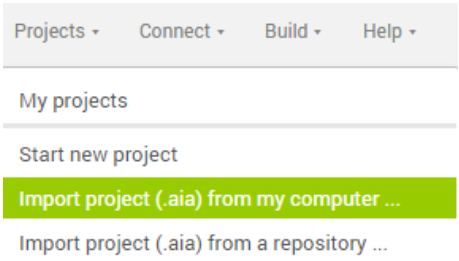
App Inventor – 5ο Μάθημα (Κορώνα γράμματα- επέκταση)

- ✓ Λογικός τελεστής not
- ✓ Δομή επιλογής If…then …else
- ✓ Λογικές μεταβλητές
- ✓ Animation

Θα επεκτείνουμε την εφαρμογή Κορώνα – Γράμματα που δόθηκε σαν δραστηριότητα στο πρώτο μάθημα. Καταρχήν θα την μετατρέψουμε σε ένα απλό παιχνίδι, στο οποίο ο παίκτης θα επιλέγει Κορώνα ή Γράμματα και στη συνέχεια θα ρίχνει το νόμισμα. Επίσης, θα προσθέσουμε animation κατά την ρίψη του νομίσματος, για να μοιάζει το νόμισμα σαν να γυρίζει.



1. Στο http://ai2.appinventor.mit.edu/ ανοίγουμε την εφαρμογή *CoinFlip* από το πρώτο μάθημα. Στην περίπτωση που δεν την είχαμε ολοκληρώσει, στον φάκελο που βρήκαμε τις σημειώσεις αυτές, υπάρχει το αρχείο *CoinFlip.aia* το οποίο και θα πρέπει να εισάγετε στη λίστα των Project σας.

Βήμα 1: Προσθήκη κουμπιών

2. Αφού ανοίξουμε θα προσθέσουμε δύο κουμπιά τα οποία θα επιτρέπουν στον χρήστη πριν στρίψει το νόμισμα να επιλέγει κορώνα ή γράμματα. Τα δύο κουμπιά θα μπουν μέσα σε ένα αντικείμενο **HorizontalArrangement** ώστε να τοποθετηθούν το ένα δίπλα στο άλλο. Σβήνουμε το εξορισμού κείμενο από το κάθε κουμπί και εφαρμόζουμε τις επόμενες ιδιότητες ώστε να μοιάζει η εικόνα του κινητού μας με το διπλανό στιγμιότυπο.

από την **ομάδα**	μεταφέρουμε το **αντικείμενο**	του δίνουμε το **όνομα**	μεταβάλλουμε τις **ιδιότητες**
Layout	HorizontalArrangement	CoinArea	Height : 100 pixels
UserInterface	Button	HeadButton	Image : 1.jpg Width : 60 pixels Height : 60 pixels
UserInterface	Button	TailButton	Image : 2.jpg Width : 60 pixels Height : 60 pixels

Επιπρόσθετα στο *CoinArea* θα θέσουμε το **Width** στο *Fill parent* και τα **AlignHorizontal** και **AlignVertical** στο *Center*.

Βήμα 2: Προσθήκη ετικέτας

3. Θα προσθέσουμε μια ετικέτα για να εμφανίζουμε μήνυμα επιτυχίας στον χρήστη όταν μαντεύει σωστά το νόμισμα.

από την **ομάδα**	μεταφέρουμε το **αντικείμενο**	του δίνουμε το **όνομα**	μεταβάλλουμε τις **ιδιότητες**
UserInterface	Label	WinMessageLabel	Font Size : 40 Text Alignment : center Text : TextColor : Red Width : Fill Parent

Βήμα 3: Επιλογή χρήστη

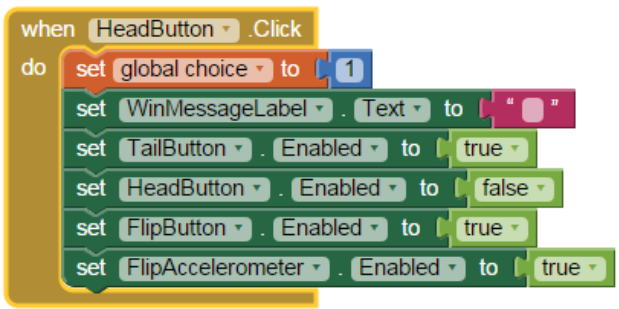
4. Αρχικά θα δημιουργήσουμε μια μεταβλητή που θα αποθηκεύει την επιλογή του χρήστη. Η μεταβλητή αυτή θα παίρνει την τιμή 1 όταν ο χρήστης επιλέγει κορώνα και την τιμή 2 όταν ο χρήστης επιλέγει γράμματα. Θα ονομάσουμε τη νέα μας μεταβλητή *choice* και θα της δώσουμε αρχικά την τιμή 0.

