```
import processing.serial.*;
import processing.sound.*;
SoundFile file;
Serial myPort;
String data,alldata1;
String a_shoot=" ",d_mov=" ";
int a_rot=0,d_rot=0;

class Button
{
  String label; // button label
  float x;      // top left corner x position
  float y;      // top left corner y position
  float w;      // width of button
  float h;      // height of button
  Button(String labelB, float xpos, float ypos, float widthB, float heightB)
  {
    label = labelB;
    x = xpos;
    y = ypos;
    w = widthB;
    h = heightB;
  }
  void Draw()
  {
    fill(255,165,0);
    stroke(0);
    rect(x, y, w, h, 5);
    textAlign(CENTER, CENTER);
    fill(0);
    textSize(25);
    text(label, x + (w / 2), y + (h / 2));
  }
  boolean MouseIsOver()
  {
    if (mouseX > x && mouseX < (x + w) && mouseY > y && mouseY < (y + h))
    {
      return true;
    }
    return false;
  }
}
```

```
class defender
{
  //defender vars
  int s=5;
  float ang = -PI; //initial rotation
  float x = width/2; //initial position X
  float y = 200; //initial position Y
  float rotateVel = 0.05; // rotation speed
  float radius = 2; //movement speed
  int defenderz = 14*s;

  //defender moving vars
  boolean d_movingForward = false;
  boolean d_movingBackwards = false;
  boolean d_turningRight = false;
  boolean d_turningLeft = false;

  void Draw()
  {
    //draw defender
    fill(255);
    pushMatrix();
    translate(x, y);
    rotate(ang);
    fill(141,85,36);
    ellipse(9*s,0,4*s,4*s);
    ellipse(-9*s,0,4*s,4*s);
    fill(200,50,59);
    ellipse(0,0,16*s,8*s);
    fill(141,85,36);
    ellipse(3*s,0,2*s,2*s);
    ellipse(-3*s,0,2*s,2*s);
    triangle(-s,-2*s,s,-2*s,0,-4.5*s);
    fill(0,0,0);
    ellipse(0,0,6*s,6*s);
    popMatrix();
    ac-=1;
    if((ab==true)&&(ac>=7*fr))
    {
      strokeWeight(3);
      stroke(255);
      noFill();
      ellipse(x,y,24*s,24*s);
```

```
  for(int c=0;c<3;c++)
  {
    float tdist=((y-yb[c])*(y-yb[c]))+((x-xb[c])*(x-xb[c]));
    if(tdist<=(144*s*s))
    {
      xb[c] = -100;
      yb[c] = -100;
      angb[c] = -100;
      radiusb = 5;
    }
  }
}
if(ac<7*fr)
{
  ab=false;
}
strokeWeight(1);
stroke(0);
fill(0);
//check movement of defender
float moveX = radius * sin(ang);
float moveY = -radius * cos(ang);
if (d_movingForward)
{
  x += moveX;
  y += moveY;
}
if (d_movingBackwards)
{
  x -= moveX;
  y -= moveY;
}
if (d_turningRight)
{
  ang += rotateVel;
}
if (d_turningLeft)
{
  ang -= rotateVel;
}
//keep defender inside (wrap around)
if (x<-defenderz)x=width+defenderz;
if (x>width+defenderz)x=-defenderz;
```

```
  if (y<-defenderz)y=height+defenderz;
  if (y>height+defenderz)y=-defenderz;
 }

 void dead()
 {
  if(ab==false)
  {
   for(int c=0;c<3;c++)
   {
    float tdist=((y-yb[c])*(y-yb[c]))+((x-xb[c])*(x-xb[c]));
    if(tdist<=(100*s*s))
    {
     i=-1;
     for(int c1=0;c1<3;c1++)
     {
      xb[c1] = -100;
      yb[c1] = -100;
      angb[c1] = -100;
      radiusb = 5;
     }
     littscene(4);
    }
   }
  }
 }
}

class attacker
{
 //attacker vars
 int s=5;
 float ang = 0; //initial rotation
 float x = width/2; //position X
 float y = height-200; //position Y
 float rotateVel = 0.05; // rotation speed
 int attackerz = 14*s;

 //defender moving vars
 boolean a_turningRight = false;
 boolean a_turningLeft = false;

 void Draw()
```

```
{
  for(int c=0;c<3;c++)
  {
    if (angb[c] != -100)
    {
      //convert polar coordinates to cartesian in order to move (towards pointing angle)
      float moveX = radiusb * sin(angb[c]);
      float moveY = -radiusb * cos(angb[c]);
      xb[c] += moveX;
      yb[c] += moveY;
    }
    if (((xb[c]>width)||(xb[c]<0))||((yb[c]>height)||(yb[c]<0)))
    {
      //bullet variables (off-screen)
      xb[c] = -100;//initial x
      yb[c] = -100;//initial y
      angb[c] = -100;//initial angle
