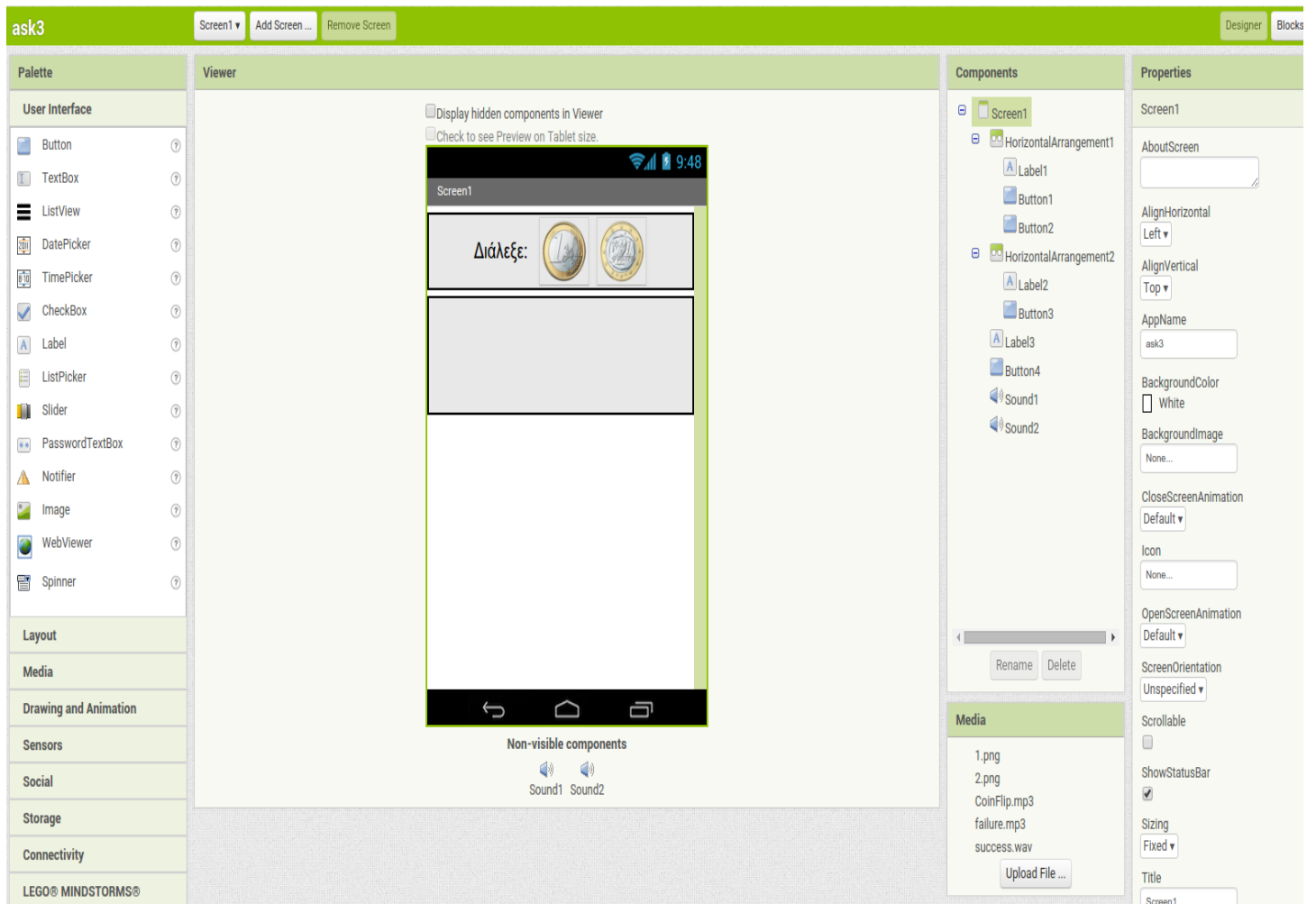


# ΣΗΜΕΙΩΣΕΙΣ APPINVENTOR DESIGNER



Στο χώρο της εργαλειοθήκης “**Palette**” του Appinventor υπάρχουν 9 κατηγορίες αντικειμένων που μπορούν να χρησιμοποιηθούν στην εκάστοτε εφαρμογής μας και είναι οι: **1) User Interface, 2) Layout, 3) Media, 4) Drawing and Animation, 5) Sensors, 6) Social, 7) Storage, 8) Connectivity, 9) Lego MindStorms**. Στις ασκήσεις μας θα χρησιμοποιήσουμε λίγα βασικά αντικείμενα από τις 6 πρώτες κατηγορίες.

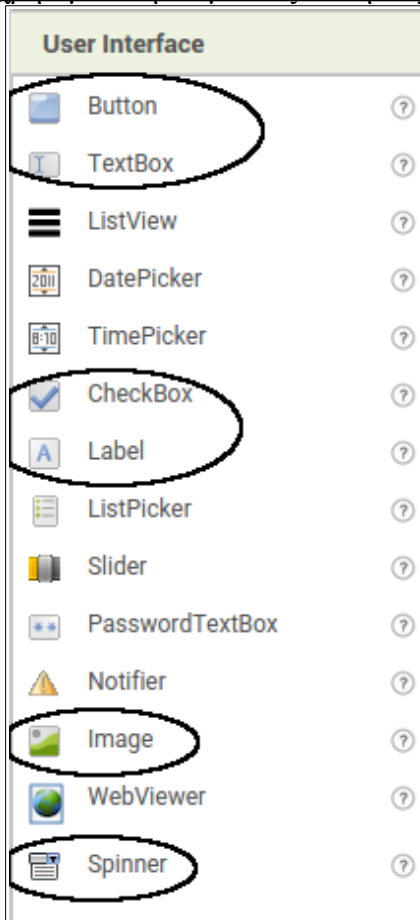
Όταν βρισκόμαστε σε κατάσταση σχεδίασης (Designer), τα αντικείμενα της εργαλειοθήκης μπορούν να συρθούν στο χώρο “**Viewer**” (ουσιαστικά δηλαδή στην οθόνη του κινητού μας) κι έτσι να χρησιμοποιηθούν στην εφαρμογή που κατασκευάζουμε.

Τα αντικείμενα που τοποθετούνται στην περιοχή Viewer εμφανίζονται και στην περιοχή “**Components**” σε μια δενδροειδή μορφή. Η μορφή είναι δενδροειδής μιας και ορισμένα αντικείμενα εμπεριέχονται σε άλλα και σίγουρα όλα εμπεριέχονται στο αντικείμενο Screen1 (ή Screen2, ... αν έχουμε πολλαπλές οθόνες).

Στην περιοχή “**Media**” εμφανίζονται όλα τα εξωτερικά αρχεία (ήχοι και εικόνες κυρίως) που απαιτεί η εφαρμογή μας και τα οποία έχουμε “ανεβάσει” εμείς οι ίδιοι επιλέγοντας το κουμπί Upload File.

Τέλος στην περιοχή “**Properties**” εμφανίζονται όλες οι ιδιότητες του αντικειμένου που είναι επιλεγμένο και οι αρχικές τιμές που αυτές έχουν κατά τη σχεδίαση της εφαρμογής. Σημειωτέον ότι οι τιμές των ιδιοτήτων αυτών πιθανότατα αλλάζουν με τη χρήση block κώδικα κατά την εκτέλεση της εφαρμογής.

Αντικείμενα της κατηγορίας **User Interface**. Κυκλωμένα είναι αυτά που πιθανότατα θα χρησιμοποιήσουμε στις ασκήσεις μας. Γι' αυτά παρακάτω φαίνονται βασικές τους ιδιότητες-γεγονότα:



## Button

Αντικείμενο-Κουμπί με την ικανότητα να ανιχνεύει κλικ. Πολλές πτυχές της εμφάνισης ενός κουμπιού μπορούν να αλλάξουν, καθώς επίσης και το κατά πόσον αυτό μπορεί να πατηθεί ή όχι.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Text** (οποιοδήποτε κείμενο σε " " που αναγράφεται στο κουμπί),

**Enabled** (true-false, δείχνει αν μπορεί να πατηθεί ή όχι)

**Visible** (true-false, δείχνει αν είναι ορατό ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

**Image** (όνομα αρχείου μέσα σε " " της εικόνας που φαίνεται πάνω στο κουμπί)

**FontSize** (ακέραια τιμή που δείχνει μέγεθος του font γραμμάτων του κειμένου του κουμπιού)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set Button1.Text to "Πάτησε με!"

Set Button1.Image to get global img (το img είναι μεταβλητή)

Set Button1.Visible to false (κάνει το Button1 αόρατο)

**Γεγονότα** (που σχετίζονται με Buttons):

Όταν κλικαρισθεί (πατηθεί και αφεθεί)	Όταν πατηθεί παρατεταμένα κι ύστερα αφεθεί	Την ώρα που είναι πατημένο	Την ώρα που αφήνεται
when Button1 .Click do	when Button1 .LongClick do	when Button1 .TouchDown do	when Button1 .TouchUp do

## Label

Μια ετικέτα εμφανίζει ένα κομμάτι κειμένου στην οθόνη του κινητού, το οποίο καθορίζεται από την ιδιότητα Text. Άλλες βασικές ιδιότητες που μπορούν να ρυθμιστούν σε Designer ή Blocks Editor, ελέγχουν την εμφάνιση και τη θέση του κειμένου.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Text** (οποιοδήποτε κείμενο σε " " το οποίο δείχνει η ετικέτα),

**Visible** (true-false, δείχνει αν είναι ορατή ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

**FontSize** (ακέραια τιμή που δείχνει μέγεθος του font γραμμάτων του κειμένου της ετικέτας)

**TextColor, BackgroundColor** (χρώμα, βλ. Built-in στοιχεία Appinventor)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set Label1.Text to "Πάτησε με!"

