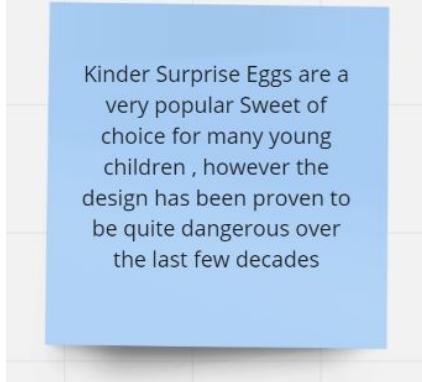

p5.js portfolio

By Shaan Rehsi

Food Project

- Kinder Surprise Egg



Research Project

HYPOTHESIS:

The Kinder egg packaging is dangerous for kids. Since the late 90s to the early 00s, there has been some controversy towards the toys inside the chocolate egg. Its small parts in the toys have been deemed a choking hazard and therefore been banned in countries such as the US.

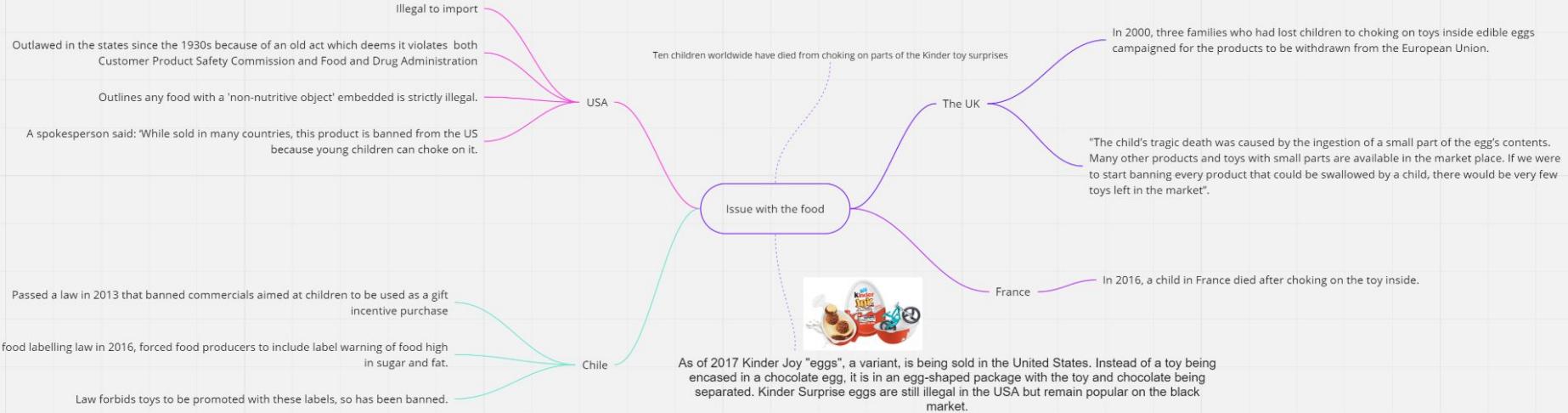
How design can alter the packaging to make products safer?