5. Επίσης θα χρειαστούμε μια μεταβλητή *coin* που θα αποθηκεύει το αποτέλεσμα από το στρίψιμο του νομίσματος, δηλαδή τον τυχαίο αριθμό 1 ή 2. Η μεταβλητή *coin* θα έχει και αυτή αρχικά την τιμή 0.

6. Στη συνέχεια θα πρέπει να υλοποιήσουμε την παρακάτω συμπεριφορά κατά το άγγιγμα των κουμπιών. Σημειώνεται ότι, δεδομένου ότι, τα δύο παρακάτω σενάρια έχουν ελάχιστες διαφορές μπορούμε να χρησιμοποιήσουμε την επιλογή **Duplicate** αφού φτιάξουμε το πρώτο σενάριο και κάνωντας δεξί click πάνω του.

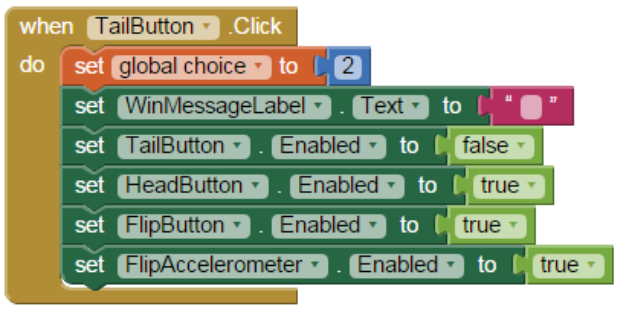
Όταν ο χρήστης επιλέγει το κουμπί της κορώνας, θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή *choice* θα παίρνει την τιμή 1.
- Το κείμενο της ετικέτας για το αποτέλεσμα του στριψίματος θα γίνεται το κενό «».
- Τα κουμπία επιλογής για την κορώνα, *HeadButton* θα απενεργοποιείται (ιδιότητα **Enabled**) ώστε ο χρήστης να μην μπορεί να τα ξαναεπιλέξει ενώ για τα γράμματα, *TailButton*, θα ενεργοποιείται.
- Το κουμπί για το στρίψιμο του νομίσματος αλλά και ο αντίστοιχος σένσορας θα γίνοντε ενεργά.



Όταν ο χρήστης επιλέγει το κουμπί γράμματα θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή *choice* θα παίρνει την τιμή 2.
- Το κείμενο της ετικέτας για το αποτέλεσμα του στριψίματος θα γίνεται το κενό «».
- Τα κουμπία επιλογής για τα γράμματα, *TailButton* θα απενεργοποιείται (ιδιότητα



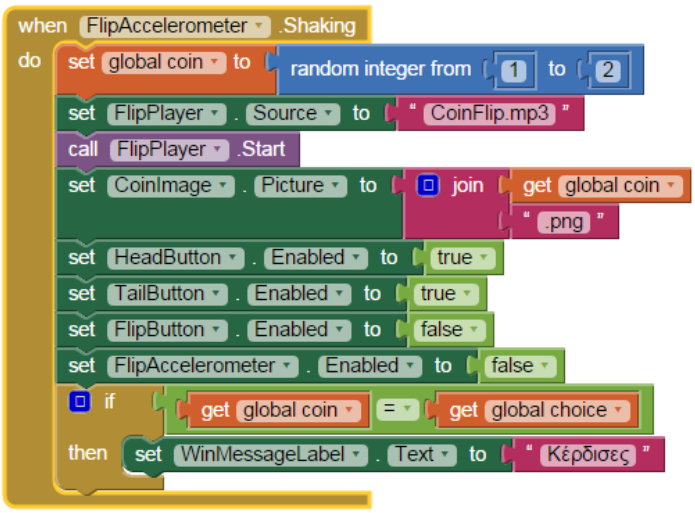
Enabled) ώστε ο χρήστης να μην μπορεί να τα ξαναεπιλέξει ενώ για την κορώνα, *HeadButton*, θα ενεργοποιείται

- Το κουμπί για το στρίψιμο του νομίσματος, FlipButton, και ο σένσορας FlipAccelerometer θα γίνοντε ενεργά.

