

Κτύπα την κατσαρίδα (Mole Mash +)

Φύλλο Εργασίας

Προγραμματισμός Εφαρμογής στο AppInventor

<http://ai2.appinventor.mit.edu>

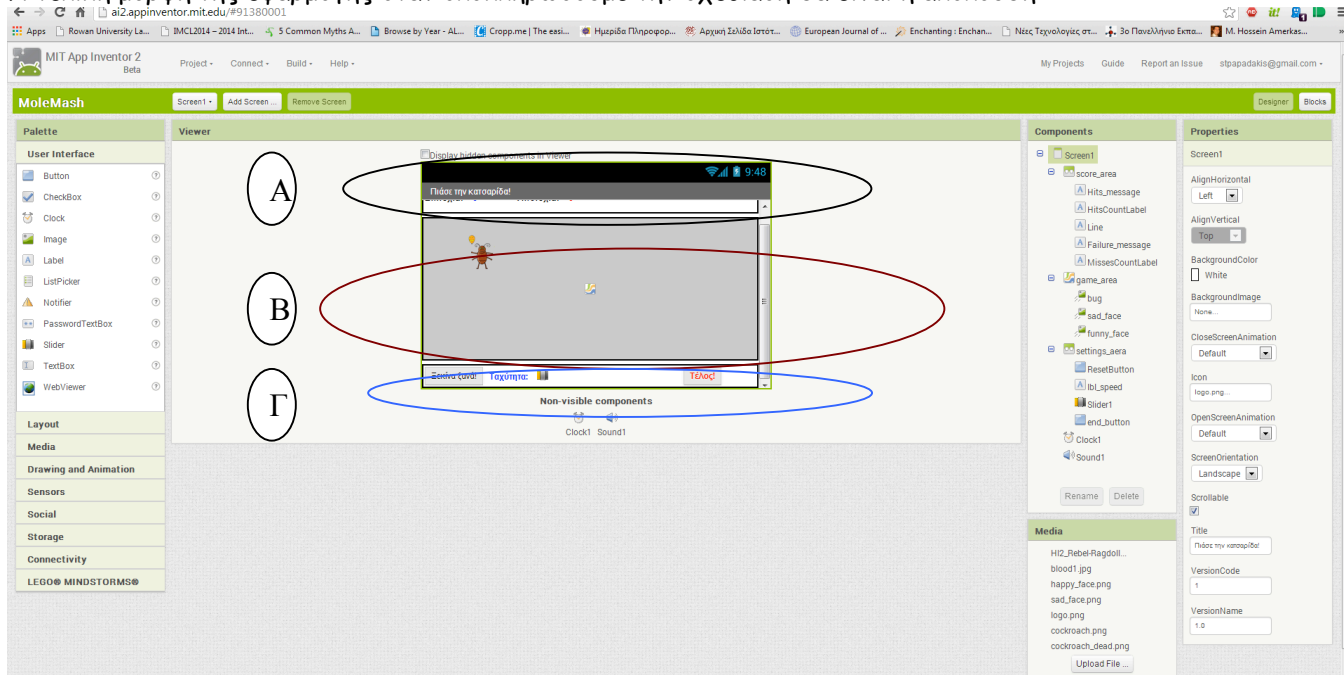
Ακολουθώντας τα βήματα αυτού του φύλλου εργασίας, θα αναπτύξουμε μια μικρή εφαρμογή για Android συσκευές (κινητά ή tablets), η οποία θα μας εξοικειώσει με χρήσιμες λειτουργίες που θα αξιοποιήσουμε και στη συνέχεια. Θα αναπτύξουμε σταδιακά μια εφαρμογή στην οποία θα προσπαθήσουμε να 'πετύχουμε' ένα αντικείμενο που κινείται με τυχαίο τρόπο στην οθόνη της έξυπνης φορητής συσκευή μας. Ανάλογα τον αριθμό των επιτυχημένων ή αποτυχημένων προσπαθειών θα κερδίζουμε ή θα χάνουμε (ανάλογα πότε θα φτάσουμε στον αριθμό 10). Ανά πάσα στιγμή θα έχουμε την δυνατότητα να αυξομειώνουμε την ταχύτητα με την οποία κινείται το αντικείμενο στην οθόνη, να επανεκκινούμε ή να τερματίζουμε την εφαρμογή μας.