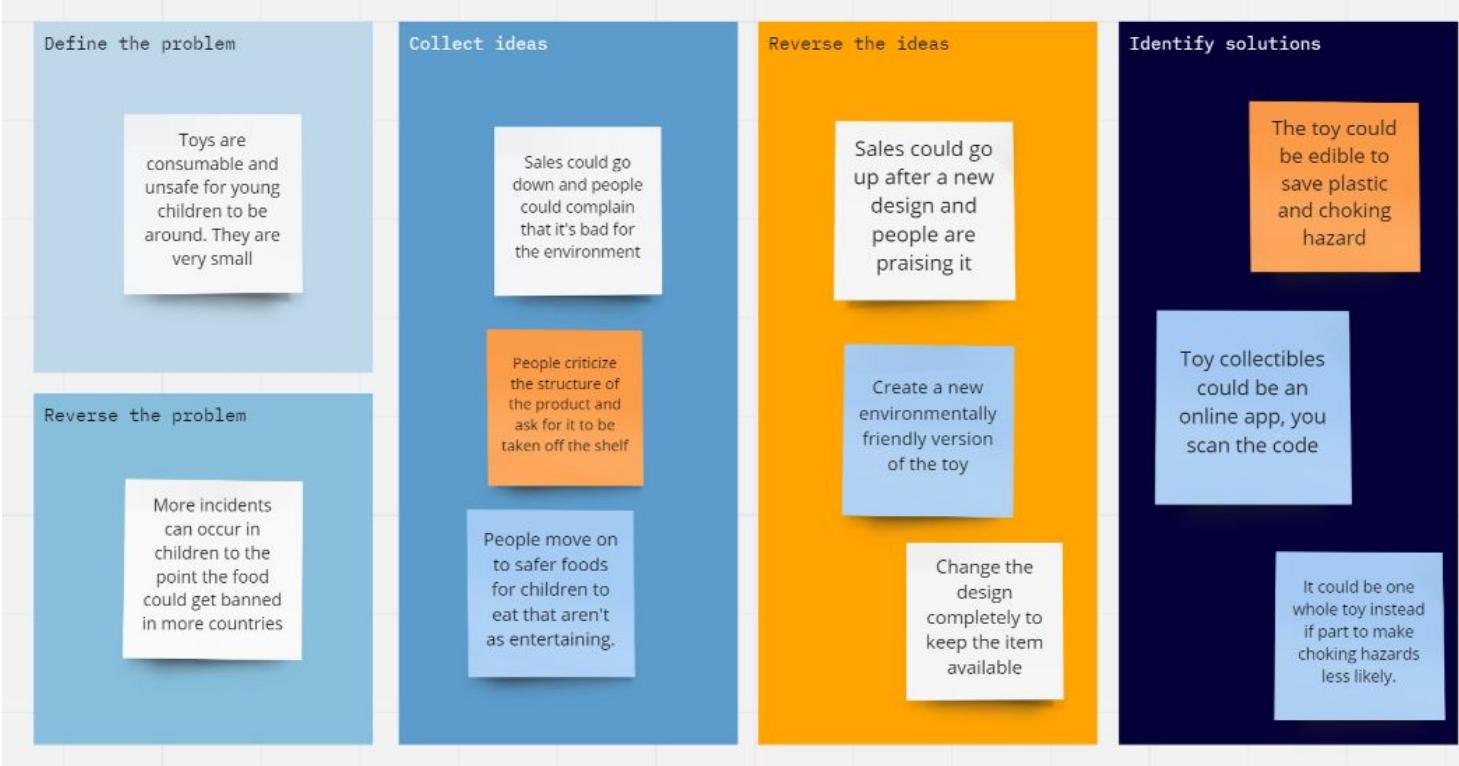


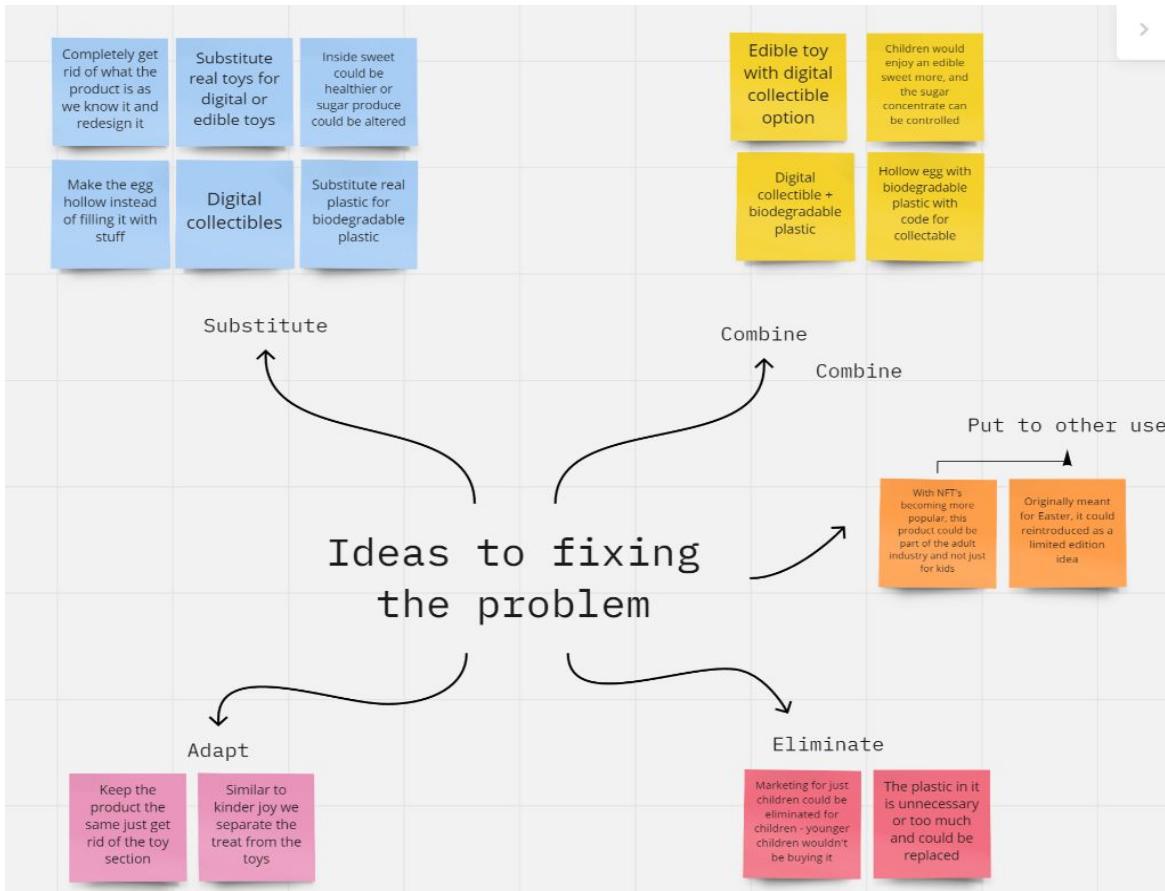
Biodegradable plastic - 74% of consumers are more likely to pay more for sustainable packaging

