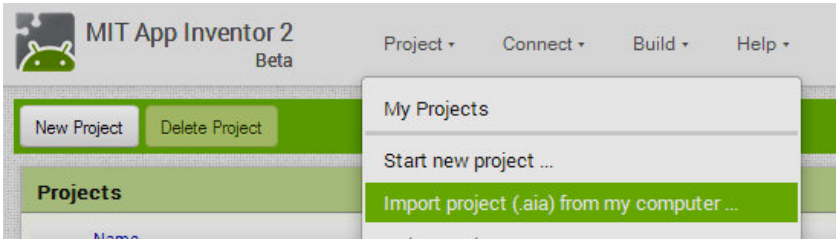
Σε αυτό το φύλλο εργασίας θα ολοκληρώσουμε, βήμα - βήμα μια **μισοτελειωμένη εφαρμογή** για Android συσκευές (κινητά ή tablets).

Πρόκειται για ένα παιχνίδι, όπου ο ήρωας μας είναι μια μαϊμού, η οποία προσπαθεί να πιάσει όσες πιο πολλές μπανάνες μπορεί, έχοντας μόνο 3 ζωές.

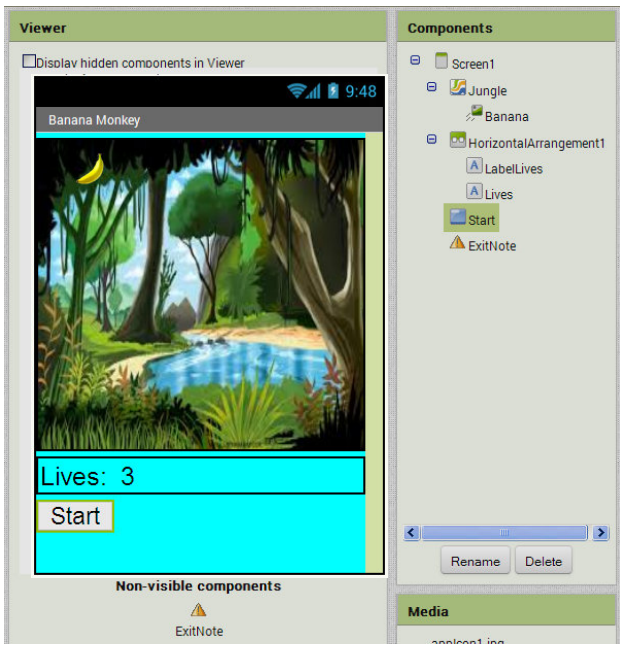
Ορισμένα κομμάτια της εφαρμογής έχουν ήδη σχεδιαστεί και προγραμματιστεί και εμείς καλούμαστε να συμπληρώσουμε τα υπόλοιπα. Τα ολοκληρωμένα κομμάτια με τις αναλυτικές οδηγίες τους παραθέτονται στις τελευταίες σελίδες του φύλλου εργασίας.

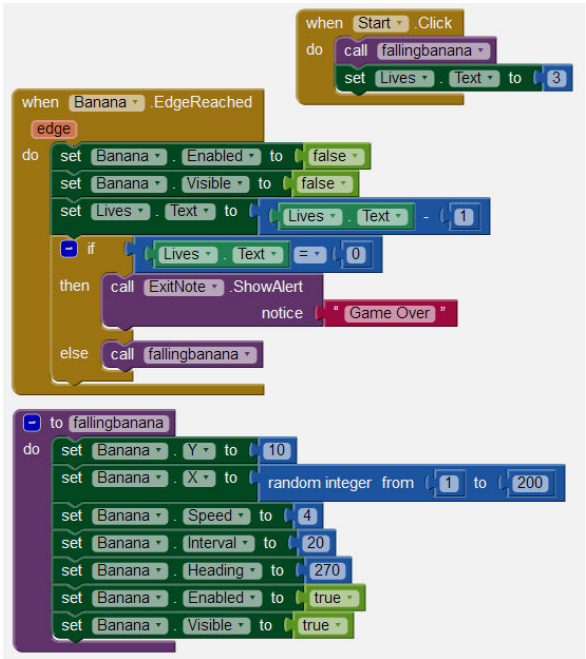
## **Δραστηριότητα 1** Εισαγωγή project από τον υπολογιστή

Ξεκινώντας, πληκτρολογούμε το site του AppInventor και δίνοντας το όνομα χρήστη και τον κωδικό πρόσβασης, εισερχόμαστε στον λογαριασμό μας.
Στην συνέχεια, **εισάγουμε** στο προγραμματιστικό περιβάλλον **την μισοτελειωμένη εφαρμογή**, την οποία **θα βρούμε στην επιφάνεια εργασίας** (με όνομα **BananaGame.aia**).




Αμέσως βλέπουμε τα **components** που υπάρχουν ήδη τοποθετημένα στην οθόνη (**Viewer**) που αποτελεί την διεπαφή – interface της εφαρμογής στην ενότητα **Designer**, και εάν επιλέξουμε την ενότητα **Blocks** *(κουμπιά πάνω δεξιά)* βλέπουμε τα blocks με τον κώδικα για κάθε component.



