

Interazione e Multimedia – Laboratorio Processing

Prova 14 febbraio 2018

Lo studente svolga gli esercizi richiesti usando Processing. È suggerito l'uso del solo editor interno di Processing. Lo studente può fare uso solo ed esclusivamente delle Reference fornite con l'editor. Non sono richieste librerie esterne. Il tempo a disposizione è di **2 ore e 25 minuti**.

*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 **Heart** – Un **Heart** è identificato da una **posizione**, da una **velocità in pixel sx** lungo x e da una **velocità sy** lungo y.
- Gli **Heart** possono essere **vivi** o **morti**. Quando un Heart viene **istanziato** è sempre **vivo**. La forma di un Heart è quella di un **cuore** senza bordi. Le sue dimensioni lo rendono **approssimativamente** inscritto all'interno di un rettangolo di dimensioni **60 × 60**; un cuore vivo è **rosso scuro**, mentre un cuore morto è **grigio**.

Suggerimento di San Valentino ♥: si potrebbe disegnare il cuore tramite due curve di Bezier.

Un Heart deve prevedere un metodo che ad ogni ciclo di **draw** ne modifica la posizione di un valore pari alla sua velocità, a prescindere dal fatto che sia **vivo** o **morto**. **Solo** se lo Heart è **morto**, prima di modificare la posizione, la velocità lungo y viene modificata in modo tale da **aumentare verso il basso** di una quantità pari a **0.1 + f(q)** pixel. Dove $f(q) = \frac{(q \% 21)}{80}$. Il valore di **q** è uguale alle ultime 5 cifre del numero di matricola.

Si preveda un metodo per controllare se il puntatore del mouse si trova approssimativamente all'interno della forma che rappresenta l'Heart e un metodo per cambiarne lo stato.

Si implementi uno sketch Processing in cui:

- Si apre una finestra di dimensione 512×512 e **framerate 60**. Lo **sfondo** della finestra deve essere realizzato in modo tale che gli oggetti che si muovono su di esso simulino il rilascio di una scia. Il colore deve essere un **blu molto scuro**.
- Ad ogni ciclo di **draw**, **tutti** gli Heart istanziati si spostano in base alla loro velocità e al fatto che siano **vivi** o **morti**. Inoltre, con una probabilità pari al 5%, può essere generato un **nuovo Heart**. L'Heart generato dovrà avere posizione iniziale lungo X **casuale**, ma tale che sia compresa all'interno della finestra. La posizione lungo Y deve essere sempre la stessa, ma scelta in modo che **sia poco fuori** la parte bassa della finestra. La velocità lungo X deve essere casuale compresa tra -5 e 5. Infine, la velocità lungo Y deve avere valore assoluto casuale compreso tra 3 e 7; il **segno** deve essere scelto dallo studente, in modo che tale velocità sia **verso l'alto**.
- Quando si clicca con il mouse su un Heart **vivo**, esso passa allo stato **morto**. Nulla accade se si clicca sugli Heart **morti**.

[15] Esercizio 2

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

- **randomIm()** – Deve avere come parametri di input due interi. L'output della funzione è un'immagine RGB di dimensioni pari a quelli dei due interi in input. Ogni pixel di questa immagine deve avere un colore generato casualmente. Le intensità di colore per ogni canale saranno comprese tra 0 e un valore **massimo** uguale alla somma dei codici ASCII delle iniziali (minuscole) dello studente.

Esempio per Mario Rossi: l'intensità sarà compresa tra 0 e 223 ('m' + 'r' = 109 + 114 = 223).

- **combine()** – Deve avere come parametro di input due immagini RGB e un valore **a** compreso tra 0 e 1. L'output della funzione è una nuova immagine, calcolata facendo la media **pesata** tra le due immagini di input. Il peso della prima è **a**, mentre il peso della seconda è **(1-a)**. Con "media" tra due immagini si intende la media delle intensità dei colori pixel a pixel.

Esempio per un pixel in posizione (x,y): se A(x,y) e B(x,y) hanno rispettivamente i colori (50, 10, 20) e (100, 20, 40), la loro media pesata con a = 0.5 è (75, 15, 30).

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 **768×256**. Inoltre si inizializza una variabile **alfa** con valore **0.5**.
- **In ogni momento, durante l'esecuzione dello sketch**, nella parte sinistra della finestra si visualizza un'immagine ottenuta la funzione **randomIm()** con parametri **256** e **256**. Nella parte centrale della finestra si visualizza invece l'immagine aperta. Infine, nella parte destra della finestra, si visualizza un'immagine ottenuta con la funzione **combine()** passando come parametri: **l'immagine originale, l'immagine ottenuta con randomIm()** e il valore **alfa**.
- Da qualche parte, dove si vuole, si visualizza il valore di **alfa**.
- Alla pressione del **tasto '+'** si aumenta di **0.1** il valore di alfa. Viceversa, alla pressione del **tasto '-'** si decrementa di **0.1** tale valore. Si **limiti** comunque questo valore tra un **minimo di 0** e un **massimo di 1**.