

Übungsblatt 4- EIPNF WS

Aufgabe 1:

Gegeben sei die Klasse Ball. Die Klasse besitzt die Felder: x , y und r , die Position und Größe bestimmen und die Felder dx und dy , die die Bewegungsrichtung beschreiben.

Dem Konstruktor werden Parameter für Position und Größe übergeben, die Bewegungsrichtung wird zufällig generiert.

Die Methode `move` bewegt den Ball und ändert gegebenenfalls bei Kollision mit den Seiten die Richtung. `plot` zeichnet einen Kreis an der aktuellen Position mit dem entsprechenden Radius.

1. Deklariere zwei Objekte der Klasse Ball
2. Initialisiere diese in `setup`
3. Führe die Methoden `move` und `plot` auf beiden Objekten in `draw` aus

```
class Ball {
  float dx,dy,x,y,r;
  public Ball (float x, float y, float r) {
    this.x = x;
    this.y = y;
    this.r = r;
    dx = random(-5,5);
    dy = random(-5,5);
  }

  void move () {
    x += dx;
    y += dy;
    if (x+r>width) {
      x = width-r;
      dx *= -1;
    } else if (x-r<0) {
      x = r;
      dx *= -1;
    }
    if (y+r>height) {
      y = height-r;
      dy *= -1;
    } else if (y-r<0) {
      y = r;
      dy *= -1;
    }
  }

  void plot () {
    ellipse(x,y,2*r,2*r);
  }
}
```

Aufgabe 2:

Erweitere nun die Klasse Ball um folgende Methoden:

1. `calcDistance` : diese Methode berechnet den Abstand zu einem anderen Ball. Übergebe dafür als Parameter ein Objekt der Klasse Ball und berechne mit Hilfe des Satzes von Pythagoras ($a^2 + b^2 = c^2$) die Distanz.

Das Quadrat einer Zahl lässt sich mit der Funktion `sq(2)`; (->4) und die Wurzel mit `sqrt(4)`; (->2) berechnen.

Die berechnete Distanz wird als `float` zurückgegeben.

2. Erweitere die vorherige Aufgabe: Überprüfe in `draw` vor den Aufrufen von `move` und `plot` ob die Distanz zwischen beiden Bällen kleiner als die Summe ihrer Radien ist.

Falls dies der Fall ist wird die Füllfarbe auf rot gesetzt, andernfalls auf weiß.