

Code Save

Project Name : Bitmoji Reunion

Copy and Paste Code Below

↓↓

↓↓

//When ready, bitmoji draw code will be up top and we will call it below

```
var DrawShirt = function(x,y,h) {
```

```
  var p = h/200 ;
```

```
  fill(0, 10, 87);
```

```
  arc(x+(p*75), y+(p*150), p*100, p*40, 180, 360); // shirt
```

```
  fill(255, 255, 255);
```

```
  textSize(14*p);
```

```
  text("C X M", x+(p*55), y+(p*135), p*100, p*100);
```

```
};
```

```
var DrawHead = function(x,y,h) {
```

```
  var p = h/200 ;
```

```
  fill(0, 0, 0);
```

```
  ellipse(x+(p*75), y+(p*65), p*80, p*90); // hair
```

```
  rect(x+(p*35),y+(p*60),p*79,p*70);
```

```
  fill(186, 160, 109);
```

```
  rect(x+(p*70),y+(p*125),p*10,p*10); //neck
```

```
  fill(186, 160, 109);
```

```
  ellipse(x+(p*75), y+(p*75), p*75, p*100); // face
```

```
  fill(255, 255, 255);
```

```
  ellipse(x+(p*64), y+(p*70), p*10, p*4); // left eye
```

```
  ellipse(x+(p*86), y+(p*70), p*10, p*4); // right eye
```

```
  fill(74, 43, 0);
```

```
  ellipse(x+(p*64), y+(p*70), p*4, p*4); // left eye
```

```
  ellipse(x+(p*86), y+(p*70), p*4, p*4); // right eye
```

```
  noFill();
```

```
  arc(x+(p*64), y+(p*68), p*15, p*6, 180, 360); // eyebrow
```

```
  arc(x+(p*86), y+(p*68), p*15, p*6, 180, 360); // eyebrow
```

```
  fill(0, 0, 0);
```

```
  triangle(x+(p*74), y+(p*100), x+(p*75), y+(p*85), x+(p*76), y+(p*100)); //nose
```

```
  noFill();
```

```
  arc(x+(p*75), y+(p*105), p*30, p*15, 10, 170); // mouth
```

```

    fill(0, 0, 0);
    arc(x+(p*75), y+(p*60), p*75, p*70, 178, 362); // hair bangs
};

```

```

var DrawBitmoji = function(x,y,h) {
    var p = h/200;
    x = x - (75*p);
    y = y - (90*p);
    DrawHead(x,y,h);
    DrawShirt(x,y,h);
};

```

```

DrawBitmoji(100,100,153);

```

```

//Player A

```

```

//Code divider for bitmojis of player A and B

```

```

//Player B

```

```

var drawBitmoji = function(bitmojiScale, x, y) {
    noStroke();
    fill(176, 110, 80);
    ellipse(x + 200 * bitmojiScale, y + 152 * bitmojiScale, 70 * bitmojiScale, 86 * bitmojiScale); //
head
    fill(0, 0, 0);
    ellipse(x + 201 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 207 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 192 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 183 * bitmojiScale, y + 104 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 214 * bitmojiScale, y + 98 * bitmojiScale, 36 * bitmojiScale, 36 * bitmojiScale);
    ellipse(x + 223 * bitmojiScale, y + 110 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 231 * bitmojiScale, y + 123 * bitmojiScale, 17 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 174 * bitmojiScale, y + 113 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 246 * bitmojiScale, y + 134 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 166 * bitmojiScale, y + 125 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);
    ellipse(x + 246 * bitmojiScale, y + 119 * bitmojiScale, 26 * bitmojiScale, 31 * bitmojiScale);
    ellipse(x + 231 * bitmojiScale, y + 104 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 200 * bitmojiScale, y + 97 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 177 * bitmojiScale, y + 97 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 172 * bitmojiScale, y + 107 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 154 * bitmojiScale, y + 121 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 153 * bitmojiScale, y + 135 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 247 * bitmojiScale, y + 137 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 152 * bitmojiScale, y + 146 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);
    ellipse(x + 247 * bitmojiScale, y + 148 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);

