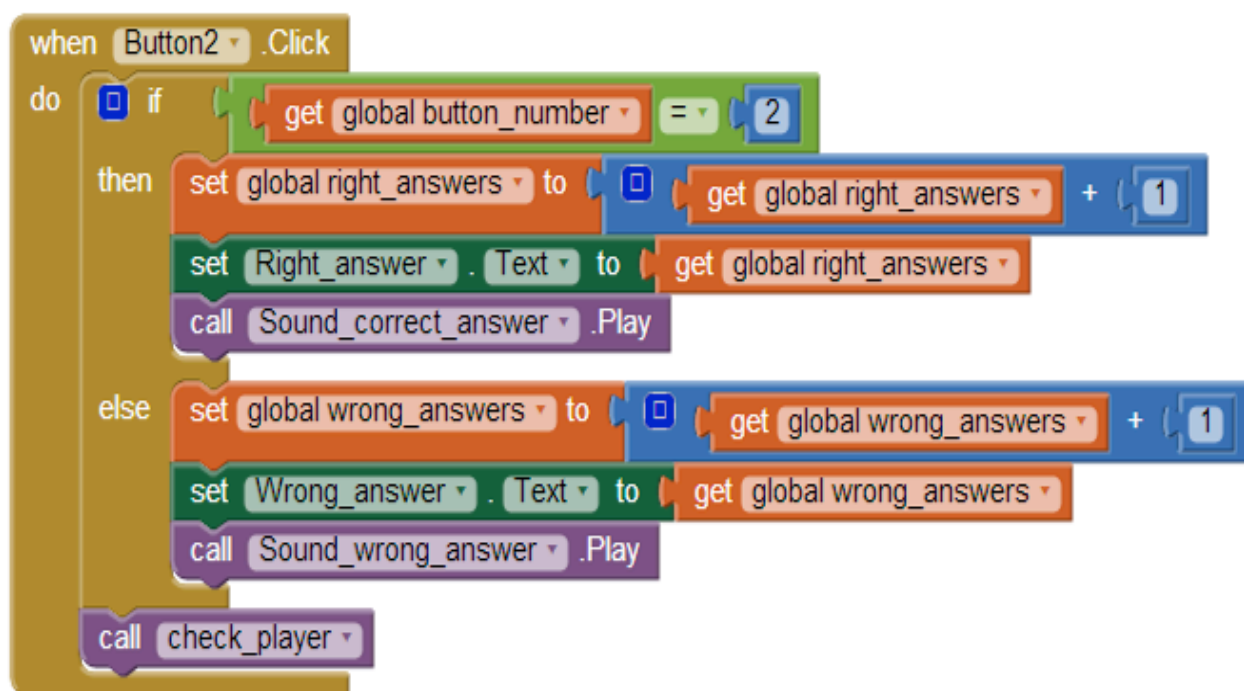
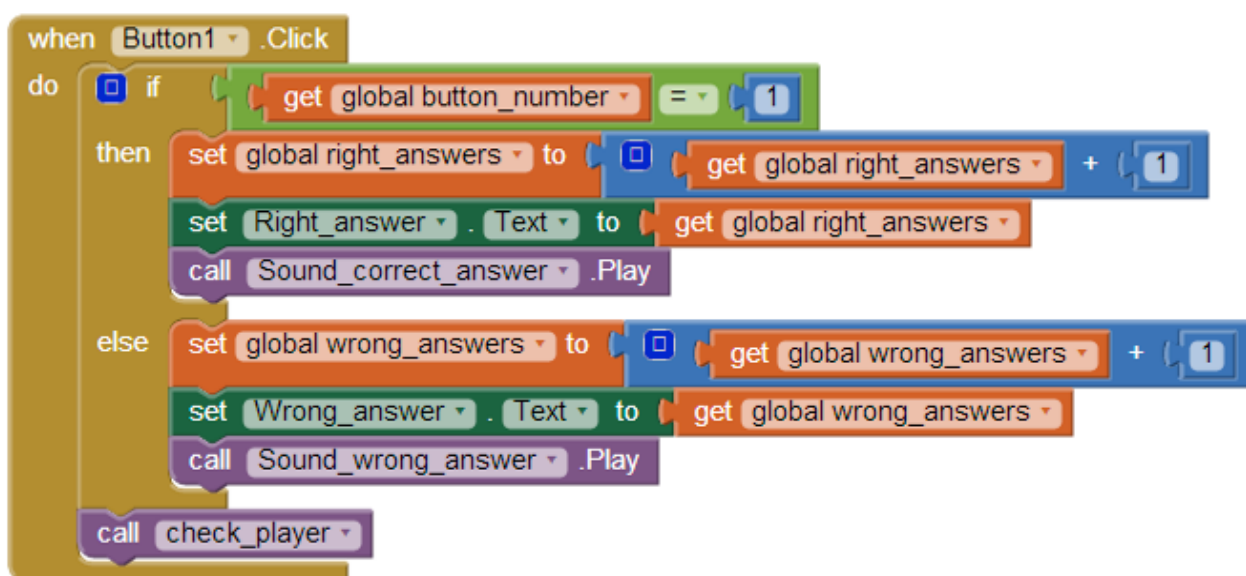
Τέλος, υπάρχει η περίπτωση το παιχνίδι να μην έχει ολοκληρωθεί (δηλαδή να μην έχει φτάσει ο χρήστης τις 4 σωστές ή λανθασμένες απαντήσεις) οπότε τότε απλά καλούμε την διαδικασία proceed προκειμένου να συνεχιστεί το παιχνίδι.

