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.

The programming language I used was p5js. The purpose of the program is for entertainment. My video illustrates playing easy mode. My game is not finished, due to the isColliding() function that works half of the time, so that is why some balls go through the paddle. When in medium and hard, the balls would increase.

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)* Each of the algorithm's work by checking if the standards are met, like if the ball hits the wall or if the ball is touching the paddle in a certain way. When in combinations with other algorithms, the job is more complete. For example, if one algorithm makes sure that the ball doesn't disappear from the right side of the screen, then the algorithm for the top, bottom, and the left screen will make it so that the ball never leaves the screen, so it is more "complete". This algorithms make it so the balls disappear when needed, for the buttons to work, and for the paddle to appear.

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. My abstraction helped me manage the complexity of the code by getting rid of excessive code. For example, the code would be longer if I didn't have the ball class.

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.

```
//  Danny Ramirez
//       8/21/19
//  This is a comment
//sketch.js
var balls = []
var paddle;
var gameState = 1
var btnEasy, btnHard, btnMedium;
var gameMode;
var score, health;
var lose, win;
//  The setup function function is
called once when your program
begins
function setup() {
  var cnv = createCanvas(800,
800);

cnv.position((windowWidth-width)/2,
30);
  fill(200, 30, 150);
  loadButtons(3);
  loadBalls(3);
  score = 0;
  health = 5;//variables
}// end of setup

//  The draw function is called @ 30
fps
function draw() {
  background(20, 20, 20, 100);
  if(gameState === 1){
```

```
//Danny Ramirez
//9/25/19
// paddle.js
class Paddle{
  constructor(x, y, w, h){
    this.loc = createVector(x, y);
    this.clr = color(0, 0, 250)
    this.w = w;
    this.h = h;//variables
  }

  run(){
   this.render();//render is called
  }//end of run

  render(){
   fill(this.clr);
   rect(mouseX - 70, this.loc.y,
this.w, this.h);//show paddle and
move
   this.clr = color(random(250),
random(250), random(250))//gives
rainbow effect
  }//end of render

  }//end of paddle class
```

```
//Danny Ramirez
//9/25/19
//button.js
class Button{
 constructor(x, y, w, h, msg, clr){
  this.x = x;
```

```
//Danny Ramirez
//9/25/19
//ball.js
class Ball{
  constructor(x, y, dx, dy, w, h){
    this.loc = createVector(x,y);
    this.vel = createVector(dx,dy);
    this.acc = createVector(0,0.01);
    this.clr = color(random(255),
random(255), random(255))
    this.s = 20;// variables
  }

  run(){
    this.checkEdges();
    this.update();//these are called in
    this.render();
  }//end of run

  checkEdges(){
  if(this.loc.x < 0){
    this.vel.x = -this.vel.x;
  }
  if(this.loc.x > width){
    this.vel.x = -this.vel.x;
  }
  if(this.loc.y < 0){
    this.vel.y = -this.vel.y;
  }
  if(this.loc.y > height){
    this.vel.y = -this.vel.y;
  }//checks if the balls hit edge
```