Designer

Βήμα 1

Σχεδίαση της εφαρμογής

Η τελική μορφή της εφαρμογής όταν ολοκληρώσουμε την σχεδίαση θα είναι η ακόλουθη:

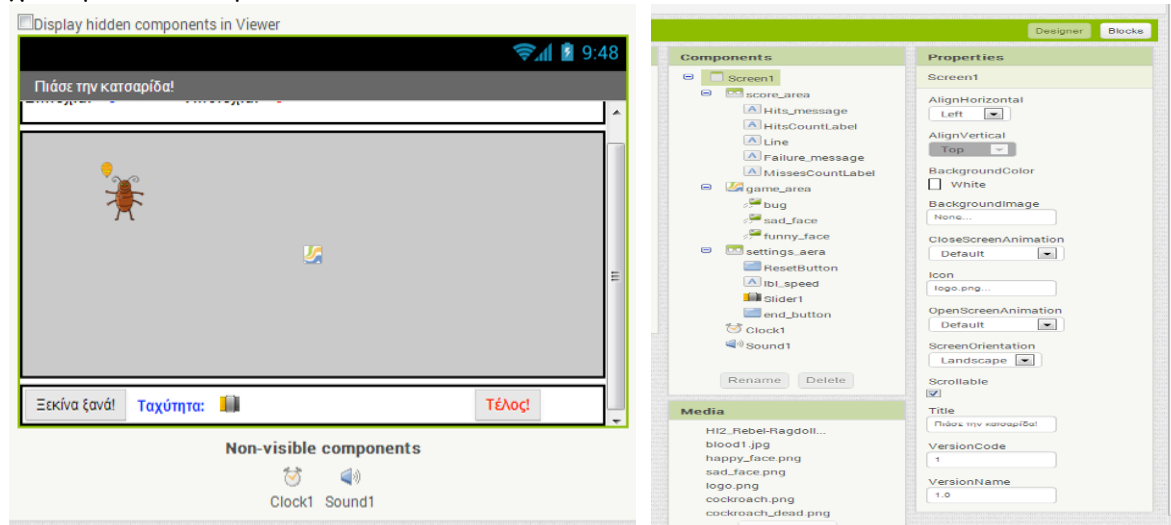


Όπως φαίνεται στο σχήμα που ακολουθεί, η εφαρμογή μας θα αποτελείται από 3 διακριτά μέρη. Το Α μέρος ή την πάνω περιοχή (score_area) (στην οποία θα κρατάμε το σκορ, το Β μέρος ή την μεσαία περιοχή (game_area) η οποία αποτελείται από τον καμβά και το αντικείμενο το οποίο θα προσπαθήσουμε να κτυπήσουμε και το Γ μέρος ή την κάτω περιοχή (settings_area) στην οποία θα υπάρχουν τα κουμπιά για την επανεκκίνηση και τον τερματισμό της εφαρμογής καθώς και αυξομείωσης της ταχύτητας του αντικειμένου.

Για τη δημιουργία του οπτικού κομματιού της εφαρμογής θα χρειαστεί να εισάγουμε μια εικόνα που παριστάνει το αντικείμενο που κινείται, μια εικόνα που αντικαθιστά το αντικείμενο καθώς το χτυπάμε, ένα αρχείο ήχο που θα είναι ο ήχος που θα ακούγεται καθώς χτυπάμε το αντικείμενο, μια εικόνα με ένα χαρούμενο πρόσωπο που θα εμφανίζεται όταν ο χρήστης θα κερδίζει το παιχνίδι, μια εικόνα με ένα λυπημένο πρόσωπο που θα εμφανίζεται

όταν ο χρήστης θα χάσει το παιχνίδι, και τέλος μια εικόνα ως εικονίδιο της εφαρμογής μας όταν θα την εγκαθιστούμε στην έξυπνη φορητή μας συσκευή.