Set Label1.FontSize to 20

**Γεγονότα** (που σχετίζονται με Labels) δεν υπάρχουν...

## Image

Αντικείμενο για την εμφάνιση εικόνων. Η εικόνα που εμφανίζεται, καθώς και άλλες πτυχές της εμφάνισης της εικόνας μπορούν να προσδιοριστούν στον Designer ή στον Blocks Editor.

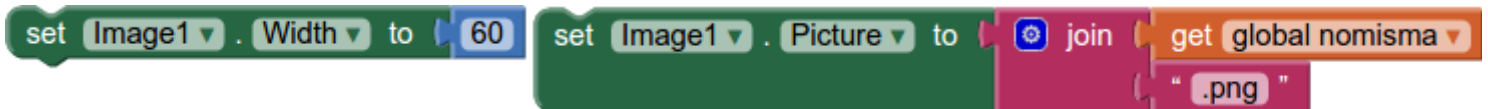
**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Picture** (όνομα αρχείου μέσα σε " " της εικόνας που εμφανίζεται),

**Visible** (*true-false*, δείχνει αν είναι ορατή ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει *pixels* πλάτους-ύψους)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές, π.χ.:



**Γεγονότα** (που σχετίζονται με Image) δεν υπάρχουν.

## TextBox

Ένα πλαίσιο κειμένου για το χρήστη για την εισαγωγή κειμένου. Η αρχική ή εισαγόμενη από τον χρήστη τιμή κειμένου βρίσκεται στην ιδιότητα Text. Αν είναι κενό, η ιδιότητα Hint, η οποία εμφανίζεται ως αχνό κείμενο στο πλαίσιο, μπορεί να παρέχει στο χρήστη με οδηγίες για το τι πρέπει να πληκτρολογήσετε. Η ιδιότητα NumbersOnly περιορίζει το πληκτρολόγιο για να δέχεται μόνο αριθμό ως είσοδο. Άλλες ιδιότητες επηρεάζουν την εμφάνιση του πλαισίου κειμένου (TextAlignment, BackgroundColor, κλπ) και αν μπορεί να χρησιμοποιηθεί (Enabled).

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Text** (οποιοδήποτε κείμενο δίνει ο προγραμματιστής από Designer ή Blocks ή ο χρήστης),

**Hint** (οποιοδήποτε κείμενο βάζει ο προγραμματιστής που αχνοφαίνεται ως υπόδειξη για το τι πρέπει να συμπληρωθεί στο πλαίσιο κειμένου από το χρήστη),

**Enabled** (*true-false*, δείχνει αν μπορεί να συμπληρωθεί ή όχι)

**Visible** (*true-false*, δείχνει αν είναι ορατό ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει *pixels* πλάτους-ύψους)

**FontSize** (ακέραια τιμή που δείχνει μέγεθος του font γραμμάτων του κειμένου του κουμπιού)

**TextColor, BackgroundColor** (χρώμα, βλ. Built-in στοιχεία Appinventor)

**NumbersOnly** (*true-false*, δείχνει αν δέχεται μόνο αριθμούς ή οποιοδήποτε κείμενο)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set Button1.NumbersOnly to true

**Γεγονότα** (που σχετίζονται με TextBox): **When TextBox1.GotFocus do**, **When TextBox1.LostFocus do** (δε θα τα χρησιμοποιήσουμε στις απλές μας ασκήσεις)

**Διαδικασίες** (που σχετίζονται με TextBox): **call TextBox1.HideKeyboard**, **call TextBox1.RequestFocus** (δε θα τα χρησιμοποιήσουμε στις απλές μας ασκήσεις)

## Spinner

Ένα αντικείμενο spinner εμφανίζει τη δυνατότητα επιλογής από μια λίστα στοιχείων (με αναδυόμενο παράθυρο όταν πατηθεί). Τα στοιχεία αυτά μπορούν να τεθούν στο Designer ή στο Blocks ορίζοντας την ιδιότητα `ElementsFromString` ως μια ακολουθία επιλογών χωριζόμενων με κόμμα. π.χ. "επιλογή1,επιλογή2,επιλογή3".

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**ElementsFromString** (οποιοδήποτε κείμενο ως ακολουθία επιλογών χωριζόμενων με "," δίνει ο προγραμματιστής από Designer ή Blocks ),

**Prompt** (οποιοδήποτε κείμενο βάζει ο προγραμματιστής που φαίνεται ως υπόδειξη για το τι πρέπει να κάνει ο χρήστης ή τι εμπεριέχει η λίστα επιλογών από την οποία πρέπει να επιλέξει )

**SelectionIndex** (οποιοσδήποτε ακέραιος μεταξύ του 1 και του πλήθους των επιλογών της λίστας. Τίθεται από τον προγραμματιστή στο Blocks ή από το χρήστη όταν επιλέγει μεταξύ των στοιχείων της λίστας).

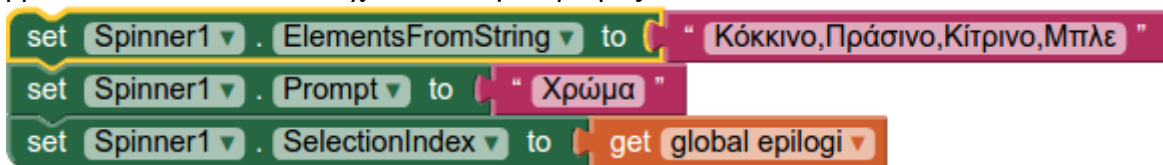
**Selection** (οποιοδήποτε κείμενο αντιστοιχεί σε μια από τις επιλογές της λίστας στοιχείων. Τίθεται από τον προγραμματιστή στο Designer ή στο Blocks, ή από τον χρήστη όταν επιλέγει μεταξύ των στοιχείων της λίστας).

**Visible** (true-false, δείχνει αν είναι ορατό ή όχι)

**Width ή Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.:



**Γεγονότα** (που σχετίζονται με Spinner): **When Spinner1.AfterSelecting Selection do** (δε θα το χρησιμοποιήσουμε στις απλές μας ασκήσεις)

**Διαδικασίες** (που σχετίζονται με Spinner): **call Spinner1.DisplayDropdown** (εμφανίζει το αναδυόμενο παράθυρο)

## CheckBox

Κουτάκι που δημιουργεί ένα γεγονός όταν ο χρήστης κάνει κλικ πάνω του. Υπάρχουν πολλές ιδιότητες που επηρεάζουν την εμφάνισή του, που μπορούν να ρυθμιστούν σε Designer ή Blocks Editor.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Checked** (true-false, δείχνει αν είναι κλικαρισμένο ή όχι)

**Text** (οποιοδήποτε κείμενο δίνει ο προγραμματιστής από Designer ή Blocks για το CheckBox),

**Enabled** (true-false, δείχνει αν μπορεί να κλικαριστεί ή όχι)

**Visible** (true-false, δείχνει αν είναι ορατό ή όχι)

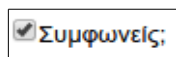
**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

**FontSize** (ακέραια τιμή που δείχνει μέγεθος του font γραμμάτων του κειμένου του κουμπιού)

**TextColor, BackgroundColor** (χρώμα, βλ. Built-in στοιχεία Appinventor)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.




Π.χ.: Set CheckBox1.Text to "Συμφωνείς;"



Set CheckBox1.Checked to true

**Γεγονότα** (που σχετίζονται με CheckBox): **When CheckBox1.Changed do, When CheckBox1.GotFocus do, When CheckBox1.LostFocus do** (Το 2ο και 3ο δε θα τα χρησιμοποιήσουμε στις απλές μας ασκήσεις)

Αντικείμενα της κατηγορίας **Layout**. Κυκλωμένα είναι αυτά που πιθανότατα θα χρησιμοποιήσουμε στις ασκήσεις μας. Γι' αυτά παρακάτω φαίνονται βασικές τους ιδιότητες:

Layout	HorizontalArrangement
 TableArrangement	Στοιχείο μορφοποίησης – περιοχή στην οποία ό,τι αντικείμενο τοποθετείται εμφανίζεται από τα αριστερά προς τα δεξιά. Τη χρησιμοποιούμε για να τοποθετήσουμε μέσα της buttons, images, labels κλπ σε οριζόντια διάταξη.
 VerticalArrangement	Στην περιοχή components εμφανίζεται ως γονικό αντικείμενο (parent) όσων αντικειμένων τοποθετηθούν μέσα σε αυτή.
 HorizontalArrangement	

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**AlignHorizontal** ή **AlignVertical** (ακέραιος 1-3. 1=Αριστερή ή πάνω στοίχιση αντικειμένων, 2=Στο κέντρο, 3=Δεξιά ή κάτω στοίχιση),

**Visible** (true-false, δείχνει αν είναι ορατή ή όχι αυτή και τα αντικείμενα που περιέχει)

**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

**Image** (όνομα αρχείου μέσα σε " " της εικόνας που φαίνεται ως background στην περιοχή)

**BackgroundColor** (χρώμα, βλ. Built-in στοιχεία Appinventor)

Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.











Π.χ.: Set HorizontalArrangement1.AlignHorizontal to 2 (στοιχίζει οριζόντια τα αντικείμενα της στο κέντρο)

## VerticalArrangement

Στοιχείο μορφοποίησης – περιοχή στην οποία ό,τι αντικείμενο τοποθετείται εμφανίζεται από πάνω προς τα κάτω. Τη χρησιμοποιούμε για να τοποθετήσουμε μέσα της buttons, images, labels κλπ σε κάθετη διάταξη. Στην περιοχή components εμφανίζεται ως γονικό αντικείμενο (parent) όσων αντικειμένων τοποθετηθούν μέσα σε αυτή.

