

Solutions for Section #2

Portions of this handout by Eric Roberts, Patrick Young, Jeremy Keeshin, Mehran Sahami, Nick Troccoli, and Kat Gregory

Solution 1: Bouncing Balls

```
function createNewBall() {
  let ball = GOval(0, 0, DIAMETER, DIAMETER);
  ball.setFilled(true);
  ball.setColor(randomColor());
  return ball;
};

/* Main program */
function BouncingBalls() {
  let gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT);
  let clickAction = function(e) {
    let xv = randomReal(MIN_X_VEL, MAX_X_VEL),
        yv = 0,
        ball = createNewBall();
    gw.add(ball);

    let checkForCollision = function() {
      // Determine if ball has dropped below the floor
      if (ball.getY() > GWINDOW_HEIGHT - DIAMETER) {
        // Change ball's Y velocity to now bounce upwards
        yv = -yv * BOUNCE_REDUCE;

        // Assume bounce will move ball an amount above the floor
        // equal to the amount it would have dropped below the floor.
        let diff = ball.getY() - (GWINDOW_HEIGHT - DIAMETER);
        ball.move(0, -2 * diff);
      }
    };

    let step = function() {
      if (ball.getX() < GWINDOW_WIDTH) {
        yv += GRAVITY;
        ball.move(xv, yv);
        checkForCollision();
      } else {
        clearInterval(timer);
      }
    };

    let timer = setInterval(step, TIME_STEP);
  };

  gw.addEventListener("click", clickAction);
}
```

Thought questions: Why does `clickAction` need to be defined within the `BouncingBalls` function, and why do `step` and `checkForCollision` need to be defined within `clickAction`?

Solution 2: Spoonerisms

```
function spoonerism(phrase) {
  let sp1 = phrase.indexOf(' ');
  let sp2 = phrase.lastIndexOf(' ');
  let orig1 = phrase.substring(0, sp1);
  let orig2 = phrase.substring(sp2 + 1);
  let middle = phrase.substring(sp1, sp2 + 1);

  let vp1 = findFirstVowel(orig1);
  let vp2 = findFirstVowel(orig2);
  let transformed1 = orig2.substring(0, vp2) + orig1.substring(vp1);
  let transformed2 = orig1.substring(0, vp1) + orig2.substring(vp2);
  return transformed1 + middle + transformed2;
}

/**
 * Function: findFirstVowel
 * -----
 * Returns the index of the first lowercase vowel, or -1 if no lowercase
 * vowel could be found.
 */
function findFirstVowel(str) {
  for (let i = 0; i < str.length; i++) {
    if (isEnglishVowel(str.charAt(i))) {
      return i;
    }
  }
}

/**
 * Function: isEnglishVowel
 * -----
 * Returns true if and only if the provided string is of length 1, and
 * its one character is a lowercase vowel.
 */
function isEnglishVowel(ch) {
  return ch.length === 1 && "aeiou".indexOf(ch) >= 0;
}
```