Digital collectibles would be more popular with adults - the rise of NFT's - there could be a profit for the future of this brand

1 in 3 parents admits to having thrown away toys in working conditions - a lot of toys are plastic







Mouse Clicked and Setup()

```
let er;
let er2;

function setup() {
  createCanvas(600, 800);
  textSize(customFont);

  for (var i = 0; i < 20; i++) {
    eggs[i] = new Egg(random(width), random(height));
  }
  er = new EggRing(width * 0.15, height * 0.1, 0.1, 60);
  er2 = new EggRing(width * 0.15, height * 0.1, 0.1, 70);
}

function preload(){
  customFont = loadFont("Napolitana.otf");
}

function draw() {
  background("#49B1F2");
  //drop eggs
  for (var i = 0; i < eggs.length; i++) {
    eggs[i].update();
    eggs[i].display();
    eggs[i].intersects();
  }
  bottomegg();
  topegg();
}

function mousePressed() {
  if (!mouseIsPressed == true) {
    fill(0);
    textAlign(CENTER);
    fill(0, 255, mouseX, mouseY);
    stroke(255, mouseX, mouseY);
    strokeWeight(10);
    textSize(50);
    text("CLICK ME", 300, 340);
    textSize(50);
    fill(0, 255, mouseX, mouseY);
    stroke(255, mouseX, mouseY);
    strokeWeight(10);
    text("SURPRISE", 300, 520);
  }
  er.transmit();
}
```

- Wanted to go with with theme of the packaging with my poster to sell the item
- Used mouseIsPressed as the surprise feature

```

function topegg(){
  if (mouseIsPressed == true) {
    stroke(0);
    text("CLICK ME", 300, 400);
    textSize(20);
    stroke(0);
    fill(139,76,57); // brown
    textSize(20);
  }
  else {
    fill(255);
  }
  strokeWeight(1);
  arc(eggX, eggY, eggW, eggH, 3.1, 6.3, CLOSE);
  line(eggX-(eggW/2), eggY, eggX-eggW/5, eggY+(eggH/6));
  line(eggX-eggW/5, eggY+(eggH/6), eggX, eggY);
  line(eggX, eggY, eggX+eggW/5, eggY+(eggH/6));
  line(eggX+eggW/5, eggY+(eggH/6), eggX+(eggW/2)-1, eggY);
  noStroke();
  triangle(eggX-(eggW/2), eggY, eggX-eggW/5, eggY+(eggH/6), eggX, eggY);
  triangle(eggX, eggY, eggX+eggW/5, eggY+(eggH/6), eggX+(eggW/2), eggY);
  textAlign(CENTER);
}

```



Functions for what happens to the two parts of the egg and design

```

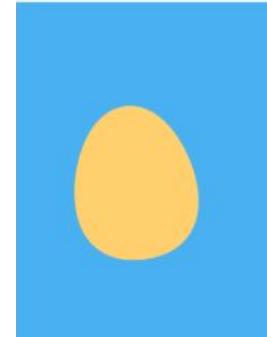
function bottomegg(){
  if (mouseIsPressed == true) {
    fill(255, 215, 0); // White
  }
  else {
    fill("red"); // Black
  }
  arc(eggX, eggY, eggW, eggH, 6.3, 3.1, CLOSE);
  line(eggX-(eggW/2), eggY, eggX-eggW/5, eggY+(eggH/6));
  line(eggX-eggW/5, eggY+(eggH/6), eggX, eggY);
  line(eggX, eggY, eggX+eggW/5, eggY+(eggH/6));
  line(eggX+eggW/5, eggY+(eggH/6), eggX+(eggW/2)-1, eggY);
  noStroke();
  noFill();
}

