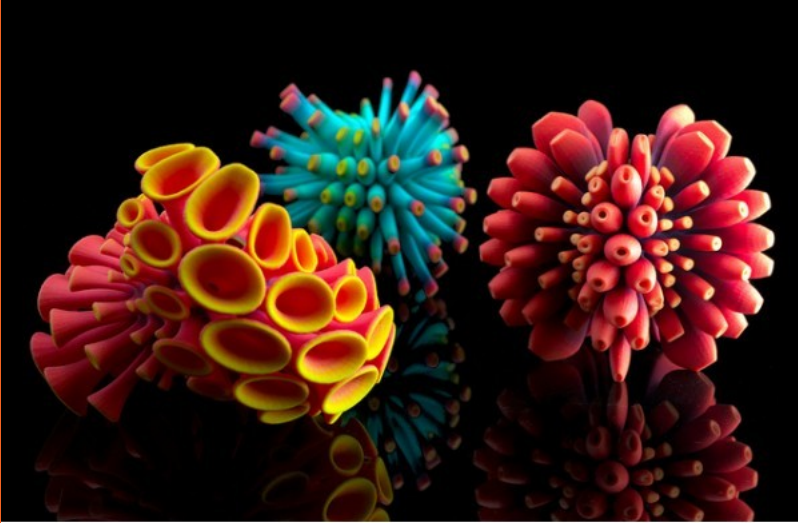


Processing



di Roberta Molinari



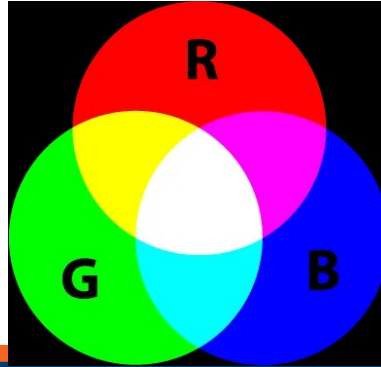
Processing

- Linguaggio OOP basato su Java
- Nato per arti visive
- Semplice
- Interfacciabile con Arduino
- Licenza libera
—GPL

Esempio iterazione con un kinect



Modello del colore RGB o modello additivo



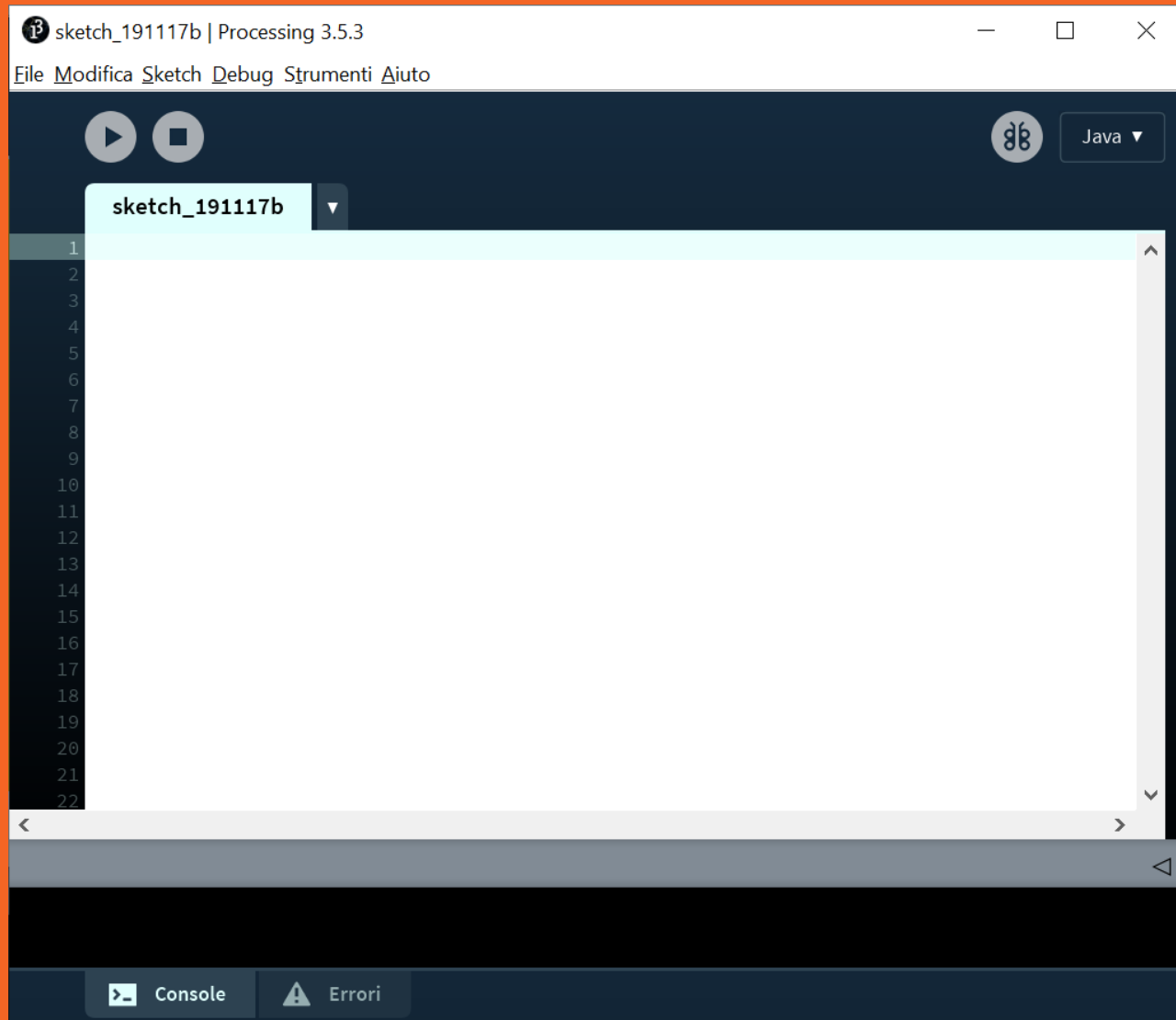
Combinando la luce Red Green e Blue si
ottengono tutti i possibili colori.

