

# Interazione e Multimedia – Laboratorio Processing

**Prova 17 aprile 2019**

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 **80 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.*

## Esercizio 1

- a) [10] Implementare una classe **Shield** con le seguenti caratteristiche: la classe è identificata da una **posizione** e da una **velocità  $sx$  lungo  $x$** . Appare come un **triangolo** isoscele **verde**, **senza bordi** di altezza pari a circa **50px**. Ha un metodo per cambiare la sua posizione lungo l'asse  $x$  di un valore pari a  **$sx$**  per ciclo di draw. Quando lo **Shield** esce dalla parte destra dello schermo rientra dalla parte sinistra e viceversa.
- b) [5] Estendere tramite **ereditarietà** la classe **Shield** definendo **GreatShield**. **GreatShield** possiede **anche** un attributo **scale** - di default **0.5**. Il colore è **rosso** e il **bordo giallo**. Inoltre, **GreatShield** appare sempre **riscaldato**, rispetto al suo centro, di un fattore pari a **scale**. A sua volta **scale** varia continuamente, crescendo da un minimo di **0.5** fino a un massimo di **2** e poi calando da **2** fino a **0.5**. La variazione è di **0.05** per ciclo di draw.

### [OBBLIGATORIO PER VALUTARE I PUNTI PRECEDENTI]

Alla pressione di **"R"** o **"r"** lo sketch viene resettato. Implementare uno sketch in cui si apre una finestra di dimensioni **256x512**. Ad ogni ciclo di draw la finestra viene aggiornata.

**Per rendere (a) valutabile** - Nella metà superiore istanziare un oggetto di tipo **Shield** che si muove in base al comportamento definito. La posizione iniziale è casuale e compresa nella metà superiore della finestra, mentre la velocità iniziale è casuale e compresa tra -5 e 5.

**Per rendere (b) valutabile** - Nella metà inferiore della finestra va istanziato un oggetto **GreatShield** che si muove e cambia dimensione in base al comportamento definito.

## Esercizio 2

- c) [10] Implementare la funzione con prototipo **PImage gammaOp(PImage I, float g)**. Input e output sono a scala di grigi. L'output è una nuova immagine ottenuta applicando l'operatore puntuale gamma all'immagine in input, con gamma uguale a **g**.
- d) [5] Implementare la funzione con prototipo **PImage gammaOp(PImage I, float g, int L, int T)**. Come (c), ma dopo aver applicato l'operatore puntuale viene tracciato un quadrato **viola** di lato **L**, con bordo di spessore **T** e con centro corrispondente a quello dell'immagine. I pixel all'interno del quadrato rimangono inalterati.

### [OBBLIGATORIO PER VALUTARE I PUNTI PRECEDENTI]

Alla pressione di **"R"** o **"r"** lo sketch viene resettato. Implementare uno sketch in cui si apre una finestra di dimensioni **768x256**. Si apre inoltre un'immagine **Im**, si converte a scala di grigi e si ridimensiona a **256x256**. Si visualizza infine questa immagine nella parte sinistra della finestra.

**Per rendere (c) valutabile** - Nella parte centrale della finestra, si visualizza l'immagine ottenuta usando **gammaOp()** con input l'immagine originale **Im** e **g** casuale compreso tra 1.5 e 4.

**Per rendere (d) valutabile** - Nella parte destra della finestra, si visualizza l'immagine ottenuta usando **gammaOp()** con input l'immagine originale **Im**, **g** casuale tra 1.5 e 4, **L** casuale tra 50 e 250, **T** casuale tra 3 e 15.