Connor Spackman
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.

   I am programming in javascript but I am using the p5js library. The purpose of my program is to supply hours of fun to kids and computer science teachers grading my project. The video illustrates the player choosing a difficulty playing the game trying to keep the balls from falling off the screen. The video shows the player losing and restarting the game at the same difficulty.

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)*

```
function booleanState(){ // a method to check if the game is over
 var check=0;
  for(var i=0; i<balls.length; i++){
   if(balls[i] == null) check++;
  }
  if(check===balls.length){
   return false;
  }
 }
```

This algorithm uses for loops and if statements to traverse an array to determine if there are still balls left in the array. If there are no balls left in the array this algorithm is used to change game states and end the game.

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.
    a. Abstraction helped me manage the complexity of the program by simplifying the code and allowing to not have to code repeatedly. Abstraction saved me lots of time and made my program cleaner and more efficencient.

```
class paddle{
 constructor(l,x,y){// constructor fot the method
  this.loc = createVector(x,y);
  this.l = l;
  this.clr= color(51,109,255);
 }
  run(){
   this.move();
   this.render();

  }
  render(){ // a method to render the paddle
   fill(this.clr);
   rect(this.loc.x,this.loc.y,this.l,30);
  }
  move(){
   if(keyIsDown(65)||keyIsDown(LEFT_ARROW)){// a method to move the paddle left
 if(this.loc.x>0&&this.loc.x<width){
    this.loc.x=this.loc.x-20;
   }
    }
   if(keyIsDown(68)||keyIsDown(RIGHT_ARROW)){ // a method to move the paddle right
   if(this.loc.x+this.l>0&&this.loc.x+this.l<width){
   this.loc.x=this.loc.x+20;
   }
  }
 }
}
```

