```
var stillPlaying = true; //Says that you're still playing the game
var currentScene = 0; //Sets the current scene
var bitmojiX = 0; //Sets the bitmoji x value to 0
var bitmojiY = 0; //Sets the bitmoji y value to 0
var bitmoji2X = 0; //Sets the second bitmoji x value to 0
var bitmoji2Y = 0; //Sets the second bitmoji y value to 0
var CPUX = 150; //Sets the x value of the CPU to 150
var CPUY = 30; //Sets the y value of the CPU to 30
var puck1Speed = 2; //Sets the speed of the first level puck
var puck2Speed = 4; //Sets the speed of the second level puck
var puck3Speed = 6; //Sets the speed of the third level puck

//Both partners bitmoji code
//Josh's bitmoji code
var drawBitmojiFace = function (bitmojiX,bitmojiY,bitmojiHeight) {
noStroke();
fill(240, 210, 166);
ellipse(bitmojiX,bitmojiY,bitmojiHeight/150*80,bitmojiHeight/150*100); //head
fill(255,255,255);
ellipse(bitmojiX-(bitmojiHeight/150*15),bitmojiY-(bitmojiHeight/150*2),bitmojiHeight/150*12,bitm
ojiHeight/150*12); //left eye white
ellipse(bitmojiX+(bitmojiHeight/150*15),bitmojiY-(bitmojiHeight/150*2),bitmojiHeight/150*12,bitm
ojiHeight/150*12); //right eye white
fill(150,120,80);
ellipse(bitmojiX-(bitmojiHeight/150*14.5),bitmojiY-(bitmojiHeight/150*1.5),bitmojiHeight/150*4.5,
bitmojiHeight/150*4.5); //left eye color
ellipse(bitmojiX+(bitmojiHeight/150*14.5),bitmojiY-(bitmojiHeight/150*1.5),bitmojiHeight/150*4.5,
bitmojiHeight/150*4.5); //right eye color
stroke(0,0,0);
fill(240,210,156);
bezier(bitmojiX,bitmojiY,bitmojiX+(bitmojiHeight/150*21),bitmojiY+(bitmojiHeight/150*22),bitmoji
X-(bitmojiHeight/150*8),bitmojiY+(bitmojiHeight/150*20),bitmojiX-(bitmojiHeight/150*4),bitmojiY+
(bitmojiHeight/150*15)); //nose
fill(255,255,255);
arc(bitmojiX,bitmojiY+(bitmojiHeight/150*28),bitmojiHeight/150*30,bitmojiHeight/150*13,1,180);
//mouth
line(bitmojiX-(bitmojiHeight/150*14),bitmojiY+(bitmojiHeight/150*27.5),bitmojiX+(bitmojiHeight/1
50*14),bitmojiY+(bitmojiHeight/150*27.5)); //top of mouth
};

var drawBitmojiHair = function (bitmojiX,bitmojiY,bitmojiHeight) {
fill(122, 74, 27);
```

```
quad(bitmojiX-(bitmojiHeight/150*41),bitmojiY,bitmojiX-(bitmojiHeight/150*34),bitmojiY-(bitmojiHeight/150*32),bitmojiX-(bitmojiHeight/150*11),bitmojiY-(bitmojiHeight/150*50),bitmojiX-(bitmojiHeight/150*40),bitmojiY-(bitmojiHeight/150*1)); //left hair
quad(bitmojiX+(bitmojiHeight/150*43),bitmojiY-(bitmojiHeight/150*2),bitmojiX+(bitmojiHeight/150*33),bitmojiY-(bitmojiHeight/150*32),bitmojiX+(bitmojiHeight/150*13),bitmojiY-(bitmojiHeight/150*43),bitmojiX+(bitmojiHeight/150*40),bitmojiY-(bitmojiHeight/150*3)); //right hair
ellipse(bitmojiX,bitmojiY-(bitmojiHeight/150*37),bitmojiHeight/150*62,bitmojiHeight/150*28); //top hair
};

var drawBitmojiShirt = function (bitmojiX,bitmojiY,bitmojiHeight) {
noStroke();
fill(166, 166, 166);
quad(bitmojiX-(bitmojiHeight/150*54),bitmojiY+(bitmojiHeight/150*38),bitmojiX-(bitmojiHeight/150*60),bitmojiY+(bitmojiHeight/150*95),bitmojiX-(bitmojiHeight/150*1),bitmojiY+(bitmojiHeight/150*95),bitmojiX-(bitmojiHeight/150*1),bitmojiY+(bitmojiHeight/150*51)); //left part of shirt
quad(bitmojiX+(bitmojiHeight/150*52),bitmojiY+(bitmojiHeight/150*38),bitmojiX+(bitmojiHeight/150*58),bitmojiY+(bitmojiHeight/150*95),bitmojiX-(bitmojiHeight/150*1),bitmojiY+(bitmojiHeight/150*95),bitmojiX-(bitmojiHeight/150*1),bitmojiY+(bitmojiHeight/150*51)); //right part of shirt
stroke(25, 14, 240);
strokeWeight(3);
line(bitmojiX-(bitmojiHeight/150*40),bitmojiY+(bitmojiHeight/150*60),bitmojiX,bitmojiY+(bitmojiHeight/150*60)); //part of J
line(bitmojiX-(bitmojiHeight/150*20),bitmojiY+(bitmojiHeight/150*60),bitmojiX-(bitmojiHeight/150*20),bitmojiY+(bitmojiHeight/150*85)); //part of J
line(bitmojiX-(bitmojiHeight/150*20),bitmojiY+(bitmojiHeight/150*85),bitmojiX-(bitmojiHeight/150*37),bitmojiY+(bitmojiHeight/150*85)); //part of J
