

App Inventor – 6ο Μάθημα (Γκαλερί φωτογραφιών)

✓ Υποπρόγραμμα – διαδικασία (procedure)

Θα κατασκευάσουμε μια εφαρμογή που θα δίνει τη δυνατότητα να παρουσιάζουμε ένα σύνολο από φωτογραφίες. Ο χρήστης της εφαρμογής θα έχει τη δυνατότητα να μετακινηθεί από τη μια φωτογραφία στην επόμενη με 3 διαφορετικούς τρόπους. Αγγίζοντας ένα κουμπί, σύροντας το δάκτυλό του πάνω στην οθόνη και αυτόματα κάθε 3 δευτερόλεπτα.

Βήμα 1: Δημιουργία έργου και ρυθμίσεις οθόνης

1. Στο <http://ai2.appinventor.mit.edu/> ξεκινάμε δημιουργώντας ένα νέο Project με το όνομα PhotoGallery (**Projects – Start new project**).
2. Το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη, *Screen1*, και θα χρειαστεί να μεταβάλλουμε τις ιδιότητές της που φαίνονται και στη συνέχεια.



επιλέγουμε το
αντικείμενο

μεταβάλλουμε τις
ιδιότητες

Screen1

AlignHorizontal: Center

Screen Orientation: Portrait

Scrollable: No

Title: PhotoGallery

Βήμα 2: Προσθήκη αρχείων

3. Ας προσθέσουμε στο έργο μας τα αρχεία εικόνας και τον ήχο που θα χρησιμοποιεί η εφαρμογή μας. Τα απαραίτητα αυτά αρχεία θα τα βρείτε στο φάκελο που βρίσκονται και οι παρούσες σημειώσεις. Υπενθυμίζετε ότι η προσθήκη γίνεται από το πλαίσιο **Media** και με το κουμπί **Upload File...**



ανεβάζουμε το αρχείο με
όνομα

σύντομη
περιγραφή

1.jpg

Εικόνα για προβολή στο άλμπουμ

2.jpg

Εικόνα για προβολή στο άλμπουμ

3.jpg

Εικόνα για προβολή στο άλμπουμ

4.jpg

Εικόνα για προβολή στο άλμπουμ

5.jpg

Εικόνα για προβολή στο άλμπουμ

6.jpg

Εικόνα για προβολή στο άλμπουμ

next.png

Κουμπί επόμενο

previous.png

Κουμπί προηγούμενο

Averil White - Death of the Hero.mp3

Μουσικό κομμάτι εφαρμογής

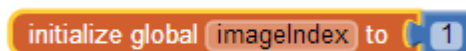
Βήμα 3: Προσθήκη αντικειμένων στο γραφικό περιβάλλον

4. Θα χρειαστούμε το αντικείμενο του καμβά, πάνω στο οποίο θα εμφανίζονται οι φωτογραφίες του άλμπουμ. Επιπλέον θα προσθέσουμε τα κουμπιά για μετακίνηση σε προηγούμενη και επόμενη φωτογραφία, το ρολόι για την αυτόματη εναλλαγή και ένα αντικείμενο Player για την αναπαραγωγή της μουσικής. Τέλος σε μια ετικέτα θα εμφανίζουμε τον αριθμό της εικόνας που απεικονίζεται εκείνη τη στιγμή στην οθόνη. Στην επόμενη λίστα ακολουθούν οι απαιτούμενες ενέργειες.
- **Layout – HorizontalArrangement**
 - Όνομα: *ButtonsArea*
 - Width: *Fill Parent*
 - AlignHorizontal: *Center*
 - **User Interface – Button**
 - Όνομα: *PreviousButton*
 - Image: *previous.png*
 - Text:
 - **User Interface – Label**
 - Όνομα: *PicNumberLabel*
 - FontSize: *30*
 - Text: *1*
 - TextAlignment: *center*
 - Width: *100 pixels*
 - **User Interface – Button**
 - Όνομα: *NextButton*
 - Image: *next.png*
 - Text:
 - **Drawing and Animation – Canvas**
 - Όνομα: *GalleryCanvas*
 - BackgroundImage: *1.jpg*
 - **Sensors – Clock**
 - Όνομα: *TimerClock*
 - TimeInterval: *3000*
 - **Media – Player**
 - Όνομα: *MusicPlayer*
 - Loop: *Yes*
 - Source: *Averil White - Death of the Hero.mp3*