**Properties - Ιδιότητες** (ίδιες με την Horizontal Arrangement και τίθενται με αντίστοιχες εντολές Set):

Αντικείμενα της κατηγορίας **Media**. Κυκλωμένα είναι αυτά που πιθανότατα θα χρησιμοποιήσουμε στις ασκήσεις μας. Γι' αυτά παρακάτω φαίνονται βασικές τους ιδιότητες:

Media	Sound
 Player	Ένα αντικείμενο πολυμέσων που παίζει αρχεία ήχου και προαιρετικά δονείται για τόσα χιλιοστά του δευτερολέπτου όσα καθορίζονται στο Blocks. Το όνομα του αρχείου ήχου που θα παίζει πρέπει να καθοριστεί με την ιδιότητα Source είτε στο Designer ή στο Blocks. Είναι καλύτερο για μικρά αρχεία ήχου, όπως ηχητικά εφέ. Για μεγαλύτερου χρονικού διαστήματος ήχους, όπως τραγούδια να χρησιμοποιείτε το αντικείμενο Player.
 Sound	<b>Properties - Ιδιότητες</b> (και τιμές που μπορούν να πάρουν):
 TextToSpeech	<b>Source</b> (οποιοδήποτε κείμενο μέσα σε " " ή μεταβλητή με κείμενο που παραπέμπει σε αρχείο ήχου π.χ. mp3, wav που είναι φορτωμένο στην εφαρμογή)
 YandexTranslate	<b>MinimumInterval</b> (ακέραιος που εκφράζει τα χιλιοστά του δευτερολέπτου που πρέπει να περάσουν για να μπορεί να ξανακουστεί ήχος που αντιστοιχεί στο ίδιο αντικείμενο ήχου)
 VideoPlayer	
 Camcorder	
 Camera	
 SpeechRecognizer	
 ImagePicker	
 SoundRecorder	

Οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set Sound1.Source to "success.mp3"

Set Sound1.MinimumInterval to 1000 (πρέπει να περάσει 1sec για να ξανακληθεί η διαδικασία Sound1.Play)

**Διαδικασίες** (που σχετίζονται με Sound1):

*call Sound1.Play* (παίζει τον ήχο που αντιστοιχεί στο αντικείμενο Sound1)

*call Sound1.Stop* (σταματάει να παίζει τον ήχο που αντιστοιχεί στο αντικείμενο Sound1)

*call Sound1.Pause* (σταματάει προσωρινά τον ήχο που αντιστοιχεί στο αντικείμενο Sound1)

*call Sound1.Resume* (ξαναρχίζει τον ήχο που αντιστοιχεί στο αντικείμενο Sound1 απ' όπου είχε σταματήσει με κλήση διαδικασίας Sound1.Pause)

*call Sound1.Vibrate* (παίρνει σαν είσοδο ακέραιο που εκφράζει msec και δονεί το κινητό για αντίστοιχο χρονικό διάστημα κατά το παίξιμο του ήχου που αντιστοιχεί στο Sound1)

## Player

Αντικείμενο πολυμέσων που παίζει ήχους και ελέγχει τη δόνηση του κινητού. Το όνομα του αρχείου ήχου που θα παίζει πρέπει να καθοριστεί με την ιδιότητα Source είτε στο Designer ή στο Blocks. Αυτό το αντικείμενο είναι καλύτερο για μεγάλα αρχεία ήχου, όπως τραγούδια, ενώ το αντικείμενο Sound είναι πιο αποδοτικό για μικρά αρχεία, όπως τα ηχητικά εφέ.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Source** (οποιοδήποτε κείμενο μέσα σε " " ή μεταβλητή με κείμενο που παραπέμπει σε αρχείο ήχου π.χ. mp3, wav που είναι φορτωμένο στην εφαρμογή)

**Volume** (ακέραιος 0-100 που εκφράζει την ένταση του ήχου)

**Loop** (true-false, καθορίζει αν ο ήχος θα αρχίσει από την αρχή όταν ολοκληρωθεί το παίξιμό του)

Οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

π.χ. Set Player1.Loop to true (θα παίζει κατ' επανάληψη τον ήχο)

**Γεγονότα** (που σχετίζονται με Player): **When Player1.Completed do** (Γεγονός που συμβαίνει όταν ολοκληρώνεται το παίξιμο του ήχου)

**Διαδικασίες** (που σχετίζονται με Player):




*call Player1.Start* (παίζει τον ήχο που αντιστοιχεί στο αντικείμενο Player1)

*call Player1.Stop* (σταματάει να παίζει τον ήχο που αντιστοιχεί στο αντικείμενο Player1)

*call Player1.Pause* (σταματάει προσωρινά τον ήχο που αντιστοιχεί στο αντικείμενο Player1)

*call Player1.Vibrate* (παίρνει σαν είσοδο ακέραιο που εκφράζει msec και δονεί το κινητό για αντίστοιχο χρονικό διάστημα κατά το παίξιμο του ήχου που αντιστοιχεί στο Player1)

Αντικείμενα της κατηγορίας **Drawing and Animation**. Κυκλωμένα είναι αυτά που πιθανότατα θα χρησιμοποιήσουμε στις ασκήσεις μας. Γι' αυτά παρακάτω φαίνονται βασικές τους ιδιότητες:

Drawing and Animation	Canvas
 Canvas	Αντικείμενο που αντιστοιχεί σε ένα διδιάστατο ορθογώνιο χώρο όπου μπορεί κάποιος να ζωγραφίσει και στοιχεία εικόνων μπορούν να κινηθούν. Κάθε σημείο του Canvas προσδιορίζεται από ένα ζευγάρι τιμών (x, y):
 ImageSprite	το x αντιστοιχεί στον αριθμό των pixels από την αριστερή ακμή του Canvas
 Ball	το y αντιστοιχεί στον αριθμό των pixels από την πάνω ακμή του Canvas.

Υπάρχουν γεγονότα που υποδεικνύουν πότε και που έχει πατηθεί ένας Canvas ή ένα ImageSprite ή Ball έχει συρθεί. Επίσης υπάρχουν μέθοδοι για σχεδιασμό σημείων, γραμμών και κύκλων.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**BackgroundColor - PaintColor** (χρώμα, βλ. Built-in στοιχεία. Καθορίζει το χρώμα φόντου του Canvas και το χρώμα που θα χρησιμοποιηθεί για ζωγραφική σε αυτόν),

**BackgroundImage** (όνομα αρχείου μέσα σε " " της εικόνας που φαίνεται ως φόντο στον Canvas)

**Visible** (true-false, δείχνει αν είναι ορατός ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

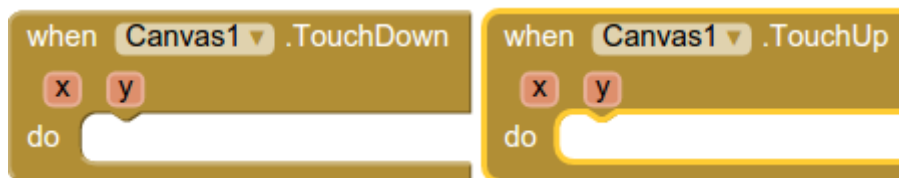
**LineWidth** (ακέραια τιμή που υποδεικνύει το πάχος γραμμής που σχεδιάζεται με αντίστοιχη διαδικασία στον Canvas)

**FontSize** (ακέραια τιμή που υποδεικνύει το μέγεθος του font του κειμένου που σχεδιάζεται στον Canvas)

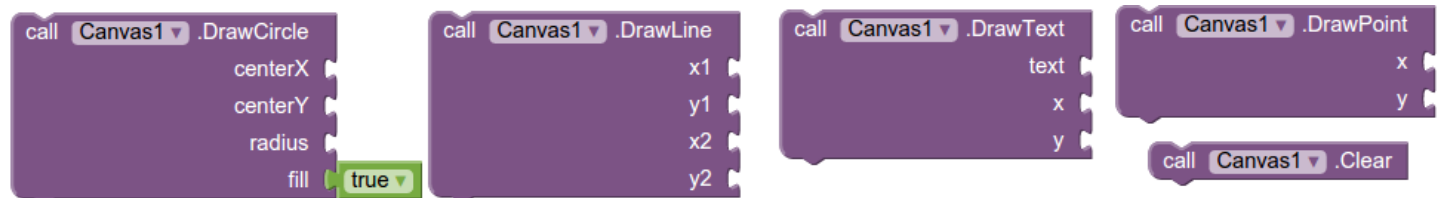
Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set Canvas1.BackgroundImage to "Grass.jpg"

**Γεγονότα** (που σχετίζονται με Canvas και χρησιμοποιήσαμε στην άσκηση AngryDog). Οι εντολές που θα τοποθετούνται μέσα σε αυτά τα γεγονότα μπορούν να χρησιμοποιήσουν τις τιμές των x, y όπου πατήθηκε ή "ξε-πατήθηκε" ο καμβάς (get x, get y)



**Διαδικασίες** (που σχετίζονται με Canvas):



Χρησιμοποιούνται για να σχεδιάσουμε με το δάκτυλο πάνω στον Canvas κύκλους, γραμμές, κείμενο ή σημείο στα σημεία όπου πατήθηκε το δάκτυλό μας ή μεταξύ των σημείων που πατήθηκε και σύρθηκε το δάκτυλό μας. Επίσης για καθαρισμό του Canvas.

Πιθανότατα δε θα προλάβουμε να κάνουμε άσκηση όπου να τις χρησιμοποιήσουμε στα πλαίσια του μαθήματος.

## ImageSprite

Αντικείμενο που μπορεί να τοποθετηθεί μέσα σε έναν Canvas όπου μπορεί να αλληλεπιδράσει σε πατήματα και συρσίματα του δακτύλου ή με άλλα ImageSprites ή Balls καθώς και με τις άκρες του Canvas. κινούμενο βάσει των τιμών των ιδιοτήτων του. Το τι εικόνα θα έχει εξαρτάται την ιδιότητά του Picture .