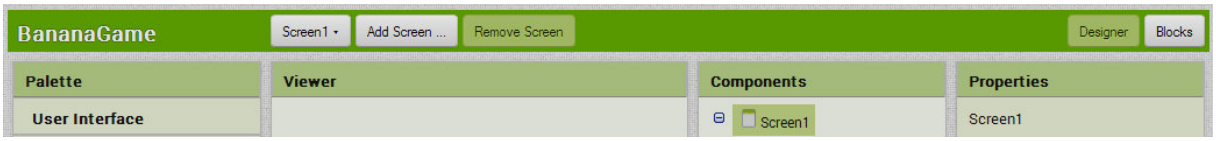



*Για να δείτε πως τοποθετήθηκαν τα αρχικά components στο Viewer και τον τρόπο προγραμματισμού με blocks ανατρέξτε στις τελευταίες σελίδες.*

## Δραστηριότητα 2 Προσθήκη των υπόλοιπων components στην ενότητα Designer

Στο στάδιο αυτό θα προσθέσουμε ένα sprite με την εικόνα της μαϊμούς, δύο labels που θα αποθηκεύουν το σκορ και ένα button για έξοδο από την εφαρμογή.

Η ενότητα Designer αποτελείται από τα τμήματα: **Palette**, **Viewer**, **Components** και **Properties**.




Επιστρέφουμε στο **Designer** και από το **Palette** επιλέγουμε από την ομάδα αντικειμένων **Drawing and Animation** ένα **ImageSprite** και το σέρνουμε στην περιοχή του **Viewer** (πάνω στην εικόνα της ζούγκλας).
Στο τμήμα **Components** πατάμε το κουμπί **Rename** (βρίσκεται στο κάτω μέρος) και μετονομάζουμε το ImageSprite σε **"Monkey"**.
Τέλος στο τμήμα **Properties** επιλέγουμε την ιδιότητα **Picture** και φορτώνουμε το αρχείο (**upload file**) με την εικόνα της μαϊμούς **"monkeyWin.gif"**, που θα βρούμε στην **επιφάνεια εργασίας**, καθώς και αλλάζουμε τιμές στις ιδιότητες X και Y, που δείχνουν τις συντεταγμένες (**X:120**, **Y:220**).
Ομοίως δουλεύουμε και για τα δύο labels και το button.

| από την ομάδα | μεταφέρουμε το component | του δίνουμε το όνομα (Rename) | μεταβάλουμε τις ιδιότητες |
|---|---|---|---|
| Drawing and Animation | ImageSprite *(πάνω στην ζούγκλα)* | Monkey | Picture: monkeyWin.gif<br>X: 120, Y: 220 |
| User Interface | Button *(κάτω από το κουμπί Start)* | Exit | FontSize: 28<br>Text: Exit |
| User Interface | Label *(δίπλα από το "Lives: 3")* | LblScore | FontSize: 28<br>Text: Score: |
| User Interface | Label *(δίπλα από το "Score: ")* | N_Score | FontSize: 28<br>Text: 0 |

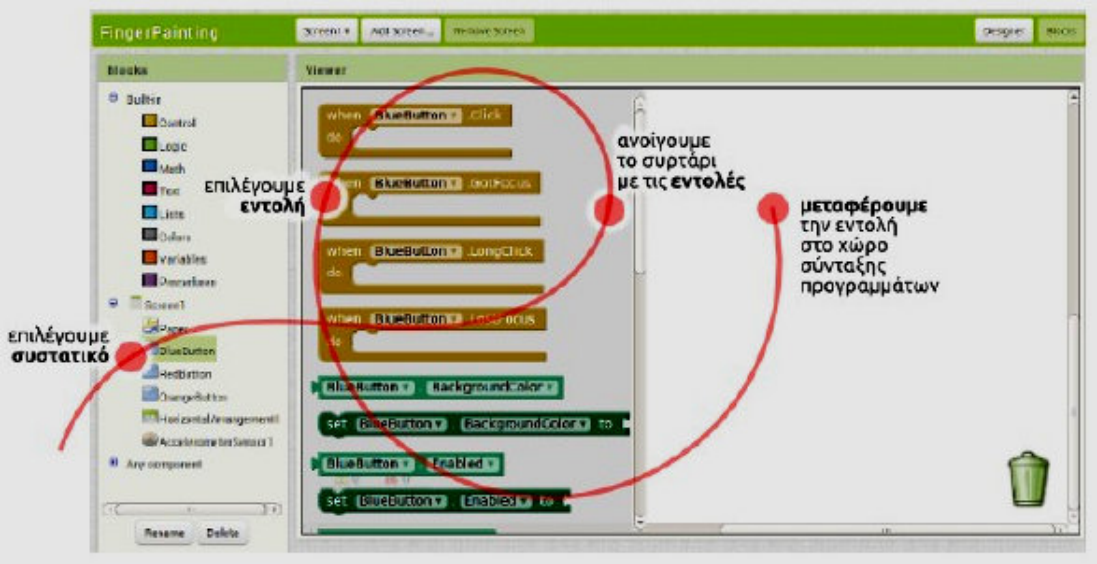
## Δραστηριότητα 3 Προγραμματισμός στην ενότητα Blocks

Στο στάδιο αυτό θα δουλέψουμε με τα blocks της εφαρμογής για να ολοκληρώσουμε τον κώδικα του παιχνιδιού.

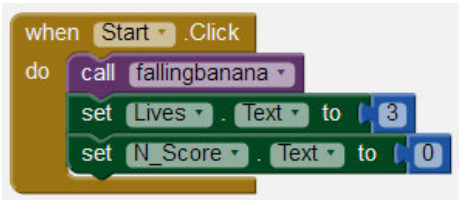
Μεταβαίνουμε λοιπόν στο **Blocks** (το κουμπί μετάβασης βρίσκεται πάνω δεξιά) για να συσχετίσουμε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσουμε, προσθέτοντας τις κατάλληλες εντολές.