```

```

stroke(0, 0, 0);
fill(176, 110, 80);
rect(x + 175 * bitmojiScale, y + 133 * bitmojiScale, 18 * bitmojiScale, 17 * bitmojiScale); //
glasses left
rect(x + 204 * bitmojiScale, y + 133 * bitmojiScale, 18 * bitmojiScale, 17 * bitmojiScale); //
glasses right
line(x + 193 * bitmojiScale, y + 144 * bitmojiScale, x + 205 * bitmojiScale, y + 143 *
bitmojiScale); // nose bridge
line(x + 176 * bitmojiScale, y + 143 * bitmojiScale, x + 146 * bitmojiScale, y + 137 *
bitmojiScale); // left glasses
line(x + 259 * bitmojiScale, y + 132 * bitmojiScale, x + 223 * bitmojiScale, y + 144 *
bitmojiScale); // right glasses

fill(92, 54, 54);
ellipse(x + 185 * bitmojiScale, y + 142 * bitmojiScale, 7 * bitmojiScale, 7 * bitmojiScale); // left
eye
ellipse(x + 214 * bitmojiScale, y + 142 * bitmojiScale, 7 * bitmojiScale, 7 * bitmojiScale); // right
eye
fill(176, 110, 80);
triangle(x + 192 * bitmojiScale, y + 160 * bitmojiScale, x + 206 * bitmojiScale, y + 159 *
bitmojiScale, x + 198 * bitmojiScale, y + 151 * bitmojiScale
);
noStroke();
arc(x + 212 * bitmojiScale, y + 169 * bitmojiScale, 23 * bitmojiScale, 36 * bitmojiScale, 1, 361);
// nose shaping
arc(x + 193 * bitmojiScale, y + 170 * bitmojiScale, 9 * bitmojiScale, 25 * bitmojiScale, 1, 361); //
nose shaping
arc(x + 199 * bitmojiScale, y + 171 * bitmojiScale, 9 * bitmojiScale, 25 * bitmojiScale, 1, 361); //
nose shaping
stroke(0, 0, 0);
fill(255, 255, 255); // teeth
arc(x + 200 * bitmojiScale, y + 175 * bitmojiScale, 30 * bitmojiScale, 11 * bitmojiScale, 7, 391);
// mouth

fill(217, 158, 208);
quad(
x + 176 * bitmojiScale,
y + 194 * bitmojiScale,
x + 223 * bitmojiScale,
y + 194 * bitmojiScale,
x + 300 * bitmojiScale,
y + 351 * bitmojiScale,
x + 100 * bitmojiScale,

```

```

    y + 350 * bitmojiScale
); // dress
rect(x + 116 * bitmojiScale, y + 167 * bitmojiScale, 30 * bitmojiScale, 88 * bitmojiScale); //
sleeve
rect(x + 253 * bitmojiScale, y + 167 * bitmojiScale, 30 * bitmojiScale, 88 * bitmojiScale); //
sleeve

    textSize(32 * bitmojiScale);
    fill(0);
    text("L", x + 178 * bitmojiScale, y + 259 * bitmojiScale);
    textSize(32 * bitmojiScale);
    fill(0);
    text("R", x + 199 * bitmojiScale, y + 259 * bitmojiScale);
};

```

//here is where bitmoji code will end when we implement it

```

//declare var variables
//stage phases
var phase = 0;
var currentPhase = 0;
var WL = 0; //win/lose
var i = 0;
var gate = 0; //gate activation for stages
var starters = 1;
var AtoB = 0;
var AorB = 0;

```

//Time vars

```

var startTime;
var elapsedTime;
var timeMS = 0;

```

//Bitmoji possitional code Decloration

```

//A
var XofA = 0;
var YofA = 0;

```

//B

```

var XofB = 0;
var YofB = 0;