Μέσα στο αντικείμενο **HorizontalArrangement** θα τοποθετήσετε τα δύο κουμπιά, **Button**, και την ετικέτα, **Label** όπως φαίνεται και στο παραπάνω στιγμιότυπο της εφαρμογής.

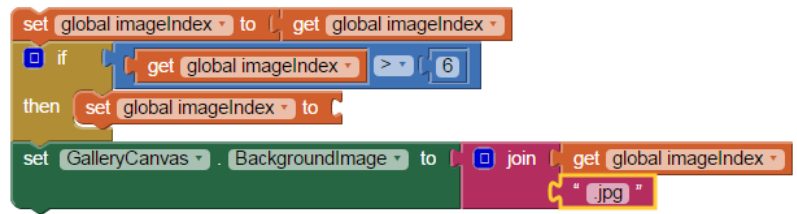
Βήμα 4: Εναλλαγή των εικόνων του άλμπουμ

5. Έχοντας σχεδιάσει, στο **Designer**, τα συστατικά της εφαρμογής μας προχωράμε στον κώδικα που θα ορίζει την συμπεριφορά τους στο **Blocks**.
6. Για να επιτύχουμε την εναλλαγή των εικόνων θα χρειαστεί να δημιουργήσουμε μια μεταβλητή, η οποία θα παίρνει διαδοχικά τις τιμές από 1 μέχρι 6 και ξανά από την αρχή. Η τρέχουσα τιμή της μεταβλητής θα μας δείχνει και τη φωτογραφία που πρέπει να απεικονίσουμε πάνω στον καμβά. Για το λόγο αυτό και τα αρχεία των εικόνων έχουν ονομαστεί κατάλληλα με αριθμούς από το 1 μέχρι το 6 και την κατάληξη φυσικά του αρχείου εικόνας *.jpg*. Ας ονομάσουμε τη μεταβλητή μας *imageIndex* με αρχική τιμή 1.



```
initialize global imageIndex to 1
```

7. Η εφαρμογή μας θα αυξάνει κατά 1 την τιμή της μεταβλητής κάνοντας έλεγχο, όταν ξεπεράσει την τιμή 6, να της δίνει ξανά την τιμή 1. Με τον τρόπο αυτό οι εικόνες θα προβάλλονται κυκλικά.



8. Στη συνέχεια η εφαρμογή θα ανανεώνει την εικόνα φόντου του αντικειμένου *GalleryCanvas* και το κείμενο της ετικέτας με τον αριθμό της εικόνας που προβάλλεται. Στο διπλανό στιγμιότυπο φαίνεται η μορφή του κώδικα.

Παρατηρούμε ότι θα πρέπει να τοποθετήσουμε τις παραπάνω εντολές κάτω από τα 3 διαφορετικά γεγονότα, δηλαδή το άγγιγμα του κουμπιού, την πυροδότηση του ρολογιού και το σύρσιμο του δάκτυλου στην οθόνη. Κάτι τέτοιο δε φαίνεται και πολύ λογικό, αφού θα περιγράψουμε ακριβώς τις ίδιες ενέργειες και στις 3 περιπτώσεις. Με τον τρόπο αυτό φορτώνουμε την εφαρμογή μας με τον ίδιο κώδικα ξανά και ξανά, κάνοντας την δυσανάγνωστη και λιγότερο κατανοητή σε κάποιον που τη διαβάζει. Σκεφτείτε την περίπτωση που οι εντολές δεν επαναλαμβάνονται 3, αλλά 10 ή 50 φορές!! Επιπλέον σκεφτείτε να πρέπει να αλλάξετε κάτι στον κώδικα αυτόν, σε πόσα σημεία θα πρέπει να το κάνετε.