Στα αριστερά της οθόνης διακρίνουμε κάποιες ομάδες πλακιδίων (blocks). Πρώτα βρίσκουμε τα ενσωματωμένα πλακίδια (**Built-in blocks**). Στη συνέχεια, βλέπουμε τα **blocks των components** που έχουμε προσθέσει στην εφαρμογή μας.

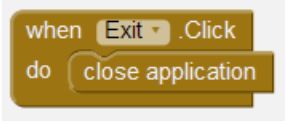
*Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή πρακτικά συναρμολογούμε **blocks εντολών**. Για κάθε block που θέλουμε να προσθέσουμε ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης, ανοίγει το αντίστοιχο **«συρτάρι» με τις διαθέσιμες εντολές**, αναζητούμε και επιλέγουμε το block που χρειαζόμαστε και το **σέρνουμε στον χώρο σύνταξης των προγραμμάτων (Viewer)**.*




Αρχικά, θα συμπληρώσουμε στο **ήδη υπάρχον block μέσα στο Viewer "when Start.Click"** *(block για το συμβάν πατήματος κουμπιού),* την αρχικοποίηση του component **"N_Score"** (ιδιότητα **Text**), όπως έχει ήδη γίνει για το component "Lives", ως εξής:
**Επιλέγουμε (κάνουμε κλικ) το** component **N_Score** από το τμήμα blocks αριστερά της οθόνης. Αυτόματα ανοίγει η συρταριέρα και αναζητούμε την εντολή **"set N_Score.text to"**, την οποία σέρνουμε και **κουμπώνουμε** μέσα στο block **"when Start.Click"**, έτσι ώστε να προστεθεί στα άλλα δύο.
Από τα **Built-in Blocks** επιλέγουμε το **Math** και σέρνουμε το πρώτο πλακίδιο με το **"0"** για να **κουμπώσει** δίπλα στο **"set N_Score.text to"**.
Έτσι το block παίρνει τη μορφή:



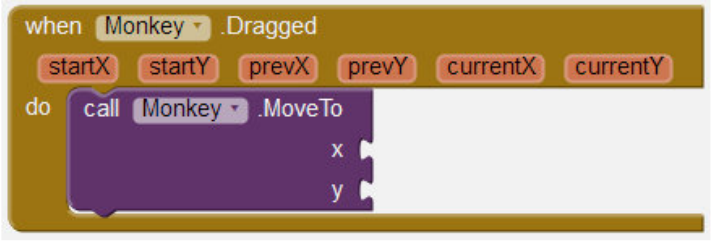

Ομοίως προγραμματίζουμε το συμβάν που θέλουμε να εκτελεστεί μόλις πατήσουμε το κουμπί **"Exit"** (δηλ. να τερματιστεί η εφαρμογή).
**Επιλέγουμε** από το τμήμα blocks, το component **"Exit"**. Ανοίγει η συρταριέρα, επιλέγουμε το block **"when Exit.Click"**, και το σέρνουμε στο **Viewer**.
Από τα **Built-in Blocks** επιλέγουμε το **Control**, βρίσκουμε την εντολή **"close application"** και την **κουμπώνουμε** μέσα στο **"when Exit.Click"**.




## Δραστηριότητα 4 — Κίνηση της μαϊμούς

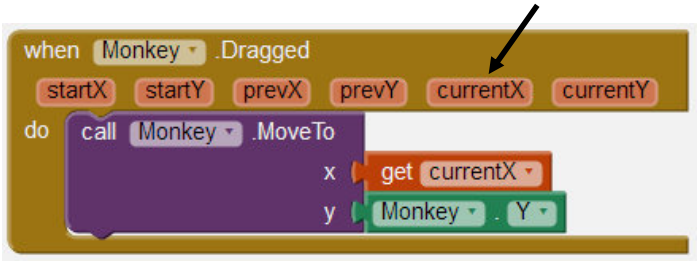
Ακουμπάμε την μαϊμού με το δάκτυλό μας και το σέρνουμε πάνω στην οθόνη (δεξιά - αριστερά) και το ακολουθεί η μαϊμού.

**Επιλέγουμε το** component **"Monkey"**, ανοίγει το αντίστοιχο συρτάρι και μεταφέρουμε το κατάλληλο για το συμβάν κίνησης block **"when monkey.Dragged"**.
Επιλέγουμε πάλι το component **"Monkey"**, εντοπίζουμε το block **"call Monkey.MoveTo"** (procedure – μωβ χρώμα) και το **κουμπώνουμε** μέσα στο block **"when monkey.Dragged"**.




*Παρατήρηση: Σημειώστε ότι το block "when monkey.Dragged" επιστρέφει κάποιες τιμές: Τα αρχικά x και y, τα προηγούμενα x και y και τα τωρινά x και y (μεταβλητές). Αυτά τα νούμερα προσδιορίζουν που ακριβώς είχε ακουμπήσει το δάχτυλό του ο χρήστης στην οθόνη την αμέσως προηγούμενη στιγμή (ή την αρχική στιγμή όταν πρωτοξεκίνησε η εφαρμογή), καθώς και το σημείο που ακουμπάει τώρα το δάκτυλό του ο χρήστης, έτσι ώστε να μπορέσει το sprite να πάει από το προηγούμενο σημείο στο τωρινό, ακολουθώντας στην ουσία το δάχτυλο του χρήστη.*

Πως θα προσδιορίσουμε που θα πάει η μαϊμού; Βάζοντας τον δείκτη του ποντικιού πάνω στο **"currentX"** του block **"when Monkey.Dragged"** εμφανίζεται η εντολή **"get currentX"**, την οποία σέρνουμε και κουμπώνουμε δίπλα στο **X** (η τωρινή τιμή του x).
Στην συνέχεια ανατρέχουμε στο component **"Monkey"**, ανοίγει το συρτάρι, επιλέγουμε το block **"Monkey.Y"** (έχει πράσινο χρώμα), το σέρνουμε και το κουμπώνουμε στο **Y**.