```

```

// round obstacle arrays
var round1 = [];
var round2 = [];

var BarPara = [];

//Obstacle 1
var Obstacle1 = function (x,y) {
  fill(0, 0, 0);
  rect (x,y,130,16);
};

//Obstacle 2
var Obstacle2 = function (x,y) {
  fill(0, 0, 0);
  rect (x+45,y,130,16);
};

//Obstacle 3
var Obstacle3 = function (x,y) {
  fill(0, 0, 0);
  rect (x+32.5,y,110,16);
};

//generate array course numbers

var generateCourseNumbers = function(roundNumber) {
  for (var i=0; i<5; i++) {
    roundNumber.push (round(random(0.5,3.4)));
  }
};

//generate course

var CourseCreate = function (Player, roundNumber) {
  if (Player === "A") {
    AtoB = 25;
    fill(199, 189, 82);
    ellipse((25+87.5), 40, 20, 20);
  }
  if (Player === "B") {
    AtoB = 200;
    fill(199, 189, 82);
  }
};

```

```

        ellipse((200+87.5), 40, 20, 20);
    }

    for (var i=0; i < roundNumber.length; i++) {
        if (roundNumber[i] === 1){
            Obstacle1 (AtoB, (355-40)-(i*62));
        }
        if (roundNumber[i] === 2){
            Obstacle2 (AtoB, (355-50)-(i*62));
        }
        if (roundNumber[i] === 3){
            Obstacle3 (AtoB, (355-50)-(i*62));
        }
    }
};

```

```

var XaYnXbY_Reset = function () {
    //A
    XofA = 0;
    YofA = 0;
    //B
    XofB = 0;
    YofB = 0;
};

```

//background setup code

```

var backgroundScreen = function () {
    background (0,0,0);

```

```

//Screen space
fill(255, 255, 255);
rect (25,25,350,350);
};

```

```

var backgroundPlayerScreen = function () {
    background (0,0,0);
    //player A Side
    fill(255, 255, 255);
    rect (200,25,175,350);
    //Player B Side
    fill(255, 255, 255);
    rect (25,25,175,350);
};

```

```

// main code
backgroundScreen ();
//code start button to enable phase 1;

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";
};

Button.prototype.draw = function() {
    fill(227, 227, 227);
    rect(this.x, this.y, this.width, this.height, 5);
    fill(0, 0, 0);
    textSize(20);
    textAlign(LEFT, TOP);
    text(this.label, this.x+10, this.y+this.height/4);
};

Button.prototype.isMouseInside = function() {
    return mouseX > this.x &&
        mouseX < (this.x + this.width) &&
        mouseY > this.y &&
        mouseY < (this.y + this.height);
};

//buttons set up

var startButton = new Button({x: 50, y: 50, label: "Start Game"});
var restartButton = new Button({x: 200, y: 50, label: "Restart"});
startButton.draw();

//goldenGate Check
//mouse check

mouseClicked = function() {
    if (startButton.isMouseInside()) {
        if (phase === 0) {
            phase = 1;
            starters = 0;

```

```

        //println ("button");
        round1 = [];
        round2 = [];
        generateCourseNumbers (round1);
        generateCourseNumbers (round2);
    }
}
if (restartButton.isMouseInside()) {
    if (phase === -1) {
        phase = 1;
        starters = 0;
        //println ("button 2");
        round1 = [];
        round2 = [];
        generateCourseNumbers (round1);
        generateCourseNumbers (round2);
    }
}
};
//var test = 0;

// timer var function

mouseClicked = function() {
    if (startButton.isMouseInside()) {
        if (phase === 0) {
            phase = 1;
            starters = 0;
            startTime = millis(); // Record the start time
            round1 = [];
            round2 = [];
            generateCourseNumbers (round1);
            generateCourseNumbers (round2);
        }
    }
    if (restartButton.isMouseInside()) {
        if (phase === -1) {
            phase = 1;
            starters = 0;
            startTime = millis(); // Record the start time
            round1 = [];
            round2 = [];
            generateCourseNumbers (round1);
            generateCourseNumbers (round2);
        }
    }
}