Τέλος θα εισάγουμε και δυο αντικείμενα (τα οποία δεν θα εμφανίζονται στο οπτικό κομμάτι της εφαρμογής), ένα αντικείμενο τύπου ρολόι ως μετρητή το οποίο θα μας καθορίζει πόσο γρήγορα ή αργά θα κινείται το αντικείμενο στην οθόνη, και ένα αντικείμενο τύπου ήχου το οποίο θα αναπαραγάγει τον ήχο που θέλουμε κάθε φορά που θα χτυπάμε το αντικείμενο.



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

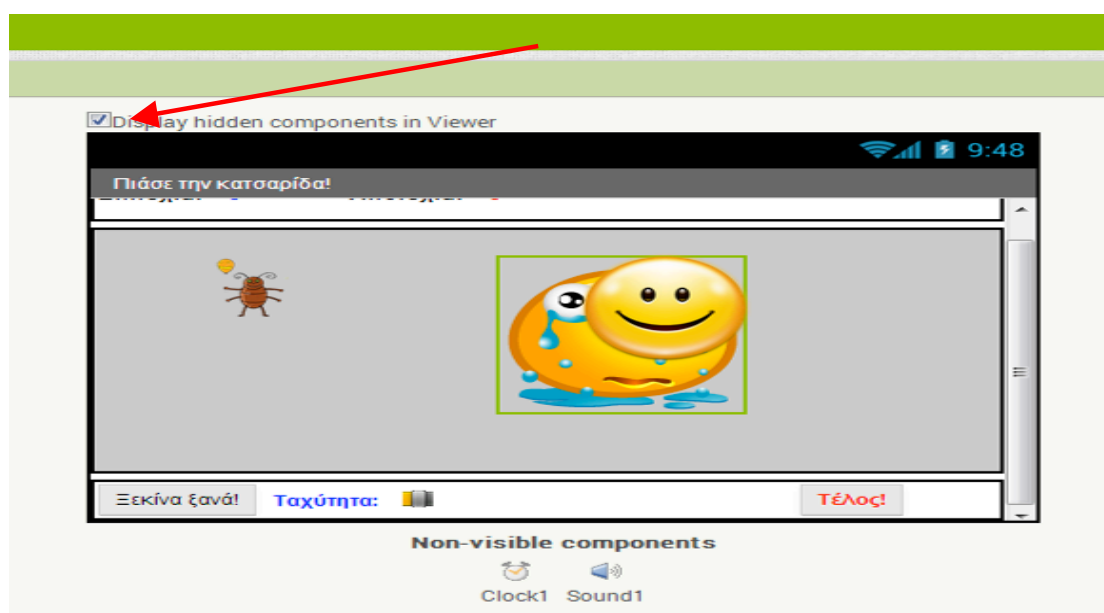
Τύπος συστατικού	Ομάδα παλέτας	Όνομα συστατικού	Μεταβολή ιδιοτήτων
Screen1	-	Screen1	Icon: bug.png Title: Πιάσε την κατσαρίδα!
Horizontal arrangement	Layout	Score_area	Width: Fill parent Height: 30 pixels
Label	User interface	Hits_message	Text: επιτυχία FontBold:Nai FontItalic:Nai
Label	User interface	Hits_count_label	Text: 0 FontBold:Nai Textcolor: blue
Label	User interface	line	Text: *_*_*
Label	User interface	Failure_message	Text: Αποτυχία FontBold:Nai FontItalic:Nai
Label	User interface	Missescountlabel	Text: 0 FontBold:Nai Textcolor: Red
Canvas	Drawing animation &	Game_area	Background colour: Light Grey Width: Fill parent Height: 200 pixels
Imagesrpitem	Drawing animation &	bug	Picture: cockroach.png Width: 40 pixels Height: 50 pixels
Imagesrpitem	Drawing animation &	Sad_face	Picture: sad_face.png Visible: No
Imagesrpitem	Drawing &	Funny_face	Picture: funny_face.png

