

# Interazione e Multimedia – Laboratorio Processing

**Prova 26 luglio 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 **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 implementi in Processing la seguente funzione con le caratteristiche dichiarate:

- **convolutionNBox()** – Deve avere come parametri di input un'immagine a scala di grigi (PImage) **I** e un intero positivo **s**. L'output della funzione è una nuova immagine a scala di grigi, ottenuta calcolando la convoluzione tra l'immagine **I** e un filtro di media NBox di dimensione **s**.
- **absDiff()** – Deve avere come parametri di input due immagini a scala di grigi (PImage) **I** e **J**. L'output **R** della funzione è una nuova immagine a scala di grigi, ottenuta calcolando il valore assoluto della differenza punto a punto tra l'immagine **I** e l'immagine **J**. Cioè, il nuovo valore per il pixel di coordinate (x,y) sarà  $R(x,y) = |I(x,y) - J(x,y)|$ .

Si implementi uno sketch Processing in cui:

- Si apre un'immagine a scala di grigi di dimensioni **256×256** (oppure si modifica dopo l'apertura in modo che diventi a scala di grigi e di dimensioni **256×256**) e una finestra di dimensioni **768×300**. Si inizializza una variabile intera **S** con valore pari a **11**.
- In ogni momento, nella parte in alto a sinistra della finestra viene visualizzata l'immagine originale. Nella parte in alto al centro viene visualizzata l'immagine ottenuta con la funzione convolutionNBox(), con input l'immagine originale e intero **S**. Nella parte in alto destra, viene invece visualizzata l'immagine ottenuta utilizzando la funzione absDiff(), con parametri l'immagine originale e l'immagine posta al centro della finestra (quella ottenuta con convolutionNBox).
- Nell'area inferiore della finestra, che non viene occupata dalle immagini, si inserisce il seguente testo: "Original" sotto l'immagine a sinistra; "NBox **S**" sotto l'immagine centrale (con **S** ci si riferisce al valore della variabile); "Difference" sotto l'immagine a sinistra. Il testo deve essere bianco, e lo sfondo nero. Inoltre deve essere posizionato opportunamente senza sfiorare la finestra e senza che si sovrapponga con le immagini.
- Se viene premuto il tasto "+" della tastiera, **S** viene incrementata di **2**, mentre alla pressione del tasto "-" questa sarà decrementata di **2**. Si limiti comunque il valore di **S** tra un minimo di **1** e un massimo di **21**.

**SUGGERIMENTO da Berna:** Stare attenti a non fare errori nel calcolo degli elementi del kernel NBox di dimensione **s**.

## [15] Esercizio 2

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

- Una classe **Petal** – Un oggetto Petal è identificato da una **posizione**, un colore, una velocità lungo **X e Y**, un **angolo** e uno **stato**. Un Petal è rappresentato graficamente da una curva di Bezier i cui 4 punti di controllo sono disposti ai vertici di un triangolo isoscele di base 100 e altezza 50 pixel. L'altezza è relativa al **vertice** individuato dalla **posizione** del Petal, che coincide col punto di controllo 1 e il punto di controllo 4 della curva (questi due sono sovrapposti e si hanno in totale tre vertici/punti di controllo). Inoltre, il Petal viene ruotato di un valore pari al suo **angolo** attorno alla sua **posizione**. La curva è disegnata **senza bordi** e la sua area riempita con il **colore** che caratterizza il Petal.

Il Petal deve avere un metodo per modificare la sua **posizione** ad ogni ciclo di draw di un valore pari alle rispettive **velocità** lungo **X e Y**. Lo stesso metodo deve modificare anche il valore delle **velocità** di un valore pari a **0.3 pixel** per ciclo. Tutto questo deve avvenire solo se il Petal è **attivo**. Quando viene istanziato, il Petal è nello **stato di inattività**, cioè non si muove.

Si deve quindi prevedere un altro metodo che permetta di **attivare** il Petal e contemporaneamente assegnarne le velocità iniziali.

- Una classe **Flower** – Un oggetto Flower è identificato da una **posizione**, da un **colore** e da un insieme di 6 **Petal (che controlla)**. Un Flower è rappresentato graficamente da un cerchio giallo (**corolla**) di diametro pari a 25 pixel, centrato nella **posizione** del Flower, e dai 6 Petal. I Petal sono caratterizzati nel modo seguente: inizialmente la **posizione** e il **colore** di ogni **Petal** sono gli stessi del **Flower**, tuttavia, ognuno di essi è orientato diversamente. Nello specifico, l'angolo di ogni Petal cambia di 60° rispetto al Petal adiacente. In questo modo i 6 Petal copriranno 360°. La **corolla** deve essere in primo piano rispetto ai Petal. Quando il Flower viene istanziato i suoi Petal sono **inattivi**.

Il Flower deve prevedere un metodo che permetta di verificare se un determinato punto si trova all'interno della corolla. Inoltre, deve prevedere un metodo che permetta di **attivare** i Petal. Quando questo metodo è invocato, si sceglie uno a **caso** dei Petal ancora **inattivi** del Flower e si chiama il suo metodo di **attivazione**, assegnando alle velocità lungo X e Y due valori casuali (uno per ciascuna velocità) compresi tra -5 e 1. Se non ci sono più Petal **inattivi** da attivare, non accade nulla.

Si implementi uno sketch Processing in cui:

- Si apre una finestra di dimensione 512×512 con sfondo **bianco**.
- Ad ogni ciclo di draw, ogni Flower si occupa di controllare i suoi Petal, disegnandoli e richiamando i loro metodi di movimento e attivazione.
- Alla pressione di un tasto del mouse, viene verificato se il puntatore si trova all'interno della **corolla** di un Flower. Se ciò avviene, il Flower corrispondente **attiverà** uno a caso dei suoi **Petal inattivi** che si "staccherà" e inizierà a muoversi secondo le regole che lo caratterizzano.
- Alla pressione del tasto CTRL, viene introdotto un nuovo Flower nella scena. Questo sarà istanziato in posizione casuale X compresa tra 50 e 450 e posizione casuale Y compresa tra 50 e 450. Il colore del Flower deve essere una **sfumatura di rosa** generata casualmente.
- Alla pressione del tasto ALT lo sketch viene resettato.

**SUGGERIMENTO da Berna:** Si potrebbero gestire i Petal attivi e inattivi di un Flower con due diverse strutture dati, ma è solo una possibilità. Per le sfumature di rosa, basta fissare i valori per due dei tre canali RGB e scegliere casualmente il terzo in un certo range...