Βήμα 5: Διαδικασία

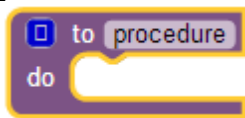
Για να επιλύσουμε το παραπάνω πρόβλημα μπορούμε να τοποθετήσουμε τις εντολές για την εναλλαγή των εικόνων μέσα σε μία διαδικασία.

Η διαδικασία είναι ένα υποπρόγραμμα, δηλαδή ένα μικρό και αυτόνομο τμήμα της εφαρμογής στο οποίο τοποθετούμε μια σειρά εντολών. Στην ουσία χρησιμοποιώντας διαδικασίες δημιουργούμε νέες εντολές που μπορούμε στη συνέχεια να καλούμε (χρησιμοποιούμε) από οποιοδήποτε σημείο του προγράμματος.

Η χρήση υποπρογραμμάτων έχει μια σειρά πλεονεκτήματα, όπως:

- Διάσπαση του προβλήματος σε μικρότερα τμήματα και επίλυση των προ-βλημάτων αυτών με αυτόνομα τμήματα εντολών.
- Μείωση του χρόνου ανάπτυξης και τροποποιήσεων της εφαρμογής, αφού δεν χρειάζεται να επαναλαμβάνουμε όμοια τμήματα εντολών σε διάφορα σημεία της εφαρμογής.
- Ο κώδικας της εφαρμογής είναι πιο ευανάγνωστος και «συμπαγής».

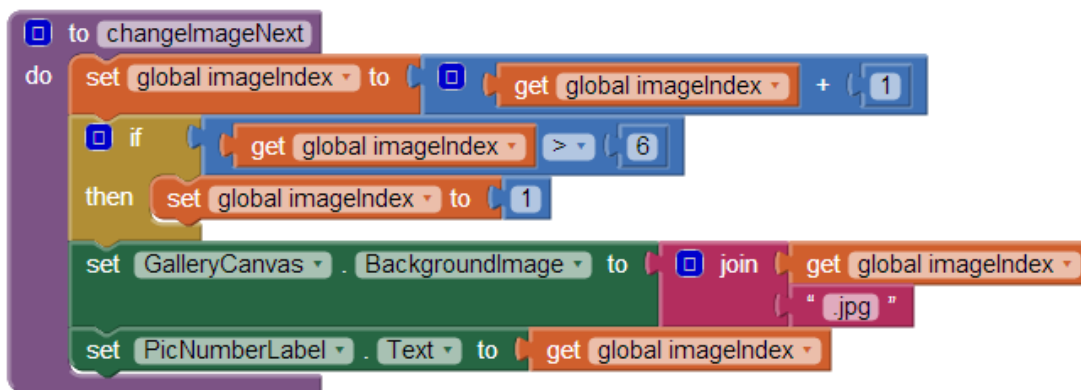
9. Για να δημιουργήσουμε μια νέα διαδικασία μεταβαίνουμε στην ομάδα **Procedures** (Διαδικασίες) ... και επιλέγουμε την εντολή **to [procedure] do**



10. Στη συνέχεια θα μετονομάσουμε τη διαδικασία μας από *procedure* σε *changeImageNext*, ώστε να έχει κάποιο συγκεκριμένο νόημα για εμάς το όνομά της και θα τοποθετήσουμε τις εντολές για την

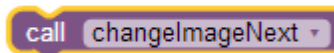


εναλλαγή των εικόνων μέσα σε αυτή. Το τελικό αποτέλεσμα φαίνεται στο επόμενο στιγμιότυπο. Σημειώνουμε ότι με την εντολή *if* εμποδίζουμε την μεταβλητή *imageIndex* να γίνει μεγαλύτερη από 6 ξαναγυρνώντας την στην τιμή 1, δηλαδή από την 6^η εικόνα μας ξαναγυρίζει στην 1^η.