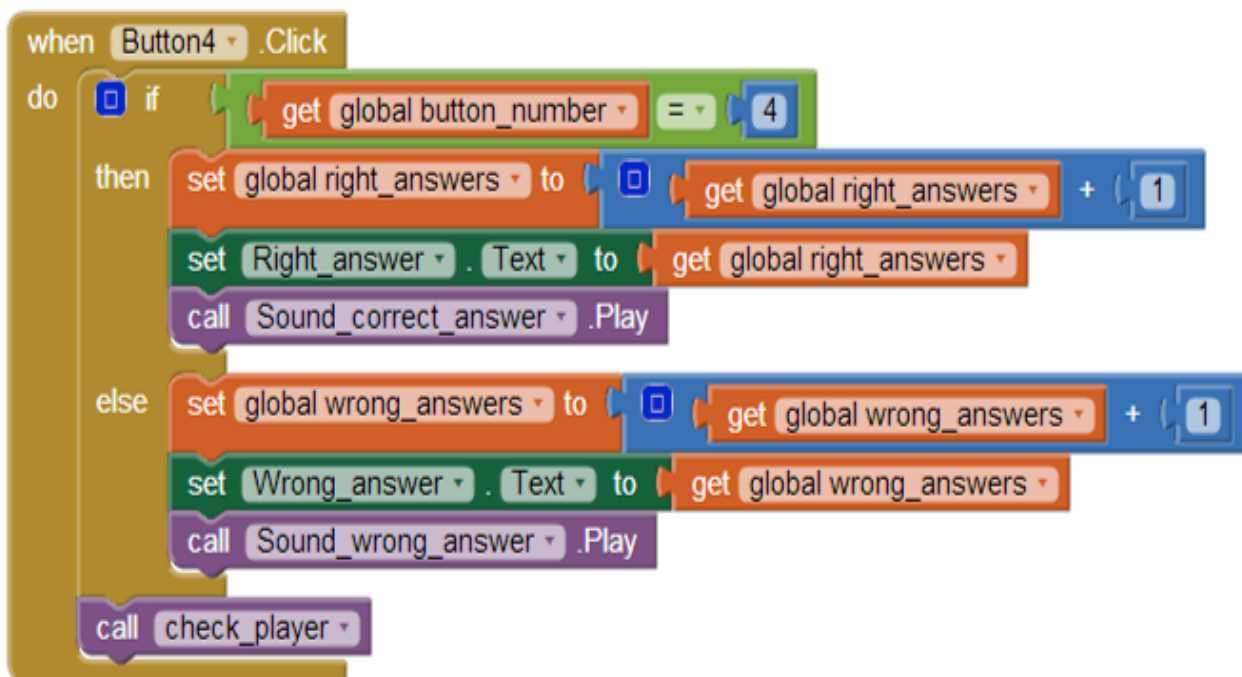
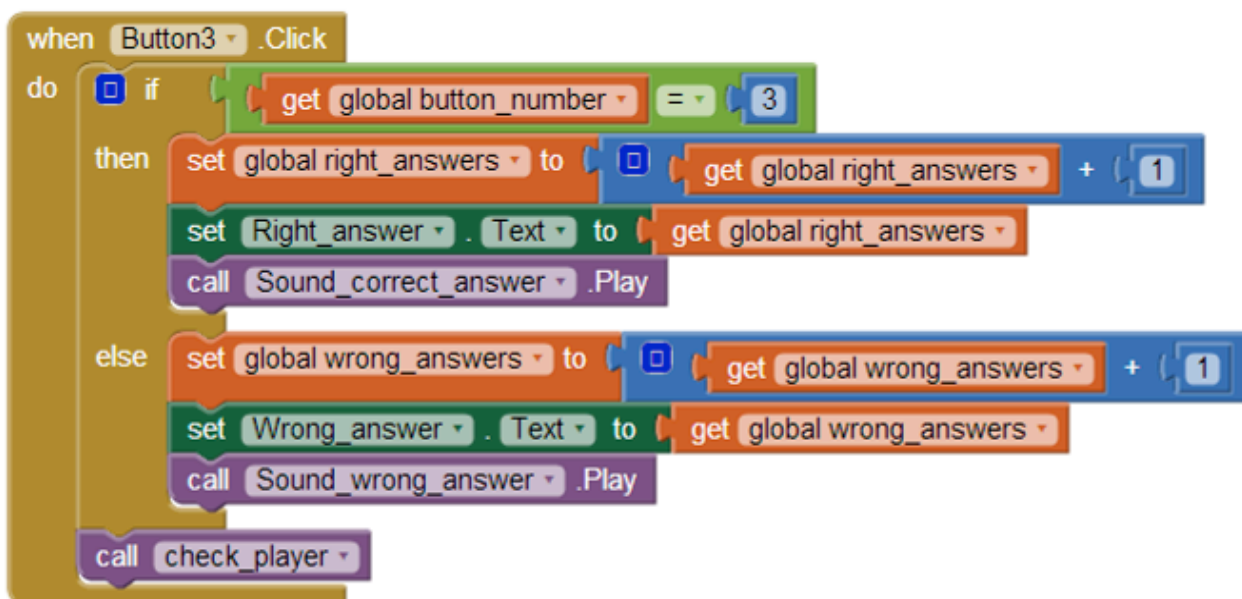