line(bitmojiX-(bitmojiHeight/150*37),bitmojiY+(bitmojiHeight/150*85),bitmojiX-(bitmojiHeight/150*37),bitmojiY+(bitmojiHeight/150*80)); //part of J
line(bitmojiX+(bitmojiHeight/150*10),bitmojiY+(bitmojiHeight/150*60),bitmojiX+(bitmojiHeight/150*10),bitmojiY+(bitmojiHeight/150*85)); //part of K
line(bitmojiX+(bitmojiHeight/150*10),bitmojiY+(bitmojiHeight/150*72.5),bitmojiX+(bitmojiHeight/150*30),bitmojiY+(bitmojiHeight/150*60)); //part of K
line(bitmojiX+(bitmojiHeight/150*10),bitmojiY+(bitmojiHeight/150*72.5),bitmojiX+(bitmojiHeight/150*30),bitmojiY+(bitmojiHeight/150*85)); //part of K
};

var drawBitmojiHat = function (bitmojiX,bitmojiY,bitmojiHeight) {
noStroke();
fill(158, 158, 158);
ellipse(bitmojiX,bitmojiY-(bitmojiHeight/150*38),bitmojiHeight/150*70,bitmojiHeight/150*30);
//base of hat
fill(0,0,0);
```

```
ellipse(bitmojiX,bitmojiY-(bitmojiHeight/150*25),bitmojiHeight/150*57,bitmojiHeight/150*20);
//brim of hat
};

var drawBitmoji = function (bitmojiX,bitmojiY,bitmojiHeight) {
drawBitmojiFace (bitmojiX,bitmojiY,bitmojiHeight);
drawBitmojiHair (bitmojiX,bitmojiY,bitmojiHeight);
drawBitmojiShirt (bitmojiX,bitmojiY,bitmojiHeight);
drawBitmojiHat (bitmojiX,bitmojiY,bitmojiHeight);
};

//Anthony's bitmoji code
var drawBitmoji2 = function(x,y,bitmojiHeight) {


noStroke();

var drawBitmojiHead = function() {

fill(224, 172, 105);

ellipse(x+0*(bitmojiHeight/100),y+0*(bitmojiHeight/100),83*(bitmojiHeight/100),100
*(bitmojiHeight/100)); //head


fill(0, 0, 0);
quad(x-50*(bitmojiHeight/100),y+0*(bitmojiHeight/100),x-37*(bitmojiHeight/100),y-32
*(bitmojiHeight/100),x-10*(bitmojiHeight/100),y-50*(bitmojiHeight/100),x-30
*(bitmojiHeight/100),y-1*(bitmojiHeight/100)); //left hair

quad(x+49*(bitmojiHeight/100),y-2*(bitmojiHeight/100),x+34*(bitmojiHeight/100),y-32
*(bitmojiHeight/100),x+13*(bitmojiHeight/100),y-50*(bitmojiHeight/100),x+35
*(bitmojiHeight/100),y-3*(bitmojiHeight/100)); //right hair

ellipse(x+0*(bitmojiHeight/100),y-37*(bitmojiHeight/100),62*(bitmojiHeight/100),28
*(bitmojiHeight/100)); //top hair


fill(255,255,255);

ellipse (x-12*(bitmojiHeight/100),y+0*(bitmojiHeight/100),13*(bitmojiHeight/100),10
*(bitmojiHeight/100)); //left eye back
```

```
ellipse (x+15*(bitmojiHeight/100),y+0*(bitmojiHeight/100),13*(bitmojiHeight/100),10
*(bitmojiHeight/100)); //right eye back

fill(115,23,23); //brown eyes fill

ellipse(x-10*(bitmojiHeight/100),y+0*(bitmojiHeight/100),6*(bitmojiHeight/100),4
*(bitmojiHeight/100)); //left eye

ellipse(x+17*(bitmojiHeight/100),y+0*(bitmojiHeight/100),6*(bitmojiHeight/100),4
*(bitmojiHeight/100)); //right eye


stroke(0, 0, 0);
fill(224, 172, 105);

bezier(x+0*(bitmojiHeight/100),y+0*(bitmojiHeight/100),x+21*(bitmojiHeight/100),y
+22*(bitmojiHeight/100),x-8*(bitmojiHeight/100),y+20*(bitmojiHeight/100),x-4
*(bitmojiHeight/100),y+15*(bitmojiHeight/100)); //nose


fill(255, 255, 255);

arc(x+3*(bitmojiHeight/100),y+25*(bitmojiHeight/100), 30*(bitmojiHeight/100), 13
*(bitmojiHeight/100),1,180); //mouth

line(x-14*(bitmojiHeight/100),y+24*(bitmojiHeight/100),x+19*(bitmojiHeight/100),y
+24*(bitmojiHeight/100)); //top of mouth


noStroke();
fill(59, 80, 102);

arc(x+0*(bitmojiHeight/100), y-20*(bitmojiHeight/100), 85*(bitmojiHeight/100), -70
*(bitmojiHeight/100), 5, 186); //hat

fill(0, 0, 0);
arc(x+0*(bitmojiHeight/100), y-25*(bitmojiHeight/100), 30*(bitmojiHeight/100), -20
*(bitmojiHeight/100), 5, 186); //hat hole

};
drawBitmojiHead();
```

```
var drawBitmojiBody = function() {

fill(224, 172, 105);

rect(x-9*(bitmojiHeight/100),y+45*(bitmojiHeight/100),23*(bitmojiHeight/100),15
*(bitmojiHeight/100)); //neck


fill(116, 158, 109);

rect(x-30*(bitmojiHeight/100),y+55*(bitmojiHeight/100),66*(bitmojiHeight/100),80
*(bitmojiHeight/100)); //shirt

triangle(x-60*(bitmojiHeight/100), y+80*(bitmojiHeight/100), x-30*(bitmojiHeight       /100),
y+56*(bitmojiHeight/100), x-30*(bitmojiHeight/100), y+80*(bitmojiHeight/100     )); //right shirt