```



```
    }  
  }  
};
```

```
frameRate(50);
```

```
keyReleased = function() {  
  if (keyCode === 18) {  
    if (AorB === 0) {  
      AorB = 1;  
      //println ("toggle " + AorB);  
    }  
    else if (AorB === 1) {  
      AorB = 0;  
      //println ("toggle " + AorB);  
    }  
  }  
};
```

```
draw = function () {  
  //timer var code  
  
  if (phase === 1 || phase === 2) {  
    elapsedTime = millis() - startTime; // Calculate elapsed time  
    timeMS = (round(elapsedTime/1000));  
  }  
}
```

```
  //testing code  
  //test = test + 1;  
  //println (test);
```

```
//left-right-up-down format
```

```
//A B switch L-Shift
```

```
/*
```

```
  if (keysReleased && (keyCode === 18)) {  
    if (AorB === 0) {  
      AorB = 1;  
      println ("toggle " + AorB);  
    }  
    else if (AorB === 1) {  
      AorB = 0;  
      println ("toggle " + AorB);  
    }  
  }  
}
```

```

    }
}
*/
//37 left arrow
if (keyIsPressed && (keyCode === 37)) {
    if (AorB === 0) {
        XofA=XofA-1;
    }
    if (AorB === 1) {
        XofB=XofB-1;
    }
}
//39 = right arrow
if (keyIsPressed && (keyCode === 39)) {
    if (AorB === 0) {
        XofA=XofA+1;
    }
    if (AorB === 1) {
        XofB=XofB+1;
    }
}
//39 = up arrow
if (keyIsPressed && (keyCode === 38)) {
    if (AorB === 0) {
        YofA=YofA-1;
    }
    if (AorB === 1) {
        YofB=YofB-1;
    }
}
//40 = down arrow
if (keyIsPressed && (keyCode === 40)) {
    if (AorB === 0) {
        YofA=YofA+1;
    }
    if (AorB === 1) {
        YofB=YofB+1;
    }
}

//Barriers check

if ((YofA >= 2) || (YofB >= 4)) {
    phase = -1;
}

```

```

    WL = 0;
}
if ((YofA <= -325) || (YofB <= -325)) {
    phase = -1;
    WL = 0;
}
if ((XofA >= 78.5) || (XofB >= 78.5)) {
    phase = -1;
    WL = 0;
}
if ((XofA <= -78.5) || (XofB <= -78.5)) {
    phase = -1;
    WL = 0;
}

if (phase === 1) {
    backgroundPlayerScreen();
    CourseCreate("A", round1);
    CourseCreate("B", round1);

    // Display the timer on the screen
    fill(255, 0, 255);
    textSize(10);
    text("Time: " + timeMS + "s", 10, 10);

    //insert main code round 1 A
    DrawBitmoji(112.5+XofA, 360+YofA, 40);
    //insert main code round 1 B
    drawBitmoji(0.1, 268+XofB, 336+YofB);

}
if (((360+YofA)>25 && (360+YofA)<50) && ((360+YofB) >25 && (360+YofB)<50)) {
    if (((112.5+XofA-10)>100) && ((112.5+XofA-10) < 120)) && (((290+XofB) > 275) &&
((290+XofB) < 300))) {
        gate = 1;
    }
}
if (gate === 1) {
    phase = 2;
    gate = 0;
    XaYnXbY_Reset();
}
else if (phase === 2) {
    backgroundPlayerScreen();

```

```

CourseCreate("A", round2);
CourseCreate("B", round2);

// Display the timer on the screen
fill(255, 0, 255);
textSize(10);
text("Time: " + timeMS + "s", 10, 10);

//insert main code round 2 A
DrawBitmoji(112.5+XofA,360+YofA,40);
//insert main code round 2 B
drawBitmoji(0.1, 268+XofB, 336+YofB);
if (((360+YofA)>25 && (360+YofA)<50) && ((360+YofB) >25 && (360+YofB)<50)) {
    if (((112.5+XofA-10)>100) && ((112.5+XofA-10) < 120)) && (((290+XofB) > 275) &&
((290+XofB) < 300))) {
        gate = 1;
    }
}
if (gate === 1) {
    phase = -1;
    gate = 0;
    WL = 1;
    XaYnXbY_Reset();
}
}
else if (phase === -1) {
    backgroundScreen();
    //insert end code
    XaYnXbY_Reset();

//Display WL (win/lose) status
if (WL === 1) {
    fill(0, 0, 0);
    textSize(50);
    text("You Win!", 150, 200);

// Display the timer on the screen
fill(255, 0, 255);
textSize(10);
text("Time: " + timeMS + "s", 10, 10);

// Display confetti as a reunion effect

for (var j = 0; j < 100; j++) {

