CMSC141: Object-Oriented Programming

Lab 4: A Functional Game

due October 12th/13th, 2017

This lab assignment asks you to modify a simple bouncing-ball game. Your modifications should improve the game in some sense. After your changes, the game should be more fun to play.

WARM-UP: Go through the `draw` function with a partner and add comments to describe what it is doing.

Make the following changes to the provided sketch:

1. reorganize the sketch using functions (e.g. `drawPaddle`, `moveBall`);
2. make a change in terms of the appearance of the game;
3. a point tally should be displayed (you decide how points are awarded or deducted);
4. consider creating a second paddle and score;
5. in pairs, consider using a makey-makey to provide an alternate interface to your game.

Learning Objectives

- ❏ Modify an existing program.
- ❏ Refactor code to use functions.
- ❏ Reflect on a minimalistic game.

Deliverables

- ❏ Your program should start with a comment that includes your name, email, date, assignment description & collaboration statement.
- ❏ Bring a hardcopy of your programs (the source code, not the graphics) to your next lab period.
- ❏ Be prepared to run the Processing sketches and demonstrate your "Theory of the Program."

```
float paddle_x;
float paddle_step;
float paddle_h = 16;
float paddle_w = 5 * paddle_h;
float ball_x, ball_y;
float ball_x_step, ball_y_step;
float ball_r = 13;

void setup() {
  size(600, 300);
  paddle_x = width/2;
  paddle_step = 0;
  reset();
}

void reset() {
  ball_x = random(ball_r, width - ball_r);
  ball_y = random(ball_r, height/2 );
  ball_x_step = random(-3, 3);
  ball_y_step = random(1, 3);
}

void keyPressed() {
  if (key == CODED && keyCode == LEFT) {
    paddle_step = -3;
  } else if (key == CODED && keyCode == RIGHT) {
    paddle_step = 3;
  } else if (key == ' ') {
    reset();
  }
}

void keyReleased() {
  paddle_step = 0;
}

// https://yal.cc/rectangle-circle-intersection-test/
// returns true if a rectangle (rx, ry, rw, rh) intersects with a circle
(cx,cy,cr)
boolean rectIsOnCircle(float rx, float ry, float rw, float rh,
                       float cx, float cy, float cr) {
  float nx = max(rx, min(cx, rx + rw));
  float ny = max(ry, min(cy, ry + rh));
  float dx = cx - nx;
  float dy = cy - ny;
  return dx * dx + dy * dy < cr * cr;
}
```

```
void draw() {
  background(196);

  // COMMENT
  paddle_x = paddle_x + paddle_step;

  // COMMENT
  if (ball_x - ball_r < 0 || ball_x + ball_r > width) {
    ball_x_step = -ball_x_step;
  }

  // COMMENT
  if (ball_y - ball_r < 0) {
    ball_y_step = -ball_y_step;
  }

  // COMMENT
  if (rectIsOnCircle(paddle_x, height - paddle_h, paddle_w, paddle_h,
  ball_x, ball_y, ball_r)) {
    ball_y_step = -ball_y_step;
  }

  // COMMENT
  ball_x = ball_x + ball_x_step;
  ball_y = ball_y + ball_y_step;

  // COMMENT
  noStroke();
  fill(196, 0, 0);
  ellipse(ball_x, ball_y, ball_r*2, ball_r*2);

  // COMMENT
  stroke(24);
  fill(64);
  rect(paddle_x, height-paddle_h, paddle_w, paddle_h);
}
```