	animation		Visible: No
Horizontal arrangement	Layout	Settings_area	Width: Fill parent Height: 30 pixels
Button	User interface	Resetbutton	Text: Ξεκίνα ξανά!
Label	User interface	Lbl_speed	Text: Ταχύτητα FontBold:Nai Textcolor: blue
Slider	User interface	Slider1	Colorleft: Orange Colorright: Grey Maxvalue: 5000 Minvalue: 500 Thumbposition: 100 Width: 200 pixels
Button	User interface	End_button	Text: Τέλος! FontBold:Nai Textcolor: Red

Επίσης θα εισάγουμε και δυο 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
Clock	User interface	Clock1	TimerAlwaysFires: Nai TimerEnabled: Nai TimerInterval: 1000
Sound	Media	Sound1	MinimumInterval: 500 Source: H12_Rebel-Ragdoll485-573931361.mp3

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



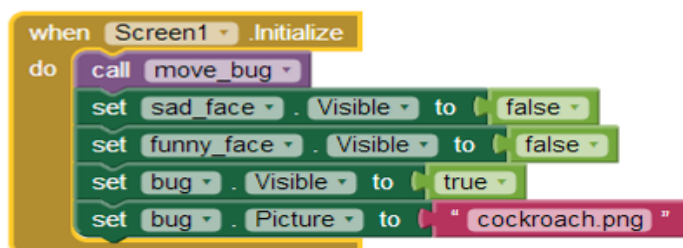
Blocks

Στη συνέχεια, θα δούμε τον κώδικα που πρέπει να ενσωματώσουμε στην εφαρμογή μας, προκειμένου να είναι λειτουργική και να αποκρίνεται στα συμβάντα του χρήστη.

1. Διαδικασία μετακίνησης του αντικειμένου (teacher) με τυχαίο τρόπο στη οθόνη.



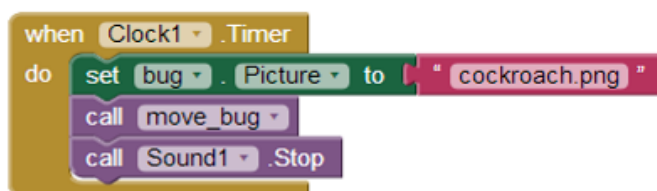
2. Αρχικοποίηση της εφαρμογής



3. Χρήση του μετρητή.

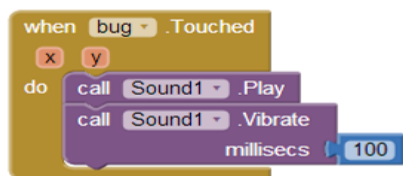
Κάθε φορά που ενεργοποιείται ο μετρητής (ρολόι) (το οποίο θυμίζουμε ότι το έχουμε ρυθμίσει εμείς κατά τη φάση σχεδίασης ή με την μετακίνηση της μπάρας ολίσθησης) γίνονται τα εξής:

- αλλάζει η εικόνα του αντικειμένου στην αρχική, διότι δεν πρέπει να ξεχνάμε ότι αν χτυπάμε το αντικείμενο αλλάζει σε αίμα, οπότε θέλουμε να βλέπουμε εκ νέου την αρχική εικόνα μας.
- καλούμε την διαδικασία με όνομα 'move teacher', η οποία μας μετακινεί το αντικείμενο στην οθόνη
- Σταματάμε κάθε ήχο που αναπαράγουμε διότι π.χ. δε θέλουμε να ακούγεται το ωχ και να κινείται το αντικείμενο



