IDEA135: Games at Work: Procedure, Participation, and Play

# Sketch 8: p5 pong
# Due: Monday, February 27, 2017, 1:30 PM

**Create a simple ball bouncing game using `p5.js`.**

## Rules

1. Fork the sketch on github (also available at http://jsbin.com/jikavil).
2. A *score* should be visualized; you decide how points are awarded or deducted.
3. The number of *lives* should be visualized; the player should have a limited number of lives.
4. Add any other game elements you'd like:
    a. Make a change in terms of the static appearance of the game.
    b. Change the speed or size of the ball.
    c. Change the speed, size or controls of the paddle.
    d. Add another paddle.
    e. Create another type of bouncing object.
    f. Provide a *game over* screen.

## Learning Objectives

1. Modify an existing program.
2. Reflect on a minimalistic digital game.
3. Manage state in a game.
4. More practice employing interactivity with p5.js.

When you are finished, reflect in a comment at the top of your sketch about what you were trying to accomplish with your game.

```
/*
 * Skeleton of a ball bouncing /game/.
 * Keith O'Hara <kohara@bard.edu>
 */

var paddle_x, paddle_y;
var paddle_w, paddle_h;
var paddle_step;

var ball_x, ball_y;
var ball_r;
var ball_x_step, ball_y_step;


function setup() {
  createCanvas(600, 300);
  paddle_h = 16;
  paddle_w = 6 * paddle_h;
  paddle_x = width / 2;
  paddle_y = height - paddle_h;
  paddle_step = 0;
  ball_r = 13;
  reset();
}

function draw() {
  background(196);

  // move paddle
  //paddle_x += (mouseX - paddle_x) * .1;
  paddle_x = paddle_x + paddle_step;

  // is the ball hitting the right or left wall?
  if (ball_x - ball_r < 0 || ball_x + ball_r > width) {
    ball_x_step = -ball_x_step;
  }

  // hitting the top?
  if (ball_y - ball_r < 0) {
    ball_y_step = -ball_y_step;
  }

  // hitting the paddle?
  if (ball_y + ball_r > paddle_y) {
    if (ball_x >= paddle_x && ball_x <= paddle_x + paddle_w) {
      ball_y_step = -ball_y_step;
      ball_y = paddle_y - ball_r;
    }
  }

  // move ball by ball_x_step and ball_y_step
  ball_x = ball_x + ball_x_step;
  ball_y = ball_y + ball_y_step;

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

  // draw paddle
  stroke(24);
  fill(64);
  rect(paddle_x, paddle_y, paddle_w, paddle_h);
}
```

```
function 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);
}

function keyPressed() {
  if (keyCode == LEFT_ARROW) {
    paddle_step = -3;
  } else if (keyCode == RIGHT_ARROW) {
    paddle_step = 3;
  } else if (key == ' ') {
    reset();
  }
}

function keyReleased() {
  paddle_step = 0;
}
```