```

```

        fill(random(255), random(255), random(255));
        ellipse(random(width), random(height), 10, 10);
    }
}
if (WL === 0) {
    fill(0, 0, 0);
    textSize(50);
    text("You Lose :", 150, 200);

    // Display the timer on the screen
    fill(255, 0, 255);
    textSize(10);
    text("Time: " + timeMS + "s", 10, 10);
}
//check for play again button
restartButton.draw();
//if button pressed start phase 1 again
}
//println (mouseX);
//println (mouseY);
};

```

```

//Notes
//Create Barriers for testMojilMG + Gravity
//Obstacle courses of both rounds be still

```

```

//When ready, bitmoji draw code will be up top and we will call it below
var DrawShirt = function(x,y,h) {
    var p = h/200 ;

    fill(0, 10, 87);
    arc(x+(p*75), y+(p*150), p*100, p*40, 180, 360); // shirt

    fill(255, 255, 255);
    textSize(14*p);
    text("C X M", x+(p*55), y+(p*135), p*100, p*100);
};

```

```

var DrawHead = function(x,y,h) {
  var p = h/200 ;

  fill(0, 0, 0);
  ellipse(x+(p*75), y+(p*65), p*80, p*90); // hair
  rect(x+(p*35),y+(p*60),p*79,p*70);

  fill(186, 160, 109);
  rect(x+(p*70),y+(p*125),p*10,p*10); //neck

  fill(186, 160, 109);
  ellipse(x+(p*75), y+(p*75), p*75, p*100); // face

  fill(255, 255, 255);
  ellipse(x+(p*64), y+(p*70), p*10, p*4); // left eye
  ellipse(x+(p*86), y+(p*70), p*10, p*4); // right eye

  fill(74, 43, 0);
  ellipse(x+(p*64), y+(p*70), p*4, p*4); // left eye
  ellipse(x+(p*86), y+(p*70), p*4, p*4); // right eye

  noFill();
  arc(x+(p*64), y+(p*68), p*15, p*6, 180, 360); // eyebrow
  arc(x+(p*86), y+(p*68), p*15, p*6, 180, 360); // eyebrow

  fill(0, 0, 0);
  triangle(x+(p*74), y+(p*100), x+(p*75), y+(p*85), x+(p*76), y+(p*100)); //nose

  noFill();
  arc(x+(p*75), y+(p*105), p*30, p*15, 10, 170); // mouth

  fill(0, 0, 0);
  arc(x+(p*75), y+(p*60), p*75, p*70, 178, 362); // hair bangs
};

var DrawBitmoji = function(x,y,h) {
  var p = h/200;
  x = x - (75*p);
  y = y - (90*p);
  DrawHead(x,y,h);
  DrawShirt(x,y,h);
};