```

Transmit Egg movement

```
class EggRing {  
constructor(x, y, t, sp) {  
    this.x = x;  
    this.y = y;  
    this.t = t;  
    this.sp = sp;  
    this.ovoid = new EggTilt(this.x, this.y, this.t, this.sp);  
}  
  
transmit() {  
    this.ovoid.wobble();  
    this.ovoid.display();  
}  
}
```

Class for the title
and wobble
functions in my code



Egg Rain and EggTilt

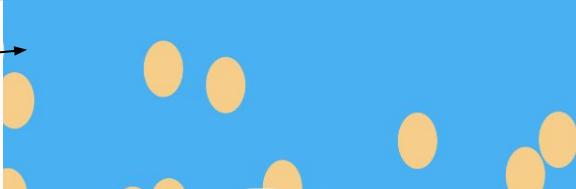
```
function Egg(x, y) {
  this.x = x;
  this.y = y;
  this.r1 = 35;
  this.r2 = 50;
  this.col = color(246, 206, 137);

  function changeColor(){
    this.col=color(random(255),random(255),random(255));
  };

  this.display = function() {
    noStroke();
    fill(this.col);
    ellipse(this.x, this.y, this.r1, this.r2);
  };

  this.update = function() {
    //this.x=this.x+ random(1);
    this.y = this.y + random(0.8);
  };

  this.intersects = function() {
    var d = dist(this.x, this.y, mouseX, 300);
    for (var i = 0; i < eggs.length; i++){
      if (d < 30) {
        this.x=mouseX;
        this.y = 280;
        this.col = color(246, 206, 137,100);
      }
    }
  }
}
```



- Added a rainfall technique

```
}  
if (this.y > height) {  
  this.y = random(0, -height);  
  this.gravity = 0;  
}  
};  
};
```

```
class EggTilt {
  constructor(xpos, ypos, t, s) {
    this.x = 50;
    this.y = 100;
    this.tilt = t;
    this.scalar = s / 100.0;
    this.angle = 0.0;
  }

  wobble() {
    this.tilt = cos(this.angle) / 8;
    this.angle += 0.1;
  }

  display() {
    noStroke();
    fill(255, mouseX, mouseY);
    push();
    translate(this.x, this.y);
    rotate(this.tilt);
    scale(this.scalar);
    beginShape();
    vertex(0, -100);
    bezierVertex(25, -100, 40, -65, 40, -40);
    bezierVertex(40, -15, 25, 0, 0, 0);
    bezierVertex(-25, 0, -40, -15, -40, -40);
    bezierVertex(-40, -65, -25, -100, 0, -100);
    endShape();
    pop();
  }
}
```

PONG Game

- Album art inspired by Flower Boy, Tyler the Creator
I chose it for the colour theme
There were elements I could draw
Something I am passionate about



```

function draw() {
  const top = color(topColor);
  const bottom = color(bottomColor);

  for(let y = 0; y < height; y ++){
    const lineColor = lerpColor(top, bottom, y / height);
    stroke(lineColor);
    line(0, y, width, y);
  }
  fill("#445310");
  noStroke();
  ellipse(310, 380, 400, 250);
  ellipse(70, 380, 400, 250);

  makeCloud(cloudx - 200, cloudy - 70);

  makeCloud(cloudx - 100, cloudy + 70);

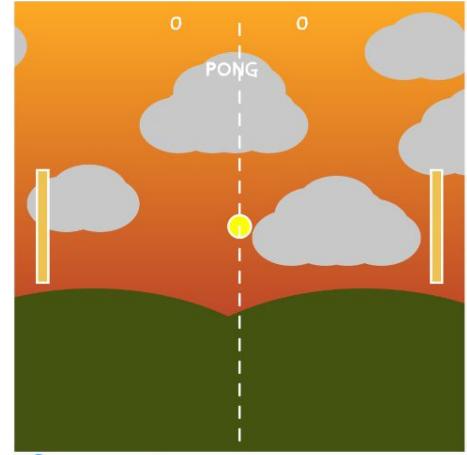
  //original shape and cloud patterns
  makeCloud(cloudx, cloudy);
  makeCloud(cloudx +50 , cloudy);
  makeCloud(cloudx+ 30 , cloudy- 30);

  //cloudshape 2
  makeCloud(cloudx + 100, cloudy + 100);
  makeCloud(cloudx + 120, cloudy + 80);
  makeCloud(cloudx + 150, cloudy + 100);

  makeCloud(cloudx + 200, cloudy - 65);
}