triangle(x+66*(bitmojiHeight/100), y+80*(bitmojiHeight/100), x+36*(bitmojiHeight       /100),
y+56*(bitmojiHeight/100), x-30*(bitmojiHeight/100),y+80*(bitmojiHeight/100      )); //left shirt


fill(0, 0, 0);
textSize(30*(bitmojiHeight/100));

text("AM",x-20*(bitmojiHeight/100),y+66*(bitmojiHeight/100),100*(bitmojiHeight/100
),100*(bitmojiHeight/100));


fill(224, 172, 105);

rect(x-50*(bitmojiHeight/100),y+80*(bitmojiHeight/100),15*(bitmojiHeight/100),30
*(bitmojiHeight/100)); //left arm

rect(x+41*(bitmojiHeight/100),y+80*(bitmojiHeight/100),15*(bitmojiHeight/100),30
*(bitmojiHeight/100)); //right arm


ellipse(x-43*(bitmojiHeight/100),y+115*(bitmojiHeight/100),20*(bitmojiHeight/100
),20*(bitmojiHeight/100)); //left hand

ellipse(x+49*(bitmojiHeight/100),y+115*(bitmojiHeight/100),20*(bitmojiHeight/100
),20*(bitmojiHeight/100)); //right hand

};
```

```
drawBitmojiBody();

};

//Function that draws the stars array background
var drawStars = function() {
var xStar = [33,115,335,250,230,20,120,200,330,120,280]; //Draws xPosition of the star
var yStar = [119,33,239,200,50,250,340,330,120,200,280]; //Draws yPosition of the star
for (var i = 0; i < xStar.length; i++) {
image(getImage("space/star"), xStar[i], yStar[i], 40, 40); //Draws the actual star
    }
};

//Function that draws the red and green barrier of the screens
var barrier = function () {
     fill(0, 255, 0);
     rect(0, 0, 10, 200);
     rect(390, 0, 10, 200);
     rect(0, 0, width, 10);
     fill(255, 0, 0);
     rect(0, 200, 10, 200);
     rect(390, 200, 10, 200);
     rect(0, 390, width, 10);
};

//All of the khan button class from what we learned in class
var Button = function(config) {
    this.x = config.x || 0;
    this.y = config.y || 0;
    this.width = config.width || 150;
    this.height = config.height || 50;
    this.label = config.label || "Click";
    this.onClick = config.onClick || function() {};
};

//Draws the button based on the top left of the screen at the x and y and width and height values
that you set for it
Button.prototype.draw = function() {
    fill(255, 255, 0);
    rect(this.x, this.y, this.width, this.height, 5);
    fill(0, 0, 0);
    textSize(19);
    textAlign(LEFT, TOP);
    text(this.label, this.x+10, this.y+this.height/4);
```

```
};

//Prototype that shows if your mouse is inside of the button
Button.prototype.isMouseInside = function() {
    return mouseX > this.x &&
           mouseX < (this.x + this.width) &&
           mouseY > this.y &&
           mouseY < (this.y + this.height);
};


//Prototype that shows if you click inside of the button
Button.prototype.handleMouseClick = function() {
    if (this.isMouseInside()) {
        this.onClick();
    }
};

//Button that changes home screen to game options screen
var gameOptions = new Button({
    x: 120,
    y: 275,
    label: "Game Options",
    onClick: function() {
        currentScene = 1;
    }
});

//Button to choose the option for one player game mode
var playerOptions1 = new Button ({
    x: 120,
    y: 200,
    label: "   One Player",
    onClick: function() {
        currentScene = 2;
    }
});

//Button to choose the level one option of the one player game
var levelOne = new Button ({
    x: 120,
    y: 150,
    label: " Level One",
    onClick: function() {
```

```
        currentScene = 5;
    }
});

//Button to choose the level two option of the one player game
var levelTwo = new Button ({
    x: 120,
    y: 230,
    label: " Level Two",
    onClick: function() {
        currentScene = 6;
    }
});

//Button to choose the level three option of the one player game
var levelThree = new Button ({
    x: 120,
    y: 310,
    label: " Level Three",
    onClick: function() {
        currentScene = 7;
    }
});

//Draws the CPU for level one of the game
var CPU1 = function () {
    fill(255,255,0);
    rect(CPUX,CPUY,50,20);

     if (CPUX <=25) {
        CPUY = 31;