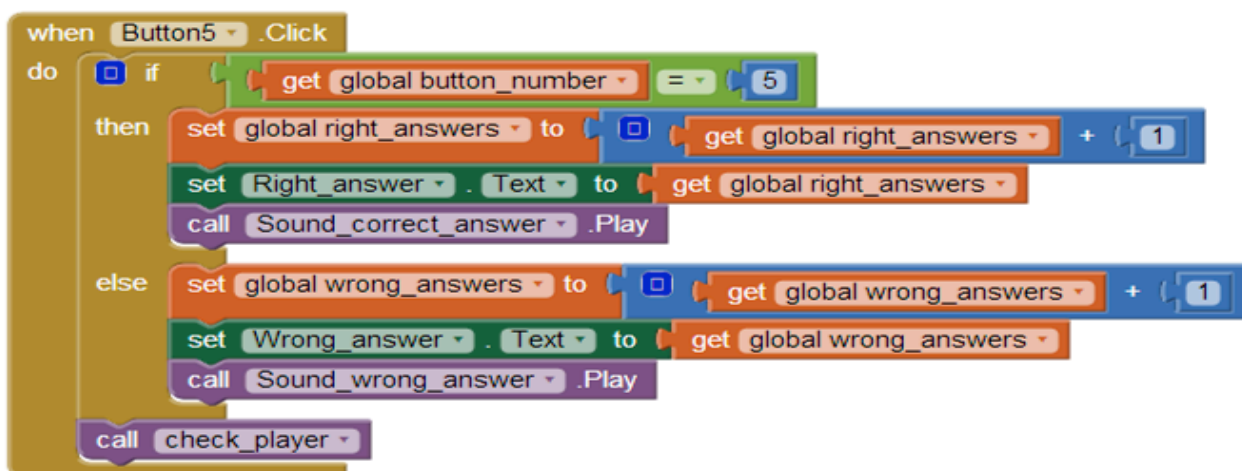
Στα παρακάτω συμβάντα, τα οποία εκτελούνται αυτόματα, όταν ο χρήστης πατήσει στο αντίστοιχο κουμπί ουσιαστικά ελέγχουμε αν ο χρήστης έχει απαντήσει ορθά ή λανθασμένα.