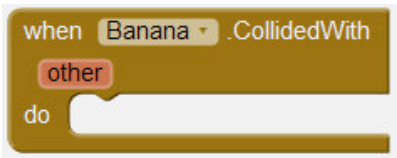

*Σέρνουμε το δάχτυλό μας δεξιά ή αριστερά και η μαϊμού ακολουθεί την κίνησή μας πάνω στον άξονα των x, ενώ παραμένει σταθερή με μια συγκεκριμένη τιμή πάνω στον άξονα y (πάνω - κάτω).*

## **Δραστηριότητα 5** Μέτρηση του σκορ

Σε αυτό το στάδιο θα ασχοληθούμε με το πώς αλληλεπιδράνε τα sprites (μαϊμού, μπανάνα) μεταξύ τους.

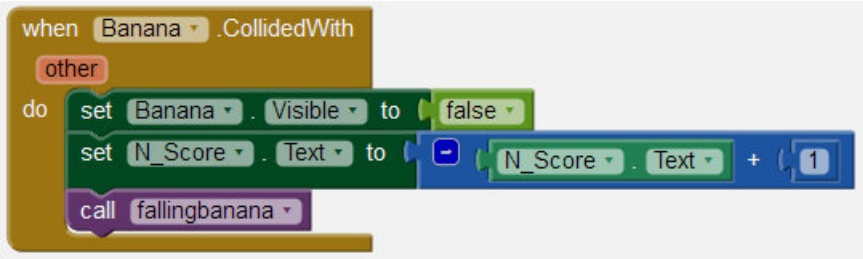
Όταν η μαϊμού πιάσει μια μπανάνα, το σκορ της αυξάνεται κατά ένα, εξαφανίζεται η μπανάνα που έπιασε και καλεί την procedure "fallingbanana" για να εμφανίσει την επόμενη σε τυχαία θέση.

Για να το πετύχουμε αυτό **επιλέγουμε** το component **"Banana"** και βγάζουμε έξω το κατάλληλο για το συμβάν σύγκρουσης block **"when Banana.CollidedWith"**, το οποίο ενεργοποιείται όταν η φιγούρα της μαϊμούς και της μπανάνας αγγίξουν η μία την άλλη.

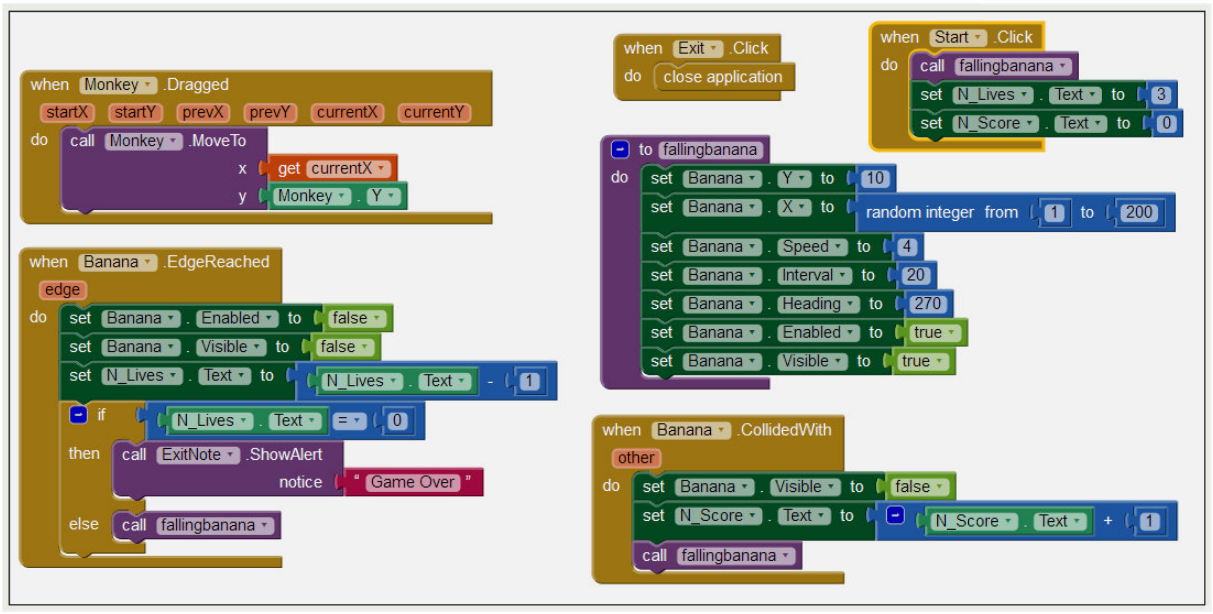



Κουμπώνουμε σε αυτό το block την εντολή για να αλλάξει η ιδιότητα **"visible"** της μπανάνας, την οποία θα βρούμε στο component **"Banana"** (**"set Banana.Visible to"**) και στην οποία προσθέτουμε την τιμή **"false"** που θα βρούμε στο τμήμα **Logic** των **Built-in blocks**.
Στην συνέχεια, αυξάνουμε την ιδιότητα **"Text"** του σκορ κατά ένα, το οποίο θα το βρούμε στο component **"N_Score"** (**"set N_Score.Text to"**).
Συμπληρώνουμε την **πράξη της πρόσθεσης** που θα βρούμε στο **Math** των **Built-in blocks**, με