   } else if (CPUX >= 325) {
        CPUY = 30;
   }
   if (CPUY === 30) {
        CPUX -= puck1Speed;
   } else if (CPUY === 31) {
        CPUX += puck1Speed;
   }
};

//Draws the CPU for level two of the game
var CPU2 = function () {
    fill(255,255,0);
```

```
    rect(CPUX,CPUY,50,20);

     if (CPUX <=25) {
       CPUY = 31;
   } else if (CPUX >= 325) {
       CPUY = 30;
   }
   if (CPUY === 30) {
       CPUX -= puck2Speed;
   } else if (CPUY === 31) {
       CPUX += puck2Speed;
   }
};

//Draws the CPU for level 3 of the game
var CPU3 = function () {
    fill(255,255,0);
    rect(CPUX,CPUY,50,20);

     if (CPUX <=25) {
       CPUY = 31;
   } else if (CPUX >= 325) {
       CPUY = 30;
   }
   if (CPUY === 30) {
       CPUX -= puck3Speed;
   } else if (CPUY === 31) {
       CPUX += puck3Speed;
   }
};

//Draws the home screen or the splash screen of our game
var drawHomeScreen = function () {
    currentScene = 0;
    background(41, 28, 28);
    drawStars();
    fill(255,255,255);
    textSize(40);
    text("Air Hockey", 100, 100);
    stroke(255,255,255);
    strokeWeight(5);
    line(85, 150, 305, 150);
    textSize(20);
    text("By: Josh Kleinberg & Anthony Martinez", 25, 175);
```

```
    drawBitmoji (350, 50, 75);
    drawBitmoji (50, 330, 75);
    drawBitmoji2 (50, 40, 40);
    drawBitmoji2 (350, 330, 40);
    gameOptions.draw ();
    barrier();
};

//Draws the game options screen for our game
var drawGameOptions = function () {
   currentScene = 1;
   background(41, 28, 28);
   drawStars();
   fill(255,255,255);
   textSize(40);
   text("Game Options", 65, 100);
   stroke(255,255,255);
   strokeWeight(5);
   line(60, 150, 325, 150);
    drawBitmoji (350, 50, 75);
    drawBitmoji (50, 330, 75);
    drawBitmoji2 (50, 40, 40);
    drawBitmoji2 (350, 330, 40);
    playerOptions1.draw ();
    barrier();
};

//Draws the instructions screen for the one player option of our game
var drawInstructions1 = function () {
   currentScene = 2;
   background(41, 28, 28);
   drawStars();
   fill(255, 255, 255);
   textSize(40);
   text("Instructions", 90, 40);
   stroke(255, 255, 255);
   strokeWeight(5);
   line(85, 90, 300, 90);
   textSize(20);
   text("Use the mousepad to move the bitmoji \nacross your side of the screen and try to\n
score on the opponents goal. First one \nto score 7 goals wins! Click anywhere to \nchoose your
player and level", 20, 150);
   drawBitmoji (350, 50, 75);
   drawBitmoji (50, 330, 75);
```

```
    drawBitmoji2 (50, 40, 40);
    drawBitmoji2 (350, 330, 40);
    barrier();
};

//Draws the screen where you can choose between levels one and three for the game
var drawChooseLevel = function () {
    currentScene = 4;
    background(41,28,28);
    drawStars();
    fill(255,255,255);
    textSize(40);
    text("Choose Your Level", 30, 65);
    stroke(255,255,255);
    strokeWeight(5);
    line(30,115,370,115);
    noStroke();
    levelOne.draw();
    levelTwo.draw();
    levelThree.draw();
    barrier();
};

//Screen where players can choose their character for the one player option
var drawChoosePlayer1 = function () {
    background(41, 28, 28);
    drawStars();
    fill(255, 255, 255);
    textSize(40);
    text("Choose Your Player", 20, 65);
    stroke(255, 255, 255);
    strokeWeight(5);
    line(20, 115, 375, 115);
    drawBitmoji (100, 250, 100);
    drawBitmoji2 (300, 250, 55);
    barrier();
};

//Draws the choose player options screen if the user chooses level one
var drawChoosePlayer101 = function () {
    currentScene = 5;
    drawChoosePlayer1();
};
```

```
//Draws the choose player options screen if the user chooses level two
var drawChoosePlayer102 = function () {
    currentScene = 6;
    drawChoosePlayer1();
};

//Draws the choose player options screen if the user chooses level three
var drawChoosePlayer103 = function () {
    currentScene = 7;
    drawChoosePlayer1();
};

// variables for puck