Βήμα 6: Κλήση Διαδικασίας

Για να εκτελεστούν οι εντολές που βρίσκονται μέσα σε μια διαδικασία θα πρέπει να την καλέσουμε, όπως συνηθίζεται να λέμε, δηλαδή να την χρησιμοποιήσουμε μέσα σε ένα ή περισσότερα σημεία της εφαρμογής. Η εντολή για την κλήση μιας διαδικασίας που αναπτύσσουμε βρίσκεται και αυτή στην ομάδα



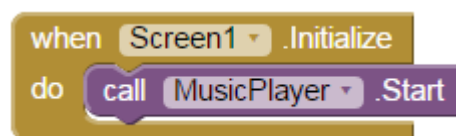
Procedures.

11. Στο σημείο αυτό το πρόβλημα έχει απλοποιηθεί πολύ, αφού το μόνο που μένει να κάνουμε είναι να καλέσουμε τη διαδικασία μας, με την εκτέλεση των 3 γεγονότων. Το πρώτο αναφέρεται στο χρονόμετρο, να περάσει ένα χρονικό διάστημα μέχρι την εναλλαγή, το δεύτερο το άγγιγμα του αντίστοιχου κουμπιού και το τρίτο το χαρακτηριστική πέρασμα του δαχτύλου από το κινητό



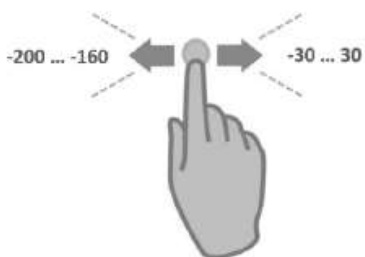
Βήμα 7: Αναπαραγωγή μουσικής

12. Με την έναρξη της εφαρμογής μπορούμε να ξεκινάμε και την μουσική χρησιμοποιώντας τον Player που εισαγάγαμε.
13. Με τον γνωστό τρόπο, Connect – AI Companion μπορείτε να δοκιμάσετε τι έχετε πετύχει μέχρι αυτή τη στιγμή.



Βήμα 8: Μετακίνηση στην προηγούμενη εικόνα

14. Θα κάνουμε τις απαραίτητες ενέργειες, ώστε ο χρήστης της εφαρμογής να μπορεί να μετακινείται στην προηγούμενη εικόνα με δύο τρόπους. Είτε αγγίζοντας το κουμπί *Προηγούμενο*, είτε σέρνοντας το δάκτυλό του προς τα αριστερά. Θα υλοποιήσουμε όπως και προηγουμένως μια διαδικασία με όνομα *changeImagePrevious* που θα εκτελεί τις απαραίτητες ενέργειες.