3. Τι θα γίνει όταν ο χρήστης καταφέρει και κτυπήσει το αντικείμενο (teacher)

- α) Θα ακουστεί ένας προκαθορισμένος ήχος (ωχ!!) όπως τον έχουμε ορίσει κατά τη φάση σχεδίασης της εφαρμογής
- β) η φορητή συσκευή θα δονείται για 1/10 του δευτερολέπτου



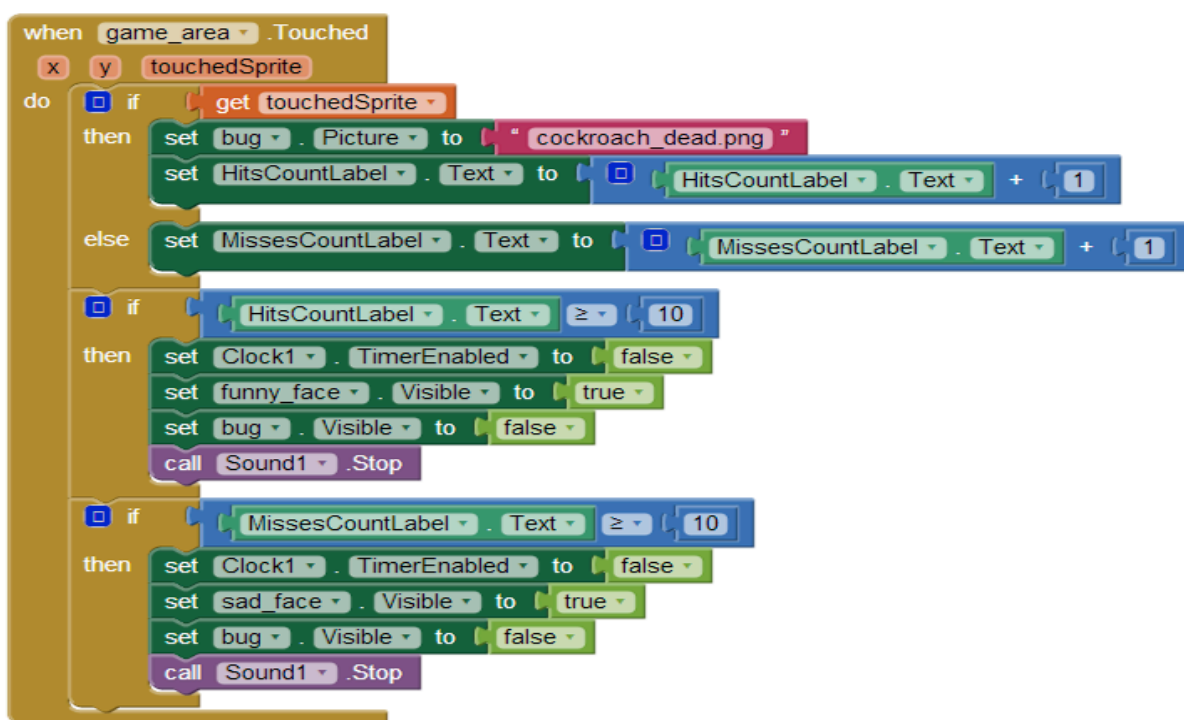
4. Συμπεριφορά του καμβά

Πως συμπεριφέρεται δηλαδή ο καμβάς (δηλαδή η οθόνη της φορητής συσκευής) κάθε φορά που ο χρήστης ακουμπάει επάνω της.