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

**Picture** (όνομα αρχείου μέσα σε " " της εικόνας που έχει το ImageSprite)

**Visible** (true-false, δείχνει αν είναι ορατό ή όχι)

**Width, Height** (ακέραια τιμή που εκφράζει pixels πλάτους-ύψους)

**Speed** (ακέραιος που δείχνει την ταχύτητα με την οποία κινείται το ImageSprite, εκφράζει τα pixels που θα μετακινηθεί σε χρόνο που υποδεικνύεται από την ιδιότητα Interval)

**Interval** (ακέραιος που υποδεικνύει κάθε πόσο χρονικό διάστημα σε msec ανανεώνεται η θέση του)

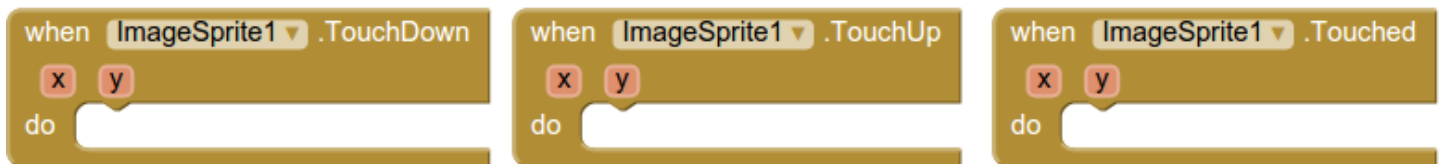
**Heading**(ακέραιος που υποδεικνύει γωνία σε μοίρες που σχηματίζει η κατεύθυνση κίνησής του με τον x'x)

**X ή Y ή Z** (υποδεικνύει τη θέση του στον Canvas)

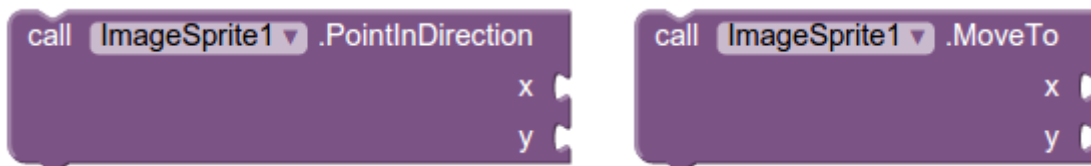
Όλες οι παραπάνω ιδιότητες τίθενται με χρήση αντίστοιχων εντολών Set και οι τιμές που μπορούν να πάρουν μπορεί να βρίσκονται και σε αντίστοιχων τύπων μεταβλητές.

Π.χ.: Set ImageSprite1.Picture to "SleepyDog.png"

**Γεγονότα** (που σχετίζονται με ImageSprite). Οι εντολές που θα τοποθετούνται μέσα σε αυτά τα γεγονότα μπορούν να χρησιμοποιήσουν τις τιμές των x, y όπου πατήθηκε ή "ξε-πατήθηκε" το ImageSprite (get x, get y)



**Διαδικασίες** (που σχετίζονται με ImageSprite, την 1η τη χρησιμοποιήσαμε στην άσκηση AngryDog):










Η πρώτη διαδικασία κάνει το ImageSprite1 να κατευθυνθεί προς την κατεύθυνση των x,y που ορίζονται.

Η δεύτερη διαδικασία μετακινεί αυτόματα το ImageSprite1 στη θέση x,y που ορίζεται.



Αντικείμενα της κατηγορίας **Sensors**. Κυκλωμένα είναι αυτά που πιθανότατα θα χρησιμοποιήσουμε στις ασκήσεις μας. Γι' αυτά παρακάτω φαίνονται βασικές τους ιδιότητες:

Sensors	AccelerometerSensor
 Clock	Μη ορατό αντικείμενο – αισθητήρας που μπορεί να εντοπίσει ταρακουνήματα του κινητού και να μετρήσει την επιτάχυνσή του στις 3 διαστάσεις.
 BarcodeScanner	
 LocationSensor	
 OrientationSensor	
 NearField	
 AccelerometerSensor	<b>Properties - Ιδιότητες</b> (και τιμές που μπορούν να πάρουν): <i>Enabled</i> (true-false, δείχνει αν είναι ενεργοποιημένος ο αισθητήρας) <i>MinimumInterval</i> (ελάχιστο χρον.διάστημα σε msec μεταξύ ταρακουνημάτων που μπορεί να ανιχνεύσει) <i>Sensitivity</i> (ακέραιος 1-3. 1=μικρή ευαισθησία 2=μέτρια 3=μεγάλη) <i>Xaccel, Yaccel, Zaccel</i> (αυτές οι ιδιότητες <u>τίθενται αυτόματα από τον αισθητήρα. δεν μπορεί να τις πειράξει ο προγραμματιστής.</u> Xaccel θετικό όταν το κινητό γέρνει δεξιά ή αρνητικό όταν γέρνει αριστερά, Yaccel θετικό όταν το πίσω μέρος του κινητού γέρνει προς τα πάνω ή αρνητικό όταν το μπροστά μέρος γέρνει προς τα πάνω, Zaccel -9,8 όταν κινητό αφημένο “ανάσκελα” σε επίπεδο ή +9,8 όταν είναι αφημένο “μπρούμυτα”)
 ProximitySensor	

**Γεγονότα** (που σχετίζονται με AccelerometerSensor). Στην άσκηση με τα ζάρια χρησιμοποιήσαμε το πρώτο.

```
when AccelerometerSensor1 .Shaking
do
```

```
when AccelerometerSensor1 .AccelerationChanged
  xAccel yAccel zAccel
do
```

## Clock

Μη ορατό αντικείμενο που παρέχει τη δυνατότητα μετρήσεων χρόνου με τη χρήση του εσωτερικού ρολογιού του κινητού. Ουσιαστικά πυροδοτεί χρονικά γεγονότα (χτύπους ρολογιού) σε καθοριζόμενα χρονικά διαστήματα για την εκτέλεση υπολογισμών και χειρισμών που σχετίζονται με το χρόνο.

**Properties - Ιδιότητες** (και τιμές που μπορούν να πάρουν):

*TimerEnabled* (true-false, δείχνει αν είναι ενεργοποιημένο το ρολόι ή όχι)

*TimerInterval* (χρονικό διάστημα σε msec μεταξύ κτύπων-γεγονότων του ρολογιού)

*TimerAlwaysFires* (true-false, δείχνει αν το ρολόι παράγει γεγονότα ακόμα κι όταν η εφαρμογή δεν φαίνεται στην οθόνη του κινητού – αλλά προφανώς εκτελείται, ή όχι)

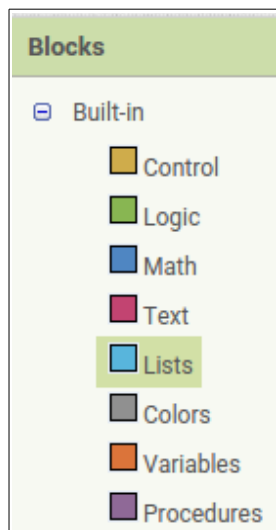
**Γεγονός** (που σχετίζονται με Clock).

```
when Clock1 .Timer
do
```

**Διαδικασίες** (που σχετίζονται με Clock είναι πάρα πολλές αλλά δε θα δούμε στις ασκήσεις μας)

Αντικείμενα της κατηγορίας **Social**. θα χρησιμοποιήσουμε στην άσκηση5 αλλά δεν θα έχουμε να τα ξέρουμε.

# BLOCKS



Στην περιοχή των Blocks – στον Blocks Editor – υπάρχουν εκτός των σχετικών blocks των αντικειμένων που έχουμε εισάγει στην εφαρμογή μας από το Designer και ενσωματωμένα στοιχεία που πιθανόν να χρειαστούν κατά τη σύνθεση του block κώδικα της εφαρμογής.

Στα ενσωματωμένα στοιχεία συναντάμε:

**Control blocks:** στοιχεία για τον έλεγχο ροής της εφαρμογής, όπως στοιχεία επιλογής (if), επανάληψης (for ή while), χειρισμού οθονών (open another ή close screen) κλπ.

**Logic blocks:** στοιχεία για λογικές πράξεις (and, or, not), καθώς και λογικές σταθερές (true, false)

**Math blocks:** στοιχεία για αριθμητικές πράξεις ("+", "-", "x", "/", "^", "modulo of ") και πράξεις σύγκρισης ("=", ">", "≥", "<", "≤", "≠"), συναρτήσεις ("square root", "absolute", "neg", "log", "e^", "round", "ceiling", "floor", "sin" -ημίτονο-, "cos" -συνημίτονο-, "tan" -εφαπτομένη-, γεννήτριες τυχαίων αριθμών ("random integer from 1 to 100")

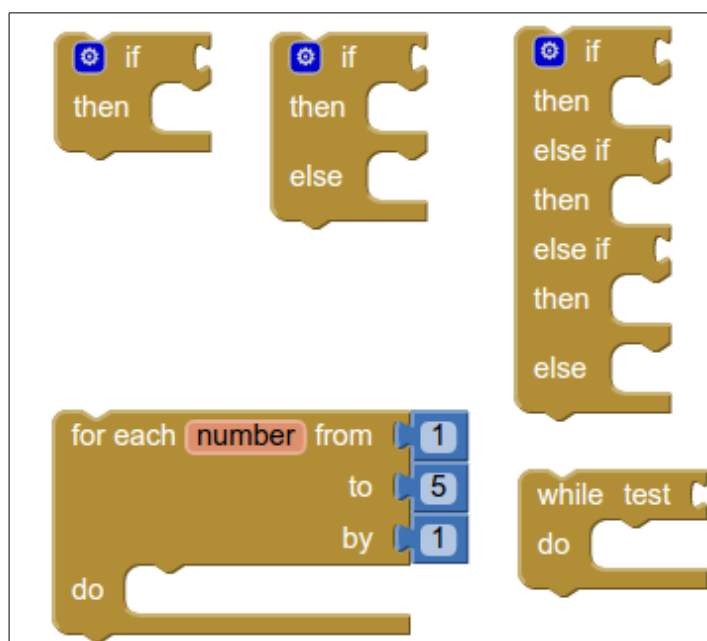
**Text blocks:** στοιχεία για σύγκριση κειμένων ("<", "=", ">"), συνένωση κειμένων ("join"), εύρεσης και αντικατάστασης κειμένου ("contains text", "replace all text", "starts at text"), εύρεσης μήκους κειμένου ("length").

