Natalie Hallmann
Mr. Ettlin
AP Com science, Period 1
3 October 2019

1) This program, using Java, is meant to be an enjoyable game using a movable paddle to hit and bounce multiple balls around the play screen. The user is able to choose from three different difficulties including easy, medium and hard. This video illustrates the selection of the easy button on the front splash screen however, my code still is not finished and therefore the playscreen is not available so it shows the transition between the splash screen and the play screen.
2)
3) Paddle game still is not finished so I cannot identify a fundamental algorithm that incorporates other algorithms as well.
4) Paddle game still is not finished so I cannot identify an abstraction.
5) Currently, I am trying to make the transition from game state 1 to game 2 better because my paddle and balls are not moving are frozen in place. Once I figure this out I will use it to help with the rest of the difficulties and move on to the end screen.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Template</title>
    <script src="libraries/p5.js" type="text/javascript"></script>
    <script src="libraries/p5.dom.js" type="text/javascript"></script>
    <script src="libraries/p5.sound.js" type="text/javascript"></script>
    <script src="sketch.js" type="text/javascript"></script>
    <script src="ball.js" type="text/javascript"></script>
    <script src="paddle.js" type="text/javascript"></script>
    <script src="but.js" type="text/javascript"></script>
    <style> body {padding: 0; margin: 0;} canvas {vertical-align: top;} </style>
  </head>


  <body>
  </body>
</html>
```

```
class Button{
  constructor(x,y,w,h,clr){
    this.loc=createVector(x,y);
    this.w=w;
    this.h=h;
    this.clr=clr
  }

  run(){
    this.render();
    this.isTouched();
  }
  render(){
    fill(this.clr);
    rect(this.loc.x,this.loc.y,this.h,this.w);

  }

  isTouched(){
    if(mouseIsPressed &&
      mouseX > this.loc.x &&
      mousex < this.loc.x+this.w &&
      mouseY > this.loc.y &&
      mouseY < this.loc.y +this.h){
        return true;
      }else{
        return false;
      }

    }

ifTouched();

if(difficulty===="easy")||if(difficulty===="medium")||if(difficulty===="hard")||if(difficulty=
==="instructions");
 if(difficulty===="easy"){
   loadObjects(5);
   gameState===2
```

```
 }

 if(difficulty===="medium"){
  loadObjects(7);
  gameState===2
 }
 if(difficulty===="hard"){
  loadObjects(11);
  gameState===2
 }
 if(difficult===="instructions"){
  function Instructions();
  gameState===3
 }


function ifTouched(){
 (if gameState===1)
 //Easy Button touched
 if(mouseIsPressed&&
 mouseX>50&&
 mouseX<125&&
 mouseY>150&&
 mouseY<470){
   difficulty="easy"
 }
 //Medium Button touched
 if(mouseIsPressed&&
 mouseX>50&&
 mouseX<325&&
 mouseY>150&&
 mouseY<470){
   difficulty="medium"
}
//Hard Button touched
if(mouseIsPressed&&
mouseX>50&&
mouseX<525&&
mouseY>150&&
```

```
mouseY<470){
  difficulty="hard"
}
//Instructions Button touched
if(mouseIsPressed&&
mouseX>50&&
mouseX<325&&
mouseY>150&&
mouseY<615){
  difficulty="instructions"
}

}

//  Natalie Hallmann
//          PaddleBall Game Project 9/16
//  This is a comment
//  The setup function function is called once when your program begins
var balls=[];
var paddle;
var ifTouched;
var score=0
var gameState =1;
var win;
var btnEasy, btnMedium, btnHard, btnBTME, btnBTMI, btnInstructions;
function setup() {
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
  background(5, 5, 5);
  function newButton();
  loadObjects();
  newButton();

}

//  The draw function is called @ 30 fps
function draw() {
  background(5, 5, 5);
  //runObjects();
```

```
if(gameState===1){
  startGame();//start splash screen
}else if(gameState===2){
  playGame();//paddle and balls screen
}else if(gameState===3){
  Instructions();
  if(btnBTMI.isTouched() === true){
    gameState = 1;
    endGame();//end splash screen
  }

} //end of draw function

function startGame(){

  //PaddleBall Text
  fill(150,150,150);
  textAlign(CENTER);
  textSize(80);
  text("Paddle Ball",400,300);

  //Score Text
  textSize(40)
  text("SCORE:0",100,790);



  //Easy Button
  btnEasy.render();
  textSize(20);
  fill(5,5,5);
  text("Easy",200,500);

  //medium button
  btnMedium.render();
  fill(5,5,5);
  text("Medium", 400,500);

  //Hard Button
```

```
  btnHard.render();
  fill(5,5,5);
  text("Hard", 600,500);

  //Instructions Button
  btnInstructions.render();
  fill(5,5,5);
  text("Instructions", 400,650);
  Instructions();


  //instructions screen
}
function Instructions(){
  textSize(25);
  textAlign(CENTER);
  text("Move the mouse in order to control the paddle and hit the balls.",400,200)
  text("you must collect all green balls in order to win and avoid the red balls.",400,300)
  text("if a red ball bounces on your paddle, you lose",400,400)