την οποία προσθέτουμε το πλακίδιο **"N_Score.Text"**, το οποίο θα βρούμε στο component **"N_Score"** με το πρώτο πλακίδιο στο **Math**, όπου σβήνουμε το **"0"** και γράφουμε **"1"**.
Τέλος κουμπώνουμε και την **procedure "fallingbanana"** (θα την βρούμε στο **Procedures** των **Built-in blocks**) για να εμφανίσει μια άλλη μπανάνα.




Όλος ο κώδικας της εφαρμογής φαίνεται στο παρακάτω σχήμα:




*Είστε ελεύθεροι να πειραματιστείτε π.χ. με την ταχύτητα που πέφτει η μπανάνα ή εάν θέλετε να προσθέσετε κι άλλο ένα sprite (π.χ. ένα κεράσι), που όταν το πιάνει η μαϊμού θα αφαιρείται ένας πόντος.*
*Άλλη ιδέα είναι να δημιουργήσετε πίστες (30 δευτερολέπτων η καθεμιά) αυξανόμενης δυσκολίας, ή ό,τι άλλο μπορείτε να φανταστείτε!!!*

Στις επόμενες σελίδες θα βρείτε πληροφορίες για το πώς προσθέσαμε τα components και πως προγραμματίσαμε τον κώδικα, για όλα εκείνα που δεν αναφέρθηκαν στις Δραστηριότητες.

## Η εφαρμογή από την αρχή

Εάν θέλουμε να δημιουργήσουμε το παιχνίδι από την αρχή μόνοι μας, δεν έχουμε παρά να εισέλθουμε στον λογαριασμό μας στο AppInventor και από το μενού να επιλέξουμε:
**Project→ Start new project…** και να ονοματίσουμε το νέο μας project.

Αρχικά, βρισκόμαστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη **διεπαφή (interface)** της εφαρμογής μας η οποία φαίνεται στο **Viewer**, προσθέτοντας τα απαραίτητα **αντικείμενα (components)** και ορίζοντας **ιδιότητες (properties)** για αυτά.

### Σχεδίαση της διεπαφής (interface) στην ενότητα Designer

Υπενθυμίζουμε ότι η ενότητα Designer αποτελείται από τα τμήματα: **Palette**, **Viewer**, **Components** και **Properties**.

Το μοναδικό component μέχρι στιγμής, είναι η **οθόνη (Screen1)**. Προτού προσθέσουμε άλλα components, κάνουμε ορισμένες απαραίτητες τροποποιήσεις στις ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties** (βρίσκεται δεξιά):
Επιλέγουμε το χρώμα του φόντου (**BackgroundColor**) να είναι γαλάζιο (Cyan), ενώ μπορούμε να φορτώσουμε μια εικόνα που μας αρέσει, για να αλλάξουμε την εικόνα με την οποία εμφανίζεται η εφαρμογή στην συσκευή μας (πατάμε στο **Icon** και επιλέγουμε **upload file** και μετά αναζήτηση – appIcon1.jpg).
Η ιδιότητα **Scrollable** να μην είναι επιλεγμένη. Στον τίτλο (**Title**) προσθέτουμε το όνομα που προτιμάμε για την εφαρμογή (στην περίπτωσή μας Banana Monkey).

### Προσθήκη του καμβά

Στο βήμα αυτό θα προσθέσουμε το component που θα αποτελέσει το χώρο όπου θα κινείται η μαϊμού και η μπανάνα και θα καθορίσουμε τις ιδιότητες του χώρου αυτού.

Ο **καμβάς (Canvas)** είναι μια ορθογώνια επιφάνεια, εντός της οποίας μπορούμε να χειριζόμαστε **φιγούρες (Sprites)** (π.χ. να κινούνται η μαϊμού και η μπανάνα) ή να σχεδιάζουμε αγγίζοντάς την.

Από το **Palette** (στα αριστερά) λοιπόν, επιλέγουμε από την ομάδα αντικειμένων **Drawing and Animation**, το component **Canvas**, και το τοποθετούμε στην οθόνη.
Καλό θα είναι να ονομάζουμε τα components που χρησιμοποιούμε με τέτοιο τρόπο, ώστε να τα αναγνωρίζουμε ευκολότερα. Στο τμήμα **Components** πατάμε το κουμπί **Rename** (βρίσκεται στο κάτω μέρος) και μετονομάζουμε το Canvas σε **"Jungle"**.