var player1Score = 0; //Sets the player 1 score to zero
var player2Score = 0; //Sets the CPU score to zero
var puck; //Sets the variable for the puck
var gameStarted = false; //Sets the game started to false
var PUCK_SPEED = 15; //Sets the speed of the puck in the game to 15
var PADDLE_HEIGHT = 40; //Sets the paddle height for the player 1 to 40
var PADDLE_WIDTH = 40; //Sets the paddle width for the player 1 to 40

//Draws the first paddle based on if player one was selected or not, and moves according to
mouse X and Mouse Y and is also constrained to the bottom half of the screen
var drawPaddle1 = function() {
    var c1X = constrain(mouseX, 45, 355);
    var c1Y = constrain(mouseY - 5, 206, 349);
    var c2Y = constrain(mouseY, 212, 355);
    fill(225, 225, 225);
    noStroke();
    ellipse(c1X, c2Y, PADDLE_WIDTH, PADDLE_HEIGHT);
    drawBitmoji(c1X, c1Y, 25);
};

//Draws the second paddle based on if player two was selected or not, and moves according to
mouse X and mouse Y and is also constrained to the bottom half of the screen
var drawPaddle2 = function() {
    var c1X = constrain(mouseX, 45, 355);
    var c1Y = constrain(mouseY - 5, 206, 349);
    var c2Y = constrain(mouseY, 212, 355);
    fill(225, 225, 225);
    noStroke();
    ellipse(c1X, c2Y, PADDLE_WIDTH, PADDLE_HEIGHT);
    drawBitmoji2(c1X, c1Y, 15);
};
```

```
//Handles the puck
var Puck = function(position, speed) {
    this.position = position; //position of the puck
    this.speed = speed || PUCK_SPEED; //speed of the puck

    this.radius = 12; //radius of the puck

    //reset function
    this.resetVelocity = function() {
        this.theta = random(0, 360);
        this.velocity = new PVector(
        this.speed*cos(this.theta), -this.speed*sin(this.theta));
    };
    this.resetVelocity();

    //draws the puck
    this.draw = function() {
        fill(255, 255, 255);
        noStroke();
        ellipse(this.position.x, this.position.y,
        this.radius*2, this.radius*2);
    };

    //handles vertical paddle collision across y axis
    this.collideWithPaddle = function(x,y) {
        if (this.position.x - this.radius < x + PADDLE_WIDTH/2 &&
        this.position.x + this.radius > x - PADDLE_WIDTH/2) {
            if (dist(0, this.position.y, 0, y) <
            PADDLE_HEIGHT/2 + this.radius) {
                if (this.position.x > x) {
                    this.position.x = x +
                    this.radius + PADDLE_WIDTH/2;
                }
                else if (this.position.x < x) {
                    this.position.x = x -
                    this.radius - PADDLE_WIDTH/2;
                }
                this.velocity.mult(new PVector(-1, 1));
            }
        }
    };

    //handles horizontal paddle collision across x axis
```

```
this.collideWithPaddle2 = function(x,y) {
    if (this.position.y - this.radius < y + PADDLE_HEIGHT/2 &&
    this.position.y + this.radius > y - PADDLE_HEIGHT/2) {
        if (dist(this.position.x, 0, x, 0) <
        PADDLE_WIDTH/2 + this.radius) {
            if (this.position.y > y) {
                this.position.y = y +
                this.radius + PADDLE_HEIGHT/2;
            }
            else if (this.position.y < y) {
                this.position.y = y -
                this.radius - PADDLE_HEIGHT/2;
            }
            this.velocity.mult(new PVector(1, -1));
        }
    }
};

//handles vertical CPU collision across y axis
this.collideWithCPU = function(x,y) {
    if (this.position.x - this.radius < x + 25 &&
    this.position.x + this.radius > x - 25) {
        if (dist(0, this.position.y, 0, y) <
        PADDLE_HEIGHT/2 + this.radius) {
            if (this.position.x > x) {
                this.position.x = x +
                this.radius + 25;
            }
            else if (this.position.x < x) {
                this.position.x = x -
                this.radius - 25;