```

- Created gradient and used Adobe Colour theme
- Used ellipses
- Created custom clouds that moved



```

textFont(customFont);
text("PONG", 170, 50);
for ( var i = 0; i < 400; i += 20){
  line(200, i, 200, i + 10);
  stroke(255);
  fill(255);
  textAlign(CENTER);
  textSize(32);
  text("white");
}

}

```

makeCloud()

```
//original shape and cloud patterns  
makeCloud(cloudx, cloudy);  
makeCloud(cloudx +50 , cloudy);  
makeCloud(cloudx+ 30 , cloudy- 30);
```

```
//cloudshape 2  
makeCloud(cloudx + 100, cloudy + 100);  
makeCloud(cloudx + 120, cloudy + 80);  
makeCloud(cloudx + 150, cloudy + 100);
```

```
makeCloud(cloudx + 200, cloudy - 65);
```

```
//cloudshape 3  
makeCloud(cloudx + 230, cloudy + 20);  
makeCloud(cloudx + 250, cloudy );  
makeCloud(cloudx + 270, cloudy + 20)
```

```
makeCloud(cloudx + 300, cloudy + 110);
```

```
cloudx += 0.1;
```

```
song.setVolume(0.5)  
song.setVolume(slider.value());  
movePaddles();  
p1.show();  
p2.show();
```

```
function makeCloud(cloudx, cloudy){  
    fill(200);  
    noStroke();  
    ellipse(cloudx, cloudy, 70, 50);  
    ellipse(cloudx + 10, cloudy + 10, 70, 50);  
    ellipse(cloudx - 20, cloudy + 10, 70, 50);  
}
```

Main cloud shape

Customised clouds so they're all different

Clouds all move



Pong Game - Files

- Created different files for the, ball, paddles and the backdrop



```
1 class Ball {
2
3     constructor(x, y, r, speed = 3) {
4         this.spawn = createVector(x, y)
5         this.speed = 3; //create location of paddles
6         this.r = r;
7         this.resetball();
8
9     }
10
11    resetball() {
12        this.pos = this.spawn.copy(); //how it reacts to being hit
13        and velocity of paddles
14        let angle = random(-PI/4, PI/4)
15        this.vel = p5.Vector.fromAngle(angle, this.speed);
16        if (random(1) > 0.5) this.vel.x *= -1;
17
18    }
19
20    outOfBounds() {
21
22        // If the ball is out of the screen,
23        // return the side, otherwise return false
24
25    }
}
```

Defines what happens when the ball leaves the canvas

```
22
23    if (this.pos.x > width + this.r) {
24        this.resetball();
25        return 'right';
26    }
27
28    if (this.pos.x < -this.r) {
29        this.resetball();
30        return 'left';
31    }
32
33    return false;
34}
35
36
37
38    hit(p1, p2) {
39        for (let pad of [p1, p2]) {
40            let padX = pad.pos.x;
41            let padY = pad.pos.y;
42            let ballX = ball.pos.x;
43            let ballY = ball.pos.y;
44            let r = this.r;
45
46            // if ball collides on x-axis
47            if ((padX - r) < (ballX) && (ballX) < (padX + pad.w + r)) {
48                // and on y-axis
49                if ((padY - r) < ballY && ballY < (padY + pad.h + r)) {
```

- The ball class, defines the speed, size and velocity of the ball
- Defines what happens if the ball is hit