Immagina in una stanza buia 3 lampade con
luce: rossa verde e blu.

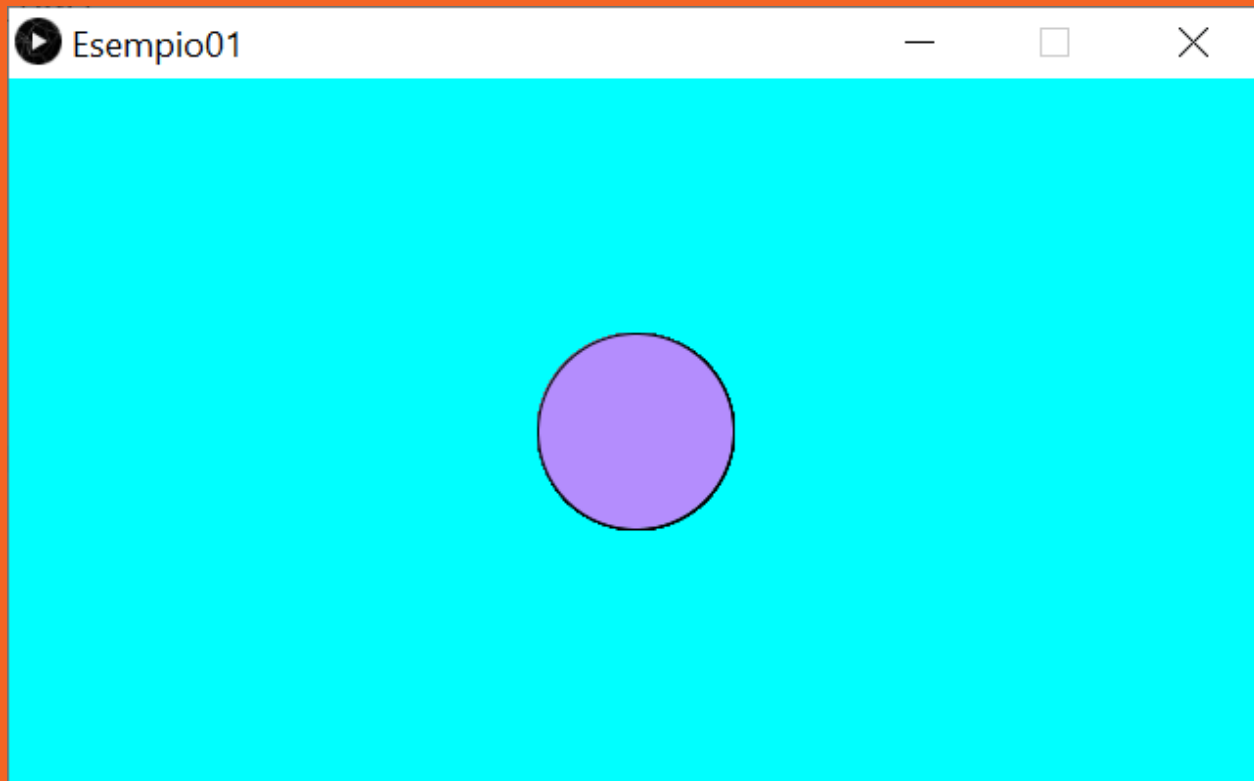
Tutte spente → nero

Tutte accese → bianco

L'IDE



Primo esercizio: cerchio che cambia colore



Classe Cerchio

Cerchio
<ul style="list-style-type: none">- dim: float- DIM_DEFAULT: float =100- cordX: float- cordY: float- colore: color- bordo: color
<ul style="list-style-type: none">+ Cerchio()+ show()+ set/getDim()+ set/getCordX()+ set/getCordX()+ set/getColore()+ set/getBordo()