Βήμα 4: Στρίψιμο νομίσματος

7. Στη συνέχεια, θα προγραμματίσουμε τη συμπεριφορά του κουμπιού για το στρίψιμο του νομίσματος. Όταν ο χρήστης αγγίζει το κουμπί για το στρίψιμο του νομίσματος, **when** FlipButton. **Click** ή κουνάει την συσκευή, **when** FlipAccelerometer **Shaking** θα πρέπει να γίνονται οι παρακάτω ενέργειες. Θα πρέπει να αλλάξετε τις ήδη υπάρχουσες εντολές μιάς και δεν πρέπει να υπάρχουν δύο ίδιες εντολές σαν τις παραπάνω για το ίδιο αντικείμενο.

- Η μεταβλητή coin θα παίρνει ως νέα τιμή έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 2.
- Θα αναπαράγεται ο ήχος του στριψίματος του νομίσματος Για να το κάνετε αυτό θα προσθέσετε ένα αντικείμενο **Player** από την ομάδα **Media**, θα του δώσετε το όνομα *FlipPlayer* και θα θέσετε την πηγή του ήχου, **Source**, στο κατάλληλο ηχητικό αρχείο.
- Η εικόνα του κουμπιού του νομίσματος θα γίνεται κορώνα ή γράμματα ανάλογα με το αποτέλεσμα στη μεταβλητή coin.
- Τα κουμπιά για την επιλογή Κορώνα ή Γράμματα θα είναι και τα δύο ενεργοποιημένα.
- Το κουμπί για το στρίψιμο του νομίσματος και ο αντίστοιχος σένσορας θα απενεργοποιούνται.

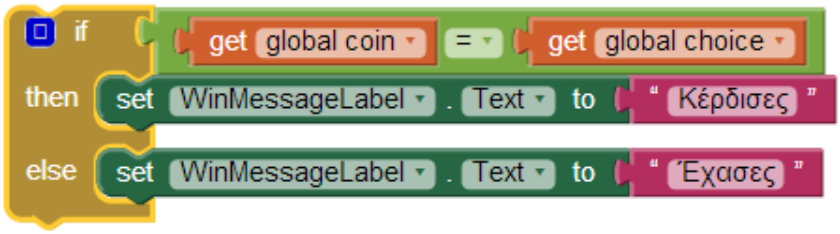


- Αν η επιλογή του χρήστη και το αποτέλεσμα του στριψίματος είναι τα ίδια τότε θα εμφανίζεται το μήνυμα «Κέρδισες» στην ετικέτα.

8. Μπορείτε στο σημείο αυτό να δοκιμάσετε την εφαρμογή σας επιλέγοντας **Connect – AI Companion**.

Βήμα 5: Η εντολή else

Παρατηρήστε ότι η εφαρμογή εμφανίζει μήνυμα επιτυχίας όταν ο χρήστης κερδίζει, αλλά δεν κάνει τίποτα σε περίπτωση που χάνει. Θα την επεκτείνουμε ώστε όταν ο χρήστης δεν μαντεύει σωστά να του εμφανίζει το μήνυμα «*Έχασες*». Πολύ συχνά χρειάζεται να περιγράψουμε τις ενέργειες που θα εκτελεστούν όταν μια λογική πρόταση είναι αληθής και ταυτόχρονα να περιγράψουμε κάποιες άλλες ενέργειες όταν η ίδια λογική πρόταση δεν ισχύει. Για να το πετύχουμε θα πρέπει να επεκτείνουμε την εντολή **if** προσθέτοντας της την εντολή



else. Για να προσθέσουμε την εντολή else κάτω από την εντολή if κάνουμε κλικ στο μπλε τετράγωνο που βρίσκεται πάνω στην εντολή **if**, όπως φαίνεται στο διπλανό στιγμιότυπο.

9. Όταν ο χρήστης δεν μαντέψει σωστά το νόμισμα κάτω από την εντολή else θα θέσουμε το κείμενο *Έχασες* στην ετικέτα *WinMessageabel*.
10. Μπορείτε να δοκιμάσετε και αυτήν την προσθήκη στην συσκευή σας.