```

```
DrawBitmoji(100,100,153);
```

```
//Player A
```

```
//Code divider for bitmojis of player A and B
```

```
//Player B
```

```
var drawBitmoji = function(bitmojiScale, x, y) {  
  noStroke();  
  fill(176, 110, 80);  
  ellipse(x + 200 * bitmojiScale, y + 152 * bitmojiScale, 70 * bitmojiScale, 86 * bitmojiScale); //  
head  
  fill(0, 0, 0);  
  ellipse(x + 201 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 207 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 192 * bitmojiScale, y + 101 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 183 * bitmojiScale, y + 104 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 214 * bitmojiScale, y + 98 * bitmojiScale, 36 * bitmojiScale, 36 * bitmojiScale);  
  ellipse(x + 223 * bitmojiScale, y + 110 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 231 * bitmojiScale, y + 123 * bitmojiScale, 17 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 174 * bitmojiScale, y + 113 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 246 * bitmojiScale, y + 134 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 166 * bitmojiScale, y + 125 * bitmojiScale, 18 * bitmojiScale, 20 * bitmojiScale);  
  ellipse(x + 246 * bitmojiScale, y + 119 * bitmojiScale, 26 * bitmojiScale, 31 * bitmojiScale);  
  ellipse(x + 231 * bitmojiScale, y + 104 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 200 * bitmojiScale, y + 97 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 177 * bitmojiScale, y + 97 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 172 * bitmojiScale, y + 107 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 154 * bitmojiScale, y + 121 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 153 * bitmojiScale, y + 135 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 247 * bitmojiScale, y + 137 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 152 * bitmojiScale, y + 146 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  ellipse(x + 247 * bitmojiScale, y + 148 * bitmojiScale, 33 * bitmojiScale, 26 * bitmojiScale);  
  
  stroke(0, 0, 0);  
  fill(176, 110, 80);  
  rect(x + 175 * bitmojiScale, y + 133 * bitmojiScale, 18 * bitmojiScale, 17 * bitmojiScale); //  
glasses left  
  rect(x + 204 * bitmojiScale, y + 133 * bitmojiScale, 18 * bitmojiScale, 17 * bitmojiScale); //  
glasses right  
  line(x + 193 * bitmojiScale, y + 144 * bitmojiScale, x + 205 * bitmojiScale, y + 143 *  
bitmojiScale); // nose bridge  
  line(x + 176 * bitmojiScale, y + 143 * bitmojiScale, x + 146 * bitmojiScale, y + 137 *  
bitmojiScale); // left glasses
```

```

    line(x + 259 * bitmojiScale, y + 132 * bitmojiScale, x + 223 * bitmojiScale, y + 144 *
bitmojiScale); // right classes

    fill(92, 54, 54);
    ellipse(x + 185 * bitmojiScale, y + 142 * bitmojiScale, 7 * bitmojiScale, 7 * bitmojiScale); // left
eye
    ellipse(x + 214 * bitmojiScale, y + 142 * bitmojiScale, 7 * bitmojiScale, 7 * bitmojiScale); // right
eye
    fill(176, 110, 80);
    triangle(x + 192 * bitmojiScale, y + 160 * bitmojiScale, x + 206 * bitmojiScale, y + 159 *
bitmojiScale, x + 198 * bitmojiScale, y + 151 * bitmojiScale
    );
    noStroke();
    arc(x + 212 * bitmojiScale, y + 169 * bitmojiScale, 23 * bitmojiScale, 36 * bitmojiScale, 1, 361);
// nose shaping
    arc(x + 193 * bitmojiScale, y + 170 * bitmojiScale, 9 * bitmojiScale, 25 * bitmojiScale, 1, 361); //
nose shaping
    arc(x + 199 * bitmojiScale, y + 171 * bitmojiScale, 9 * bitmojiScale, 25 * bitmojiScale, 1, 361); //
nose shaping
    stroke(0, 0, 0);
    fill(255, 255, 255); // teeth
    arc(x + 200 * bitmojiScale, y + 175 * bitmojiScale, 30 * bitmojiScale, 11 * bitmojiScale, 7, 391);
// mouth

    fill(217, 158, 208);
    quad(
        x + 176 * bitmojiScale,
        y + 194 * bitmojiScale,
        x + 223 * bitmojiScale,
        y + 194 * bitmojiScale,
        x + 300 * bitmojiScale,
        y + 351 * bitmojiScale,
        x + 100 * bitmojiScale,
        y + 350 * bitmojiScale
    ); // dress
    rect(x + 116 * bitmojiScale, y + 167 * bitmojiScale, 30 * bitmojiScale, 88 * bitmojiScale); //
sleeve
    rect(x + 253 * bitmojiScale, y + 167 * bitmojiScale, 30 * bitmojiScale, 88 * bitmojiScale); //
sleeve

    textSize(32 * bitmojiScale);
    fill(0);
    text("L", x + 178 * bitmojiScale, y + 259 * bitmojiScale);
    textSize(32 * bitmojiScale);

```



```
fill(0);
text("R", x + 199 * bitmojiScale, y + 259 * bitmojiScale);
};
```

//here is where bitmoji code will end when we implement it

```
//declare var variables
//stage phases
var phase = 0;
var WL = 0; //win/lose
var i = 0;
var gate = 0; //gate activation for stages
var starters = 1;
var AtoB = 0;
var AorB = 0;
```

