Chapter 8

More with classes

Practice with Zoog

```
// Another simple class – practical example
class BankAccount {
  int id;
  float amount;
  String name;

  BankAccount(int i, float amt, String nm){
    id = i;
    amount = amt;
    name = nm;
  }

  void deposit(float amt) {
    amount = amount + amt;
  }
  void withdraw(float amt) {
    amount = amount - amt;
  }
}
```

```
BankAccount steph, klay, sandra;
void setup(){
  steph = new BankAccount(120, 24.00, "Steph C.");
  klay = new BankAccount(121, 30.00, "Klay T.");
  sandra = new BankAccount(122, 25.00, "Sandra P.");

  steph.deposit(3.50);
  steph.withdraw(10.00);

  println(steph.amount);
}
```

What is the amount printed by the object steph at the end of setup?

```
// Another simple class
class MovingSquare {
   int x, y, sLength;
   int xVel, yVel;

   void update(){
      if(x > height - sLength || x < 0)
         xVel = xVel * -1;
      if(y > height - sLength || y < 0)
         yVel = yVel * -1;
      x = x + xVel;
      y = y + yVel;
      rect(x,y,sLength,sLength);
   }
}
```

```
void draw(){
   background(100);
   square1.update();
   square2.update();
}
```

```
MovingSquare square1, square2;
void setup(){
   size(400,400);
   square1 = new MovingSquare();
   square2 = new MovingSquare();
   square1.x = 0;
   square1.y = height/2;
   square1.sLength = 5;
   square2.sLength = 10;
   square2.x = height/2;
   square2.y = 0;
   square1.xVel = 2;
   square1.yVel = 2;
   square2.xVel = 1;
   square2.yVel = 1;
}
```

Create a constructor for the MovingSquare class such that x, y, sLength, xVel, and yVel can all populated in one statement (as opposed to the 5 for each object here).

```
// A Constructor added to the class
class MovingSquare {
   int x, y, sLength;
   int xVel, yVel;
   MovingSquare(int xx, int yy, int sL, int xv, int yv){
      x = xx;
      y = yy;
      sLength = sL;
      xVel = xv;
      yVel = yv;
   }
   void update(){
   . . .
   }
}
```

```
MovingSquare square1, square2;
void setup(){
   size(400,400);
   square1 = new MovingSquare(0,height/2,5,2,2);
   square2 = new MovingSquare(height/2,0,10,1,1);
}
```

Detecting Collision

Write a method called collide that takes in a MovingSquare object, and determines whether the MovingSquare parameter has collided with the MovingSquare object calling the method. If the two objects have collided, then reverse direction of the MovingSquare objects.

The instance variables you're checking are x and y, and the instance variables you're potentially changing are xVel and yVel.

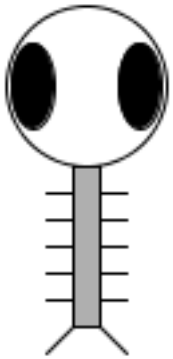
```
void collide(MovingSquare ms){
//code goes here

}
```

```
void collide(MovingSquare b){
   if((abs(b.x - x) < sLength) && (abs(b.y - y) < sLength))
   {
      if(b.xVel*xVel < 0){
        b.xVel = b.xVel * -1;
        xVel = xVel * -1;
      }
      if(b.yVel*yVel < 0){
        b.yVel = b.yVel * -1;
        yVel = yVel * -1;
      }

   }
}
```