            }
            this.velocity.mult(new PVector(-1, -1));
        }
    }
};

//handles horizontal CPU collision across x axis
this.collideWithCPU2 = function(x,y) {
    if (this.position.x - this.radius < x &&
    this.position.x + this.radius > x - 10) {
        if (dist(0, this.position.y, 0, y) <
        PADDLE_HEIGHT/2 + this.radius) {
            if (this.position.x > x) {
```

```
            this.position.x = x +
            this.radius;
        }
        else if (this.position.x < x) {
            this.position.x = x -
            this.radius;
        }
        this.velocity.mult(new PVector(-1, -1));
    }
  }
};

//collision updates for puck, paddles, and CPU
this.update = function() {

  //Handle goal collisions
  if (this.position.x >= 151 && this.position.x <= 248 && this.position.y >= 360        )
  {
    player2Score++;
    this.position = new PVector(width/2, height/2);
    gameStarted = false;
    this.resetVelocity();
  }

  if (this.position.x >= 151 && this.position.x <= 248 && this.position.y <= 40)
  {
    player1Score++;
    this.position = new PVector(width/2, height/2);
    gameStarted = false;
    this.resetVelocity();
  }

  // handle vertical collision
  if (this.position.y < 40) {
    this.position.y = 40;
    this.velocity.mult(new PVector(1, -1));
  }
  if (this.position.y > 360) {
    this.position.y = 360;
    this.velocity.mult(new PVector(1, -1));
  }

  // handle horizontal collision
  if (this.position.x < 40) {
```

```
            this.position.x = 40;
            this.velocity.mult(new PVector(-1, 1));
        }

        if (this.position.x > 360) {
            this.position.x = 360;
            this.velocity.mult(new PVector(-1, 1));
        }
            this.collideWithPaddle(mouseX, mouseY);
            this.collideWithPaddle2(mouseX, mouseY);

            this.collideWithCPU(CPUX, CPUY);
            this.collideWithCPU2(CPUX, CPUY);

        this.position.add(this.velocity);
    };
};
puck = new Puck(new PVector(width/2, height/2));

//Displays the score during the game
var scoreBoard = function() {
    noFill();
    strokeWeight(1);
    stroke(255, 0, 0);
    rect(320, 160, 20, 20);
    stroke(0, 255, 0);
    rect(320, 220, 20, 20);
    textSize(12);
    fill(255, 255, 255);
    text(player1Score, 327, 222);
    text(player2Score, 327, 162);
    if (player1Score >= 7) {
        currentScene = 14; //If the score is greater than or equal to 7, you win the game and it
displays you won
    }
    if (player2Score >= 7) {
        currentScene = 15; //If the score is greater than or equal to 7, the CPU wins and it displays
you lost
    }
};

//Draws the game screen for air hockey
var drawGameScreen = function () {
    //Includes all the lines, ellipses, etc to draw the outline of the game screen
```

```
    background(41, 28, 28);
    stroke(255, 0, 0);
    strokeWeight(5);
    line(25, 25, 375, 25);
    line(25, 25, 25, 200);
    line(375, 25, 375, 200);
    stroke(0, 255, 0);
    strokeWeight(5);
    line(25, 375, 375, 375);
    line(25, 375, 25, 200);
    line(375, 375, 375, 200);
    stroke(255, 255, 0);
    strokeWeight(3);
    line(24, 198, 376, 198);
    fill(41, 28, 28);
    ellipse(200, 200, 85, 85);
    ellipse(200, 200, 3, 3);
    strokeWeight(1);
    line(25, 260, 375, 260);
    line(25, 140, 375, 140);
    ellipse(100, 300, 40, 40);
    ellipse(300, 300, 40, 40);
    ellipse(100, 100, 40, 40);
    ellipse(300, 100, 40, 40);
    strokeWeight(3);
    arc(200, 380, 121, 102, 183, 358);
    arc(200, 20, 121, 102, 5, 178);

    //Displays the arrays that make the green and red circles show up on the screen
    stroke(255,0,0);
    fill(255, 0, 0);
    var xRedCircle =
[12,12,12,12,12,12,12,12,12,37,62,87,112,137,162,187,212,237,262,287,312,337,362,388,388,388
,388,388,388,388,388]; //Displays the red circles that show up on the game screen x values