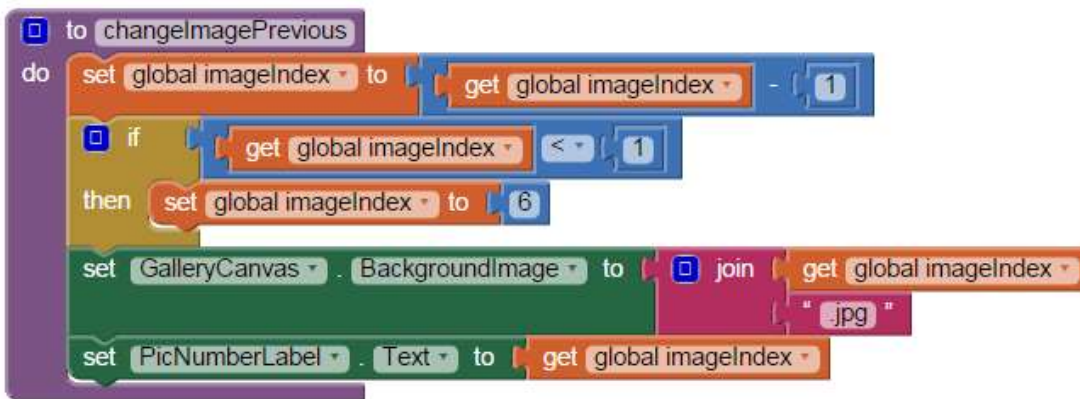
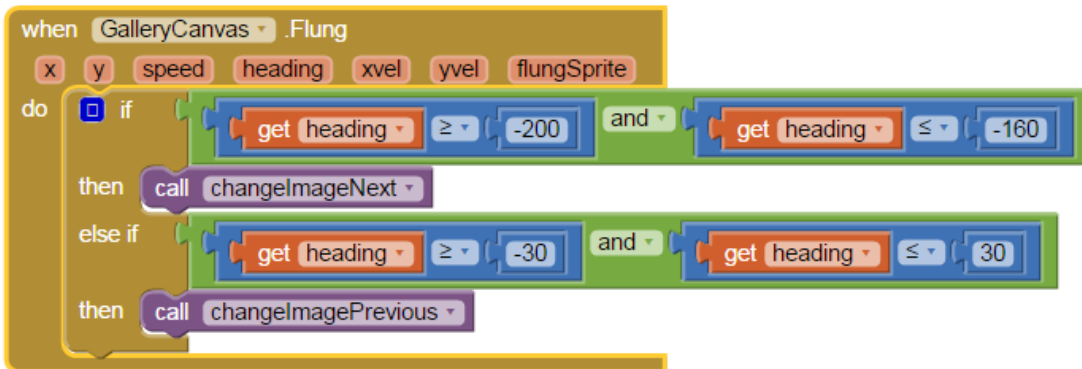
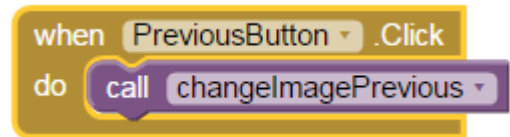
Για να ανιχνεύσουμε τη φορά προς την οποία μετακινείται το δάκτυλο του χρήστη πάνω στον καμβά θα χρειαστούμε την ιδιότητα *heading* (κατεύθυνση), την τιμή της οποίας σας παρέχει

το γεγονός **when [GalleryCanvas] Flung**.



Για να ανιχνεύσουμε την κίνηση του δάκτυλου προς τα δεξιά ελέγχουμε αν η ιδιότητα heading έχει τιμές από -30 μέχρι 30. Για να ανιχνεύσουμε την κίνηση του δάκτυλου προς τα αριστερά ελέγξτε αν η ιδιότητα heading έχει τιμές από -200 μέχρι -160.

Η διαδικασία changeImagePrevious, η κλήση της όταν γίνεται click στην PreviousButton καθώς και η ανανεωμένη εκδοχή του γεγονότος GalleryCanvas.Flung φαίνονται στις επόμενες εικόνες.

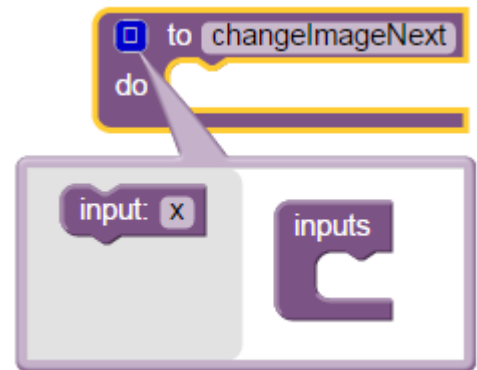


15. Δοκιμάστε και πάλι την εφαρμογή σας μέχρι το σημείο αυτό.