Βήμα 6: Προσθήκη ήχου

11. Στο **Design** και πάλι, και στην ομάδα **Media** προσθέστε στην εφαρμογή σας ένα αντικείμενο **Player**. Ονομάστε το *AppPlayer*.
12. Στην συνέχεια ανεβάστε ένα ηχητικό για γιουχάισμα, και ένα για χειροκρότημα ανάλογα αν ο παίκτης χάνει ή κερδίζει. Παραδείγματα και για τα δύο θα βρείτε στον φάκελο που βρήκατε αυτές τις σημειώσεις αλλά φυσικά μπορείτε να κατεβάσετε και τα δικά σας από κατάλληλους ιστότοπους όπως το http://www.freesfx.co.uk/. Για να ανεβάσετε οποιοδήποτε αρχείο στην εφαρμογή σας υπενθυμίζεται ότι θα κάνετε χρήση του μενού **Media** – **Upload File** και θα τα βρείτε στον κατάλληλο φάκελό.
13. Στο **Blocks** και στο σενάριο που μόλις φτιάξετε θα κάνετε την προσθήκη που φαίνεται και στο διπλανό στιγμιότυπο.

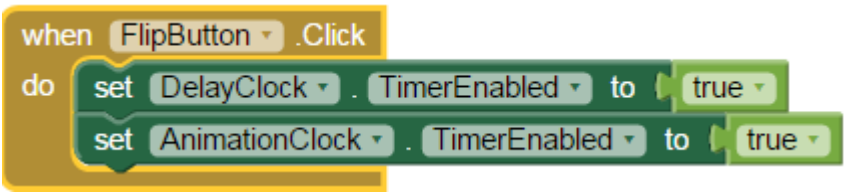
Βήμα 7: Προσθήκη Animation

14. Θα προσθέσουμε ένα απλό animation που θα αλλάζει την εικόνα του νομίσματος έτσι ώστε να δίνει την αίσθηση ότι το νόμισμα γυρίζει. Για το σκοπό αυτό θα χρειαστεί να προσθέσουμε 2 ρολόγια στην εφαρμογή. Το πρώτο θα καθορίζει τη συνολική διάρκεια του animation, για παράδειγμα 1 δευτερόλεπτο, ή 1000 χιλιοστά του δευτερολέπτου. Το δεύτερο ρολόι θα καθορίζει κάθε πότε θα γίνεται η εναλλαγή της εικόνας του κουμπιού, για παρά-δειγμα κάθε 50 χιλιοστά του δευτερολέπτου.

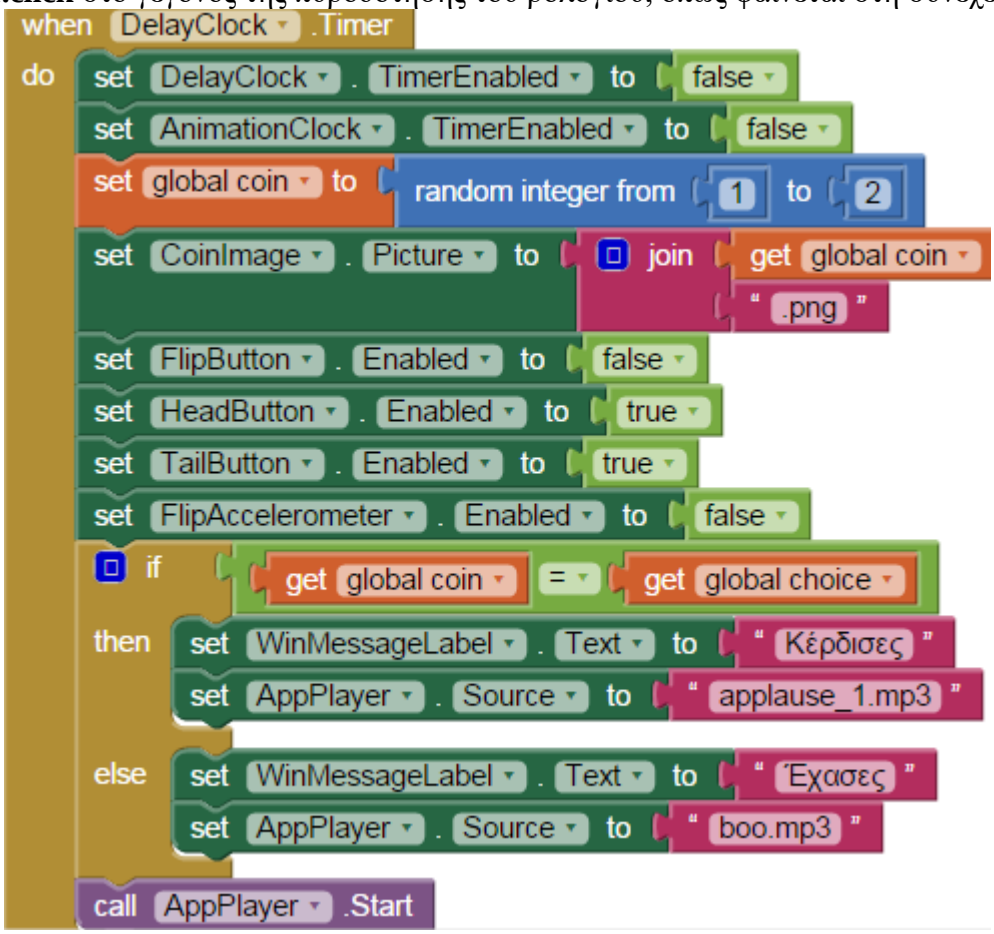
από την **ομάδα**	μεταφέρουμε το **αντικείμενο**	του δίνουμε το **όνομα**	μεταβάλλουμε τις **ιδιότητες**
UserInterface	Clock	DelayClock	TimerEnabled : ΟΧΙ TimerInterval : 1000
UserInterface	Clock	AnimationClock	TimerEnabled : ΟΧΙ TimerInterval : 50