    var yRedCircle =
[15,40,65,90,115,140,165,190,12,12,12,12,12,12,12,12,12,12,12,12,12,12,15,40,65,90,115,140,
165,190]; //Displays the red cirlces that show up on the game screen for the y values
    for (var j = 0; j<xRedCircle.length; ++j) {
        ellipse(xRedCircle[j], yRedCircle[j], 5, 5);
    } //For loop that displays the circles x and y values based on how many numbers are in the x
positions array

    stroke(0,255,0);
    fill(0,255,0);
```

```
    var xGreenCircle =
[12,12,12,12,12,12,12,12,12,37,62,87,112,137,162,187,212,237,262,287,312,337,362,388,388,388
,388,388,388,388,388,388]; //Displays the green circles that show up on the game screen x values
    var yGreenCircle =
[215,240,265,290,315,340,365,390,388,388,388,388,388,388,388,388,388,388,388,388,388,388,38
8,390,365,340,315,290,265,240,215]; //Displays the green circles that show up on the game
screen y values
    for (var k = 0; k<xGreenCircle.length; ++k) {
        ellipse(xGreenCircle[k], yGreenCircle[k], 5, 5);
    } //For loop that displays the cirlces x and y values based on how many numbers ar ein the x
positions green circles array
};

//Draws the end screen when game is finished if you won the game
var drawEndScreen1 = function () {
    currentScene = 14;
    drawGameScreen();
    textSize(30);
    fill(255, 255, 255);
    text("You won!", 143, 280);
};

//Draws the end screen when the game is finished if the CPU won the game
var drawEndScreen2 = function () {
    currentScene = 15;
    drawGameScreen();
    textSize(30);
    fill(255, 255, 255);
    text("You lost!", 143, 280);
};

//Holds the function that draws the game screen for level one if the user chooses bitmoji 1 as
their player
var drawGameScreen1 = function () {
    currentScene = 8;
    drawGameScreen();
    scoreBoard();
    drawPaddle1();
    puck.draw();
    puck.update();
    CPU1();
};
```

```
//Holds the function that draws the game screen for level one if the user chooses bitmoji 2 as
their player
var drawGameScreen2 = function () {
    currentScene = 9;
    drawGameScreen();
    scoreBoard();
    drawPaddle2();
    puck.draw();
    puck.update();
    CPU1();
};

//Holds the function that draws the game screen for level two if the user chooses bitmoji 1 as
their player
var drawGameScreen3 = function () {
    currentScene = 10;
    drawGameScreen();
    scoreBoard();
    drawPaddle1();
    puck.draw();
    puck.update();
    CPU2();
};

//Holds the function that draws the game screen for level two if the user chooses bitmoji 2 as
their player
var drawGameScreen4 = function () {
    currentScene = 11;
    drawGameScreen();
    scoreBoard();
    drawPaddle2();
    puck.draw();
    puck.update();
    CPU2();
};

//Holds the function that draws the game screen for level three if the user chooses bitmoji 1 as
their player
var drawGameScreen5 = function () {
    currentScene = 12;
    drawGameScreen();
    scoreBoard();
    drawPaddle1();
    puck.draw();
```

```
    puck.update();
    CPU3();
};

//Holds the function that draws the game screen for level three if the user chooses bitmoji 2 as
their player
var drawGameScreen6 = function () {
    currentScene = 13;
    drawGameScreen();
    scoreBoard();
    drawPaddle2();
    puck.draw();
    puck.update();
    CPU3();
};

//Mouseclicked function for all the buttons in the khan button class and all the scene changes
throughout the code
mouseClicked = function() {
    if (currentScene === 0) {
        gameOptions.handleMouseClick(); //Draws the game options screen if this button is clicked
    } else if (currentScene === 1) {
        playerOptions1.handleMouseClick(); //Draws the player 1 options screen if the button is
clicked
    } else if (currentScene === 2) {
        drawChooseLevel(); //Draws the screen where you choose your level if this button is
clicked
    } else if (currentScene === 4) {