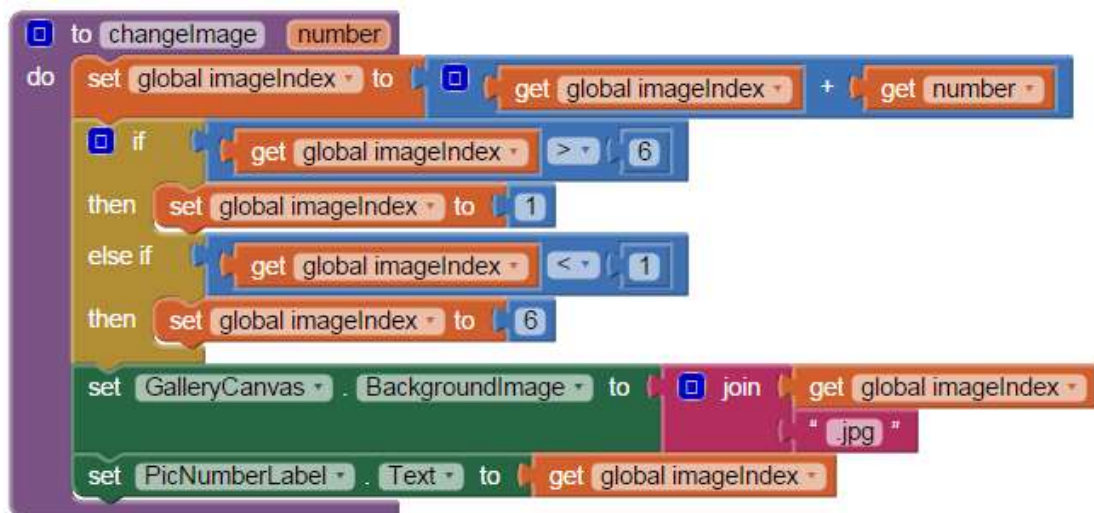
Βήμα 9: Συγχωνεύοντας τις δύο διαδικασίες σε μία

Αν παρατηρήσουμε προσεκτικά τις δύο διαδικασίες που υλοποιήσαμε για την μετακίνηση στην επόμενη και στην προηγούμενη εικόνα, θα δούμε ότι εκτελούν στην ουσία σχεδόν τις ίδιες εντολές. Αυτό που τις διαφοροποιεί είναι η αύξηση ή η μείωση του αντίστοιχου αριθμού εικόνας κατά ένα. Επίσης, διαφέρουν στον έλεγχο για το αν ο αριθμός ξεπέρασε το 6 ή έγινε μικρότερος του 1, αλλά και αυτό εξαρτάται από το αν ο αριθμός αυξάνεται ή μειώνεται. Ο κώδικάς μας απλοποιείται σε μεγάλο βαθμό εάν μπορέσουμε να φτιάξουμε μία διαδικασία η οποία θα υλοποιεί τις λειτουργίες και των δύο παραπάνω διαδικασιών. Για να το επιτύχουμε αυτό θα πρέπει να χρησιμοποιήσουμε παραμέτρους. Έχουμε τη δυνατότητα να στέλνουμε τιμές σε μια διαδικασία τη στιγμή που την καλούμε. Οι τιμές αυτές ονομάζονται παράμετροι της διαδικασίας. Στην ουσία οι παράμετροι είναι μεταβλητές που μεταφέρουν τιμές από την εφαρμογή προς τη διαδικασία.