Τέλος, μεταβάλλουμε τις ιδιότητες του καμβά Jungle ως εξής:

| από την ομάδα | μεταφέρουμε το component | του δίνουμε το όνομα (Rename) | μεταβάλουμε τις ιδιότητες |
|---|---|---|---|
| Drawing and Animation | Canvas *(μέσα στην οθόνη)* | Jungle | Width: Fill Parent<br>Height: 300 pixels<br>BackgroundImage: bckrnd.jpg |

Η επιλογή **Fill Parent** για τις ιδιότητες **Width** και **Height** του καμβά, του επιτρέπει να επεκταθεί και να καταλάβει όλο το διαθέσιμο χώρο. Έτσι, τα sprites θα μπορούν να κινούνται κατά μήκος και κατά πλάτος όλης της οθόνης.
Για να χρησιμοποιηθεί ως **φόντο** του καμβά Jungle η εικόνα με τη ζούγκλα (**bckrnd.jpg**), θα πρέπει πρώτα αυτή να «ανέβει» στο project μας. Κάνουμε κλικ στην ιδιότητα **BackgroundImage**, επιλέγουμε **Upload File...** μετά Αναζήτηση... (**Browse…**) κι επιλέγουμε το αρχείο bckrnd.jpg.

*Τα ονόματα των components πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες και πρέπει να ξεκινάνε με χαρακτήρα, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.*

## Προσθήκη sprites και λοιπών components

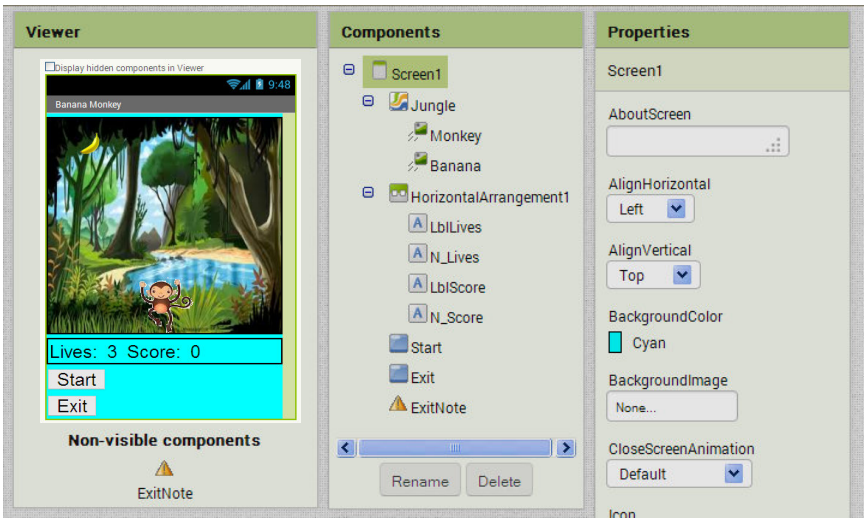
Τα **sprites** τοποθετούνται εντός του καμβά και μπορούν να αλληλεπιδρούν με τον καμβά, με άλλα sprites εντός του καμβά και φυσικά να αντιδρούν στις δικές μας ενέργειες.

Η **παλέτα** του App Inventor προσφέρει μορφοποιήσεις (**Layout**) προκειμένου να τοποθετούμε όλα τα components ομαδοποιημένα και σε μια σειρά.
Έτσι μπορούμε να σύρουμε στην οθόνη, κάτω από τον καμβά, ένα **HorizontalArrangment** component από το **Layout** και να βάλουμε μέσα τα αντικείμενα που θέλουμε.

Οπότε, με τον ίδιο τρόπο με της **Δραστηριότητας 2**, θα προσθέσουμε τα υπόλοιπα components με τα οποία δεν ασχοληθήκαμε εκεί (το **sprite "Banana"** και τα υπόλοιπα **labels** και **buttons**), αλλάζοντας τις ιδιότητες τους από το **Properties**, όπως φαίνεται στον παρακάτω πίνακα.

| από την ομάδα | μεταφέρουμε το component | του δίνουμε το όνομα (Rename) | μεταβάλουμε τις ιδιότητες |
|---|---|---|---|
| Drawing and Animation | ImageSprite *(πάνω στην ζούγκλα)* | Banana | Picture: banana1.gif<br>Width:30px, Height:30px<br>*(είναι πολύ μεγάλη, μικραίνουμε τις διαστάσεις της)* |
| Layout | HorizontalArrangment *(κάτω από τον καμβά)* | | Width: Fill Parent<br>AlignHorizontal: Left |
| User Interface | Label *(μέσα στο HorizontalArrangment)* | LblLives | FontSize: 28<br>Text: Lives: |
| User Interface | Label *(δίπλα από το "Lives: ")* | N_Lives | FontSize: 28<br>Text: 3 |
| User Interface | Button *(κάτω από το HorizontalArrangment)* | Start | FontSize: 28<br>Text: Start |
| User Interface | Notifier *(μέσα στην οθόνη, non-visible)* | ExitNote | BackgroundColor: Yellow<br>NotifierLenghth: Long |

*Το **Notifier** (ειδοποίηση) με όνομα **ExitNote** είναι ένα **Non-visible component**, το οποίο **δεν φαίνεται** κατά την διάρκεια εκτέλεσης της εφαρμογής μας, παρά μόνο όταν χάσουμε όλες τις ζωές θα εμφανίσει ένα **μήνυμα "Game Over"**.*



## Ολοκληρωμένα Blocks

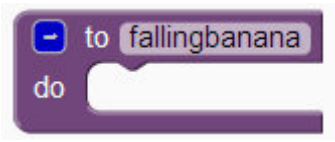
Υπενθυμίζουμε ότι για να μεταβούμε στο **Blocks** πατάμε το κουμπί που βρίσκεται πάνω δεξιά στην οθόνη). Εκεί, στα αριστερά διακρίνουμε τα ενσωματωμένα πλακίδια (**Built-in blocks**) και από κάτω τα **blocks των components** που έχουμε προσθέσει στην εφαρμογή μας. Στην μέση βρίσκεται ο χώρος σύνταξης των προγραμμάτων (**Viewer**), όπου σέρνουμε και τοποθετούμε τα **blocks ενεργειών και εντολών**.