α) Εάν έχει καταφέρει να χτυπήσει το αντικείμενο bug, τότε αλλάζει η εικόνα του αντικειμένου και από κατσαρίδα εμφανίζεται ένα χέρι που έχει πιάσει την κατσαρίδα στην οθόνη για να δείξει ότι όντως η κατσαρίδα πιάστηκε και αυξάνουμε τον μετρητή HitsCountLabel, (η οποία μετράει τον αριθμό που ο χρήστης χτυπάει σωστά το αντικείμενο teacher) κατά 1. Αν ο χρήστης δεν κατάφερε σωστά να χτυπήσει το αντικείμενο αυξάνουμε τον μετρητή MissesCountLabel, (η οποία μετράει τον αριθμό που ο χρήστης δεν χτυπάει σωστά το αντικείμενο bug) κατά 1.

β) Ελέγχεται αν η μεταβλητή HitsCountLabel, η οποία μετράει τον αριθμό που ο χρήστης χτυπάει σωστά το αντικείμενο bug έχει φτάσει το νούμερο 10. Στις 10 επιτυχημένες προσπάθειες το παιχνίδι τελειώνει. Οπότε κατά σειρά απενεργοποιούμε το ρολόι διότι δεν θέλουμε να κινείται πλέον το αντικείμενο στην οθόνη, εμφανίζουμε ένα χαρούμενο πρόσωπο στην οθόνη αφού ο χρήστης κέρδισε, κρύβουμε το αντικείμενο και σταματάμε ότι ήχο αναπαράγαμε.

γ) Ελέγχεται αν η μεταβλητή MissesCountLabel, η οποία μετράει τον αριθμό που ο χρήστης δεν χτυπάει σωστά το αντικείμενο bug έχει φτάσει το νούμερο 10. Στις 10 αποτυχημένες προσπάθειες το παιχνίδι τελειώνει. Οπότε κατά σειρά απενεργοποιούμε το ρολόι διότι δεν θέλουμε να κινείται πλέον το αντικείμενο στην οθόνη, εμφανίζουμε ένα λυπημένο πρόσωπο στην οθόνη αφού ο χρήστης έχασε κρύβουμε το αντικείμενο και σταματάμε ότι ήχο αναπαράγαμε.

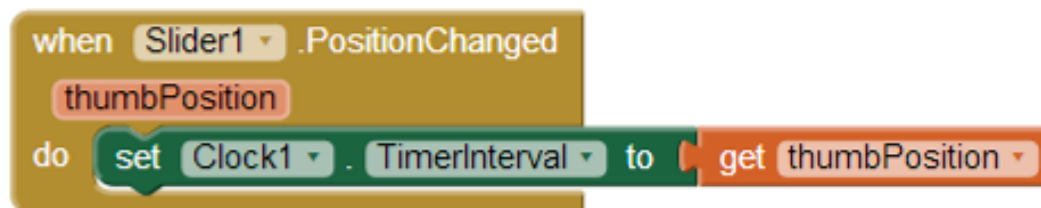


5. Μεταβολή της ταχύτητας του αντικειμένου

Μετακινώντας τη μπάρα ολίσθησης προς τα αριστερά αυξάνουμε την ταχύτητα μετακίνησης του αντικειμένου στην οθόνη (επομένως δυσκολεύουμε το παιχνίδι) ενώ όταν μετακινούμε τη μπάρα ολίσθησης προς τα δεξιά μειώνουμε την ταχύτητα μετακίνησης του αντικειμένου στην οθόνη (επομένως κάνουμε πιο εύκολο το παιχνίδι)

Πως υλοποιείται:

