

Interazione e Multimedia – Laboratorio Processing

Prova 06 marzo 2017

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 **2 ore e 45 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 **Stain** – Un oggetto Stain è identificato da una **posizione**, da un **diametro D** e da un **colore**. L'aspetto di uno Stain è di un cerchio **senza bordi** di **diametro D** e del colore che lo caratterizza.

Lo **Stain** deve avere un metodo che gli permetta di modificare la sua posizione lungo X e Y ad ogni ciclo di draw nel modo seguente: la posizione lungo Y varia di una quantità casuale compresa tra 0 e 4, mentre la posizione lungo X varia di una quantità casuale compresa tra -1 e 1.

- Una classe **Sponge** – Un oggetto Sponge è identificato da una **posizione**, una **dimensione L** e da un **colore (di default [200,200,100])**. Ha inoltre l'aspetto di un **quadrato** senza bordi di lato **L**.

Si preveda un metodo che ad ogni ciclo di draw permetta di muovere l'oggetto, modificandone la **posizione** e assegnandogli quella del **mouse**.

Si preveda anche un metodo **adsorb()** che abbia in input un oggetto di tipo **Stain**. Il metodo avrà il seguente comportamento: **se e solo se** il cerchio che rappresenta lo Stain si trova interamente nel quadrato che rappresenta l'oggetto Sponge, allora l'oggetto **Sponge modifica** i propri attributi. Nello specifico, la **dimensione** aumenta di un valore pari alla metà del **diametro** dello **Stain** in input; il **colore** dell'oggetto Sponge diventa uguale al colore medio tra il suo colore attuale e il colore dello Stain. Con "medio" si intende la media canale per canale (media rosso, media verde, ecc.). Infine viene restituito **true**. Nel caso in cui non si verifichi la condizione iniziale, non accade nulla e viene restituito **false**.

Si implementi uno sketch Processing in cui:

- Si apre una finestra di dimensione 700×700 con sfondo **azzurro chiaro**. Viene inoltre istanziato un solo oggetto Sponge di dimensione pari a 30.
- A ogni ciclo di draw la finestra viene aggiornata ed ogni Stain **presente (cioè non rimosso)** si muove, mentre l'oggetto Sponge segue il puntatore del mouse. Viene inoltre chiamato il metodo **adsorb()** di Sponge per ogni Stain **presente**; se il risultato è **true** lo Stain in input deve essere **rimosso** dallo sketch. Gestire la rimozione come si vuole (es: **nel main** o **all'interno** della classe Stain, ecc.).
- Quando si preme il tasto "S" Viene generato un nuovo Stain in posizione X casuale compresa tra 50 e 650, posizione Y uguale a -30, diametro casuale compreso tra 10 e 50, colore casuale.
- Quando si preme il tasto "R" l'oggetto Sponge viene resettato (dimensione 30 e colore di default).

[15] Esercizio 2

Si implementi in Processing la seguente **funzione** con le caratteristiche dichiarate:

- **swap()** – Deve avere come parametri di input un'immagine RGB (**PImage**) e cinque interi **x1, y1, x2, y2, s**. L'output è una nuova immagine RGB bit ottenuta scambiando la **sottoimmagine** (intorno) centrata in (**x1,y1**) di dimensione **s**, con la sottoimmagine (intorno) centrata in (**x2,y2**) di dimensione **s**.

Si implementi uno sketch Processing in cui:

- Si apre un'immagine di dimensioni **512×512** (oppure si ridimensiona in modo che diventi di dimensioni **512×512**) e una finestra di dimensioni **512×512**. L'immagine deve essere a colori; **S** con valore di default pari a **80**.
- In ogni momento, nella finestra viene visualizzata un'immagine, inizialmente quella aperta all'avvio dello sketch. Questa poi subirà delle modifiche cumulative.
- Quando si clicca la prima volta su un punto dell'immagine, viene registrata la posizione in cui è avvenuto il click. Quando si clicca la seconda volta su un punto dell'immagine viene chiamato **swap()**, passando come parametri **l'immagine attualmente** visualizzata, le **coordinate del primo click**, le **coordinate del secondo click** e il valore **S**. L'immagine viene **aggiornata** col risultato di **swap()**. Dopodiché si predispone lo sketch per accettare due **nuovi** click e si procede in questo modo all'infinito. **Attenzione!** Se le coordinate dei punti scelti sono tali che l'intorno includa regioni esterne all'immagine, queste coordinate vanno riportate entro i limiti prima di chiamare **swap()**.
- In ogni momento deve essere visualizzato un quadrato di dimensione **S**, senza colore di riempimento, solo con un bordo nero, che faccia capire qual è l'intorno considerato. Inoltre, quando si effettua il primo click, la zona interessata deve rimanere evidenziata (come si vuole, purché si capisca), **finché** non si effettua il secondo click. Gestire anche in questo caso, nello stesso modo di cui sopra, le coordinate che porterebbero l'intorno fuori dall'immagine.
- Alla pressione del tasto **'+'** il valore di **S** viene incrementato di **10**. Alla pressione del tasto **'-'** il valore di **S** viene decrementato di **10**. Si limiti comunque il valore di **S** tra **30 e 150** (estremi compresi). Ovviamente quando **S** cambia, tutto viene aggiornato di conseguenza. Questo evento particolare legato alla tastiera **NON** deve verificarsi se lo sketch è in attesa del secondo click.

SUGGERIMENTO da Berna - Esercizio 1: sappiamo che verificare se un singolo punto si trova all'interno di un rettangolo è semplice. Se invece si vuole verificare se un cerchio si trova all'interno di un quadrato bisogna ricordarsi di considerare il centro del cerchio, ma anche il suo raggio.

SUGGERIMENTO da Berna - Esercizio 2: un modo per capire se si attende il primo click o il secondo click e usare una variabile booleana che cambi di volta in volta.