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.

```
var balls =[]; // An Array for Balls
var p1;
var score=0;
var firstRun=0;
var textX;
var textY;
var gameState=3; // gameState the driving force and biggest pain in the neck of this game
var mouseState;
var b1,b2;
var track;
function setup() {
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
 textY = height/2;// Sets position of text
 textX = width/2-200;
 frameRate(60); // Forces the game to run at 60 frame cause this game is not ment for chrombooks
}

//  The draw function is called @ 30 fps
function draw() {
  if(gameState===0){
  gameStates(0); /// Logic to decide on gamemode
  }
  if(gameState===1){
  gameStates(1);
  }
  if(gameState===2){
    gameStates(2);
  }
  if(gameState===3){
    loadStartScreen();
  }

}
```

```
function loadObjects(num,lengthOfPaddle){ // a method to load Objects
for(var i =0; i<num; i++){
  balls[i] =  new ball(i,1,.035,random(100,width/1.5),0+100*i);
}
p1= new paddle(lengthOfPaddle,height/2,width/2+250);
}

function intGame(gameState){
  if(gameState===0){
    loadObjects(4,300);
  }
  if(gameState===1){
    loadObjects(6,250);
  }
  if(gameState===2){
    loadObjects(6,100)
  }
}

function runObjects(){ // A method to run objects
for(var i =0; i<balls.length; i++){
    balls[i].run();
  }
  p1.run();
}

function booleanState(){ // a method to check if the game is over
var check=0;
  for(var i=0; i<balls.length; i++){
    if(balls[i] == null) check++;
  }
  if(check===balls.length){
    return false;
  }
}

function gameStates(mode){ // A method that runs the given gamemode
    background(255,255,255,60);
    if(firstRun==0){
      intGame(gameState);
      track=gameState;
    }
    firstRun++;
    runGame();
}
  function runGame(){ // a method to run the game
    runObjects();
    textX =100000;
    textY = 100000;
    textSize(32);
    text(("Score: "+score),10,30);
    deleteBalls();
    if(booleanState()===false){
      gameState=0;
      endGame();
    }
}

function endGame(){ // a method to load the end game screen
  if(booleanState()===false){
    background(255,255,255);
    textSize(40);
    text(("Score: "+score),300,450);
    textSize(64);
    text("Game Over",300,400);
    b1 = new Button((width/2-150)+0*150,height/2+200,141,219,255,3);
    b1.ForceLoad();
    b1.Restart();
    mouseState=false;
    }
}
function loadStartScreen(numOfButtons){ // a method to load buttons
  var buttons =[];
  for(var i=0; i<3; i++){
    buttons[i] = new Button((width/2-150)+i*150,height/2+200,141,219,255,i);
    buttons[i].loadButton();
    buttons[i].CheckMouse();
    }
  fill(0,0,0);
  textSize(64);
  text("Welcome to the \nBall Game",textX,textY);
  textSize(32)
  text("Use the A and D keys \nor the Arrow Keys to move",textX,textY+130)
}
function mouseClicked(){
  mouseState = true;
}
```

```
function deleteBalls(){ // a method to delete balls from the array when the fall off the screen
 for(var i=balls.length-1; i>=0; i--){
  if(balls[i].loc.y>height){
   balls.splice(i, 1);
  }
 }
}

class paddle{
 constructor(l,x,y){// constructor fot the method
  this.loc = createVector(x,y);
  this.l = l;
  this.clr= color(51,109,255);
 }
  run(){
   this.move();
   this.render();

  }
  render(){ // a method to render the paddle
   fill(this.clr);
   rect(this.loc.x,this.loc.y,this.l,30);
  }
  move(){
   if(keyIsDown(65)||keyIsDown(LEFT_ARROW)){// a method to move the paddle left
 if(this.loc.x>0&&this.loc.x<width){
    this.loc.x=this.loc.x-20;
   }
    }
   if(keyIsDown(68)||keyIsDown(RIGHT_ARROW)){ // a method to move the paddle right
   if(this.loc.x+this.l>0&&this.loc.x+this.l<width){
   this.loc.x=this.loc.x+20;
   }
  }
 }
}

class ball{ // I cant be bothered to comment on this class good luck finding out what it does
  constructor(id,dy,grav,x,y){
   this.dy=dy;
   this.grav=grav;
   this.loc = createVector(x,y);
   this.velocity = createVector(random(1,5),this.dy);
   this.id=id;
   this.clr= color(0,0,0);
   this.acc = createVector(0,this.grav);
  }
  run(){
   this.check();
   this.checkHitbox();
   this.update();
   this.render();
  }
  check(){
   if(this.loc.x<=0){
    this.velocity.x = -this.velocity.x;
   }
   if(this.loc.x>=width){
    this.velocity.x = -this.velocity.x;
   }
   if(this.loc.y<=0){
    this.velocity.y= -this.velocity.y;
   }
   if(this.loc.y>=height){
    this.velocity.y= -this.velocity.y;
   }
  }
  update(){
   this.velocity.add(this.acc);
   this.loc.add(this.velocity)
  }
  render(){
   fill(this.clr);
   ellipse(this.loc.x, this.loc.y, 30,30);
  }
  checkHitbox(){
var vec;
   if(this.loc.x > p1.loc.x&& this.loc.x < p1.loc.x+p1.l
   && this.loc.y> p1.loc.y&& this.loc.y<p1.loc.y+30){
    this.velocity.y= -this.velocity.y;
    score++;
    this.velocity.add(vec=createVector(.2,-.5));
   }
   }

}

class Button{
 constructor(x,y,r,g,b,id){
 var strings =["Easy","Hard","Really\n Hard","Restart"];//Names of the Buttons
```

```
  this.x =x;
  this.y =y;
  this.w = 70;
  this.z= 70;
  this.r =r;
  this.g =g;
  this.b=b;
  this.id=id;
  this.str = strings[this.id];
  }
  loadButton(){ // A method to load buttons give the game is not playing
    if(gameState=== 3){
    fill(this.r,this.g,this.b);
    rect(this.x,this.y,this.w,this.z);
    fill(0,0,0);
    textSize(20);
    text(this.str,this.x,this.y,this.w,this.z)
    }
  }
  CheckMouse(){ //Method to check if the button has been clicked
    if(mouseX > this.x && mouseX < this.x+this.w
    && mouseY> this.y && mouseY <this.y+this.z &&
    mouseState=== true){
      gameState= this.id; // Sets the gamemode
    }
  }
  ForceLoad(){ // A method to load buttons no matter the gameState
    fill(this.r,this.g,this.b);
    rect(this.x,this.y,this.w,this.z);
    fill(0,0,0);
    textSize(20);
    text(this.str,this.x,this.y,this.w,this.z)
  }
  Restart(){ // A method to Restart the game
    if(mouseX > this.x && mouseX < this.x+this.w
    && mouseY> this.y && mouseY <this.y+this.z &&
    mouseState=== true){
      gameState=track;
      firstRun=0;
      score=0;
    }
  }
}
```