//Bitmoji possitional code Decloration

```
//A
var XofA = 0;
var YofA = 0;
```

//B

```
var XofB = 0;
var YofB = 0;
```

```
// round obstacle arrays
var round1 = [];
var round2 = [];
```

```
var BarPara = [];
```

```
//Obstacle 1
var Obstacle1 = function (x,y) {
  fill(0, 0, 0);
  rect (x,y,130,16);
};
```

```
//Obstacle 2
var Obstacle2 = function (x,y) {
  fill(0, 0, 0);
  rect (x+45,y,130,16);
};
```

```

//Obstacle 3
var Obstacle3 = function (x,y) {
    fill(0, 0, 0);
    rect (x+32.5,y,110,16);
};

//generate array course numbers

var generateCourseNumbers = function(roundNumber) {
    for (var i=0; i<5; i++) {
        roundNumber.push (round(random(1,3)));
    }
};

//generate course

var CourseCreate = function (Player, roundNumber) {
    if (Player === "A") {
        AtoB = 25;
        fill(199, 189, 82);
        ellipse((25+87.5), 40, 20, 20);
    }
    if (Player === "B") {
        AtoB = 200;
        fill(199, 189, 82);
        ellipse((200+87.5), 40, 20, 20);
    }

    for (var i=0; i < roundNumber.length; i++) {
        if (roundNumber[i] === 1){
            Obstacle1 (AtoB, (355-40)-(i*62));
        }
        if (roundNumber[i] === 2){
            Obstacle2 (AtoB, (355-50)-(i*62));
        }
        if (roundNumber[i] === 3){
            Obstacle3 (AtoB, (355-50)-(i*62));
        }
    }
};

var XaYnXbY_Reset = function () {
    //A

```

```

    XofA = 0;
    YofA = 0;
    //B
    XofB = 0;
    YofB = 0;
};

//background setup code

var backgroundScreen = function () {
background (0,0,0);

//Screen space
fill(255, 255, 255);
rect (25,25,350,350);
};

var backgroundPlayerScreen = function () {
background (0,0,0);
//player A Side
fill(255, 255, 255);
rect (200,25,175,350);
//Player B Side
fill(255, 255, 255);
rect (25,25,175,350);
};

// main code
backgroundScreen ();
//code start button to enable phase 1;

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";
};

Button.prototype.draw = function() {
    fill(227, 227, 227);
    rect(this.x, this.y, this.width, this.height, 5);
    fill(0, 0, 0);

```

```
    textSize(20);
    textAlign(LEFT, TOP);
    text(this.label, this.x+10, this.y+this.height/4);
};
```

```
Button.prototype.isMouseInside = function() {
    return mouseX > this.x &&
        mouseX < (this.x + this.width) &&
        mouseY > this.y &&
        mouseY < (this.y + this.height);
};
```

```
//buttons set up
```

```
var startButton = new Button({x: 50, y: 50, label: "Start Game"});
var restartButton = new Button({x: 200, y: 50, label: "Restart"});
startButton.draw();
```

```
//goldenGate Check
```

```
//mouse check
```

```
mouseClicked = function() {
    if (startButton.isMouseInside()) {
        if (phase === 0) {
            phase = 1;
            starters = 0;
            //println ("button");
            round1 = [];
            round2 = [];
            generateCourseNumbers (round1);
            generateCourseNumbers (round2);
        }
    }
    if (restartButton.isMouseInside()) {
        if (phase === -1) {
            phase = 1;
            starters = 0;
            //println ("button 2");
            round1 = [];
            round2 = [];
            generateCourseNumbers (round1);
            generateCourseNumbers (round2);
        }
    }
}
```

```

};
//var test = 0;

frameRate(50);

keyReleased = function() {
  if (keyCode === 18) {
    if (AorB === 0) {
      AorB = 1;
      //println ("toggle " + AorB);
    }
    else if (AorB === 1) {
      AorB = 0;
      //println ("toggle " + AorB);
    }
  }
};

draw = function () {
  //testing code
  //test = test + 1;
  //println (test);

  //left-right-up-down format
  //A B switch L-Shift
  /*
  if (keyIsReleased && (keyCode === 18)) {
    if (AorB === 0) {
      AorB = 1;
      println ("toggle " + AorB);
    }
    else if (AorB === 1) {
      AorB = 0;
      println ("toggle " + AorB);
    }
  }
  */

  //37 left arrow
  if (keyIsPressed && (keyCode === 37)) {
    if (AorB === 0) {
      XofA=XofA-1;
    }
    if (AorB === 1) {
      XofB=XofB-1;
    }
  }
}