**Color blocks:** block για κάθε χρώμα καθώς και block για δημιουργία χρώματος ως RGB σύνθεση ("make color").

**Variables:** blocks για την αρχικοποίηση μεταβλητών ("initialize global name to"), εκχώρηση τιμής ("set ... to") και ανάγνωση τιμής ("get ...").

**Procedures:** blocks για τη δημιουργία διαδικασιών που δεν επιστρέφουν ή επιστρέφουν τιμές.

## Control blocks



Στις ασκήσεις μας θα χρησιμοποιήσουμε κυρίως τα 3 πάνω blocks (το 2ο και 3ο παράγονται από το 1ο πατώντας το μπλε τετράγωνο).

Αντιπροσωπεύουν τις δομές επιλογής στον προγραμματισμό:

Απλή Επιλογή, Σύνθετη Επιλογή, Πολλαπλή επιλογή.

**Απλή Επιλογή If ... then ...:** ελέγχεται μια συνθήκη (λογική έκφραση που κατασκευάζεται από Logic και Math blocks με τιμή-αποτέλεσμα true ή false) και αν η τιμή της είναι true εκτελούνται οι εντολές που περιέχονται στην περιοχή του then.

**Σύνθετη Επιλογή If ... then ... else ... :** ελέγχεται μια συνθήκη και αν η τιμή της είναι true εκτελούνται οι εντολές που περιέχονται στην περιοχή του then, αλλιώς οι εντολές που περιέχονται στην περιοχή του else.

**Πολλαπλή Επιλογή If ... then ... elseif ... then ... ... else ... :** ελέγχεται η πρώτη συνθήκη και αν είναι true εκτελούνται οι εντολές που περιέχονται στην περιοχή του πρώτου then, ενώ αν είναι false εξετάζεται η δεύτερη συνθήκη. Αν εκείνη είναι true εκτελούνται οι εντολές στην περιοχή του δεύτερου then, ενώ αν είναι false εξετάζεται η τρίτη συνθήκη κ.ο.κ. Αν δεν ισχύει καμιά από τις ελεγχόμενες συνθήκες εκτελούνται οι εντολές else. Η 4η και 5η δομή ελέγχου είναι δομές επανάληψης και χρησιμοποιούνται για να εκτελέσουμε εντολές προκαθορισμένο αριθμό φορές (for) ή όσο η συνθήκη test είναι true (while). Πιθανότατα δε θα δούμε φέτος δομή επανάληψης.

### Logic blocks



Τα 2 πρώτα blocks αντιπροσωπεύουν τις 2 λογικές τιμές-σταθερές true και false

Τα 3ο, 4ο, 5ο αντιπροσωπεύουν τους τελεστές των λεγόμενων λογικών πράξεων not, and, or

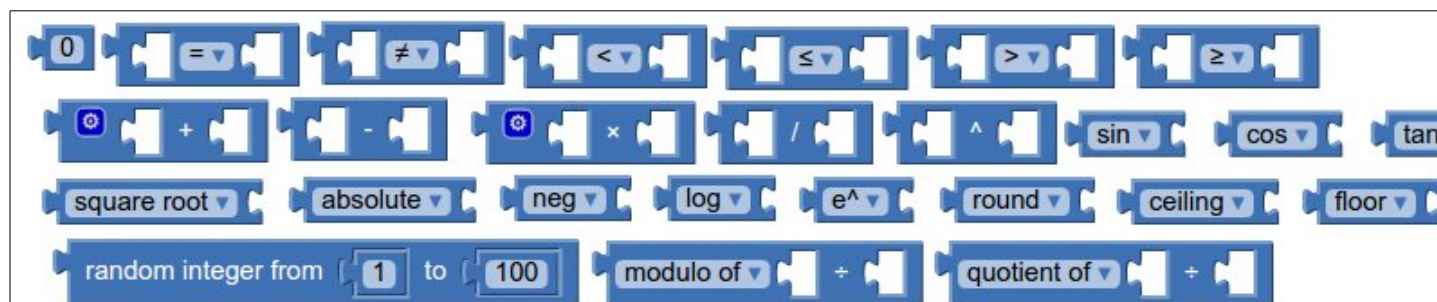
παράδειγμα λογικής πράξης  $5 > 4$  and  $3 < 2$ . Αυτή η πράξη έχει αποτέλεσμα false αφού δεν ισχύουν και οι 2 συγκρίσεις ταυτόχρονα.

παράδειγμα λογικής πράξης  $5 > 4$  or  $3 < 2$ . Αυτή η πράξη έχει αποτέλεσμα true αφού ισχύει έστω μία από τις συγκρίσεις.

παράδειγμα λογικής πράξης not ( $5 \neq 4$ ). Αυτή η πράξη έχει αποτέλεσμα false αφού το not αντιστρέφει το αποτέλεσμα της παρένθεσης  $5 \neq 4$  που είναι true, άρα το αποτέλεσμα είναι false.

Το 6ο block αντιπροσωπεύει τον έλεγχο αν 2 πράγματα είναι ή δεν είναι ίσα.

### Math Blocks



Το 1ο block αντιπροσωπεύει μια οποιαδήποτε αριθμητική σταθερά (ακέραια ή δεκαδική)

Τα 2ο – 7ο αντιπροσωπεύουν πράξεις σύγκρισης.

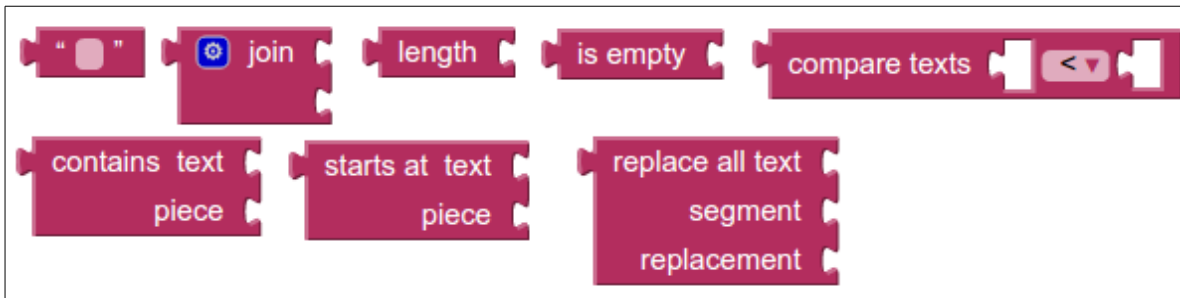
Τα 8ο – 12 αντιπροσωπεύουν αριθμητικές πράξεις. Συγκεκριμένα το 12ο την ύψωση σε δύναμη

Τα 13ο – 15ο αντιπροσωπεύουν συνηθισμένες τριγωνομετρικές συναρτήσεις

Τα 16ο – 23ο αντιπροσωπεύουν συνηθισμένες μαθηματικές συναρτήσεις

Το 24ο γεννήτρια τυχαίων ακεραίων αριθμών. Το 22ο την πράξη υπόλοιπο. Το 23ο την πράξη πηλίκο.

## Text Blocks



To 1o block αντιπροσωπεύει μια οποιαδήποτε αλφαριθμητική σταθερά δηλ. σταθερά κείμενου

To 2o block συνενώνει 2 ή περισσότερες αλφαριθμητικές σταθερές σε μία.

To 3o block εξετάζει μια μεταβλητή ή σταθερά που περιέχει κείμενο και δίνει το μήκος του κειμένου

To 4o block εξετάζει μια μεταβλητή κι επιστρέφει true αν δεν περιέχει καθόλου κείμενο ή false αν περιέχει

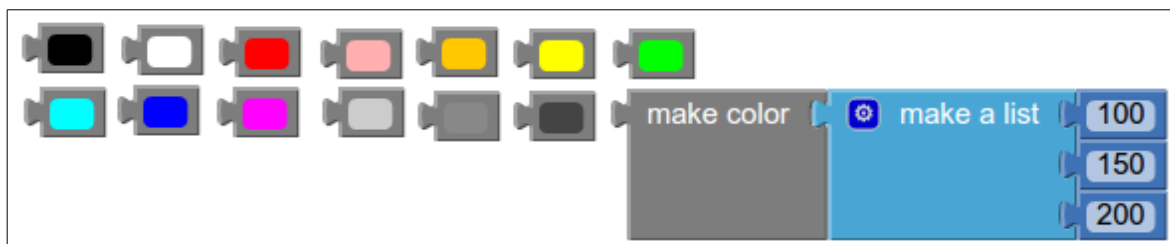
To 5o block συγκρίνει λεξικογραφικά μεταβλητές ή σταθερές κειμένου κι επιστρέφει true ή false

To 6o block εξετάζει αν μια ακολουθία piece κειμένου υπάρχει μέσα στο κείμενο text.

To 7o block επιστρέφει τη θέση μέσα στο text που εμφανίζεται για πρώτη φορά η ακολουθία κειμένου piece.

To 8o block αντικαθιστά τις ακολουθίες segment στο text με το replacement

## Color Blocks



Κάθε block αντιπροσωπεύει ένα χρώμα, εκτός το τελευταίο που αντιπροσωπεύει τη δυνατότητα μας να φτιάξουμε οποιοδήποτε χρώμα ως μια σύνθεση-list 3 αποχρώσεων των βασικών χρωμάτων : του κόκκινου R(0-255), του πράσινου G(0-255) και του μπλε B(0-255).

## Variable Blocks

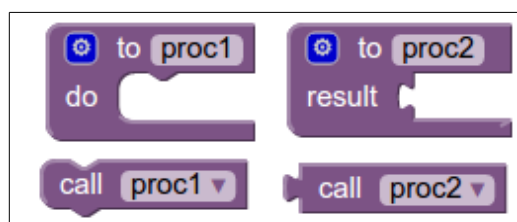


To 1o block αντιπροσωπεύει την δημιουργία και αρχικοποίηση μιας μεταβλητής δηλαδή την απόδοση σε αυτήν μιας αρχικής τιμής κατά την εκκίνηση της εφαρμογής. Ένα τέτοιο block εμφανίζεται αυτόνομο στον block κώδικά μας και δεν συνδέεται με άλλα blocks γεγονότων ή πράξεων κλπ...