        levelOne.handleMouseClick(); //Draws the screen for the level one player options if this
button is clicked
        levelTwo.handleMouseClick(); //Draws the screen for the level two player options if this
button is clicked
        levelThree.handleMouseClick(); //Draws the screen for the level three player options if this
button is clicked
    } else if (currentScene === 5) { //Draws the correct game screen for the level one options
        if (mouseX <= 150 && mouseX >= 50 && mouseY <= 300 && mouseY >=200) {
            drawGameScreen1(); //If you choose bitmoji 1, you use that bitmoji to play the game
        } else if (mouseX <= 350 && mouseX >=250 && mouseY <=300 && mouseY >=200) {
            drawGameScreen2(); //If you choose bitmoji 2, you use that bitmoji to play the game
        }
    } else if (currentScene === 6) { //Draws the correct game screen for the level two options
        if (mouseX <= 150 && mouseX >= 50 && mouseY <= 300 && mouseY >= 200) {
            drawGameScreen3(); //If you choose bitmoji 1, you use that bitmoji to play the game
        } else if (mouseX <= 350 && mouseX >= 250 && mouseY <= 300 && mouseY >= 200) {
```

```
            drawGameScreen4(); //If you choose bitmoji 2, you use that bitmoji to play the game
        }
    } else if (currentScene === 7) { //Draws the correct game screen for the level three options
        if (mouseX <= 150 && mouseX >= 50 && mouseY <= 300 && mouseY >= 200) {
            drawGameScreen5(); //If you choose bitmoji 1, you use that bitmoji to play the game
        } else if (mouseX <= 350 && mouseX >= 250 && mouseY <= 300 && mouseY >= 200) {
            drawGameScreen6(); //If you choose bitmoji 2, you use that bitmoji to play the game
        }
    }
};

//Draw function that makes the code run
draw = function() {
    if (currentScene === 0) {
        drawHomeScreen(); //Draws the home screen if the current scene is equal to 0
    } else if (currentScene === 1) {
        drawGameOptions(); //Draws the game options screen if the current scene is equal to 1
    } else if (currentScene === 2) {
        drawInstructions1(); //Draws the instructions screen if the current scene is equal to 2
    } else if (currentScene === 4) {
        drawChooseLevel(); //Draws the choose level screen if the current scene is equal to 4
    } else if (currentScene === 5) {
        drawChoosePlayer1(); //Draws the choose player 1 screen if the curent scene is equal to 5
    } else if (currentScene === 6) {
        drawChoosePlayer1(); //Draws the choose player 1 screen if the curent scene is equal to 6
    } else if (currentScene === 7) {
        drawChoosePlayer1(); //Draws the choose player 1 screen if the curent scene is equal to 7
    } else if (currentScene === 8) {
        drawGameScreen1(); //Draws the game game screen for level one player one if the current
scene is equal to 8
    } else if (currentScene === 9) {
        drawGameScreen2(); //Draws the game screen for level one player 2 if the current scene is
equal to 9
    } else if (currentScene === 10) {
        drawGameScreen3(); //Draws the game screen for level two player one if the current scene
is equal to 10
    } else if (currentScene === 11) {
        drawGameScreen4(); //Draws the game screen for level two player two if the current scene
is equal to 11
    } else if (currentScene === 12) {
        drawGameScreen5(); //Draws the game screen for level three player one if the current
scene is equal to 12
    } else if (currentScene === 13) {
```

```
        drawGameScreen6(); //Draws the game screen for level three player two if the current
scene is equal to 13
    } else if (currentScene === 14) {
        drawEndScreen1();  //Draws the end screen if you won the game and the current scene is
14
    } else if (currentScene === 15) {
        drawEndScreen2(); //Draws the end screen if you lost the game and the current scene is
15
    }
};
```