      radiusb = 5; //movement speed for bullet
    }
    stroke(42, 52, 57);
    fill(42, 52, 57);
    ellipse(xb[c], yb[c], 10, 10);
  }
  //draw attacker
  stroke(0);
  fill(255);
  pushMatrix();
  translate(x, y);
  rotate(ang);
  fill(100);
  rect(7*s,-4*s,7*s,8*s);
  fill(50);
  rect(8*s,-3*s,s,7*s);
  rect(10*s,-3*s,s,7*s);
  rect(12*s,-3*s,s,7*s);
  fill(60);
  rect(8*s,-8*s,2*s,4*s);
  fill(224, 172, 105);
  ellipse(9*s,0,4*s,4*s);
  ellipse(-9*s,0,4*s,4*s);
  fill(0,100,100);
  ellipse(0,0,16*s,8*s);
  fill(224, 172, 105);
```

```
    ellipse(3*s,0,2*s,2*s);
    ellipse(-3*s,0,2*s,2*s);
    triangle(-s,-2*s,s,-2*s,0,-4.5*s);
    fill(0,0,0);
    ellipse(0,0,6*s,6*s);
    popMatrix();
    //check movement of attacker
    if (a_turningRight)
    {
      ang += rotateVel;
    }
    if (a_turningLeft)
    {
      ang -= rotateVel;
    }
  }

  void dead()
  {
    if(nos>0)
    {
      for(int yay=0;yay<nos;yay++)
      {
        obstacle part = obstacles.get(yay);
        float tdist=((y-part.y)*(y-part.y))+((x-part.x)*(x-part.x));
        if(tdist<=(part.size*part.size/4))
        {
          littscene(1);
        }
      }
    }
  }
}

class obstacle
{
  float ang;
  float x;
  float y;
  float vel;
  float size;

  obstacle()
```

```
{
  x=random(width);
  if(x==width/2)
  {
    x+=1;
  }
  y=0;
  ang=atan((x-width/2)/( height-200));
  vel=1.0+random(speed)/10;
  size=50+random(100);
}

void Draw()
{
  fill(255,255,255);
  ellipse(x,y,size,size);
  fill(0,0,0);
  ellipse(x,y,size*0.8,size*0.8);
  fill(0,204,204);
  ellipse(x,y,size*0.6,size*0.6);
  fill(255,0,0);
  ellipse(x,y,size*0.4,size*0.4);
  fill(255,255,0);
  ellipse(x,y,size*0.2,size*0.2);
}

void dead()
{
  for(int c=0;c<3;c++)
  {
    float tdist=((y-yb[c])*(y-yb[c]))+((x-xb[c])*(x-xb[c]));
    if(tdist<=(size*size/4))
    {
      print("dead");
      dead=true;
      if(freq>1.0)
      {
        freq-=0.3;
      }
      if(speed<50)
      {
        speed+=3;
      }
```

```
      //bullet variables (off-screen)
      xb[c] = -100;//initial x
      yb[c] = -100;//initial y
      angb[c] = -100;//initial angle
      radiusb = 5; //movement speed for bullet
    }
   }
  }

  void update()
  {
   x-=(vel*sin(ang));
   y+=(vel*cos(ang));
   Draw();
   dead();
  }

}

void menu(int i)
{
  if(i==-1)
  {
   b_a = new Button("Single Player", width/2-100, height/2, 200, 80);
   b_b = new Button("Multiplayer", width/2-100, height/2+150, 200, 80);
   b_c = new Button("Exit", width/2-100, height/2+300, 200, 80);
   i=0;
   font = createFont("Felix Titling", 200);
  }
  if(i==0)
  {
   counter=0;
   gc=0;
   background(bg);
   textAlign(CENTER, CENTER);
   fill(255,0,0);
   textSize(10);
   textFont(font);
   text("Boogeyman", width/2, height/4);
   b_a.Draw();
   b_b.Draw();
   b_c.Draw();
  }
```

```
if(i==1)
{
  background(bg);
  a.Draw();
  a.dead();
  counter+=1;
  gc+=1;
  timer=50-gc/fr;
  if(timer<=0)
  {
    i=-1;
    littscene(2);
  }
  textAlign(CENTER, CENTER);
  fill(255);
  textSize(30);
  text(timer, 100, 80);
  if(counter>=freq*fr)
  {
    nos+=1;
    obstacles.add(new obstacle());
    counter=0;
  }
  if(nos>0)
  {
    for(int yay=0;yay<nos;yay++)
    {
      obstacle part = obstacles.get(yay);
      part.update();
      if(part.y>height)
      {
        obstacles.remove(yay);
        nos-=1;
        yay-=1;
      }
    }
    if(dead==true)
    {
      obstacles.remove(deadval);
      nos-=1;
      dead=false;
    }
  }
```

```
  }
  if(i==2)
  {
    background(bg);
    d.Draw();
    a.Draw();
    d.dead();
    gc+=1;
    timer=50-gc/fr;
    if(timer<=0)
    {
      i=-1;
      littscene(3);
    }
    textAlign(CENTER, CENTER);
    fill(255);
    textSize(30);
    text(timer, 100, 80);
  }
  if(i==3)
  {
    exit();
  }
}

void littscene(int d)
{
  if(d==1)
  {
    background(bg1);
