

Invasori dallo spazio

Cosa stai costruendo

Nel costruire l'app Invasori dallo Spazio farai pratica con l'utilizzo di componenti e timer di Clock, utilizzerai componenti di animazione come Image Sprites e Canvas, imposterai la visibilità e a rilevare le collisioni in App Inventor.

Programmerai un'applicazione con un'astronave che ha l'obiettivo di sparare a tutti i dischi volanti sullo schermo.

Iniziare

Collegati al sito web di App Inventor e inizia un nuovo progetto. Denominalo *InvasoriDalloSpazio*. Connettiti a un dispositivo o emulatore.

Introduzione

Questo tutorial introduce le seguenti abilità, utili per lo sviluppo futuro del gioco:

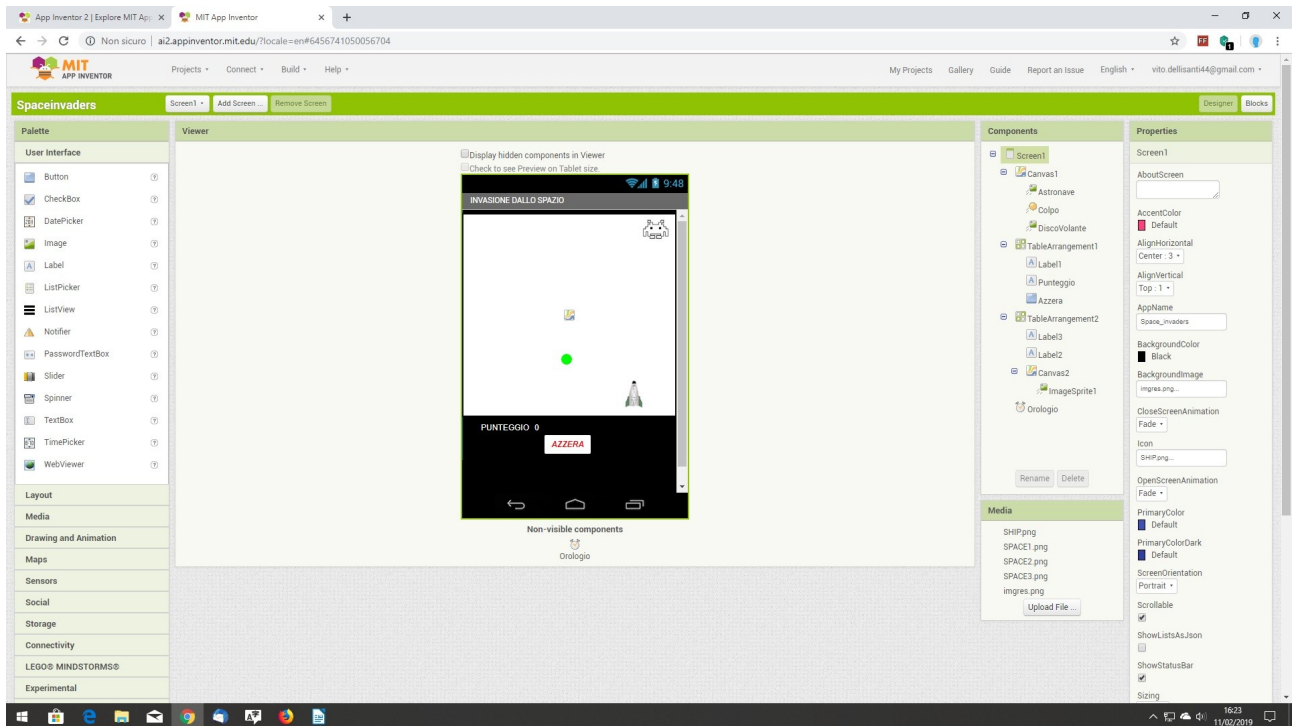
- Utilizzo del componente Orologio
- Utilizzo di Clock.Timer per spostare gli sprite
- Utilizzo di Sprite.Flung per spostare uno sprite
- Utilizzo del rilevamento delle collisioni
- Impostazione della visibilità degli sprites

Prepararsi

Per questo gioco, avrai due tipi di sprites: il primo rappresenterà una astronave che spara un colpo, mentre il secondo rappresenterà un disco volante.

