Assignment2

```
import ddf.minim.*;

Minim minim;
AudioPlayer greenSound;
AudioPlayer redSound;
AudioPlayer background;
AudioPlayer sadMusic;
AudioPlayer menuMusic;
int receiverX, receiverY, receiverW, receiverH, receiverS;
boolean up, down;
PImage credits, shooter, redBall, wallpaper, receiver, loserWallpaper, menuWallpaper, menubutton,selectmenu;
int shooterY;
int shooterDirection;
int timer;
ArrayList<Ball> balls;
ArrayList<Ball> hitBalls;
int receiverLocation;
int score = 0;
boolean isPlaying;
int attempts = 2;
int loseCounter = 1;
int stage = 1;

void setup() {

 isPlaying = true;
 size(1920, 1080);
 minim = new Minim(this);
 background  = minim.loadFile("background.mp3");
 greenSound = minim.loadFile("green.wav");
 redSound = minim.loadFile("red.wav");
 sadMusic = minim.loadFile("sadmusic.wav");
 menuMusic = minim.loadFile("background.mp3");

 rectMode(CENTER);
 receiverX = 1880;
 receiverY = height/2;
 receiverW = 50;
 receiverH = 170;
 receiverS = 10;
 smooth();
 hitBalls = new ArrayList<Ball>();
 balls = new ArrayList<Ball>();
 shooter = loadImage("shooter.png");
 credits = loadImage("credits.png");
 wallpaper = loadImage("wallpaper.jpg");
 receiver = loadImage("receiver.png");
 menubutton = loadImage("Home.png");
 menuWallpaper = loadImage("menu.png");
 selectmenu= loadImage("SelectALevel1.png");
 shooterY = 250;
 shooterDirection = 10;
 /*if (attempts>=0) {
   background.rewind();
   background.play();
 }*/
 /*if (attempts<0) {
   background.pause();
   sadMusic.rewind();
```

```
  sadMusic.play();
 }*/
 loserWallpaper = loadImage("loser.jpg");
}

void draw() {
if (stage == 1){
 menu();
}
if (stage ==2){
 background.play();
play();
}
if (stage == 3){
selectMenu();
}
if (stage==4){
 creditMenu();
}
}




void keyPressed() {
 if (key == &apos;w&apos; || key ==&apos;W&apos;) {

   up = true;
 }
 if (key == &apos;s&apos; || key ==&apos;S&apos;) {


   down = true;
 }
}



void keyReleased() {
 if (key == &apos;w&apos; || key ==&apos;W&apos;) {
   up = false;
 }
 if (key == &apos;s&apos; || key ==&apos;S&apos;) {
   down = false;
 }
}







int moveShooter() {


 if (shooterY > 900) {
   shooterDirection = -10;
   return shooterDirection;
```

```
}
 if (shooterY < 50) {
   shooterDirection = 10;
   return shooterDirection;
 } else
 return shooterDirection;
}

void decideToShoot() {
 if (timeToShoot()) {

   balls.add(new Ball(greenOrRed()));
 }
}


boolean timeToShoot() {
 if ((timer %100) == 0) {
   return true;
 }
 return false;
}
int greenOrRed() {
 float colour = random(0, 200);

 return (int)colour;
}
```

Ball

```
class Ball {
 //global variables
 float x = 300 ;
 float y = shooterY + 50 ;
 float speed = 10;
 float colourGreen;
 float colourRed;
 float colourBlue = 0;
 float location = x*y;
 boolean notHit;
 //constructor
 Ball(int colour) {
   notHit = true;
   if (colour>51 && colour < 200) {
     this.colourGreen = 255;
     this.colourRed = 0;
     //fill(0,255,0);
   }
   if (colour<50 && colour >0)
     this.colourGreen = 0;
   this.colourRed = 255;
   //fill(255,0,0);


   ellipse(this.x, this.y, 20, 20);
}

 //Functions
 void run() {
   display();
   moveBall();
```

```
  ballSound();
  //drawScore(); Feature available in the final product.
}
void moveBall() {
  if (this.x < 800) {
    this.x += speed+20;
  } else
    this.x += speed;
}
void display() {
  if (isHit()&& this.x > 1700) {
    return;
  } else
    fill(this.colourRed, this.colourGreen, colourBlue);
  ellipse(this.x, this.y, 45, 45);
}
void ballSound() {
  if (this.colourGreen == 255 && (this.x< receiverX +10 && this.x > receiverX -10)&& ((this.y > receiverY - 100)&&(this.y < receiverY +
165))) {
    greenSound.rewind();
    greenSound.play();
  }
  if (this.colourGreen == 0 && (this.x< receiverX +10 && this.x > receiverX -10)&& ((this.y > receiverY - 100)&&(this.y < receiverY +
165))) {
    redSound.rewind();
    redSound.play();
    --attempts;
  }
}
void changeColour() {
  this.colourRed = 255;
  this.colourGreen = 128;
  this.colourBlue = 0;
}
boolean isHit() {

  if ((this.x > receiverX) && ((this.y > receiverY - 100)&&(this.y < receiverY + 165))) {


    return true;
  } else

  return false;
}
// boolean isHitOnce(){
// if (((this.x > receiverX) && ((this.y > receiverY - 100)&&(this.y < receiverY + 165))) && (notBeenHit()== true)){
// this.notHit = false;
//
// return true;


// }

// return false;
// }
boolean notBeenHit() {
  return this.notHit;
}

boolean isGreen() {
```

```
  if (this.colourGreen == 255)
    return true;
  return false;
}
 boolean isRed() {
  if (this.colourRed == 255)
    return true;
  return false;
}

 void drawScore() {

  if ((this.x > receiverX) && ((this.y > receiverY - 100)&&(this.y < receiverY + 165))) {
    ++score;
  }
  fill(250);
  text("Balls Shot:", 300, 100);
  text(score, 650, 100);
}
}
```