Implementazione: Classe Cerchio

```
public class Cerchio {  
  
    private float dim;    //diametro cerchio  
    private final static int DIM_DEFAULT = 100;  
    private float cordX;  //coordinate del centro  
    private float cordY;  
    private color colore; //colore riempimento  
    private color bordo;  //colore bordo  
}
```


Implementazione: Classe Cerchio

```
public Cerchio(){  
    //piazzato al centro della finestra  
    cordX = width / 2;  
    cordY = height / 2;  
    //colore di default bianco con bordo nero  
    colore = color(255, 255, 255);  
    bordo = color(0, 0, 0);  
    dim = DIM_DEFAULT;  
}
```

Implementazione: Classe Cerchio

```
public void show() {  
    stroke(bordo); //colore bordo  
    fill(colore); //colore di riempimento  
    circle(cordX, cordY, dim);  
        //centro X,Y e diametro dim  
}
```

—

```
public void setDim(float dim){  
    if (dim >= 0) this.dim = dim; }  
  
public float getDim(){ return dim; }  
  
public void setCordX(float x){ cordX = x; }  
  
public void setCordY(float y){ cordY = y; }  
  
public float getCordX(){ return cordX; }  
  
public float getCordY(){ return cordY; }  
  
public void setColore(color colore){  
    this.colore = color(colore); }  
  
public color getColore(){ return colore;}  
  
public color getBordo(){ return bordo; }  
  
public void setBordo(color colore){bordo = colore;}
```

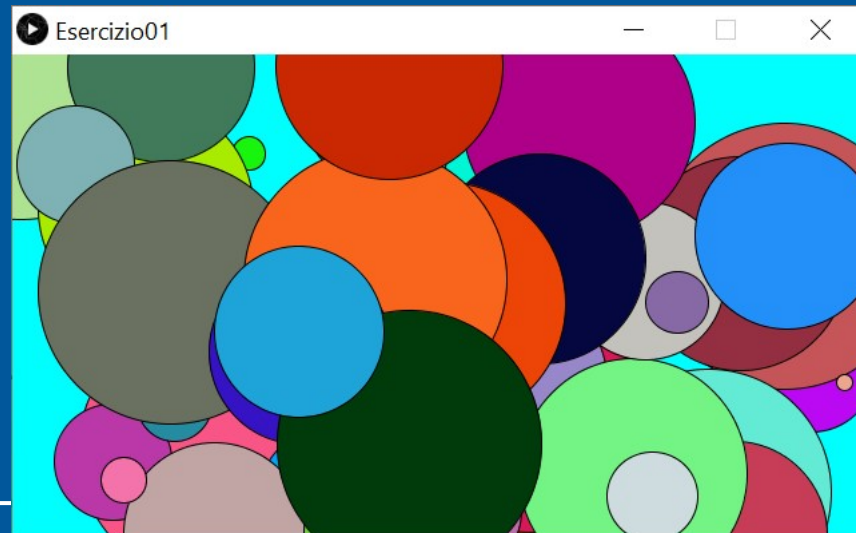
– Programma principale

```
private Cerchio c;  
  
public void settings() { size(640, 360); }  
  
public void setup() {  
    background(0,255,255); //sfondo azzurro  
    c = new Cerchio();  
}
```

Programma principale

```
public void draw(){
    c.show();
    delay(100);
    //creazione di un colore a caso
    //random restituisce un float [0..256)
    c.setColore(color(random(256),
                      random(256), random(256)));
    //stampa a console
    println("colore: ",
           red(c.getColore()), "-",
           green(c.getColore()), "-",
           blue(c.getColore()));
}
```

**Modificare il programma in modo
che si creino infiniti cerchi di colore
diverso (fisso) di dimensione
variabile da 10 a 200 e in qualunque
punto della finestra**



**Modificare il programma in modo
che si creino 4 cerchi distribuiti su 2
righe e 2 colonne
(generalizzalo in modo che cambiando
numero di cerchi e/o numero di righe, si
distribuiscano sempre in modo
opportuno)**