Αρχικά τα δύο ρολόγια είναι απενεργοποιημένα. Η ενεργοποίησή τους θα γίνεται κάθε φορά που ο παίκτης θα αγγίζει το κουμπί της ρίψης του νομίσματος. Όταν τελειώσει η κίνηση μετά το πέρας 1 δευτερολέπτου τα ρολόγια θα απενεργοποιούνται ξανά.

15. Στη συνέχεια και στο **Blocks**, προσθέτουμε κάτω από το γεγονός του αγγίγματος στο κουμπί της ρίψης τις επόμενες εντολές.



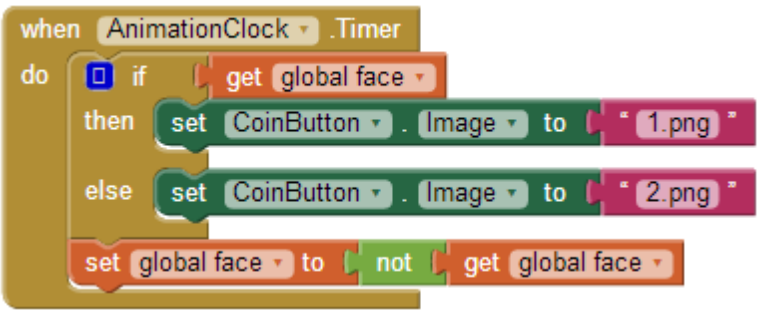
16. Οι εντολές που θα εκτελεί το ρολόι *DelayClock* θα είναι η απενεργοποίηση των ρολογιών και η ενεργοποίηση των υπολοίπων αντικειμένων του παιχνιδιού ώστε ο χρήστης να μπορεί να ξαναπαίξει. Οι εντολές αυτές θα μετακινηθούν από το γεγονός του αγγίγματος του κουμπιού, **when** *CoinButton*.**click** στο γεγονός της πυροδότησης του ρολογιού, όπως φαίνεται στη συνέχεια.



17. Το ρολόι *AnimationClock* θα μας βοηθήσει στην εναλλαγή των εικόνων του κουμπιού. Θα χρειαστεί να δημιουργήσουμε μια νέα μεταβλητή με όνομα *face* που θα παίρνει τις τιμές True/False. Θα τις δώσουμε αρχικά την τιμή false.

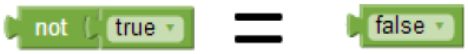
18. Στη συνέχεια κάθε φορά που θα πυροδοτείται το ρολόι *AnimationClock* θα γίνονται οι παρακάτω ενέργειες:

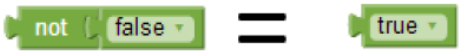
- Αν η μεταβλητή *face* έχει την τιμή false, στο κουμπί θα εμφανίζεται η εικόνα της κορώνας. Διαφορετικά θα εμφανίζεται η εικόνα των γραμμάτων.
- Η μεταβλητή *face* θα παίρνει την αντίθετη τιμή από αυτή που έχει.



Δηλαδή αν είναι true θα γίνεται false, ενώ αν είναι false θα γίνεται true.

