

Interazione e Multimedia – Laboratorio Processing

Prova 19 settembre 2016

Lo studente svolga gli esercizi richiesti usando Processing. E' suggerito l'uso del solo editor interno di Processing e delle Reference fornite con l'editor. Non sono richieste librerie esterne. Il tempo a disposizione è di **3 ore**.

*Al termine dell'esame dovrà essere consegnato un unico file zip/rar denominato con il seguente formato: **COGNOME_NOME_MATRICOLA.zip**.*

Lo zip dovrà contenere una cartella per ciascun esercizio svolto con ogni dato e file necessario alla corretta esecuzione dello sketch sviluppato.

[15] Esercizio 1

Si implementino in Processing le seguenti **classi** con le caratteristiche dichiarate :

- Una classe **Egg** – Un **Egg** è identificato da una **posizione** e da una **velocità** di caduta **sy**. Un **Egg** deve prevedere dei metodi che modifichino la posizione lungo Y.

L'aspetto di un **Egg** è quello di un ellisse senza bordi di semiasse orizzontale **15 pixel** e semiasse verticale **20 pixel** ; il colore è bianco.

L'Egg deve poter essere rimosso dalla finestra di visualizzazione.

- Una classe **Bucket** – Un **Bucket** è identificato da una **posizione**. Ha inoltre l'aspetto di un **trapezio isoscele di altezza 40 pixel, base minore 30 pixel e base maggiore 50 pixel**. Il suo colore sarà **marrone**.

Il **Bucket** deve prevedere un metodo che gli permetta di muoversi lungo X, in modo da occupare sempre la posizione X del mouse. La posizione Y invece dovrà rimanere invariata.

Si preveda pure un metodo che preso in input un **Egg** controlli se questo si trovi approssimativamente sul lato superiore del **Bucket** .

Si implementi uno sketch Processing in cui:

- Si apre una finestra di dimensione 500×500, con sfondo azzurro, e **framerate 60** e si istanzi **un solo** **Bucket** in posizione Y = 450 e in posizione X a piacere. Ad ogni ciclo di draw, la finestra viene aggiornata e tutti gli **Egg** presenti cadono in base alla loro velocità, mentre il **Bucket** segue il puntatore lungo X.
- Durante ogni ciclo di draw si introduca con una probabilità del 5% un nuovo **Egg** in posizione Y=0 e X casuale, ma compresa tra 25 e 475. La loro velocità sarà pure casuale, ma compresa tra 4 e 8.
- Quando un **Egg** entra nel **Bucket** (si verifichi con il metodo richiesto per il **Bucket**), tale **Egg** dovrà sparire dalla finestra.

[15] Esercizio 2

Si implementino in Processing le seguenti **funzioni** con le caratteristiche dichiarate :

- **channel()** – Deve avere come parametro di input un'immagine RGB (**PImage**) e un **carattere**. L'immagine di output è una nuova immagine a scala di grigi ottenuta replicando uno dei tre canali e rimpiazzando così gli altri due. In particolare, se il **carattere** passato come parametro è **'r'** si replica il rosso. Se è **'g'** il verde, se è **'b'** il blu. La funzione estrae dunque un singolo canale RGB dall'immagine.
- **changeColor()** – Deve avere come parametro di input un'immagine RGB (**PImage**), un **carattere** e un valore **intero**. L'immagine di output è una nuova immagine ottenuta sommando solo ad un canale/componente di ogni pixel il valore intero passato. Se il **carattere** è **'r'** si somma il valore solo alla componente rossa di ogni pixel. Se è **'g'** solo alla verde e se è **'b'** solo alla blu.

Si implementi uno sketch Processing in cui:

- Si apre un'immagine RGB di dimensioni **256×256** (oppure si ridimensiona in modo che diventi di dimensioni **256×256**) e una finestra di dimensioni **512×512**. Si inizializza pure un **stato** del programma in modalità **'r'**.
- Nella parte in alto sinistra della finestra si visualizza l'immagine aperta. Chiaramente questa occuperà un quarto della schermata. Nella parte in alto destra si visualizza l'immagine che si ottiene applicando **channel()** all'immagine in alto a sinistra (quella RGB) con parametro **'r'**. In basso a sinistra si fa lo stesso ma con parametro **'g'** e in basso a destra con parametro **'b'**. Si visualizzano cioè l'immagine RGB e le tre componenti R, G e B.
- Alla pressione del **tasto 'g'** si cambia lo **stato** del programma in **'g'**. Alla pressione del **tasto 'b'** si cambia lo stato in **'b'**. Lo stesso quando si preme il **tasto 'r'**. Alla pressione del **tasto '+'** si chiama la funzione **changeColor()** sull'immagine RGB, passando come carattere quello corrispondente allo **stato** e come valore intero **3**. Alla pressione del **tasto '-'** si fa lo stesso, ma passando come valore **-3**. Ovviamente l'immagine in alto a destra (e di conseguenza i suoi canali) sarà aggiornata di volta in volta. Questa parte del programma permette quindi di selezionare un canale ed aumentarne o diminuirne i valori di luminanza dei pixel.