```
//  Austin Matel
// 9/11/19
//  This is a comment
//  The setup function function is called once when your program
begins
var balls = [];
var paddle;
var gameState = 1;
var difficulty, health, score;
var win = 2;
var btnEasy, btnMedium, btnHard, btnInstructions;
var winScore;
//setup
function setup() {
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
  makeButtons();
}
//loads balls and paddles
function loadObjects(n){
  for (var i = 0; i < n; i++){
    balls[i] = new Ball(random(width), random(0,200), random(-3,3),
random(-3,3));
  }
  paddle = new Paddle(width/2, 700);
}
//title screen
function startGame(){
  win = 2;
  runButtons();
  background(200,70,100);
  runButtons();
  textSize(90);
  fill(20,20,20);
  textFont('Georgia')
  text("Paddle Game",150,200);
  fill(random(0,255), random(0,255), random(0,255));
  ellipse(340, 300, 100, 100);
  fill(random(0,255), random(0,255), random(0,255));
  ellipse(460, 300, 100, 100);
  fill(random(0,255), random(0,255), random(0,255));
  arc(400, 400, 80, 80, 0, PI, CHORD);
}
```

```
//end screen
function endGame(){
  fill(255);
  textFont('Times New Roman');
  textSize(90);
  if (win === 1){
    background(10, 200, 100);
    text("You won", 200, 200);
    btnMenu = new Button(250, 600, 300, 100, "Menu", color(10,255,10),
5);
    btnMenu.run();
    //smiley face
    fill(random(0,255), random(0,255), random(0,255));
    ellipse(340, 300, 100, 100);
    fill(random(0,255), random(0,255), random(0,255));
    ellipse(460, 300, 100, 100);
    fill(random(0,255), random(0,255), random(0,255));
    arc(400, 400, 80, 80, 0, PI, CHORD);
    }else{
      background(255, 20, 10);
      text("Game Over", 150, 200);
      btnMenu = new Button(250, 600, 300, 100, "Menu",
color(10,255,10), 5);
      btnMenu.run();
      //frowny face
      fill(random(0,255), random(0,255), random(0,255));
      ellipse(340, 300, 100, 100);
      fill(random(0,255), random(0,255), random(0,255));
      ellipse(460, 300, 100, 100);
      fill(random(0,255), random(0,255), random(0,255));
      arc(400, 400, 80, 80, PI, 0, CHORD);

    }

}
//loads the buttons
function makeButtons(){
  btnEasy = new Button(50, 500, 100, 75, "Easy", color(1, 255, 1), 1);
  btnMedium = new Button(250, 500, 100, 75, "Medium", color(255, 255,
1), 2);
  btnHard = new Button(450, 500, 100, 75, "Hard", color(255, 1, 1),
3);
```

```
  btnInstructions = new Button(650, 500, 100, 75,
"Instructions",color(0, 255, 255), 4);
}
//shows buttons
function runButtons(){
  btnEasy.run();
  btnMedium.run();
  btnHard.run();
  btnInstructions.run();
}
//game code
function playGame(){
  background(128, 128, 128, 50);
  runObjects();
  removeBall();
  if (health <= 0){
    win = 0;
    gameState = 3;
  }
  if (win === 1){
    gameState = 3;
  }
}

//  The draw function is called @  30 fps
//displaying game states
function draw() {
  if (gameState === 1){
    startGame();
  }
  if (gameState === 2){
    playGame();
  }
  if (gameState === 3){
    endGame();
  }
}

function runObjects(){
  for(var t = 0; t < balls.length; t++){
    balls[t].run();
  }
  fill(255);
  textSize(20);
  text("Score = "+ score, 20, 50);
  text("Health = "+ health, 700, 50);
  paddle.run();
}
//if the ball hits the underside of the paddle, the ball will
disappear
function removeBall(){
  for (var i = balls.length - 1; i >= 0; i--){
    if (balls[i].isColliding()){
      balls.splice(i,1);
      health = health -1;
    }
  }
}

//Austin Matel
//8/21/19
class Ball{
  constructor(x, y, dx, dy){
    this.loc = createVector(x, y);
    this.vel = createVector(dx, dy);
    this.clr = color(random(225), random(225), random(225));
    this.w = 30;
    this.ballx = dx;
    this.acc = createVector(-0.05,0.05);
  }
  //initializes all methods
  run(){
    this.checkEdges();
    this.update();
    this.render();
    this.isColliding();
  }
  //keeps the balls on the screen or teleport to the other side of the
screen
  checkEdges(){
    if (this.loc.x < 0){
      this.loc.x = width;
    }
    if (this.loc.x > width){
      this.loc.x = 0;
    }
    if (this.loc.y < 0){
      this.vel.y = -(this.vel.y);
    }
    if (this.loc.y > height){
        this.vel.y = -(this.vel.y);
    }
    //checks to see if the ball touches the paddle and makes it bounce
off
    if (this.loc.x > paddle.loc.x && this.loc.x < paddle.loc.x +