Ball

```
let padCenter = createVector(pad.pos.x + pad.w/2,
    pad.pos.y + pad.h/2)

    // Vector from center of pad to center of ball
    this.vel = this.pos.copy().sub(padCenter);
    this.vel.limit(10);
    bounceSound.play();

    // basically halve that angle so it points more to the
    center

    let a = this.vel.heading();
    if (a > -PI/2 && a < PI/2) {
        this.vel = p5.Vector.fromAngle(a/2, 5);
    } else {
        this.vel.rotate(PI);
        let a = this.vel.heading();
        this.vel = p5.Vector.fromAngle(PI + a/2, 5);
    }

}

}

}

update() {
    this.pos.add(this.vel);
}
```

```
if (go) ball.update();
ball.hit(p1, p2);
ball.show();
backdrop();
```

in sketch.js

I added change to the speed of the ball from the angle it was hit

Display colour to go
with the theme

Paddle

```
1▼ class Paddle {
2
3▼   constructor(x, y, w, h) {           //creates the size of the paddles
4     this.pos = createVector(x, y);
5     this.w = w;
6     this.h = h;
7     this.score = 0;
8   }
9
10▼ move(amt) {
11   this.pos.y += amt;      //locations of paddles
12   this.pos.y = constrain(this.pos.y, 10, height - 10 - this.h);
13 }
14
15▼ show() {
16   fill("#EDC454");    //design of paddles
17   stroke("white");
18   strokeWeight(2);
19   rect(this.pos.x, this.pos.y, this.w, this.h);
20 }
21
22 }
23 }
```

- Constructing position, size and design
- Used colours from the album

Backdrop

```
let textOffsetX = 50;
let textOffsetY = 10;

function backdrop() {

    stroke(80)
    strokeWeight(2);

    textSize(20);
    noStroke();
    fill(255);

    textAlign(RIGHT, TOP);
    text(p1.score, width/2 - textOffsetX, textOffsetY);

    textAlign(LEFT);
    text(p2.score, width/2 + textOffsetX, textOffsetY);

}
```

- Creates the position of the scores and the colour and size

PreLoad and Load

```
Ball.js  
PINGPONG.TTF  
Paddle.js  
ballBounce.mp3  
gameScore.wav  
index.html  
rainbow.mp3  
restart.wav  
score.mp3  
sketch.js  
style.css  
win.wav
```

```
1 /  
18 var restartGame;  
19 var boundSound;  
20 var scoreSound;  
21 let customFont;  
22  
23▼ function preload(){  
24   song = loadSound("rainbow.mp3");  
25   restartGame = loadSound("restart.wav");  
26   bounceSound = loadSound("ballBounce.mp3");  
27   scoreSound = loadSound("gameScore.wav")  
28   customFont = loadFont("PINGPONG.TTF")  
29  
30 }  
31 }
```

- I wanted to add sound effects if different things happen

- Song from the album

```
function setup() {  
  createCanvas(400, 400);  
  noSmooth();  
  alert('-> Use keys W and S and UP and DOWN arrows to move the paddles\n'+  
    '-> Press Spacebar to start each round, and R to reset the game\n'+  
    '-> Hit the ball and make your opponent miss to score!\n\n'+  
    '-> Drag the slider to adjust volume!\n\n'+  
    '-> Winner gets to 3 points!\n\n'+  
    'Good Luck!');  
  song = loadSound("rainbow.mp3", loaded);  
  slider = createSlider(0, 1, 0.1, 0.01);  
  topColor = ("#FEAA24");  
  bottomColor = ("#A31E2B");  
  
  ball = new Ball(width/2, height/2, 10, 10);  
  
  p1 = new Paddle(20, height/2 - 50, 10, 100);  
  p2 = new Paddle(width - 30, height/2 - 50, 10, 100);  
}  
  
function loaded() {  
  song.play();  
}
```

- Gives the instructions

Scoring and Winning

```
let oob = ball.outOfBounds();
if (oob) {
    // the ball stays at spawn till go = true
    go = false;
    if (oob == 'right') {
        p1.score++;
        scoreSound.play();
    } else {
        p2.score++;
        scoreSound.play();
    }
} else if (p1.score ==3){
    noStroke(); //gameover if player 1 wins
    textAlign(CENTER);
    textSize(50);
    text("PLAYER 1 WINS",200, 200);
    textSize(20);
    textAlign(CENTER);
    text("Press 'r' to play again.",200, 250);
}else if (p2.score ==3){
    noStroke(); //gameover if player 2 wins
    textAlign(CENTER);
    textSize(50);
    text("PLAYER 2 WINS!",200, 200);
    textSize(20);
    textAlign(CENTER);
    text("Press 'r' to play again.",200, 250);
}
```

If the ball go out of bounds on the opposite players side, the other player gets a point.