Imposta i componenti

Utilizza la finestra di progettazione dei componenti per creare l'interfaccia per InvasoriDalloSpazio. Al termine della progettazione dovresti avere uno schermo simile all'istantanea sottostante (istruzioni più dettagliate sotto l'immagine).



Per creare questa interfaccia, inserisci i seguenti componenti nella finestra di progettazione trascinandoli dalla tavolozza dei componenti nel visualizzatore e imposta le proprietà dei componenti come descritto di seguito:

Tipo	Componenti disponibili	Nome	Scopo	Azione
Stage	Disegno e animazione	Canvas1	Lo sfondo su cui metteremo i nostri sprite	Cambia la proprietà Larghezza a "Riempi il contenitore" e la proprietà Altezza a 300. Imposta la proprietà ColoreDisegno a Nero.
Sprite	Disegno e animazione	Astronave	L'astronave nel nostro gioco	Scarica l'immagine dell'astronave ed imposta la proprietà Immagine a "SHIP.png". Imposta la proprietà Y a 230. Questo posizionerà l'astronave sul fondo dell'area Stage
Sprite	Disegno e animazione	DiscoVolante	Il disco volante nel nostro gioco	Scarica le immagini del disco volante (SPACE1.png, SPACE2.png e SPACE3.png) e imposta la proprietà Immagine a "SPACE1.png".
Sprite	Disegno e animazione	Colpo	Il colpo sparato dall'astronave	Cambia ColoreDisegno in Verde e la proprietà Raggio a 8.
Orologio	Sensori	Orologio	Usiamo l'Orologio per spostare l'astronave	Cambia la proprietà IntervalloTimer a 10000.

Disposizione orizzontale	Impaginazione	DisposizioneOrizzontale	Conterrà i campi Label1 e Punteggio.	
CasellaDiTesto	Interfaccia utente	Label1	Conterrà la parola "PUNTEGGIO: "	Cambia la proprietà Testo in "PUNTEGGIO: "
CasellaDiTesto	Interfaccia utente	Punteggio	Conterrà il valore del punteggio raggiunto	Imposta la proprietà Testo a "0".
CasellaDiTesto	Interfaccia utente	Label3		Imposta la proprietà Testo a "Invasione dallo Spazio".
Screen1	Interfaccia utente	Screen1		Imposta la proprietà Titolo a "INVASIONE DALLO SPAZIO".
Pulsante	Interfaccia utente	Azzera	Per resettare il punteggio raggiunto e iniziare una nuova partita	Cambia la proprietà Testo in "AZZERA".

Per muovere l'astronave

In questo gioco, l'utente muoverà l'astronave da un lato all'altro. Questo significa che cambierai solo la direzione X dello sprite dell'astronave. Per fare questo userai il gestore eventi `Astronave.Trascinamento`. Quando l'astronave viene mossa, imposterai la sua proprietà X in `Xattuale` su cui abbiamo trascinato lo sprite.

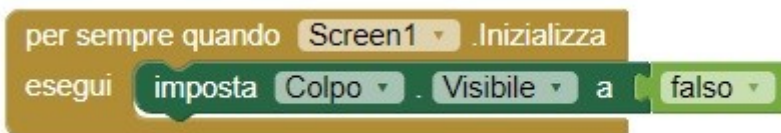


Una volta messi insieme questi blocchi, collega il tuo telefono e prova questa funzionalità!

Programmazione del comportamento del proiettile

In questo gioco ci sono diverse funzionalità da definire per il nostro proiettile. Vogliamo che sia sparato dall'astronave, entri in collisione con il disco volante, e sia invisibile dopo la collisione.

Iniziamo usando il blocco `Screen1.inizializza`. Quando lo schermo è inizializzato, programmeremo che il proiettile sia invisibile. Lo facciamo impostando la proprietà di visibilità del colpo su falso.