paddle.w && this.loc.y >= paddle.loc.y - 10 && this.loc.y <=
paddle.loc.y + paddle.h && this.vel.y > 0){
      this.vel.y = -(this.vel.y);
      score = score + 1;
      if (score === winScore){
        win = 1;
      }
    }
  }
  //allows the balls to move and accelerate
  update(){
    this.vel.add(this.acc);
    this.loc.y = this.vel.y + this.loc.y;
    this.loc.x = this.loc.x + this.ballx;
  }
  //creates the size and shape of the ball
  render(){
    fill(this.clr);
    ellipse(this.loc.x, this.loc.y, this.w, this.w)
  }
  //checks to see if the ball is touching the underside of the paddle
  isColliding(){
    if (this.loc.x > paddle.loc.x && this.loc.x < paddle.loc.x +
paddle.w && this.loc.y > paddle.loc.y && this.loc.y < paddle.loc.y +
paddle.h && this.vel.y < 0){
      return true;
    }else{
      return false;
    }
  }
}

//Austin Matel
//8/21/19
```

```
//writing parameters so that the paddle follows the mouse
class Paddle{
  constructor(x, y){
    this.loc = createVector(mouseX, mouseY);
    this.clr = color(random(225), random(225), random(225));
    this.w = 150;
    this.h = 35;
    this.acc = createVector(0,0);
  }
//shows the balls and makes sure it moves and stays on the screen
  run(){
    this.render();
    this.checkEdges();
    this.update();
  }
//makes sure the paddle stays on the screen
  checkEdges(){
    if (this.loc.x < 0){
      this.loc.x = 0;
    }
    if ((this.loc.x + this.w) > width){
      this.loc.x = width - this.w;
    }
  }
  //makes the paddle move with the mouse
  update(){
    var mouseLoc = createVector(mouseX - 50, 650);
    this.loc = p5.Vector.lerp(this.loc, mouseLoc, 0.09);
  }
  //builds the paddle based on parameters
  render(){
    fill(this.clr);
    rect(this.loc.x, 650, this.w, this.h);
  }
}

//Austin Matel
//8/21/19
//creating parameters
class Button{
  constructor(x, y, w, h, msg, clr, id){
    this.loc = createVector(x,y);
    this.h = h;
    this.w = w;
    this.msg = msg;
    this.clr = clr;
    this.id = id;
  }
  //actually creating buttons
  run(){
    this.render();
    this.isPressed();
  }
  //drawing buttons
  render(){
    fill(this.clr);
    rect(this.loc.x, this.loc.y, this.w, this.h, 20);
    fill(10);
    textSize(20);
    if (this.id === 4){
      textSize(18);
      text(this.msg, this.loc.x, this.loc.y + 30);
    }else{
      textSize(20);
      text(this.msg, this.loc.x + 10, this.loc.y + 30);
    }
  }
  //checks if the button is pressed and makes different buttons do
different things
  isPressed(){
    if (mouseX > this.loc.x && mouseX < this.loc.x + this.w && mouseY
> this.loc.y && mouseY < this.loc.y + this.h && mouseIsPressed){
      if(this.id == 1){
        gameState = 2;
        difficulty = 5;
        health = 5;
        score = 0;
        winScore = 10;
        loadObjects(difficulty);
      }
      if(this.id == 2){
        gameState = 2;
        difficulty = 10;
        health = 5;
        score = 0;
        winScore = 15;
        loadObjects(difficulty);
      }
      if(this.id == 3){
        gameState = 2;
        difficulty = 15;
        health = 3;
        score = 0;
        winScore = 20;
        loadObjects(difficulty);
      }
      if(this.id == 4){
        fill(255);
        text("Click on a difficulty and try to catch the balls on the
top of the paddle.", 90, 650);
        text("If a ball hits the bottom of your paddle, you will lose
health.", 90, 670);
        text("If you catch a certain amount balls before running out
of health, you win!", 90, 690);
      }
      if(this.id == 5){
        gameState = 1;
        balls = [];
      }
    }
  }
}
```

```
//Austin Matel
//8/21/19
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> Paddle Game </title>
    <script src="libraries/p5.js" type="text/javascript"></script>
    <script src="libraries/p5.dom.js" type="text/javascript"></script>
    <script src="libraries/p5.sound.js"
type="text/javascript"></script>
    <script src="sketch.js" type="text/javascript"></script>
    <script src="ball.js" type="text/javascript"></script>
    <script src="paddle.js" type="text/javascript"></script>
    <script src="button.js" type="text/javascript"></script>
    <style> body {padding: 0; margin: 0;} canvas {vertical-align:
top;} </style>
  </head>


  <body>
  </body>
</html>
```