To 2o block ουσιαστικά διαβάζει την τιμή που έχει μια μεταβλητή – εδώ η zari1 – στη μνήμη

To 3o block εκχωρεί , αποθηκεύει στη μεταβλητή – εδώ στη zari1 – μια τιμή.

## Procedure Blocks



To 1o και 2o block μας δίνουν τη δυνατότητα να φτιάξουμε διαδικασίες που δεν επιστρέφουν ή επιστρέφουν τιμές κατά την κλήση τους αντίστοιχα.

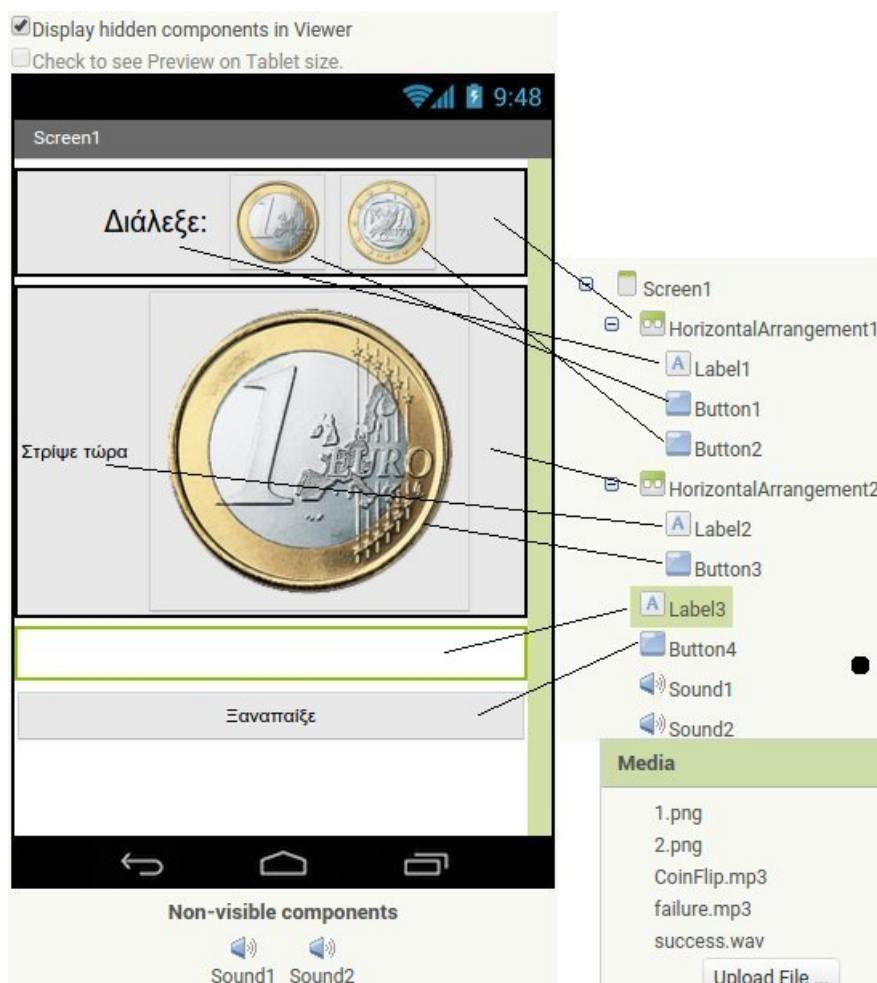
Τα 3o και 4o block καλούν αντίστοιχα αυτές τις διαδικασίες.

## Παραδείγματα Ασκήσεων

### Πώς θα μπορούσε να είναι η εκφώνηση της άσκησης 3 CoinFlip;

#### Θέμα Δ

Θέλουμε να φτιάξουμε σε Appinventor μια εφαρμογή-προσομοίωση του παιχνιδιού κορώνα γράμματα. Η σχεδίαση της εφαρμογής φαίνεται στην επόμενη εικόνα.



Αρχικά, από τη σχεδίαση και κατά την εκκίνηση της εφαρμογής τα αντικείμενα Label2, Button3, Label3 και Button4 θα είναι κρυμμένα (η ιδιότητα Visible δεν θα είναι τσεκαρισμένη). Άρα θα φαίνονται μόνο τα αντικείμενα που εμπεριέχονται στην HorizontalArrangement1 δηλαδή τα Label1, Button1, Button2.

Επίσης τα Button1 και Button3 θα έχουν σαν background Image το 1.png ενώ το Button2 το 2.png. Επίσης το αντικείμενο Sound1 θα έχει σαν πηγή ήχου το CoinFlip.mp3.

Κατά την εκκίνηση της εφαρμογής λοιπόν ο χρήστης θα μπορεί να πατήσει ένα από τα Button1, Button2 για να επιλέξει γράμματα ή κορώνα.

Στον block κώδικα θα δημιουργούνται αρχικά 2 μεταβλητές `epilogi`, `nomisma` που θα αρχικοποιούνται σε 0

**Δ1.** Όταν πατηθεί το Button1:

- α. θα εξαφανίζεται το Button2
- β. το Label1 θα εμφανίζει το κείμενο "Επέλεξες"
- γ. το Button1 θα απενεργοποιείται
- δ. τα Label2 και Button3 θα εμφανίζονται
- ε. Η μεταβλητή `epilogi` θα παίρνει την τιμή 1
- στ. Αντίστοιχες ενέργειες θα γίνονται όταν πατηθεί το Button2 (υπόδειξη: προσοχή πώς θα αλλάξουν οι ενέργειες α, γ, ε)

**Δ2.** Όταν πατηθεί το Button3:

- α. Θα ακούγεται ο ήχος του νομίσματος
- β. Το Button3 θα απενεργοποιείται.
- γ. Η μεταβλητή nomisma θα παίρνει ως τιμή μια τυχαία τιμή μεταξύ του 1 και του 2.
- δ. Η εικόνα που θα εμφανίζεται στο Button3 θα είναι η 1.png ή 2.png ανάλογα με την τιμή της μεταβλητής nomisma αντίστοιχα αν είναι 1 ή 2.
- ε. Θα ελέγχεται αν η μεταβλητή nomisma ταυτίζεται με την επιλογή και: αν ναι θα εμφανίζεται το Label3 με το μήνυμα "Κέρδισες" και θα ακούγεται ο ήχος success.wav (αντικείμενο Sound2)  
διαφορετικά θα εμφανίζεται το Label3 με το μήνυμα "Έχασες" και θα ακούγεται ο ήχος failure.mp3 (αντικείμενο Sound2)
- στ. θα εμφανίζεται το Button4

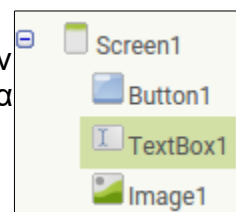
**Δ3.** Όταν πατηθεί το Button4:

- α. Θα ενεργοποιούνται τα Button1, Button2, Button3
- β. Θα εμφανίζονται τα Button1, Button2
- β. Η ιδιότητα Text του Label3 θα παίρνει την τιμή " "
- γ. Θα εξαφανίζονται τα αντικείμενα Label2, Label3, Button3, Button4

## Εξετάσεις 2014-2015

### Θέμα Γ

Επιλέξτε τη σωστή απάντηση στα Γ1-Γ5 όσον αφορά το τι λειτουργία επιτελούν τα δοσμένα κομμάτια κώδικα. Τα αντικείμενα που εμπλέκονται στα δοσμένα κομμάτια κώδικα φαίνονται στη διπλανή εικόνα.



Γ1.

- α) Κατά την φόρτωση της οθόνης η εικόνα Image1 ενεργοποιείται
- β) Κατά την φόρτωση της οθόνης η μεταβλητή TextBox1 παίρνει τιμή false
- γ) Κατά την φόρτωση της οθόνης στην μεταβλητή Enabled προστίθεται η τιμή false
- δ) Κατά την φόρτωση της οθόνης το πλαίσιο κειμένου TextBox1 απενεργοποιείται

Γ2.

- α) Κάθε φορά που πατιέται το Button1 η μεταβλητή fores παίρνει την τιμή 1
- β) Κάθε φορά που πατιέται το Button1 δημιουργείται μία επιπρόσθετη μεταβλητή με την τιμή της fores
- γ) Κάθε φορά που πατιέται το Button1, η τιμή της μεταβλητής fores αυξάνεται κατά 1
- δ) Τίποτα από τα παραπάνω

Γ3.

- α) Η μεταβλητή score αυξάνεται κατά το άθροισμα  $1+2+3+4+5+6$
- β) Η τιμή της μεταβλητή score ελέγχεται αν ανήκει στο διάστημα  $[1,6]$
- γ) Η μεταβλητή score αυξάνεται κατά έναν τυχαίο αριθμό στο  $[1,6]$  που μπορεί να δείχνει μια ζαριά
- δ) Η μεταβλητή score διαιρείται με το γινόμενο  $1*2*3*4*5*6$

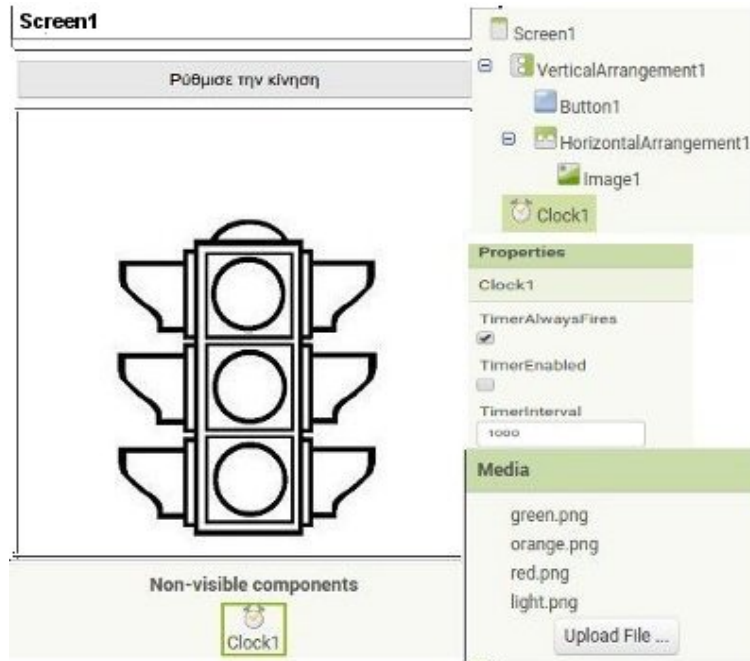
Γ4.