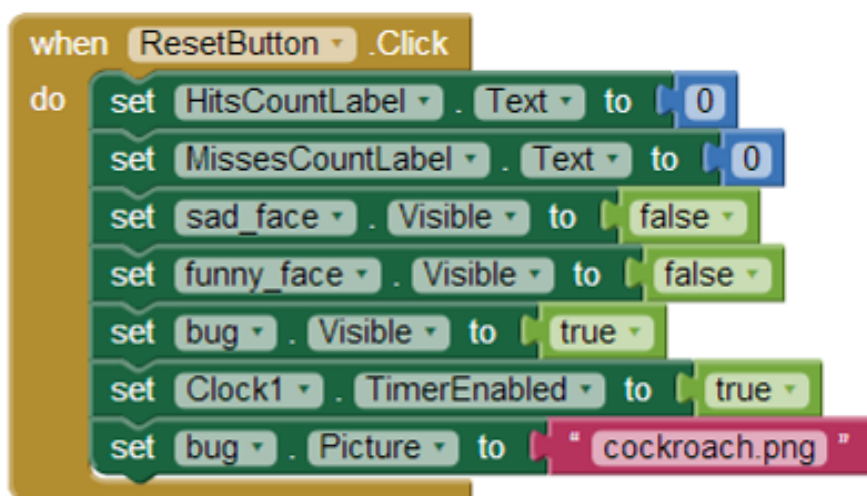
Η μπάρα ολίσθησης δέχεται τιμές σε ένα διάστημα τιμών το οποίο έχουμε ορίσει εμείς κατά τη φάση σχεδίασης ενώ θα θυμηθούμε ότι και τον ρυθμό μεταβολής εμείς τον έχουμε ορίσει εξίσου. Κάθε φορά που αλλάζει τιμή το αντικείμενο slider1 ενημερώνει τον χρόνο αντίδρασης του ρολογιού και αυτό με τη σειρά ενεργοποιείται πιο αργά ή πιο γρήγορα, επηρεάζοντας την ταχύτητα μετακίνησης του αντικειμένου στην οθόνη.

**6. Κουμπί επανεκκίνησης (Reset Button).**

Είτε χάσουμε είτε κερδίσουμε μπορεί να θέλουμε να παίξουμε το παιχνίδι ξανά.

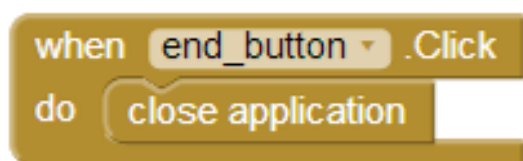
Οπότε αν πατήσει ο χρήστης το κουμπί με τίτλο 'Ξεκίνα ξανά!' τότε κατά σειρά γίνονται τα εξής:

- α) αρχικοποιούμε τον μετρητή που μετράει τις επιτυχημένες προσπάθειες με 0
- β) αρχικοποιούμε τον μετρητή που μετράει τις αποτυχημένες προσπάθειες με 0
- γ) Αν τυχόν έχουμε χάσει και έχει εμφανιστεί το λυπημένο πρόσωπο το κρύβουμε (το κάνουμε αόρατο)
- δ) Αν τυχόν έχουμε κερδίσει και έχει εμφανιστεί το χαρούμενο πρόσωπο το κρύβουμε (το κάνουμε αόρατο)
- ε) κάνουμε το αντικείμενο που εμφανίζεται στην οθόνη ορατό εκ νέου
- στ) ενεργοποιούμε εκ νέου τον μετρητή για να ξεκινήσει το αντικείμενο να κουνιέται στη οθόνη
- ζ) Θέτουμε την εικόνα του αντικειμένου στην αρχική εικόνα

