Paddle Game Questions

1. Provide a written response for your video that:
    - identifies the programming language;
    - identifies the purpose of your program; and
    - explains what the video illustrates.

My program is written in p5js and is used for entertainment. The video illustrates how the code runs and how to play the game. It shows the different screen of gameplay; start screen, play screen, and the end screen.

2. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.

.

3. Capture and paste the program code segment that implements an algorithm (marked with an oval) that is fundamental for your program to achieve its intended purpose. Your code segment must include an algorithm that integrates other algorithms and integrates mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Approximately 200 words)*

*Scroll down for the code

My paddle class functions independently because it is only code that revolves around the paddle. The code allows me to make as many paddles as a desire. The code works in combination with other because in order to keep score, there must be code detecting when the balls hit the paddle.

4. Capture and paste the program code segment that contains an abstraction you developed (marked with a rectangle in section 3 below). Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.

*Scroll down for the code

My playGame code managed the complexity of my program because it creates a new screen. It switches from my title screen and adds balls and the paddle. It uses code from my ball and paddle classes. It also changes to a new end screen when health is equal to zero.

5. Capture and paste your entire program code in this section.
    - Mark with an oval the segment of program code that implements the algorithm and integrates mathematical and /or logical concepts.
    - Mark with a rectangle the segment of program code that represents an abstraction you developed.
    - Include comments or citations for program code that has been written by someone else.

*Scroll down for the code

```
  //  Allison Smith
  //  September 20, 2019

  var ball = []
  var paddle = []
  var score = 0
  var health = 3
  var gameState = 1
  var numOfBalls = 0
  function setup() {
    var cnv = createCanvas(800, 800);
    cnv.position((windowWidth-width)/2, 30);
  }

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


//creates balls and paddle
function loadThings(n){
  for(var i=0; i < n; i++){
    ball[i] = new Ball(random(width), 0, random(-8,8), random(-8,8), i);
  }
  paddle = new Paddle(random(width/2),725);
}
//runs the ball and paddle classes
function runObjects(){
  for(var i = 0; i < ball.length; i++){
  ball[i].run();
  }
  paddle.run();
}
function startGame(){
    background(20,20,20);
//title
    fill(250, 250, 250);
    textSize(100);
    text("Paddle Game", 100, 300);
//easy
      fill(0, 250, 0);
      textSize(20);
      text("Press E for Easy", 100, 500);
       if (keyCode === 69) {
         loadThings(3);
         gameState = gameState + 1
   }
```

```
//medium
      fill(255,165,0);
      textSize(20);
      text("Press M for Medium", 325, 500);
      if (keyCode === 77) {
         loadThings(5);
         health = 4
         gameState = gameState + 1
     }
//hard
      fill(255,0 ,0 );
      textSize(20);
      text("Press H for Hard", 600, 500);
       if (keyCode === 72) {
         loadThings(10);
         health = 8
         gameState = gameState + 1
    }
//how to play
    fill(255, 100 ,0);
    textSize(20);
    text("The goal of the game is to keep as many balls in the air as you can.", 100, 650);
    text( "Once you lose all your balls, the game ends", 100, 700);
}

    function playGame(){
          background(20,20,20);
          runObjects();
          if (health === 0) {
             gameState = gameState + 1
          }
        }

    function endGame(){
      background(20,20,20);
      fill(0, 250, 0);
      textSize(32);
      text("Score: " + score, 650, 30);
      fill(250, 250, 250);
      textSize(100);
      text("You Lost :(", 150, 300);
      fill(250, 250, 250);
      textSize(50);
      text("To restart, press R", 200, 600);
      if (keyCode === 82) {
        gameState = 1
      }
    }
```

```
class Ball{
  constructor(x,y,dx,dy,id){
    this.loc = createVector(x,y);
    this.vel = createVector(dx,dy);
    this.acc = createVector(0, .2);
    this.clr = color(random(255), random(255), random(255))
  }

  run(){
    this.checkEdges();
    this.update();
    this.render();
  }

  checkEdges(){
//left edge
    if(this.loc.x<0){
    this.vel.x = -this.vel.x
    }
//right edge
    if(this.loc.x>width){
    this.vel.x = -this.vel.x
    }
//top edge
    if(this.loc.y<0){
    this.vel.y = -this.vel.y
    }

//score
    for(var i = ball.length - 1; i >= 0; i--){
      if(ball[i].isColliding()){
        ball.splice(i, 1)
        health= health - 1
      }
    }
//paddle
    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 = -this.vel.y
      score= score + 1
    }
  }

  update(){
    this.vel.add(this.acc);
    this.loc.add(this.vel);
//Health text
    fill(250, 0, 0);
    textSize(32);
    text("Health: " + health, 10, 30);
//Score text
    fill(0, 250, 0);
    textSize(32);
    text("Score: " + score, 650, 30);
  }

render(){
    fill(this.clr);
    ellipse(this.loc.x, this.loc.y, 50, 50);
  }
  isColliding(){
    if(this.loc.y > 850){
      return true
    }
  }
}
```

```
 class Paddle{
   constructor(x,y){
     this.loc = createVector(x,y);
     this.w = 250;
     this.h = 50;
   }

 run(){
   this.update();
   this.render();
 }
 update(){
     var mouseLoc = createVector(mouseX, 725);
     this.loc = p5.Vector.lerp(this.loc, mouseLoc, .09);
 }

 render(){
   fill(255, 0,0);
   rect(this.loc.x, this.loc.y, this.w, this.h);
   }
 }
```