    textAlign(CENTER, CENTER);
    fill(255,0,0);
    textSize(400);
    textFont(font);
    text("You Lose!", width/2, height/2);
    delay(1000);
    exit();
  }
  if(d==2)
  {
    background(bg2);
    textAlign(CENTER, CENTER);
    fill(255,0,0);
```

```
    textSize(400);
    textFont(font);
    text("You Win!", width/2, height/2);
    delay(1000);
    exit();
  }
  if(d==3)
  {
    background(bg2);
    textAlign(CENTER, CENTER);
    fill(255,0,0);
    textSize(400);
    textFont(font);
    text("Defender Wins!", width/2, height/2);
    delay(1000);
    exit();
  }
  if(d==4)
  {
    background(bg2);
    textAlign(CENTER, CENTER);
    fill(255,0,0);
    textSize(400);
    textFont(font);
    text("Attacker Wins!", width/2, height/2);
    delay(1000);
    exit();
  }
}

//bullet variables (off-screen)
float xb[] = {-100,-100,-100};//initial x
float yb[] = {-100,-100,-100};//initial y
float angb[] = {-100,-100,-100};//initial angle
float radiusb = 5; //movement speed for bullet

int i=-1;
defender d;
attacker a;
Button b_a,b_b,b_c;
PImage bg,bg1,bg2;
float freq=5.0;
int counter=0;
```

```
float speed=10;
boolean dead=false;
int deadval=0;
int nos=0;
ArrayList<obstacle> obstacles = new ArrayList<obstacle>();
int fr=80;
int gc=0;
int timer=50;
int ac=0;
boolean ab=false;
PFont font;
float ta,td;

void setup()
{
  bg = loadImage("bgfs.jpg");
  bg1 = loadImage("d.jpg");
  bg2 = loadImage("v.jpg");
  fullScreen();
  d=new defender();
  a=new attacker();
  frameRate(fr);
  stroke(0);
  myPort = new Serial(this,"COM3", 9600);
  file = new SoundFile(this, "Battle Music.mp3");
  file.play();
  delay(1000);
}

void draw()
{
  menu(i);
  ta=atan(a_rot/20.0);
  td=atan(d_rot/20.0);
  a.ang=ta;
  d.ang=2*td;
    if((a_shoot.equals("F"))&&((4*gc)%fr==0))
    {
      a.a_turningLeft = false;
      a.a_turningRight = false;
      for(int c=0;c<3;c++)
      {
        if(angb[c]==-100)
```

```
        {
          angb[c] = a.ang;
          xb[c] = a.x+9*a.s*cos(a.ang);
          yb[c] = a.y+9*a.s*sin(a.ang);
          break;
        }
      }
    }
    if(d_mov.equals("U"))
    {
      d.d_movingForward = true;
      d.d_movingBackwards = false;
    }
    if(d_mov.equals("D"))
    {
      d.d_movingForward = false;
      d.d_movingBackwards = true;
    }
    if(d_mov.equals("N"))
    {
      d.d_movingForward = false;
      d.d_movingBackwards = false;
    }
}

void keyPressed()
{
  if (key==' ')
  {
    for(int c=0;c<3;c++)
    {
      if(angb[c]==-100)
      {
        angb[c] = a.ang;
        xb[c] = a.x+9*a.s*cos(a.ang);
        yb[c] = a.y+9*a.s*sin(a.ang);
        break;
      }
    }
  }
  if (key=='x')
  {
    if(ac<=0)
```

```
    {
      ab=true;
      ac=8*fr;
    }
  }
  if (key=='a') a.a_turningLeft = true;
  if (key=='d') a.a_turningRight = true;
  if (keyCode==RIGHT) d.d_turningRight = true;
  if (keyCode==LEFT) d.d_turningLeft = true;
  if (keyCode==UP) d.d_movingForward = true;
  if (keyCode==DOWN) d.d_movingBackwards = true;
}

void keyReleased()
{
  if (keyCode==RIGHT) d.d_turningRight = false;
  if (keyCode==LEFT) d.d_turningLeft = false;
  if (keyCode==UP) d.d_movingForward = false;
  if (keyCode==DOWN) d.d_movingBackwards = false;
  if (key=='a') a.a_turningLeft = false;
  if (key=='d') a.a_turningRight = false;
}

void mouseClicked()
{
  if((i==-1)||(i==0))
  {
    if (b_a.MouseIsOver())
    {
      i=1;
    }
    if (b_b.MouseIsOver())
    {
      i=2;
    }
    if (b_c.MouseIsOver())
    {
      i=3;
    }
  }
}

void serialEvent( Serial myPort)
```

```
{
  data= myPort.readStringUntil('\n');
  if (data != null)
  {
    alldata1 = trim(data);
    String items[]=split(alldata1,'/');
    println(alldata1);
    if(items.length>1)
    {
      d_mov=items[0];
      a_shoot=items[1];
      a_rot=int(items[2]);
      d_rot=int(items[3]);
    }
  }
}
```