P.U.A Pharos University in Alexandria

Faculty of Engineering

“Concurrent”

“Shooting Game Project”

Presented for:

Dr.Mohamed El Khouly

Eng.Sahar Magdy

Eng.Silvia El Massery

Presented by:

Veronia Osama Ragaei Zaki 5316219

How to Play?

1) Each hit a ball +1 score.
2) If you hit a ball with the color appears on the screen +3 score.
3) If you hit the ball with number that equals the mathematical equation appears on the screen -2 score.
4) If you hit the golden ball +15 score.
5) Move gun with right and left arrow.
6) Shoot with spacebar.
7) You have 1 minute if time runs out you lose.
8) Score is out of 100.
9) Auto Shoot with up arrow key.

Notes:

By each hit the color and the mathematical equation will be changed even if you hit it wrong.

<u>Code</u>
Class Newgame

```
package newgame;

import java.awt.image.BufferedImage;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import static jdk.nashorn.internal.objects.NativeMath.random;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.Timer;
public class Newgame {
        static   int randomNumber;
        static Random rand = new Random();
        private JButton b1;
   public static int time;
   public static void main(String[] args) throws InterruptedException {
              //to load ball in the frame
               BufferedImage play_Ball=null;
               BufferedImage play_Ball2=null;
               BufferedImage play_Ball3=null;
               BufferedImage play_Ball4=null;
               BufferedImage play_Ball5=null;
               BufferedImage play_Ball6=null;
               BufferedImage play_Gold=null;
               BufferedImage play_ship= null;
               BufferedImage pro= null;
               boolean flag = true;
               //if(isGame == true){
               try{

play_Ball=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStream
("newgame/blue-removebg-removebg.png"));

play_Ball2=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/black-removebg.png"));
play_Ball3=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/green-removebg.png"));

play_Ball4=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/index-removebg.png"));

play_Ball5=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/index2-removebg.png"));

play_Ball6=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/o.png"));

play_Gold=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStrea
m("newgame/gold.png"));

play_ship=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStream
("newgame/ship.png"));

pro=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStream("new
game/pro.png"));
            }
            catch(IOException e)
            {
              System.out.println(e);
            }

      Ball b=new Ball(play_Ball,(int)random(300) ,(int)random(300),1,3 );
      Ball b1=new Ball(play_Ball2,(int)random(300) ,(int)random(300),2,4 )  ;
      Ball b3=new Ball(play_Ball3,(int)random(300),(int)random(300),5,7 )  ;
      Ball b4=new Ball(play_Ball4,(int)random(300),(int)random(300),6,8)  ;
      Ball b5=new Ball(play_Ball5,(int)random(300) ,(int)random(300),10,9 )  ;
      Ball b6=new Ball(play_Ball6,(int)random(300) ,(int)random(300),3,6 )  ;
     Ball gold=new Ball(play_Gold,(int)random(300) ,(int)random(300),14,16 )  ;
     ship s = new ship(330, 570, play_ship);
     ship sh = new ship(330, 570, play_ship);
            projectile k = new projectile(s.x_pos+40, 620,pro, s );
            frame f=new frame(b, b1, b3, b4, b5 , b6 ,gold , s, k);
            //to make ball move
            NewClass n=new  NewClass(b,f,k);
            Thread t1=new Thread(n);
```

```
  t1.setName("blue");
  NewClass n1=new  NewClass(b1,f,k);
  Thread t2=new Thread(n1);
  t2.setName("black");
  NewClass n2=new  NewClass(b3,f, k);
  Thread t3=new Thread(n2);
  t3.setName("green");
  NewClass n3=new  NewClass(b4,f, k);
  Thread t4=new Thread(n3);
  t4.setName("red");
  NewClass n4=new  NewClass(b5,f,k);
  Thread t5=new Thread(n4);
  t5.setName("yellow");
  NewClass n5=new  NewClass(b6,f,k);
  Thread t6=new Thread(n5);
  t6.setName("orange");
  NewClass g = new  NewClass(gold,f,k);
  Thread t7 = new Thread(g);
  t7.setName("gold");
  createShip shippo = new createShip(s, f);
  Thread hoho = new Thread(shippo);
  t1.start();
  t2.start();
  t3.start();
  t4.start();
  t5.start();
  t6.start();
  t7.start();
  hoho.start();
 /*Thread t1=new Thread(new NewClass(b,f,k));
Thread t2=new Thread(new  NewClass(b1,f,k));
Thread t3=new Thread(new  NewClass(b3,f, k));
Thread t4=new Thread(new NewClass(b4,f,k));
Thread t5=new Thread(new  NewClass(b5,f,k));
Thread t6=new Thread(new  NewClass(b5,f,k));
Thread hoho = new Thread(new createShip(s, f));
ExecutorService pool = Executors.newFixedThreadPool(6);
pool.execute(t1);
pool.execute(t2);
pool.execute(t3);
pool.execute(t4);
```

```
                    pool.execute(t5);
                    pool.execute(hoho);
                    pool.shutdown()*/;
                    f.addKeyListener(new KeyListener(){
                        public void keyTyped(KeyEvent e) {
                    }
            /** Handle the key pressed event from the text field. */
                      public void keyPressed(KeyEvent c) {
                         int key = c.getKeyCode();
                         if(key==KeyEvent.VK_SPACE){
                              createPro pr = new createPro(k, f);
                              Thread pt = new Thread(pr);
                              pt.start();
                         }
                      }
                      public void keyReleased(KeyEvent e) {
                      }
                  });
                      f.addKeyListener(new KeyListener(){
                        public void keyTyped(KeyEvent e) {
                       }
                      public void keyPressed(KeyEvent c) {
                         int key = c.getKeyCode();
                         if(key==KeyEvent.VK_UP && flag == true){
                             for(int i=0;i<100;i++){
                                  createPro pr = new createPro(k, f);
                                 Thread pt = new Thread(pr);
                                 pt.start();
                             }
                         }
                      }
                      public void keyReleased(KeyEvent e) {
                      }
                  });
          }
       }
```