Moving the Paddles and Key Functions

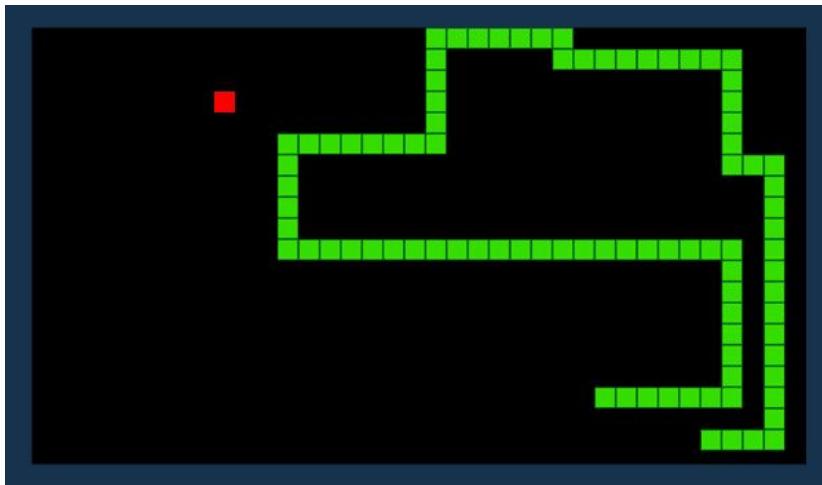
```
function movePaddles() {  
    // 87 = 'w'  
    if (keyIsDown('87')) {  
        p1.move(-5);  
    }  
  
    // 83 = 's'  
    if (keyIsDown('83')) {  
        p1.move(5);  
    }  
  
    // 75 = 'k'  
    if (keyIsDown(UP_ARROW)) {  
        p2.move(-5);  
    }  
  
    // 77 = 'm'  
    if (keyIsDown(DOWN_ARROW)) {  
        p2.move(5);  
    }  
}
```

- Number cases to control and arrow keys

```
function keyTyped() {  
    if (key == ' ') {  
        go = true;  
        restartGame.play();  
    }  
  
    if (key == 'r') {  
        p1.score = 0;  
        p2.score = 0;  
        ball.resetball();  
        go = false;  
        restartGame.play();  
    }  
  
    // for safety  
    return false;  
}
```

- Reset and starting the game

SNAKE GAME



- I enjoyed making the pong game
- Wanted to continue making games as a challenge
- Add my own adaptations

PreLoad and Slider

```
drCaterpillar.js
eat.wav
gameOver.wav
grass.jpg
index.html
sketch.js
snake.ttf
style.css

18 let customFont; //added a font in
19
20 var eatSound;
21 var startOverSound;
22 var music;
23 var musicRate = 1;
24
25 let running = true;
26 let timer = 0;
27
28▼ document.addEventListener('keydown', function(k){ //evaluates every time a
key is pressed
29   dir = k.code // variable stores keys pressed
30 })
31
32
33▼ function preload(){ //load in images and fonts
34   img = loadImage("grass.jpg")
35   customFont = loadFont("snake.ttf");
36   eatSound = loadSound("eat.wav")
37   startOverSound = loadSound("gameOver.wav")
38   music = loadSound["arcadeMusic.mp3"]
39
40 }
41
42▼ function setup() {
43   createCanvas(400, 400);
44   slider = createSlider(0, 0.1, 0.4, 0.01);
45   music.play();
46   music.setVolume(0.1);
47 }
```

- Similarly added sounds and fonts as well as images

- alert

```
function setup() {
  createCanvas(400, 400);
  slider = createSlider(0, 0.1, 0.4, 0.01);
  music.play();
  music.setVolume(0.1);

  resetSketch(); //calling the reset function
  rectMode(CENTER);
  frameRate(fr); //controls speed of snake

  alert('-> Use ARROW KEYS to move snake around'+ 
    '-> Collect the food to score and r to reset the game\n'+
    '->Use the slider to adjust the music\n'+
    '->Press p to pause and continue arrow keys to resume\n'+
    'Good Luck!'); //gives the instructions first

  gridSize = 16; //create a grid for snake
  space = width / gridSize; //number of squares can fit in a row

  snake = new Snake(); //calls the snake class

  food = new Food(); //calls food object
  //collision();
  score = createP("Score: ").position(30, -10).style("text-font: arial; font-size: 30px; opacity: 0.6"); //addin score in html paragraph
  keyTyped();
}


```

Grid and Setup()