The Zoog class



```
// Zoog's instance variables

float x, y, w, h, eyeSize;
```

```
// Zoog's constructor


Zoog(float tempX, float tempY, float tempW, float
tempH, float tempEyeSize) {
   x = tempX;
   y = tempY;
   w = tempW;
   h = tempH;
   eyeSize = tempEyeSize;
}
```

```
// Zoog's instance method (jiggle)

// Move Zoog
void jiggle(float speed) {
   // Change the location of Zoog randomly
   x = x + random(-1, 1)*speed;
   y = y + random(-1, 1)*speed;
   // Constrain Zoog to window
   x = constrain(x, 0, width);
   y = constrain(y, 0, height);
}
```

```
// Zoog's instance method (display)

// Display Zoog
void display() {
   // Set ellipses and rects to CENTER mode
   ellipseMode(CENTER);
   rectMode(CENTER);
   // Draw Zoog's arms with a for loop
   for (float i = y - h/3; i < y + h/2; i += 10) {
      stroke(0);
      line(x - w/4, i, x + w/4, i);
   }
   // Draw Zoog's body
   stroke(0);
   fill(175);
   rect(x, y, w/6, h);
   . . .
```

```
// Zoog's instance method (display)
   . . .
   // Draw Zoog's head
   stroke(0);
   fill(255);
   ellipse(x, y - h, w, h);

   // Draw Zoog's eyes
   fill(0);
   ellipse(x - w/3, y - h, eyeSize, eyeSize*2);
   ellipse(x + w/3, y - h, eyeSize, eyeSize*2);

   // Draw Zoog's legs
   stroke(0);
   line(x - w/12, y + h/2, x - w/4, y + h/2 + 10);
   line(x + w/12, y + h/2, x + w/4, y + h/2 + 10);
}
```

```
// Declaring and initializing Zoog. This is called
// Instantiation


. . .
Zoog zoog;
void setup() {
   size(200, 200);
   zoog = new Zoog(100, 125, 60, 60, 16);
}


. . .
```

```
// Declaring and initializing Zoog. This is called
// Instantiation


. . .
void draw() {
   background(255);

   // mouseX position determines speed factor
   float factor = constrain(mouseX/10, 0, 5);
   zoog.jiggle(factor);
   zoog.display();
}
. . .
```

```
Zoog zoog1, zoog2, zoog3;
void setup(){
   size(600,600);
   zoog1 = new Zoog(100, 125, 60, 60, 16);
   zoog2 = new Zoog(50, 50, 60, 60, 16);
   zoog3 = new Zoog(150, 150, 60, 60, 16);
}
void draw(){
   float factor = constrain(mouseX/10, 0, 5);
   zoog1.jiggle(factor);
   zoog2.display();
   zoog3.display();
}
```

How many Zoogs will show up in the window?

A. 0

B. 1

C. 2

D. 3

E. 4

```
Zoog zoog1, zoog2, zoog3;
void setup(){
   size(600,600);
   zoog1 = new Zoog(100, 125, 60, 60, 16);
   zoog2 = new Zoog(50, 50, 60, 60, 16);
   zoog3 = new Zoog(150, 150, 60, 60, 16);
}
void draw(){
   float factor = constrain(mouseX/10, 0, 5);
   zoog1.jiggle(factor);
   zoog2.display();
   zoog3.display();
}
```

Will the zoogs that show up be moving or stationary?

A. Moving

B. Stationary

More methods practice…

Fill in the blank so that this function returns true if the circle at x1, y1, with radius r1, has collided with the circle at x2, y2, with radius r2.

```
boolean collided(int x1, int y1, int r1, int x2, int
y2, int r2){
   return _____;
}
```

A. dist(x1, y1, x2, y2) >= r1+r2

B. abs(x1-x2) >= r1+r2 && abs(y1-y2) >= r1+r2

C. A and B will both work

D. Neither A nor B work but one would be correct if the >= was <=

E. Neither A nor B work but both would be correct if the >= was <=

What is printed by the following program?

```
void setup() {
   println("func1 = " + func1(func2(5)));
}
int func1(int x) {
   println("func1 x = " +x);
   return 2*x;
}
int func2(int x) {
   int y = func1(x);
   println("func2 x = " + x);
   return y;
}
```

What is printed by the following program?

```
int x=5;
void setup() {
   int z = timesTwo(x);
   println(x); //which x is being printed?
   println(z);
}
int timesTwo(int y) {
   int x = y * 2;
   println(x);
   return x;
}
```