}
//making Buttons
function newButton(){
  btnEasy = new Button(125,470,50,150,color(200,0,0));
  btnMedium = new Button(325,470,50,150,color(0,200,0));
  btnHard = new Button(525,470,50,150,color(0,0,200));
  btnInstructions = new Button (325,615,50,150,color(150,150,150));
  btnBTMI = new Button (325,610,150,50,color(150,150,150));
  btnReplay = new Button (70,100,150,50,color(150,150,150));
  btnBTME = new button ()
}
//gameState2
function playGame(){
  runObjects();
  fill (550);
  textSize(40)
```

```
  text("SCORE:0",100,790);
  runObjects();
  if (checkRed() === true|| balls.length === 0){
    gameState= 4;
    win = 'yes';
  } else if( score < 0 ){
    gameState = 4;
    win = 'no';
  }
  function endgame(){
    if (win === "yes"){//score>0
      textSize(95);
      fill(155,155,155);
      text ("You Won!",325,400);
    }else if (win === 'no'){//score<0
      textSize(95);
      fill(155,155,155);
      text ("You Lost!",325,400);
    }
  }
  fill(200, 200, 200) //Main Menu
  btnBTME.run();
  fill(100, 150, 100);
  textSize(30);
  text("To Main Menu", 600, 500)
  if (btnBTME.isTouched()){
    gameState =1;
  }
}//end endgame
function loadObjects(n){
  paddle= new Paddle(50,400,95,95);
  for(var i=0; i< n; i++){
    balls[i] =new Ball(random(width),0, random(-5,5),random(-5,5));
  }
}
function runObjects(){
  paddle.run();
  for(var i=0; i<balls.length; i++){
    balls[i].run();
```

```
    }
    fill(200, 200, 200) //Main Menu
    btnBTME.run();
    fill(100, 150, 100);
    textSize(30);
    text("To Main Menu", 600, 500)
    if (btnBTME.isTouched()){
      gameState =1;

    }


class Button{
  constructor(x,y,w,h,clr){
    this.loc=createVector(x,y);
    this.w=w;
    this.h=h;
    this.clr=clr
  }

  run(){
    this.render();
    this.isTouched();
  }
  render(){
    fill(this.clr);
    rect(this.loc.x,this.loc.y,this.h,this.w);

  }

  isTouched(){
    if(mouseIsPressed &&
      mouseX > this.loc.x &&
      mousex < this.loc.x+this.w &&
      mouseY > this.loc.y &&
      mouseY < this.loc.y +this.h){
        return true;
      }else{
        return false;
```

```
    }

   }

ifTouched();

if(difficulty===="easy")||if(difficulty===="medium")||if(difficulty===="hard")||if(difficulty=
==="instructions");
 if(difficulty===="easy"){
   loadObjects(5);
   gameState===2
 }

 if(difficulty===="medium"){
   loadObjects(7);
   gameState===2
 }
 if(difficulty===="hard"){
   loadObjects(11);
   gameState===2
 }
 if(difficult===="instructions"){
   function Instructions();
   gameState===3
 }


function ifTouched(){
  (if gameState===1)
  //Easy Button touched
  if(mouseIsPressed&&
  mouseX>50&&
  mouseX<125&&
  mouseY>150&&
  mouseY<470){
    difficulty="easy"
  }
  //Medium Button touched
  if(mouseIsPressed&&
```

```
  mouseX>50&&
  mouseX<325&&
  mouseY>150&&
  mouseY<470){
    difficulty="medium"
}
//Hard Button touched
if(mouseIsPressed&&
mouseX>50&&
mouseX<525&&
mouseY>150&&
mouseY<470){
  difficulty="hard"
}
//Instructions Button touched
if(mouseIsPressed&&
mouseX>50&&
mouseX<325&&
mouseY>150&&
mouseY<615){
  difficulty="instructions"
}

}
class Paddle {
  constructor(x, y, w, h){
    this.loc = createVector(x, y);
    this.w=w;
    this.h= h;
    this.clr = color(random(255), random(255), random(255));
    this.w=200
  }//constructor

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

  update(){
```

```
var mouseLoc= createVector (mouseX,600);
this.loc = p5.Vector.lerp(this.loc,mouseLoc, 0.09);

}//update

 render(){

  fill(this.clr);
  rect(this.loc.x, this.loc.y,200, 40);
 }//render



}//class
class Ball {
 constructor(x, y, dx, dy, id){
  this.loc = createVector(x, y);
  this.vel = createVector(dx, dy);
  this.acc = createVector(0,0.5,);
  this.clr = color(random(255), random(255), random(255));
  // this.w=25
 }

 run(){
  this.checkedges();
  this.update();
  this.render();
  this.Bounce();
 }

 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
  }
 }
 update(){
  this.vel.add(this.acc);
  this.loc.add(this.vel);
 }

 render(){

  fill(this.clr);
  ellipse(this.loc.x, this.loc.y,20,20);
 }


 IsColliding(){
  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){
    return true;
   } else{
    return false;
   }
  }


  Bounce(){
   if (this.IsColliding()=== true){
    this.vel.x = -this.vel.x;
    this.vel.y= - this.vel.y;
   }
  }
}
```