

Test di laboratorio per il modulo Processing del corso “Interazione e Multimedia”
Esempi “tipici” dei compiti che potremmo assegnare.

“COMPITO TIPO” N.3

a) Realizzare uno sketch Processing che si componga di un'area bianca di $N \times N$ pixel (lo sketch dovrà continuare a funzionare correttamente anche se si varia nel codice il valore intero assegnato alla variabile N), questa è l'“arena”.

N pixel dell'arena sono colorati di rosso, essi sono “il cibo” per le creature che si muovono nell'arena.

Le “creature” sono rappresentate da cerchietti pieni di raggio 3 pixel di colore giallo.

Ogni creatura è caratterizzata da una posizione (x,y) e una energia che è un intero tra 100 e 0.

Quando l'energia è 0 la creatura è morta. Ad ogni step una creatura si muove di un numero random tra 2 e 10 pixel e consuma 5 livelli di energia. Se uno dei puntini “cibo” finisce dentro il cerchietto della pallina l'energia si aumenta di 15 unità (ma non supera mai 100) e il cibo sparisce.

Inizialmente ci sono $N/4$ creature. Ad ogni click del pulsante sinistro del mouse le creature ancora “vive” “saltano” nella posizione successiva e aggiornano le loro energie.

Si richiede inoltre che lascino sotto forma di spezzata di segmenti il percorso di ciascuna creatura.

Quando tutte le creature sono morte appare un avviso.

Sebbene non obbligatorio si suggerisce di utilizzare una implementazione object oriented per le creature.

b) Realizzare uno sketch Processing in cui la canvas è divisa orizzontalmente in due regioni: la zona inferiore sarà la “tela” e quella superiore il pannello di controllo.

Si può trascinare il mouse senza schiacciare tasti nella zona della tela senza lasciare traccia.

Se si schiaccia il pulsante sinistro si lascia una scia connessa rossa, se si schiaccia il pulsante destro si lascia una scia connessa blu.

Nel pannello di controllo vengono visualizzate le seguenti informazioni: posizione x,y del mouse, stato dei pulsanti del mouse (0,R,L) e numero di frame trascorsi dall'inizio dell'animazione (modulo 1000). Il layout del pannello di controllo è a piacere.

“COMPITO TIPO” N.4

a) Il nostro “universo” è composto da una sequenza lineare di $k=20$ cellette.

La prima celletta ha per vicini la seconda e la k -esima. La seconda ha per vicini la prima e la terza e così via.

Ogni celletta si trova in uno di due possibili stati: Rosso, Verde. Ad ogni passo temporale lo stato di ciascuna celletta cambia a seconda del suo stato nell'istante precedente e a seconda dello stato della celletta che lo segue e che lo precede. Le regole sono le seguenti:

- a) se sono Rosso e ho due vicini Verdi, divento Verde;
- b) se sono Rosso e ho due vicini Rossi, resto Rosso;
- c) se sono Rosso e ho un vicino Verde e uno Rosso, divento Verde;
- a) se sono Verde e ho due vicini Verdi, resto Verde;
- b) se sono Verde e ho due vicini Rossi, divento Rosso;
- c) se sono Verde e ho un vicino Verde e uno Rosso, divento Rosso;

Il nostro universo si può rappresentare graficamente come una sequenza di quadretti colorati adiacenti. Si inizializzi a caso un universo e lo si faccia evolvere per venti passi.

Si visualizzi lo stato dell'universo nei vari istanti disponendoli su righe successive ottenendo alla fine una visualizzazione di venti righe ciascuna composta di una sequenza di venti quadretti.

b) Si crei una canvas quadrata. In essa sono rappresentati due cerchi concentrici di raggio differente con centro corrispondente al centro della canvas. Lungo ciascun cerchio, come se fosse un binario, si muove un cerchietto colorato di raggio arbitrario. I due cerchietti sono connessi da un segmento.

Le posizioni iniziali dei due cerchietti sono casuali, ma i due cerchietti hanno la stessa velocità angolare. Ad ogni passo vengono aggiornate le posizioni dei due cerchietti e del segmento che li connette.

Quando si premono i tasti “a”/ “s” della tastiera il cerchietto che gira sul cerchio più piccolo diminuisce/ aumenta la propria velocità. La combinazione dei tasti “q”/”w” della tastiera agisce allo stesso modo sull'altro cerchietto.

Osservazione: si può scegliere se implementare questo esercizio direttamente o dotandosi di una classe “cerchietto” e utilizzandola. Nel caso di implementazione ad oggetti è facile generalizzare la proposta a k cerchietti che si muovono nelle loro “orbite”.