```

```

    }
}
//39 = right arrow
if (keysPressed && (keyCode === 39)) {
    if (AorB === 0) {
        XofA=XofA+1;
    }
    if (AorB === 1) {
        XofB=XofB+1;
    }
}
//39 = up arrow
if (keysPressed && (keyCode === 38)) {
    if (AorB === 0) {
        YofA=YofA-1;
    }
    if (AorB === 1) {
        YofB=YofB-1;
    }
}
//40 = down arrow
if (keysPressed && (keyCode === 40)) {
    if (AorB === 0) {
        YofA=YofA+1;
    }
    if (AorB === 1) {
        YofB=YofB+1;
    }
}

//Barriers check

if ((YofA >= 2) || (YofB >= 4)) {
    phase = -1;
    WL = 0;
}
if ((YofA <= -325) || (YofB <= -325)) {
    phase = -1;
    WL = 0;
}
if ((XofA >= 78.5) || (XofB >= 78.5)) {
    phase = -1;
    WL = 0;
}

```

```

if ((XofA <= -78.5) || (XofB <= -78.5)) {
    phase = -1;
    WL = 0;
}

if (phase === 1) {
    backgroundPlayerScreen();
    CourseCreate("A", round1);
    CourseCreate("B", round1);
    //insert main code round 1 A
    DrawBitmoji(112.5+XofA,360+YofA,40);
    //insert main code round 1 B
    drawBitmoji(0.1, 268+XofB, 336+YofB);

    if (((360+YofA)>25 && (360+YofA)<50) && ((360+YofB) >25 && (360+YofB)<50)) {
        if (((112.5+XofA-10)>100) && ((112.5+XofA-10) < 120)) && (((290+XofB) > 275) &&
((290+XofB) < 300))) {
            gate = 1;
        }
    }
    if (gate === 1) {
        phase = 2;
        gate = 0;
        XaYnXbY_Reset();
    }
}

if (phase === 2) {
    backgroundPlayerScreen();
    CourseCreate("A", round2);
    CourseCreate("B", round2);
    //insert main code round 2 A
    DrawBitmoji(112.5+XofA,360+YofA,40);
    //insert main code round 2 B
    drawBitmoji(0.1, 268+XofB, 336+YofB);
    if (((360+YofA)>25 && (360+YofA)<50) && ((360+YofB) >25 && (360+YofB)<50)) {
        if (((112.5+XofA-10)>100) && ((112.5+XofA-10) < 120)) && (((290+XofB) > 275) &&
((290+XofB) < 300))) {
            gate = 1;
        }
    }
    if (gate === 1) {
        phase = -1;
        gate = 0;
        WL = 1;
    }
}

```

```

        XaYnXbY_Reset();
    }
}
if (phase === -1) {
    backgroundScreen();
    //insert end code
    XaYnXbY_Reset();

    //Display WL (win/lose) status
    if (WL === 1) {
        fill(0, 0, 0);
        textSize(50);
        text("You Win!", 150, 200);

        // Display confetti as a reunion effect

        for (var j = 0; j < 100; j++) {
            fill(random(255), random(255), random(255));
            ellipse(random(width), random(height), 10, 10);
        }
    }
    if (WL === 0) {
        fill(0, 0, 0);
        textSize(50);
        text("You Lose :)", 150, 200);
    }
    //check for play again button
    restartButton.draw();
    //if button pressed start phase 1 again
}
//println (mouseX);
//println (mouseY);
};

```

```

//Notes
//Create Barriers for testMojilMG + Gravity
//Obstacle courses of both rounds be still

```