Έχοντας ήδη αναφέρει πως προγραμματίζουμε τους κώδικες των blocks **"when Exit.Click"**, **"when Monkey.Dragged"**, **"when Banana.CollidedWith"** και το μισό **"when Start.Click"** στις **Δραστηριότητες 3, 4 και 5**, τώρα θα ασχοληθούμε με τις υπόλοιπα συμβάντα και εντολές που δεν έχουν ειπωθεί.

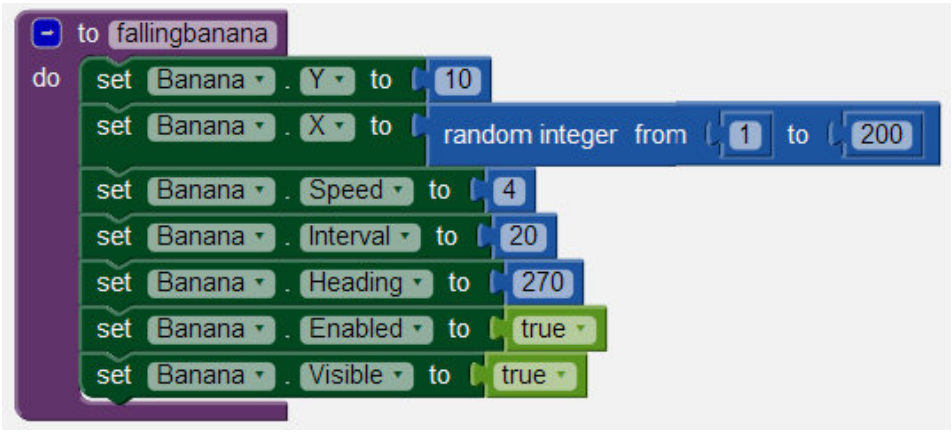
Ας ξεκινήσουμε με την **procedure "fallingbanana"**, που βλέπουμε ότι χρησιμοποιείται συχνά.

Στο παιχνίδι που φτιάχνουμε πρέπει να πέφτουν πολλές μπανάνες για να τις πιάνει η μαϊμού και να αυξάνει το σκορ της. Ή αλλιώς να πέφτει η ίδια μπανάνα πολλές φορές. Άρα είναι προγραμματιστικά συνετό να δημιουργήσουμε μια procedure που θα καλείται συνεχώς και θα εκτελεί τις εντολές για να πέφτει κάθε φορά η μπανάνα.

Στα αριστερά της οθόνης, στα **Built-in blocks**, επιλέγουμε το **Procedures** και από το συρτάρι σέρνουμε στο **Viewer** το **"to procedure do"**. Το μετονομάζουμε ως **"fallingbanana"**.

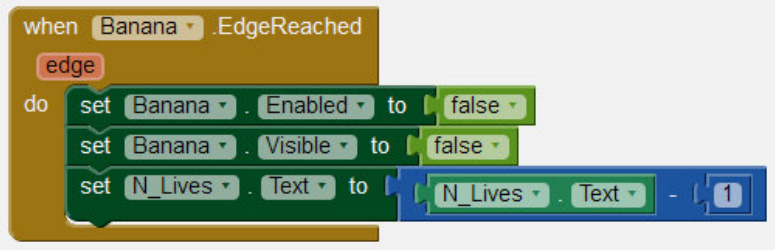

Στην συνέχεια επιλέγουμε το component **"Banana"**, βρίσκουμε τις **εντολές** που χρειάζονται να εκτελεστούν και τις **κουμπώνουμε** μέσα **στο block της procedure**.
Συγκεκριμένα, χρησιμοποιούμε την εντολή **"set Banana.Visible to"** για να αλλάξουμε την ιδιότητα της σε **"true"**, δηλαδή να **εμφανίζεται** (το οποίο θα βρούμε εάν επιλέξουμε στην **Built-in** ομάδα των blocks το **Logic**), γιατί εάν θυμάστε κάθε φορά που την έπιανε η μαϊμού, η μπανάνα εξαφανιζόταν (**Banana.Visible = false**).
Όμοια, δηλώνουμε τις **συντεταγμένες** από όπου θα πέσει η μπανάνα, την **ταχύτητα**, την **φορά**.



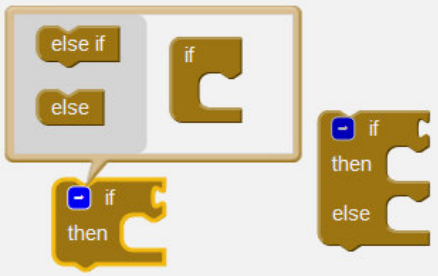

*Κάθε φορά που καλείται η **procedure "fallingbanana"**, θα εμφανίζει μια μπανάνα από σταθερό ύψος (**Y=10**) και από ένα τυχαίο πλάτος (**random X από 1 έως 200**),και η οποία θα πέφτει με μια σταθερή ταχύτητα **"speed"** (τα νούμερα τα συμπληρώνουμε εάν επιλέξουμε στην **Built-in** ομάδα των blocks το **Math**).*