Successivamente, vogliamo assicurarci che il proiettile ricompaia quando spariamo dall'astronave. Quando tocchiamo l'astronave, vogliamo che il proiettile inizi a dirigersi verso il disco volante. Lo faremo usando il gestore eventi `Astronave.Toccato`. Quando l'astronave viene toccata, non vogliamo solo rendere visibile il colpo sparato, ma vogliamo anche impostarne la velocità e la direzione. La direzione è un valore compreso tra 0 e 360 che indica in quale direzione si muoverà lo sprite. 0/360 va a sinistra, 90 in alto, 180 a destra e 270 in basso. La velocità è misurata in pixel/sec.

```

per sempre quando Astronave .Toccato
  x y
  esegui
    imposta Colpo .Visibile a vero
    imposta Colpo .Velocità a 5
    imposta Colpo .Direzione a 90

```

L'ultima cosa che dobbiamo programmare è cosa succede quando il proiettile colpisce il disco volante. Useremo il gestore eventi `Colpo.HaToccato`. Questo evento viene chiamato ogni volta che il proiettile entra in collisione con un altro sprite. Poiché la nostra astronave è bloccata in una Y nella parte inferiore dello schermo, il proiettile non entrerà mai in collisione con essa, ma solo con il disco volante. In caso di collisione vogliamo che succedano due cose. 1. Il punteggio dovrebbe aumentare di 1. 2. Il proiettile dovrebbe diventare invisibile.

```

per sempre quando Colpo .HaToccato
  altro
  esegui
    imposta Colpo .Visibile a falso
    imposta Punteggio .Testo a Punteggio .Testo + 1

```

Se hai iniziato a provare questo gioco, potresti aver notato che una volta sparato il proiettile, non sembra che tu possa spararlo di nuovo. Abbiamo bisogno di programmare il proiettile in modo tale che torni di fronte all'astronave prima che venga sparato. Possiamo farlo usando il blocco `Colpo.VaiA`.

```

per sempre quando Astronave .Toccato
  x y
  esegui
    esegui Colpo .VaiA
      x
      y
    imposta Colpo .Visibile a vero
    imposta Colpo .Velocità a 5
    imposta Colpo .Direzione a 90

```

Ora, riprova!

Potresti aver notato che se manchi il disco volante, il proiettile si sposta nella parte superiore dello schermo e rimane bloccato fino a quando non riprendi a muoverti. Per far sparire il proiettile quando colpisce il bordo superiore della nostra area, dobbiamo usare il `Colpo.HaToccatoBordo`

```
per sempre quando Colpo .HaToccatoBordo
  bordo
  esegui imposta Colpo .Visibile a falso
```

Per programmare la richiesta di azzeramento punteggio

A volte, gli utenti potrebbero voler riavviare il gioco e ripristinare il loro punteggio. Quando ciò accade, dobbiamo riportare il punteggio a 0.

```
per sempre quando Azzera .Cliccato
  esegui imposta Punteggio .Testo a 0
```

Aumentiamo la difficoltà - Cambiamo la posizione del disco volante

Rendiamo il gioco un po' più impegnativo! Ora, quando il proiettile si scontra con il disco volante, cambiamo la posizione del disco volante. Il disco volante manterrà lo stesso valore Y quindi dovremo solo cambiare la X. Possiamo farlo usando il blocco casuale.

```
per sempre quando Colpo .HaToccato
  altro
  esegui imposta DiscoVolante .X a intero casuale tra 0 e Canvas1 .Larghezza - DiscoVolante .Larghezza
  imposta Colpo .Visibile a falso
  imposta Punteggio .Testo a Punteggio .Testo + 1
```

Per renderlo ancora più difficile, quando il Timer si spegnerà cambieremo anche la posizione del disco volante, l'immagine del disco volante ed il colore al testo della Label2.

```
when Orologio .Timer
  do
    set DiscoVolante .X to random integer from 0 to Canvas1 .Width - DiscoVolante .Width
    set DiscoVolante .Picture to pick a random item list make a list "SPACE1.png" "SPACE2.png" "SPACE3.png"
    set Label2 .TextColor to pick a random item list make a list yellow magenta red white blue yellow
```