- α) Στο πλαίσιο κειμένου TextBox1 εμφανίζεται η τιμή της μεταβλητής score
- β) Η μεταβλητή score παίρνει την τιμή που περιέχεται στο πλαίσιο κειμένου TextBox1
- γ) Στο πλαίσιο κειμένου TextBox1 εμφανίζεται η μέγιστη δυνατή τιμή
- δ) Τίποτα από τα παραπάνω

Γ5.

- α) Αν η μεταβλητή score έχει τιμή πάνω από 100 τότε εμφανίζεται στην οθόνη
- β) Αν η μεταβλητή score έχει τιμή πάνω από 100 τότε η τιμή του score αλλάζει και η νέα του τιμή είναι κάτω από 100
- γ) Αν η μεταβλητή score έχει τιμή πάνω από 100 τότε εμφανίζεται στην οθόνη το 200
- δ) Αν η μεταβλητή score έχει τιμή πάνω από 100 τότε στο πλαίσιο κειμένου εμφανίζεται μια τιμή κάτω από 100

## Θέμα Δ



Θέλουμε να φτιάξουμε σε AppInventor μια εφαρμογή-προσομοίωση ενός φαναριού κυκλοφορίας το οποίο με το πάτημα ενός κουμπιού να ανάβει πράσινο για 40 δευτερόλεπτα, έπειτα πορτοκαλί για 3 δευτερόλεπτα κι έπειτα κόκκινο για 30 δευτερόλεπτα κοκ. Η σχεδίαση της εφαρμογής φαίνεται στην επόμενη εικόνα. Καλείστε να φτιάξετε τον block κώδικα που υλοποιεί τα Δ1-Δ3.

**Δ1.** Δημιουργεί τη μεταβλητή *time* και την αρχικοποιεί σε 0. (μον. 2)

**Δ2.** Όταν πατηθεί το κουμπί ενεργοποιεί το ρολόι, μηδενίζει τη μεταβλητή *time* και εμφανίζει την εικόνα *green.png* (μον. 7)

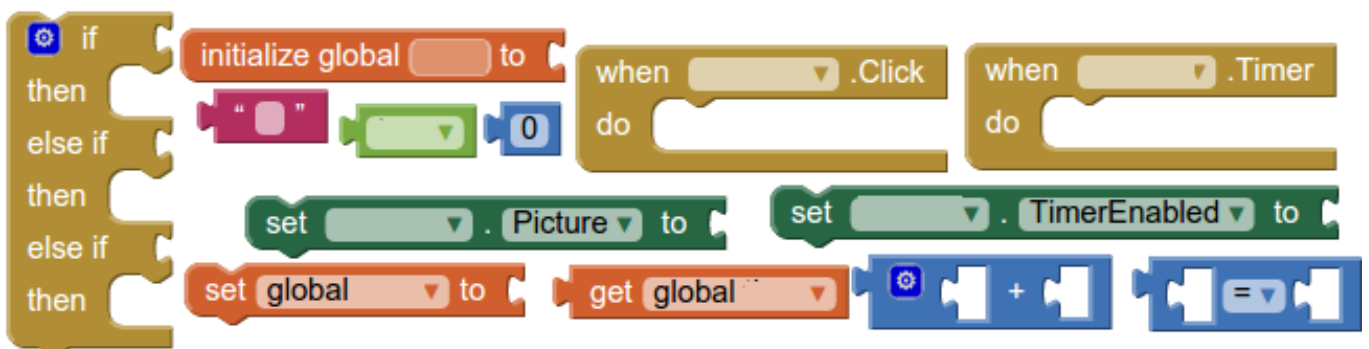
**Δ3.** Κάθε φορά που περνάει 1 δευτερόλεπτο: (μον. 3)

α) Θα αυξάνει η μεταβλητή *time* κατά 1. (μον. 3)

β) Θα ελέγχει τη μεταβλητή *time* και αν είναι 40 τότε θα εμφανίζει την εικόνα *orange.png*, αλλιώς αν είναι 43 τότε θα εμφανίζει την εικόνα *red.png*, αλλιώς αν είναι 73 θα μηδενίζει τη μεταβλητή *time* και θα εμφανίζει την εικόνα *green.png* (μον. 10)

**(Μονάδες 25)**

**Σημ.:** Παρακάτω φαίνονται τα blocks που απαιτούνται τα οποία πρέπει να συνενωθούν αφού συμπληρωθούν κατάλληλα όπου έχουν κενά. Μπορεί να χρησιμοποιηθούν πάνω από 1 φορές.





## Παραδείγματα Ασκήσεων

### Διαγώνισμα Β' τετρ. 2014-2015

#### ΘΕΜΑ Α

#### A1. Τι κάνουν οι παρακάτω εικονιζόμενες εντολές; Επιλέξτε το σωστό (μον.6)

initialize global x to 0

- α) Μπορεί να εκτελεστεί όταν υπάρχει ένα γεγονός (π.χ. πάτημα ενός κουμπιού) και θέτει τη μεταβλητή x να έχει την τιμή 0.
- β) Δημιουργεί τη μεταβλητή x και βάζει σε αυτήν αρχική τιμή 0 κατά την έναρξη εκτέλεσης της εφαρμογής.
- γ) Αποδίδει το όνομα x στον αριθμό 0. Όπου χρησιμοποιηθεί έπειτα το x στην εφαρμογή θα έχει τιμή 0.
- δ) Δημιουργεί τη μεταβλητή με το όνομα global και της δίνει την τιμή 0.

set global w to get global x

- α) Βάζει την τιμή της μεταβλητής w στη μεταβλητή x
- β) Βάζει την τιμή της μεταβλητής x στη μεταβλητή w
- γ) Βάζει την τιμή της μεταβλητής set στη μεταβλητή get
- δ) Εμφανίζει τις τιμές των μεταβλητών w και x

set global w to get global w + 1

- α) Αυξάνει την τιμή της μεταβλητής w κατά 1
- β) Εμφανίζει την τιμή της μεταβλητής w αφού την αυξήσει κατά 1
- γ) Εμφανίζει την τιμή της μεταβλητής w και κατόπιν την αυξάνει κατά 1
- δ) Εμφανίζει στην οθόνη το label με όνομα w που έχει ως τιμή την τιμή της μεταβλητής w αυξημένη κατά 1

set Clock1 . TimeInterval to 500

- α) Το ρολόι clock1 θα παράγει γεγονότα κάθε 500 δευτερόλεπτα
- β) Το ρολόι clock1 θα παράγει γεγονότα κάθε μισό δευτερόλεπτο
- γ) Το ρολόι clock1 θα παράγει γεγονότα κάθε 500 χτύπους του ρολογιού του επεξεργαστή του κινητού
- δ) Το ρολόι του επεξεργαστή θα παράγει γεγονότα κάθε μισή ώρα

set Image1 . Visible to false

- α) Θέτει τη μεταβλητή Image1 να είναι ψευδής
- β) Θέτει τη μεταβλητή Visible να μην έχει την τιμή Image1
- γ) Εμφανίζει την εικόνα που αντιστοιχεί στο αντικείμενο Image1
- δ) Αποκρύβει την εικόνα που αντιστοιχεί στο αντικείμενο Image1.

set Label1 . Text to "exam"

- α) Βάζει στη μεταβλητή Label1 την τιμή exam
- β) Θέτει την ιδιότητα Label1 του αντικειμένου με όνομα Text να είναι ίση με exam
- γ) Θέτει την ιδιότητα Text του αντικειμένου Label1 να έχει την τιμή exam.
- δ) Βάζει στη μεταβλητή exam την τιμή Label1Text

**A2. Τι κάνουν τα παρακάτω σύνολα εντολών; Επιλέξτε το σωστό.** (μον. 9)

```
initialize global x to true
when Button1 .Click
do
  set Image1 . Visible to get global x
  set Image2 . Visible to not get global x
  set global x to not get global x
```

Όταν πατηθεί 2014 φορές το κουμπί Button1, ποια εικόνα θα είναι ορατή;

- α) Η εικόνα που αντιστοιχεί στο αντικείμενο Image1
- β) Η εικόνα που αντιστοιχεί στο αντικείμενο Image2
- γ) Καμία από τις δύο
- δ) Και οι δύο

```
initialize global w to 0
when Button3 .Click
do
  set global w to get global w + 1
```

- α) Εμφανίζουν στην οθόνη του κινητού τη μεταβλητή w αυξημένη κατά 1 όταν πατηθεί το Button3
- β) Εμφανίζουν ως κείμενο πάνω στο κουμπί Button3 την τιμή της μεταβλητής w αυξημένη κατά 1
- γ) Θέτουν την τιμή της μεταβλητή w να είναι το πόσες φορές έχει πατηθεί το κουμπί Button3.

```
initialize global no to 0
when Screen1 .Initialize
do
  set Clock1 . TimerInterval to 100
when Clock1 .Timer
do
  set global no to random integer from 1 to 49
  set Label1 . Text to get global no
```

