

# Interazione e Multimedia – Laboratorio Processing

*Prova 8 maggio 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 30 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 **funzioni** con le caratteristiche dichiarate:

- **imMean()** – Deve avere come parametro di input un'immagine RGB. L'output della funzione è un colore RGB, ottenuto dalla media dei colori di ogni pixel che costituiscono l'immagine di input. Per media tra colori RGB, si intende la media canale per canale.
- **regionMean ()** – Deve avere come parametro di input un'immagine RGB, e tre interi **x**, **y** e **n**. L'output deve essere un colore, ottenuto utilizzando la funzione **imMean()** sulla sottoimmagine centrata in **x** e **y**, di dimensione **n × n**. Lo studente può gestire come preferisce la situazione in cui la sottoimmagine include regioni esterne all'immagine originale.

Si implementi uno sketch Processing in cui:

- Si apre un'immagine RGB di dimensioni **512×512** (oppure si ridimensiona in modo che diventi di dimensioni **512×512**) e una finestra di dimensioni **1024×512**, si inizializza inoltre una variabile **K** con valore pari a 10 moltiplicato per l'ultima cifra del numero di matricola dello studente +1.
- **In ogni momento, durante l'esecuzione dello sketch**, nella parte sinistra della finestra si visualizza l'immagine originale. Inoltre viene applicata la funzione **regionMean()**, passando come parametri di input l'immagine originale, **x** e **y** pari alle coordinate del pixel dell'immagine originale su cui si trova il **mouse** in quel momento, e infine il valore **K**. Nella parte destra della finestra si visualizza un rettangolo di dimensioni **512×512**, del **colore** restituito dalla funzione **regionMean()** applicata come descritto sopra. **Attenzione!** Se le coordinate **x** e **y** sono tali che l'intorno includa regioni esterne all'immagine, queste coordinate vanno riportate entro i limiti prima di chiamare **regionMean()**.
- Inoltre, deve essere sempre mostrato un quadrato di dimensione **K × K**, senza colore di riempimento e dal bordo di un colore a scelta dello studente, che permetta di capire qual è la regione che in quel momento è considerata nel calcolo della media.
- Ogni volta che viene premuto il tasto **'+'** il valore di **K** aumenta di **10**, mentre alla pressione del tasto **'-'** viene decrementato di **10**. Si limiti comunque questo valore tra un minimo di 10 e un massimo di 250.

## [15] Esercizio 2

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

- Una classe **Bullet** – Un Bullet è identificato da una posizione. Il suo aspetto è quello di un cerchio viola senza bordi di diametro pari a 10 pixel.

Un Bullet deve prevedere un metodo che ad ogni ciclo di draw ne modifica la posizione lungo **y** di un valore pari a 10 in valore assoluto. Il segno deve essere scelto in modo che lo spostamento avvenga verso l'alto.

- Una classe **Asteroid** – Un Asteroid è identificato da una **posizione** e da due velocità, lungo **x** e lungo **y**. Inoltre, questo può essere in due stati, **on** e **off**. Lo stato di default è **on**. Il suo aspetto è quello di un **cerchio** di diametro pari a **50 pixel**, con **bordo giallo** di spessore pari alla penultima cifra del numero di matricola dello studente. Il **colore** del cerchio è **rosso scuro** se lo stato è **on**, **grigio** altrimenti.

Un Asteroid deve prevedere un metodo che ad ogni ciclo di draw gli permetta di modificare la sua posizione di un valore pari alle sue velocità. **Solo** quando lo stato è **off**, questo metodo deve modificare la **velocità lungo y** di una quanti pari a 0.2 in valore assoluto. Il segno deve essere scelto affinché tale accelerazione sia verso il basso.

Infine, un Asteroid deve prevedere un metodo che preso in input un Bullet, verifichi se questo lo abbia colpito. In tal caso lo stato dell'Asteroid passa allo stato **off**. Se lo stato è già **off** non accade nulla.

- Una classe **Gun** – Un Gun è identificato da una posizione. Il suo aspetto è quello di un rettangolo **grigio** di base pari a **20 pixel** e altezza pari a **70 pixel** centrato nella posizione che caratterizza il **Gun**.

Un Gun deve prevedere un metodo che ad ogni ciclo di draw ne modifica la posizione lungo **x** assegnandogli quella del mouse. La posizione **y** non viene invece modificata. Un Gun deve inoltre prevedere un metodo **shot()** che gli permetta di generare un Bullet con posizione prossima alla parte alta del rettangolo che costituisce il Gun.

Si implementi uno sketch Processing in cui:

- Si apre una finestra di dimensione 512×512 e **framerate 60**. Lo **sfondo** della finestra deve essere di colore **azzurro**. Si istanzia poi **un solo** Gun nella parte più bassa della finestra.
- Ad ogni ciclo di **draw**, **tutti** gli Asteroid e i Bullet istanziati si muovono secondo le loro velocità, mentre il Gun segue il mouse. Si verifichi per ogni Asteroid se esso è colpito da uno dei Bullet con il metodo apposito.
- Ad ogni ciclo, con probabilità pari al 5%, viene generato un nuovo Asteroid in **posizione x pari a -50**, **posizione y** casuale tra 0 e 256, velocità lungo **x** casuale tra 2 e 7 e velocità lungo **y** casuale tra -2 e 2.
- Alla pressione del tasto '**A**' viene chiamato il metodo **shot()** del Gun. Il nuovo Bullet generato si unirà agli altri presenti nella scena.