Πιο αναλυτικά, όταν ο χρήστης πατήσει το κουμπί1, σημαίνει ότι θεωρεί ότι η ορθή απάντηση είναι στο κουμπί 1. Αν όντως η ορθή απάντηση είναι το κουμπί 1 τότε αυξάνεται η μεταβλητή που μετρά τις ορθές απαντήσεις κατά 1 (διότι δεν πρέπει να ξεχνάμε ότι κάποια στιγμή πρέπει το παιχνίδι να τελειώσει). Επίσης ο χρήστης ενημερώνεται για τον αριθμό των ορθών του απαντήσεων, μέσω της μεταβλητής `right_answer`, ενώ ακούγεται και ένας ευχάριστος ήχος, επιβεβαιωτικός της ορθής απάντησης του. (`sound_correct_answer.play`) Σε περίπτωση που ο χρήστης απαντήσει λάθος, τότε αντίστοιχα αυξάνεται η μεταβλητή που μετρά τις λανθασμένες απαντήσεις κατά 1 (διότι δεν πρέπει να ξεχνάμε ότι κάποια στιγμή πρέπει το παιχνίδι να τελειώσει). Επίσης ο χρήστης ενημερώνεται για τον αριθμό των λανθασμένων του απαντήσεων, μέσω της μεταβλητής `wrong_answer`, ενώ ακούγεται και ένας όχι τόσο ευχάριστος ήχος, επιβεβαιωτικός της λανθασμένης απάντησης του (`sound_wrong_answer.play`).

Στην συνέχεια καλείται η διαδικασία `check_player`, η οποία αν θυμόμαστε ελέγχει αν ο χρήστης έχει προσεγγίσει τον προκαθορισμένο αριθμό σωστών ή λανθασμένων απαντήσεων προκειμένου να τελειώσει το παιχνίδι.

