

Sistemi Multimediali e Laboratorio

Roberto Ranon

Esercitazione n.1 del 27/02/2018: semplici figure e colori, prime animazioni, uso di classi.

Nota: per svolgere questi esercizi, dovete aver scaricato e installato l'ultima versione di Processing (attualmente, la 3.3.6, ma qualsiasi versione dalla 2.x dovrebbe andare bene). Se state lavorando su una delle macchine del laboratorio, dovrete trovare Processing tra i programmi disponibili in ambiente Windows. Se ci sono problemi, avvisatemi durante la lezione o con un mail a roberto.ranon@uniud.it.

Le slide che presentano brevemente i concetti e i comandi necessari sono disponibili all'indirizzo <https://users.dimi.uniud.it/~roberto.ranon/slides/processing-intro.pdf> (o sul sito elearning.uniud.it). Per maggiori informazioni, consultate la guida di riferimento di Processing: basta scrivere nell'editor l'istruzione, selezionarla, cliccare su di essa con il tasto destro, e scegliere "Find in Reference".

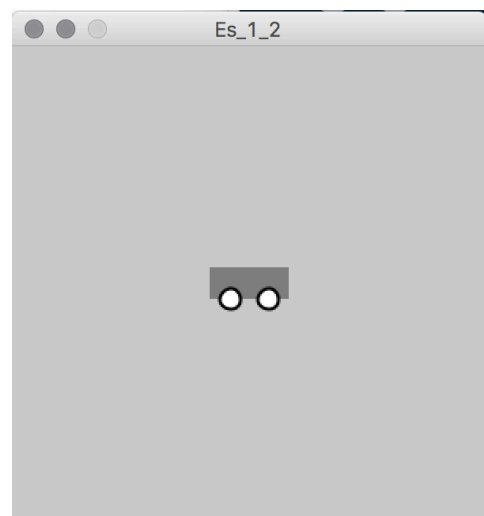
Ci aspettiamo che, al termine del laboratorio, abbiate completato almeno gli esercizi fino al n. 8. Se non è così, cercate di completarli a casa.

1. (verificare che Processing funzioni) Scrivete ed eseguite il seguente sketch:

```
fill(0,0,255);  
rect(0,0,100,200);  
fill(255,0,0,255);  
rect(0,0,200,40);  
fill(255,0,0,191);  
rect(0,50,200,40);
```

Dopo aver eseguito il programma, create un'applicazione e verificate il suo funzionamento.

2. Disegnate l'automobile in figura tramite una sequenza di istruzioni. per quanto riguarda il sistema di coordinate usato da Processing. Le funzioni che vi servono sono **size**, **rect**, **ellipse**, **fill**, **stroke**, **strokeWeight**, **noStroke**, ed eventualmente **rectMode**,



3. Ristrutturate lo sketch precedente, in modo che sia composto da due funzioni:

- **void setup()**, che imposta la dimensione della finestra, e
- **void draw()**, contenente il codice per il disegno dell'automobile.

Lo sketch di questo esercizio e quello dell'esercizio precedente dovrebbero ottenere il medesimo risultato, ma ora l'automobile verrà ridisegnata ad ogni aggiornamento della finestra. Ricordate che la funzione **setup()** viene eseguita una volta all'avvio dello sketch, la funzione **draw()** viene chiamata circa 60 volte al secondo fino al termine del programma.

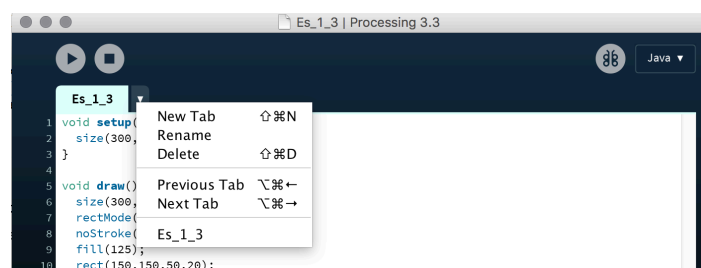
4. Modificate lo sketch precedente in modo che l'automobile: (i) appaia all'inizio vicino al bordo sinistro della finestra; (ii) si sposti verso destra di 1 pixel ad ogni ridisegno della finestra (quindi, circa 60 pixel al secondo). Vi servirà la funzione **background()**, di cui potete trovare qualche esempio d'uso nelle slide o una spiegazione completa nella guida di riferimento di Processing. Per svolgere l'esercizio, vi conviene prima riscrivere l'esercizio precedente in modo che sia costituito da tre funzioni: **setup()**, **draw()**, e una funzione che disegna l'automobile alle coordinate fornite come argomenti.
5. Modificate lo sketch precedente in modo che l'automobile, una volta uscita dal bordo destro della finestra, ricompaia dal bordo sinistro.
6. Visto che quando programiamo in Processing stiamo in realtà scrivendo codice Java, possiamo definire classi e utilizzare oggetti (uno sketch Processing è in realtà una sottoclasse di [PApplet](#)). Ad esempio, se nel nostro sketch usiamo più volte un certo elemento grafico, possiamo definire una classe per rappresentarlo. Create dunque una classe **Car**, con:

- due variabili intere, **x** e **y**, che specificano la posizione dell'automobile;
- un costruttore, che permette di specificare la posizione iniziale dell'automobile,
- un metodo **position(int x, int y)**, per impostare la posizione dell'automobile;
- un metodo **display()**, che disegna l'automobile nella posizione specificata da **x** e **y**

Naturalmente, i nomi delle variabili e dei metodi sono solo dei suggerimenti, in quanto del tutto arbitrari.

La classe **Car** può essere definita all'inizio dello sketch, oppure in un file separato, da aggiungere allo sketch tramite la freccia in basso sotto la toolbar, e il comando "New Tab". **Importante:** date al file il nome della classe che state per definire. Dopo aver creato la classe **Car**, modificate il codice

dell'esercizio precedente in modo che, per disegnare l'automobile che si sposta da sinistra verso destra, crei e usi un oggetto della classe **Car**.



7. Scrivete uno sketch che disegni 20 automobili (ferme) in posizioni casuali, all'interno della finestra. Nelle slide, troverete una spiegazione di base della funzione **random**. In particolare, **random(n)** ritorna un numero reale casuale tra 0 (incluso) e **n** (non incluso), mentre **random(low, high)** genera un numero reale casuale tra **low** (incluso) e **high** (non incluso). **Attenzione:** potreste dover convertire il risultato ritornato da **random** al tipo di dato **int**. Verificate il corretto funzionamento eseguendo il programma più volte: ad ogni esecuzione, le auto dovrebbero trovarsi in posizioni diverse.
8. Modificate lo sketch precedente in modo che le 20 automobili si muovano verso destra con velocità costante di circa 60 pixel al secondo, e ricompaiano dal bordo sinistro non appena uscite dal bordo destro. **Suggerimento:** usate un array di oggetti di tipo **Car** per poter accedere alle 20 auto e spostarle. In generale, gli oggetti vanno creati una volta sola, e quindi, tipicamente, nel metodo **setup**.
9. Modificate lo sketch precedente in modo che:
- ogni auto venga disegnata con un colore casuale. Per memorizzare il colore di un'auto, potete usare il tipo di dato **color**, ad es.
color c1 = color(204, 153, 0);
 - ogni auto si muova con una velocità stabilita all'inizio in modo casuale, tra 1 e 4 pixel per frame (cioè, per chiamata della funzione **draw**).