Προκειμένου να αντιστρέψουμε την τιμή μιας λογικής μεταβλητής χρησιμοποιούμε τον τελεστή **not** που βρίσκεται στην ομάδα **Logic**. Η λειτουργία του είναι πολύ απλή. Αν μια λογική μεταβλητή έχει την τιμή true τότε βάζοντας μπροστά της τον λογικό τελεστή **not** παίρνει την τιμή false, και αντιστρόφως.





Μπορείτε να δοκιμάσετε την προσθήκη κίνησης στην εφαρμογή σας.

Δραστηριότητες

1. Αναπτύξτε μια εφαρμογή που θα λειτουργεί σαν αντίστροφη μέτρηση. Αρχικά ο χρήστης θα ορίζει τα δευτερόλεπτα που θα μετρά αντίστροφα η εφαρμογή. Στη συνέχεια θα ενεργοποιεί την αντίστροφη μέτρηση με το άγγιγμα ενός κουμπιού.

Η εφαρμογή θα εμφανίζει έναν έναν τους αριθμούς των δευτερολέπτων μετρώντας αντίστροφα. Όταν τα δευτερόλεπτα μηδενιστούν η εφαρμογή θα παίζει έναν ήχο δική σας επιλογής. Για παράδειγμα αν ο χρήστης δώσει σαν αριθμό δευτερολέπτων το 10, η εφαρμογή θα πρέπει ανά δευτερόλεπτο να εμφανίζει έναν έναν τους αριθμούς:

10 … 9 ….8….7….6….5….4….3….2….1….0

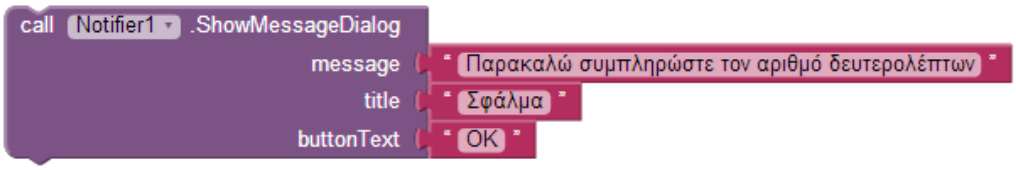
Μπορείτε να έχετε και ένα κουμπί επανεκκίνησης που όταν το πατάτε θα επανέρχετε η εφαρμογή στην αρχική της κατάσταση όπου θα μπορείτε να θέσετε από την αρχή τον αριθμό των δευτερολέπτων.

Φροντίστε να είναι ορατά κάθε στιγμή τα αντικείμενα που χρειάζονται, δηλαδή κατά την αντίστροφη μέτρηση δεν θα εμφανίζεται το κουτί για τα δευτερόλεπτα ούτε το κουμπί για την ενεργοποίηση της μέτρησης.

Η εφαρμογή σας θα πρέπει να ελέγχει αν ο χρήστης έχει γράψει κάποιον αριθμό στο πεδίο κειμένου, των δευτερολέπτων ή αν το έχει αφήσει κενό. Στη δεύτερη περίπτωση θα του εμφανίζεται ένα μήνυμα σφάλματος ζητώντας να διορθώσει το λάθος του.

Θα χρειαστείτε:

- Το αντικείμενο **Clock** της ομάδας **Sensors** για να μετρά το πέρασμα ενός δευτερολέπτου
- Το αντικείμενο **Notifier** της ομάδας **UserInterface**. Στο τμήμα των εντολών θα χρησιμοποιήσετε την επόμενη εντολή



- Κάποιο αρχείο ήχου για την έκρηξη. Είστε ελεύθεροι να χρησιμοποιήσετε οποιοδήποτε της αρεσκείας σας. Στον φάκελο που βρίσκονται τα μαθήματα αυτά μπορείτε να βρείτε κάποια έτοιμα παραδείγματα από το site http://www.freesfx.co.uk/
  - time_bomb.mp3
  - explosion_sl12.mp3
- Μπορείτε να χρησιμοποιήσετε και κάποιο εικονίδιο για βόμβα αρχικά και έκρηξη που θα εμφανίζεται την κατάλληλη στιγμή. Μπορείτε να βρείτε έτοιμες εικόνες από το site http://www.freeimages.com/.
  - bomb.png
  - explosion.png

Ερωτήσεις

Πηγή: http://www.sepchiou.gr/