**Όρια καμβά – Δομή Επιλογής (If):**

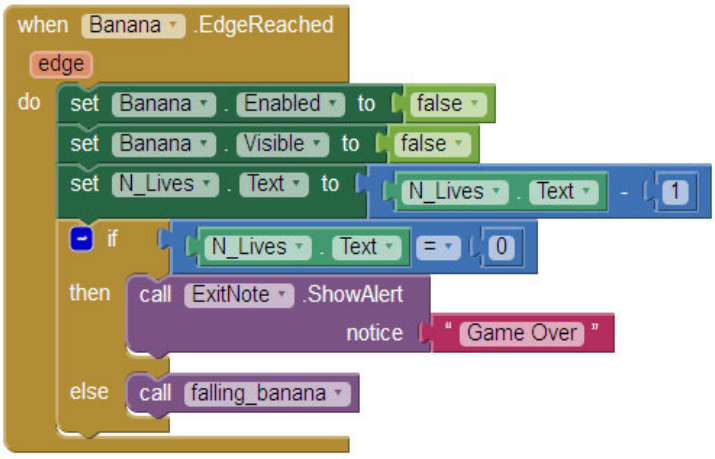
Μέχρι στιγμής έχουμε δει τον τρόπο που κινούνται οι φιγούρες μας και τι γίνεται όταν **πιάσει η μαϊμού την μπανάνα**. Ας δούμε τώρα, τι συμβαίνει **εάν χάσει η μαϊμού μια μπανάνα**. Αυτήν την εκδοχή την επεξεργαζόμαστε με ένα συμβάν, το οποίο ενεργοποιείται όταν η μπανάνα βγει από τα όρια του καμβά ("**when Banana.EdgeReached**").
Συγκεκριμένα στο παιχνίδι μας, επειδή η μπανάνα απλώς πέφτει, το να βγει από τα όρια του καμβά σημαίνει ότι θα έχει φτάσει στο κάτω άκρο του καμβά και η μαϊμού δεν θα έχει προλάβει να την πιάσει.
Οπότε, **επιλέγουμε το** component **"Banana"**, ανοίγει το αντίστοιχο συρτάρι και μεταφέρουμε το κατάλληλο για το συμβάν block **"when Banana.Edgereached"**.
Επιλέγουμε πάλι το component **"Banana"**, εντοπίζουμε τις εντολές **"set Banana.Enabled to"** και **"set Banana.Visible to",** τις κουμπώνουμε μέσα στο block και ορίζουμε τις τιμές τους σε **"false"** (από το **Logic** στα **Built-in blocks**).
Επίσης μειώνουμε της ζωές της μαϊμούς κατά 1. Από το component **"N_Lives"** κουμπώνουμε την εντολή **"set N_Lives.Text to"** και από το **Built-in block "Math"** συμπληρώνουμε με την πράξη της **αφαίρεσης**.




Εάν τώρα, μηδενιστούν οι ζωές τελειώνει το παιχνίδι, αλλιώς καλεί την procedure "fallingbanana" για να συνεχίσει να στέλνει μπανάνες.
Αυτό το διαχειριζόμαστε με μια **εντολή if**, την οποία την βρίσκουμε στο **Control** των **Built-in blocks**. Εάν πατήσουμε στο **μπλε σημάδι** πάνω δεξιά στην If μπορούμε να **σύρουμε** στο block ένα **else** ή ένα elseif.



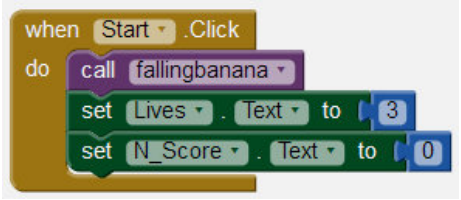

Επιλέγουμε από το **Math** μια **ισότητα** και το **"Game Over"** το συμπληρώ - νουμε από το **Text** των **Built-in blocks**.




*Το component **Notifier** (στην δική μας περίπτωση το **"ExitNote"**), χρησιμοποιείται πολύ από τον προγραμματιστή για να **εμφανίζει ερωτήματα**, όταν ο χρήστης πρέπει να απαντήσει σε κάτι, ή για να εμφανίζει **μηνύματα** και **προειδοποιήσεις** όταν είναι σημαντικό ο χρήστης να μάθει κάτι.*

**Κουμπί Start:**
Το κουμπί Start χρησιμεύει για να ξεκινάει από την αρχή το παιχνίδι (δηλαδή να αρχίσει να πέφτει η μπανάνα) και να αρχικοποιεί το σκορ και της ζωές της μαϊμούς.

Όπως ακριβώς δουλέψαμε στην **Δραστηριότητα 2** για το **"set N_Score.text to"**, έτσι δουλεύουμε και για το component **N_Lives**. Το επιλέγουμε, ανοίγει η συρταριέρα, βρίσκουμε την εντολή **"set N_Lives.text to"**, την σέρνουμε και την **κουμπώνουμε** μέσα στο block **"when Start.Click"**. Συμπληρώνουμε πάνω του το πρώτο πλακίδιο στο **Math (Built-in blocks)**, όπου σβήνουμε το **"0"** και γράφουμε **"3"**.
Τέλος κουμπώνουμε και την **procedure "fallingbanana"** (θα την βρούμε στο **Procedures** των **Built-in blocks**) για να ξεκινήσει η ροή της μπανάνας.




*Φύλλο Εργασίας: Banana Monkey*

Καρδαρά Μαρία – Λεμονιά, ΠΕ20 – Εκπαιδευτικός Πληροφορικής,
Υπεύθυνη Εργαστηρίου Πληροφορικής

Χαλδαίου Πηγή, ΠΕ20 – Εκπαιδευτικός Πληροφορικής