Loser_Screen

```
void lose() {
 background.pause();
 background(loserWallpaper);
 if (loseCounter == 1)
   sadMusic.rewind();
 sadMusic.play();
 loseCounter++;
}
```

Menu

```
void menu() {
 /*background.rewind();
 background.play();*/
 background(menuWallpaper);

}
void mousePressed() {

 if (stage == 1){
  if ((mouseY < 293 && mouseY > 225)&&(mouseX>630 && mouseX < 1375) ){
    /*background.rewind();
    background.play();*/
    stage = 2;

  }
  if ((mouseY < 640 && mouseY >575 )&&( mouseX >630 && mouseX <1375 )){
    exit();
  }
  if ((mouseY<407 && mouseY > 342)&& (mouseX>630 &&mouseX<1375)){
   stage = 3;
  }
  if ((mouseY<519 && mouseY > 457)&& (mouseX>630 &&mouseX<1375)){
   stage = 4;
  }
}
 if (stage == 2){
   if ((mouseY < 70 && mouseY > 20 )&&(mouseX > 20 && mouseX < 70) ){
     background.pause();
```

```
    stage =1;
  }
}
 if (stage == 3){
  if ((mouseY<122 && mouseY >60)&&(mouseX > 26&& mouseX <86)){
    stage = 1;
  }
   if ((mouseY<588 && mouseY >295)&&(mouseX > 130&& mouseX <407)){
    background  = minim.loadFile("background.mp3");
    wallpaper=loadImage("wallpaper.jpg");

    //stage = 2;
  }
  if ((mouseY<588 && mouseY >295)&&(mouseX > 530&& mouseX <808)){
    background = minim.loadFile("desert.wav");
    wallpaper= loadImage("desert.jpg");


    //stage = 2;
  }
  if ((mouseY<588 && mouseY >295)&&(mouseX > 926&& mouseX <1204)){
    background = minim.loadFile("night.mp3");
    wallpaper= loadImage("night.jpg");
    //stage = 2;
  }
  if ((mouseY<588 && mouseY >295)&&(mouseX > 1343&& mouseX <1624)){
    wallpaper= loadImage("snow.jpg");
    background = minim.loadFile("snow.mp3");
    //stage = 2;
  }
}

 if (stage ==4){
  if ((mouseY<122 && mouseY >60)&&(mouseX > 26&& mouseX <86)){
    stage = 1;
  }

}

}
```

Receiver

```
void drawReceiver() {

 image (receiver, receiverX, receiverY, receiverW, receiverH);
}
void drawMenu(){
 image(menubutton, 20,20, 50,50);
}
void moveReceiver() {
 if (up) {
   if (receiverY > 10)
     receiverY -= receiverS;
 }
 if (down) {
   if (receiverY < 900)
     receiverY += receiverS;
 }
}
```

SelectLevel

```
void selectMenu() {
  /*background.rewind();
 background.play();*/
 background(selectmenu);

}
```

credits

```
void creditMenu(){
/* background.rewind();
background.play();*/
 background(credits);
}
```

play

```
void play(){


  if (attempts>=0) {

   receiverLocation = receiverX * receiverY;

   background(wallpaper);
   drawReceiver();
   moveReceiver();
   drawMenu();
   image(shooter, 50, shooterY += moveShooter(), 250, 250);

   timer += 1;
   decideToShoot();
   for (Ball ball : balls) {
     ball.run();
   }
 } else
   lose();
}
```