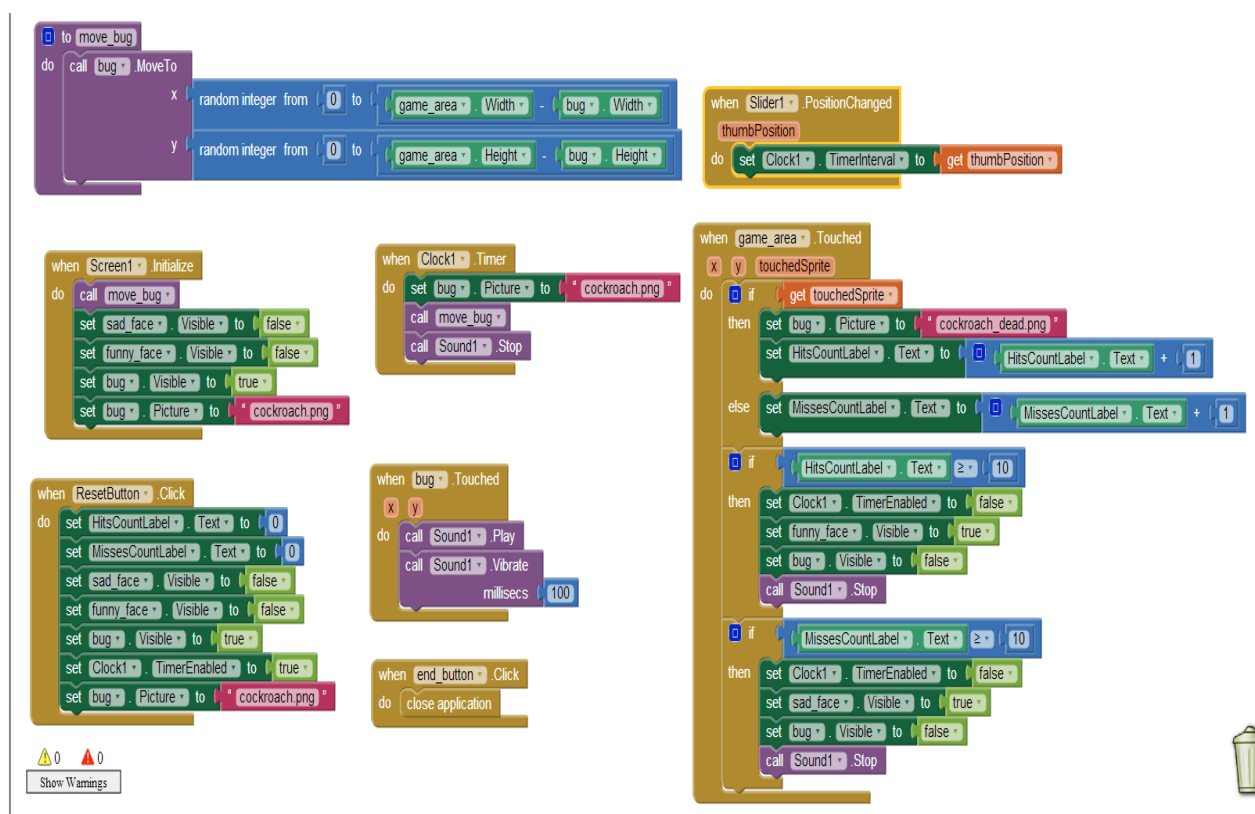


7. Κουμπί κλεισίματος εφαρμογής (End Button).

Στη περίπτωση που ανά πάσα στιγμή θέλουμε να εγκαταλείψουμε την εφαρμογή απλά πατάμε το κουμπί και μέσω της εντολής close application τερματίζεται η εφαρμογή



Ο ολοκληρωμένος κώδικας της εφαρμογής σας, λογικά θα πρέπει να μοιάζει με τον ακόλουθο:

**Ερωτήσεις για προβληματισμό**

Δε θα ήταν χρήσιμο αν για κάποιο λόγο να θέλατε προσωρινά να παγώσετε την εκτέλεση της εφαρμογής, δίχως να χάσετε σκορ κλπ να μπορούσατε να το κάνετε; Μπορείτε να σκεφτείτε πως θα το υλοποιήσετε;

Συμβουλή: προφανώς αφού βάλετε ένα νέο κουμπί θα θέλατε κάθε φορά που το πατάει ο χρήστης το ρολόι να σταματάει. Ίσως η δυσκολία σας είναι πως θα ενεργοποιήσετε εκ νέου την εφαρμογή από το σημείο που είχατε σταματήσει, δηλαδή δε θα πατήσετε το κουμπί Reset.

Καλή επιτυχία!!!!

Screenshots από διάφορα στάδια της εφαρμογής