16. Για να προσθέσουμε μια ή περισσότερες παραμέτρους σε μια διαδικασία κάνουμε κλικ στο μπλε τετράγωνο που βρίσκεται στα αριστερά του ονόματος της διαδικασίας. Ας επιλέξουμε τη διαδικασία με όνομα *changeImageNext*. Από το παράθυρο που εμφανίζεται, σέρνουμε το πλακίδιο **input** μέσα στο πλακίδιο **inputs**. Βάζουμε τόσα πλακίδια, όσες και οι παράμετροι που θέλουμε να προσθέσουμε στη διαδικασία. Στη θέση του x βάζουμε το όνομα που θέλουμε να έχει κάθε παράμετρος. Στην εφαρμογή μας θα έχουμε μία παράμετρο την οποία και θα ονομάσουμε *number*.
17. Επιπρόσθετα θα μετονομάσουμε την διαδικασία μας από *changeImageNext* σε *changeImage* μιάς και πλέον δεν θα μας πηγαίνει μόνο στην επόμενη εικόνα αλλά και στην προηγούμενη ανάλογα με την τιμή της παραμέτρου. Τέλος θα σβήσουμε εντελώς από την εφαρμογή μας την διαδικασία *changeImagePrevious*. Η διαδικασία αυτή θα υλοποιείται από την *changeImage*.
18. Η πρώτη αλλαγή που θα κάνουμε στις εντολές της διαδικασίας είναι να προσθέτουμε την τιμή της παραμέτρου *number* αντί να προσθέτουμε το 1.



19. Επιπλέον θα προσθέσουμε στην εντολή **if** και τους 2 ελέγχους που υπήρχαν στις δύο διαδικασίες για να μην ξεφύγει η εικόνα από τα όρια 1 – 6. Η τελική μορφή της μοναδικής πλέον διαδικασίας *changeImage* φαίνεται στο επόμενο στιγμιότυπο.

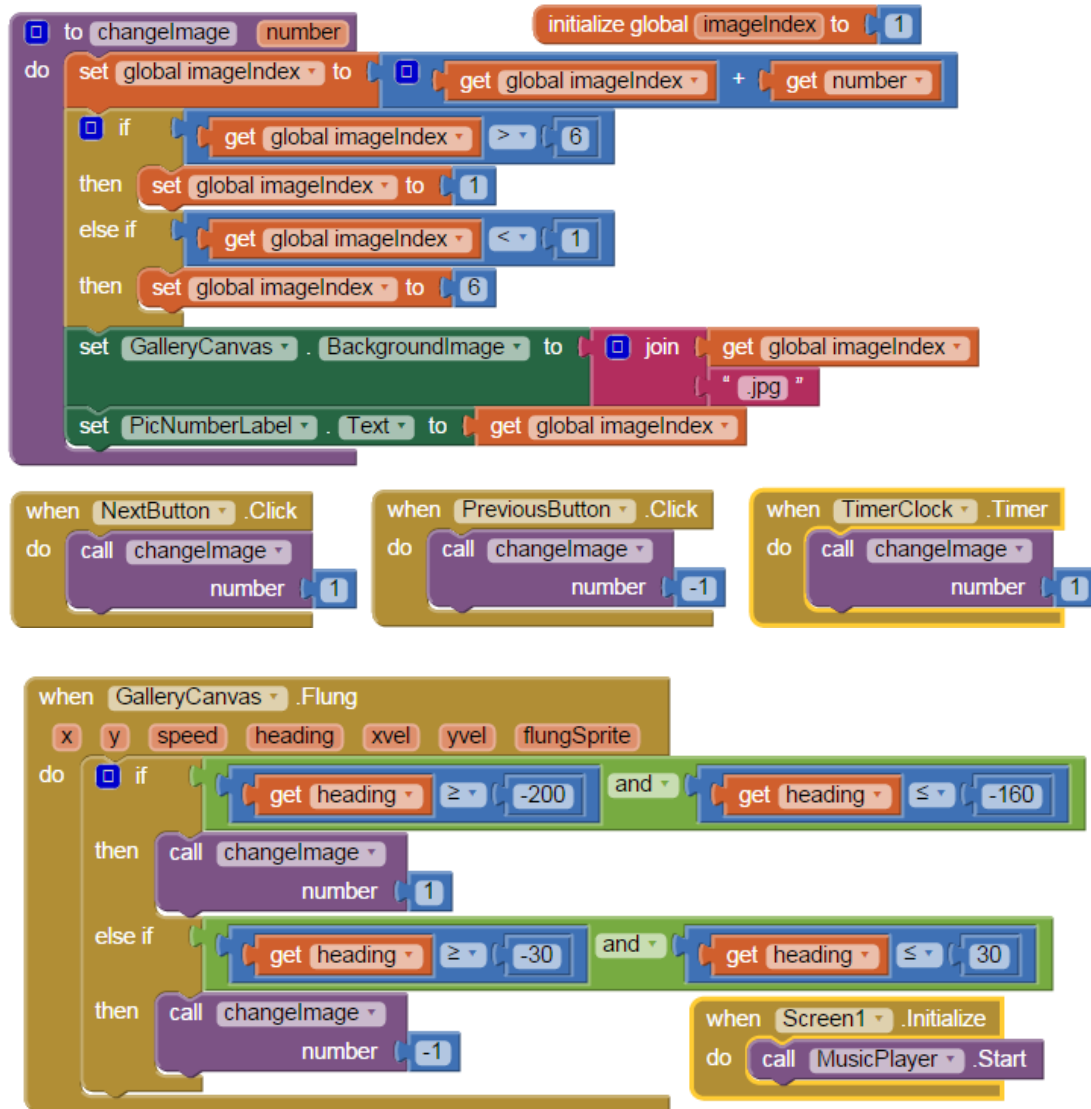
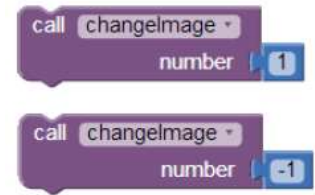