- α) Δείχνει στην οθόνη του κινητού στη θέση του label1 έναν τυχαίο ακέραιο αριθμό από το 1 μέχρι το 49 κάθε 0.1 sec.
- β) Προσθέτει στη μεταβλητή no τυχαίους ακεραίους αριθμούς από το 1 μέχρι το 49 κάθε 0.1 sec
- γ) Κάθε 100sec εμφανίζει ότι κείμενο περιέχεται στη μεταβλητή no
- δ) Ό,τι περιέχεται στο Label1 εμφανίζεται 49 φορές μέσα σε 100 sec

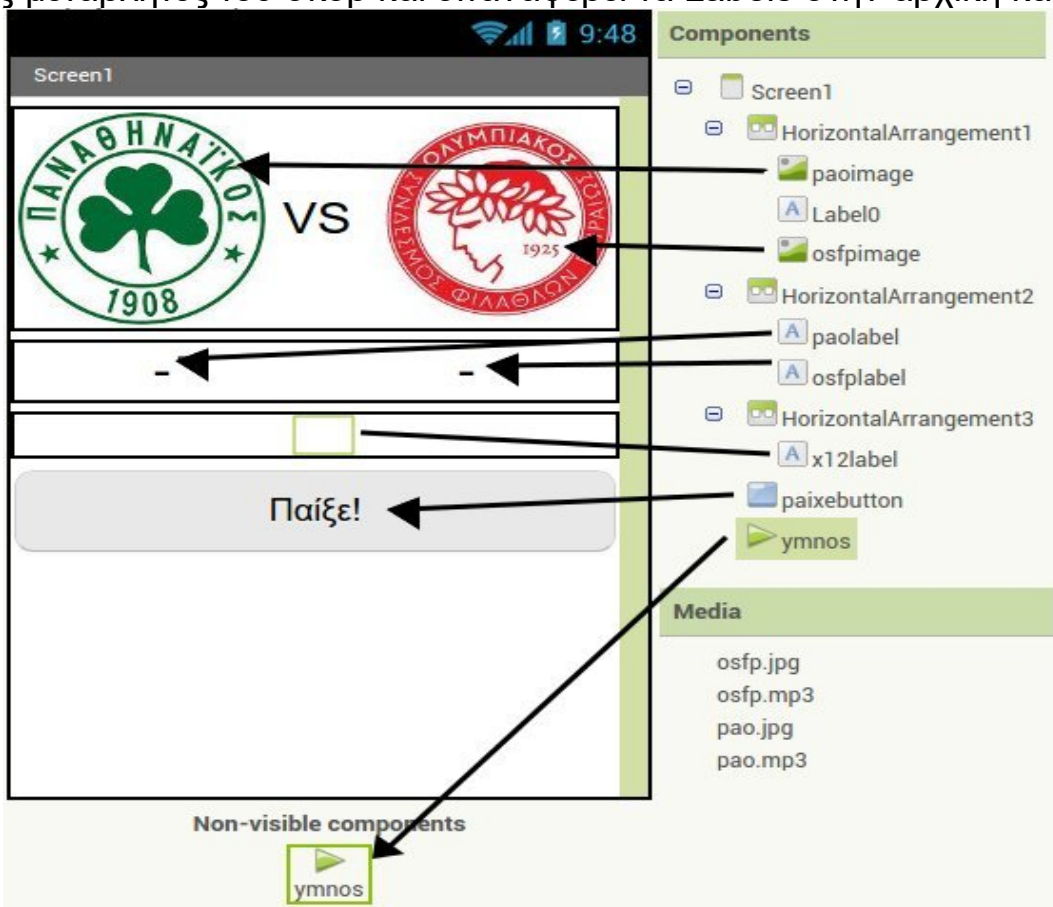
## ΘΕΜΑ Β (μον. 5)

### B1. Δίνεται παρακάτω η σχεδίαση μιας εφαρμογής η οποία:

Όταν πατηθεί το κουμπί δημιουργείται ένα τυχαίο σκορ, δηλαδή τα γκολ που σημείωσε κάθε ομάδα αποθηκεύονται σε 2 μεταβλητές (μέχρι 5 γκολ η κάθε ομάδα). Το σκορ εμφανίζεται ακριβώς κάτω από τις εικόνες των ομάδων εκεί όπου αρχικά υπάρχουν οι δύο παύλες ( - - ).

Ελέγχονται οι 3 περιπτώσεις για το σκορ και αν κέρδισε ο Παναθηναϊκός τότε εμφανίζεται κάτω από το σκορ το μήνυμα “Σημειώσατε 1” και ακούγεται ο ύμνος του Παναθηναϊκού, αλλιώς αν κέρδισε ο Ολυμπιακός τότε εμφανίζεται κάτω από το σκορ το μήνυμα “Σημειώσατε 2” και παίζει ο ύμνος του Ολυμπιακού, αλλιώς απλά εμφανίζεται το μήνυμα “Σημειώσατε X : ισοπαλία”.

Παρατεταμένο πάτημα του κουμπιού σταματάει τον ήχο που πιθανά ακούγεται, μηδενίζει τις μεταβλητές του σκορ και επαναφέρει τα Labels στην αρχική κατάσταση.



Συνδυάστε τα παρακάτω blocks αφού τα συμπληρώσετε κατάλληλα όπου έχουν κενό και δημιουργήστε έτσι τον κώδικα της εφαρμογής. Στην παρένθεση φαίνεται ο αριθμός των φορών που θα χρησιμοποιηθεί καθένα.

initialize global  to  (2)

set  to  (4)    get  (6)

random integer from  to  (2)

set  . Text  to  (8)

set  . Source  to  (2)

if  (1)

then  (8)

else if  (1)

then  (2)

else  (2)

when  . Click (1)

do  (1)

when  . LongClick (1)

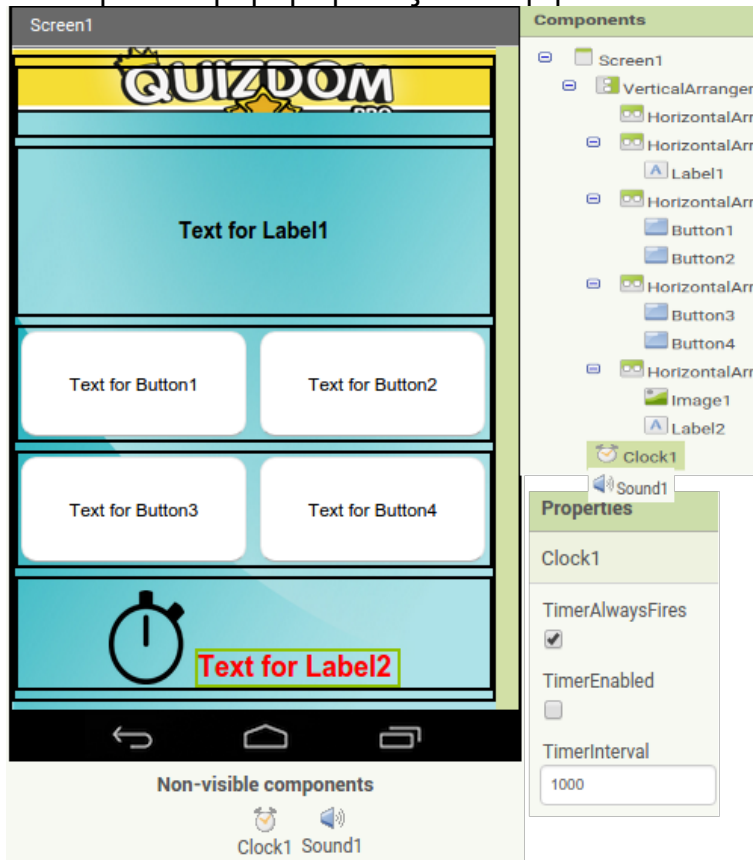
do  (1)

call  . Stop (1)

call  . Start (2)

Θέμα Δ

Θέλουμε να προσομοιώσουμε τη λειτουργία μιας ερώτησης γνωστού παιχνιδιού γνώσεων για Android με τη χρήση του Appinventor. Η ερώτηση που θέλουμε να προσομοιώσουμε είναι: “Ποια είναι η πρωτεύουσα της Ιταλίας;” και οι τέσσερις πιθανές απαντήσεις είναι “Βενετία”, “Μιλάνο”, “Ρώμη”, “Φλωρεντία” με σωστή απάντηση προφανώς το “Ρώμη”.



Η σχεδίαση της εφαρμογής φαίνεται στην επόμενη εικόνα.

Καλείστε να φτιάξετε τον block κώδικα που υλοποιεί τα παρακάτω:

**Δ1.** Φτιάχνει τη μεταβλητή **xronos** και της δίνει την τιμή 20 (μον. 2)

**Δ2.** Κατά της φόρτωση της οθόνης:

- α) Εμφανίζει την ερώτηση στο Label1 (μον. 2)
- β) θέτει τις πιθανές απαντήσεις σε καθένα από τα κουμπιά Button1, ... Button4 (μον. 4)
- γ) θέτει την τιμή της μεταβλητής χρόνος στο Label2 (μον. 2)
- δ) ενεργοποιεί το ρολόι Clock1 (μον. 2)

**Δ3.** Μειώνει τη μεταβλητή χρόνος κατά 1 όποτε περνάει 1 δευτερόλεπτο και την εμφανίζει στο Label2 (μον. 4).

**Δ4.** Αν πατηθεί το σωστό κουμπί τότε αυτό πρασινίζει και παίζει ο ήχος του αντικειμένου Sound1, ενώ αν πατηθεί ένα λάθος κουμπί αυτό κοκκινίζει και πρασινίζει το αντίστοιχο σωστό. Σε κάθε περίπτωση απενεργοποιείται το ρολόι (μον. 9)

(Μονάδες 25)

**Σημ.:** Παρακάτω φαίνονται τα blocks που απαιτούνται τα οποία πρέπει να συνενωθούν αφού συμπληρωθούν κατάλληλα όπου έχουν κενά. Μπορεί να χρησιμοποιηθούν πάνω από 1 φορές.