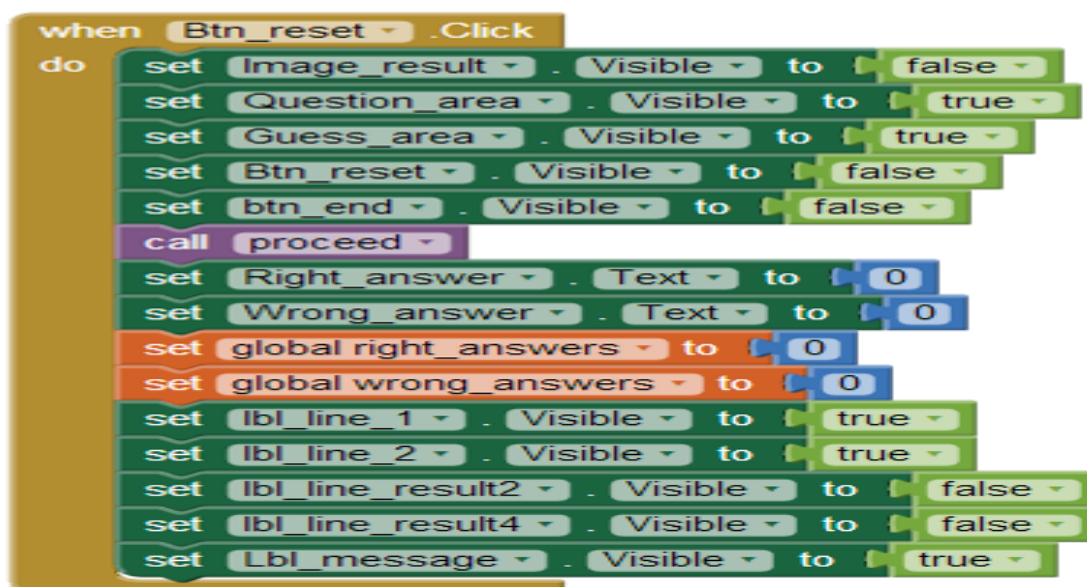




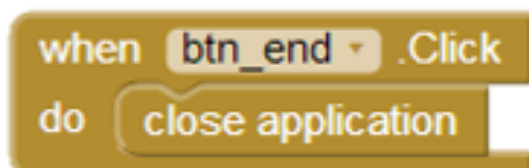


Στην συνέχεια θα δημιουργήσουμε την διαδικασία btn_reset, η οποία εκτελείται αυτόματα, όταν ο χρήστης πατήσει το κουμπί reset. Ο χρήστης θα πατήσει το κουμπί reset, οπότε θελήσει να παίξει εκ νέου το παιχνίδι. Για το λόγο αυτό πρέπει να γίνουν ορισμένες ενέργειες όπως:

- κρύβει την εικόνα επιτυχίας ή αποτυχίας
- επαναφέρει το οπτικό κομμάτι της εφαρμογής στην αρχική του μορφή προκειμένου να εμφανίζει την εικόνα ερώτησης και τα 5 κουμπιά των απαντήσεων
- κρύβει τόσο το κουμπί επανεκκίνησης της εφαρμογής (reset) όσο και αυτό του τερματισμού της
- καλεί τη διαδικασία proceed προκειμένου να ξεκινήσει η εφαρμογή
- αρχικοποιεί τους μετρητές σωστών και λανθασμένων απαντήσεων
- αρχικοποιεί κατάλληλα τα υπόλοιπα οπτικά τμήματα της εφαρμογής



Η παρακάτω διαδικασία, η οποία ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί τερματισμού (end) καλεί την εντολή συστήματος close application, η οποία όπως λέει το όνομα της τερματίζει την εφαρμογή.



Διάφορα screenshots από την εφαρμογή

Βρες την σωστή απάντηση!

Σωστές απαντήσεις: 0 Λάθος απαντήσεις: 0

Προσπάθησε να βρεις την σωστή απάντηση.
Μόνο μια απάντηση είναι σωστή.

ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ

ΕΛΑΦΟΝΗΣΟΣ

ΡΕΘΥΜΝΟ

ΚΡΗΤΗ

ΜΠΑΛΟΣ

Βρες την σωστή απάντηση!

Σωστές απαντήσεις: 0 Λάθος απαντήσεις: 1

Προσπάθησε να βρεις την σωστή απάντηση.
Μόνο μια απάντηση είναι σωστή.

ΕΛΑΦΟΝΗΣΟΣ

ΜΠΑΛΟΣ

ΡΕΘΥΜΝΟ

ΗΡΑΚΛΕΙΟ

ΧΑΝΙΑ

Βρες την σωστή απάντηση!

Σωστές απαντήσεις: 0 Λάθος απαντήσεις: 3

Προσπάθησε να βρεις την σωστή απάντηση.
Μόνο μια απάντηση είναι σωστή.

ΜΠΑΛΟΣ

ΡΕΘΥΜΝΟ

ΕΛΑΦΟΝΗΣΟΣ

ΧΑΝΙΑ

ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ

Saving screenshot.

Βρες την σωστή απάντηση!

Σωστές απαντήσεις: 0 Λάθος απαντήσεις: 4

Προσπάθησε να βρεις την σωστή απάντηση.
Μόνο μια απάντηση είναι σωστή.

Ξεκίνα ξανά!

Τέλος

Βρες την σωστή απάντηση!

Σωστές απαντήσεις: 4 Λάθος απαντήσεις: 3

Ξεκίνα ξανά!

Τέλος