Βήμα 9: Κλήση διαδικασίας με παράμετρο

20. Η διαδικασία που υλοποιήσαμε πρέπει κάθε φορά που καλείται να συνοδεύεται από έναν αριθμό, που είναι η τιμή που θέλουμε να δώσουμε στην παράμετρο *number*. Επομένως, στα σημεία που θέλουμε

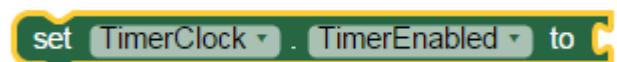
να μετακινούμαστε στην επόμενη εικόνα, θα δίνουμε σαν τιμή το +1, ενώ στα σημεία που θέλουμε να γίνεται μετακίνηση στην προηγούμενη εικόνα, θα δίνουμε σαν τιμή το -1.

21. Η εφαρμογή σας είναι έτοιμη για δοκιμή. Η τελική της μορφή φαίνεται στη συνέχεια.



Δραστηριότητες

1. Παρατηρήστε ότι όταν μετακινούμαστε στην επόμενη εικόνα με το δάκτυλο ή με το κουμπί Επόμενο, ο μετρητής του χρονομέτρου δεν ξεκινάει να μετράει από τη νέα αρχή 3 δευτερό-λεπτα, οπότε η μετάβαση στην επόμενη εικόνα μπορεί να γίνει σε μικρότερο χρονικό διάστημα. Κάντε τις απαραίτητες ενέργειες για να διορθωθεί το πρόβλημα. Μπορείτε να κάνετε χρήση της εντολής **set TimerClock . TimerEnabled** προσθέτοντας **false** και **true** πριν και μετά το κάθε σενάριο ώστε να επανεκκινήσετε ουσιαστικά το ρολόι.



2. Τι θα αλλάζατε στον κώδικά σας εάν θα θέλατε να προσθέσετε άλλη μία εικόνα στην εφαρμογή σας; Μπορείτε να κατεβάσετε μία αφίσα αγαπημένης σας ταινίας από το διαδίκτυο και να την προσθέσετε στην εφαρμογή σας δίνοντας της jpg εικόνας το όνομα 7.jpg.

Πηγή: <http://www.sepchiou.gr/>