- If the snake dies the Game Over screen takes place

```
function draw() {
  image(img, 0, 0) //drawing the images and words
  music.setVolume(slider.value());
  noStroke();
  textAlign(customFont);
  fill(0, 0, 0, 120);
  textSize(30);
  textAlign(CENTER);
  text("SNAKE", 330, 40)

  if(!snake.dead){ //calls methods if the snake is alive
    snake.move();
    snake.edges();
    snake.eat();
    snake.tail();
    snake.show();
    food.show();
  }else{
    noStroke(); //gameover if snake is dead
    textAlign(customFont);
    fill(200);
    textSize(50);
    textAlign(CENTER);
    text("Game Over", 200, 200);
    textSize(20);
    textAlign(CENTER);
    text("Press 'r' to play again.", 200, 250);
  }
}


```

```
noFill();
stroke(0,100,0); //edges/ border of game
strokeWeight(space);

rect(width / 2, height / 2, width, height)

score.html("Score: " + snake.length); //score on the screen
}
```

- Used html paragraph to show score

Class Snake

```
class Snake{          //snake object
    constructor(){
        this.pos = createVector(200, 200);
        this.length = 0;      //create length starting point
        this.posHistory = [this.pos] // first position to array
        this.dead = false;   //keeps track if snake is dead or not
    }

    move(){
        if(dir === "ArrowRight"){           //if statement to control snakes
            movement
                this.pos.x += space;
        } else if(dir === "ArrowLeft") {
            this.pos.x -= space
        }else if(dir === "ArrowUp"){
            this.pos.y -= space
        }else if (dir === "ArrowDown"){
            this.pos.y += space
        }
    }
}
```

- Movement and directions

- If the snake touches the edges the game is over

```
tail(){    //method to keep track of array, every fram position should be
    stored in array
    this.posHistory.push(this.pos.copy());
    if (this.posHistory.length > this.length){
        this.posHistory.splice(0, 1); //controls length of snake
    }
}
show(){
    noStroke();
    fill( 0 );
    for (var i = 0; i < this.posHistory.length; i++){
        rect(this.posHistory[i].x, this.posHistory[i].y, space - 5); //shows
        the new length of snake, - 5 to create space between them
    }
}
```

```
edges() {      //keep the snake on the canvas
    if (this.pos.x === 0 || this.pos.x === width || this.pos.y === 0 ||
    this.pos.y === height){
        this.dead = true;
        startOverSound.play(); //id all conditions are true then snake should
die
    }

    for(var i = 0; i < this.posHistory.length - 1; i++){ //if the snake
hits itself = game over
        if(this.pos.x === this.posHistory[i].x && this.pos.y ===
this.posHistory[i].y){
            this.dead = true;
            startOverSound.play();
        }
    }
}

eat() {        //if snake eats food, the newPos is called
    if(this.pos.x === food.x && this.pos.y === food.y){
        food.newPos();

        this.length += 1;
        eatSound.play();
        fr += 0.5;           //frame rate increases speed the more food you get
        music.rate(musicRate);
        musicRate += 0.05;
        frameRate(fr);
        //snake gets longer as it eats food
    }
}
```

The music rate and speed increases the longer the length of the tails gets

Class Food

```
class Food {          //food object
  constructor(){    //food appears in random coordinates
    this.x = floor(random(1, gridSize)) * space;
    this.y = floor(random(1, gridSize)) * space;
    this.moved = false;
  }
  newPos(){        //moves food to random position
    this.moved = false; //tried to find position that isn't inside the snake
    while(!this.moved){
      var newX = floor(random(1, gridSize)) * space;
      var newY = floor(random(1, gridSize)) * space;
      for (var i = 0; i < snake.posHistory.length; i++){
        if (newX === snake.posHistory[i].x && newY === snake.posHistory[i].y)
{
          break
        }else{
          if(i === snake.posHistory.length -1){
            this.x = newX
            this.y = newY
            this.moved = true;
          }
        }
      }
    }
}
```

- The food class starts in any position and moves to any new position

```
  show(){ //shows the food
    noStroke();
    fill(255, 50, 50);
    rect(this.x, this.y, space / 2);
  }
}
```

Reset

```
function resetSketch(){      //resets the sketch if r has been pressed
  fr = 3;    //resets framerate back to normal
  music.rate(1);
  snake = new Snake();

  food = new Food();

}
function update() {
  if (!running) return
  // everything after this point doesn't run
}

function keyTyped () {
  if(key === 'r') {
    resetSketch();
  }
  if(key === 'p'){
    running = !running // flip the boolean
    noStroke();
    fill(200);
    textSize(50);
    textAlign(CENTER);
    text("PAUSED",200, 200);
    textSize(20);
    textAlign(CENTER);
    text("press arrows to resume",200, 250);
  }
}
```

- The game resets and the speed and music go back to the beginning
- A pause function added