Functions

Public void keyPressed(KeyEvent c):

It waits to make an event on a key pressed and in the code if spacebar is pressed a small ball is fired from the ship.

public void keyReleased(KeyEvent e) :

It is responsible to make nothing when the key that was pressed released as to not fire balls even if the spacebar is pressed.

public static void main(String[] args) throws InterruptedException :

we make six balls (In a pool of threads) and a ship so the main only creates seven threads and calls the frame to start drawing in 2D the six balls and the ship.

Code
Class NewClass

```
package newgame;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.ArrayList;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.Timer;
public class NewClass implements Runnable{
Ball b;
frame f;
frame s;
projectile p;
static int arr[];
static int intArray[];
static int score = 0;
static int k = 0;
 static int temp;
int x;
int y=0;
static int randomNumber;
static int num;
static int num2;
static int sum;
int flag = 0;
Random rand = new Random();
public static boolean isGame = true;
     public static void paint(Graphics g , int score) {
            g.drawString("Score: " + score, 10, 10);
        }
    public NewClass(Ball b,frame f, projectile p) {
        this.b=b;
        this.f=f;
        this.p=p;
    }
 @Override
   public void run() {
        while(isGame == true){{
                     try{
                         b.update();
                          try{
                             f.screen.repaint();
                          }
                         catch (NullPointerException e){
                                System.out.println(e);
                          }
                         Thread.sleep(15);
                     }
                    catch (InterruptedException ex){
                       Logger.getLogger(Ball.class.getName()).log(Level.SEVERE, null,ex);
                     }

               String str[] = {"Red" , "Black" , "Blue" , "Yellow" , "Orange" , "Green"};
                int arr[] = {5, 2, 1, 4, 6, 3 };

                         if(flag == 0 ){
                           flag +=10;
                           randomNumber=rand.nextInt((5 - 0) + 1) + 0;
                           num = rand.nextInt((3 - 0) + 1) + 0;
                           num2 = rand.nextInt((3 - 1) + 1) + 1;

                           try {
                               Thread.sleep(100);
                           } catch (InterruptedException ex) {

Logger.getLogger(NewClass.class.getName()).log(Level.SEVERE, null, ex);
                           }
                         s = new frame(score , k);
                         s.paint(randomNumber);
                         s.rand(num, num2);
                         }
                         sum = (num + num2);
                         if((p.x_pos>=b.x_pos-50 && p.x_pos<= b.x_pos+50) &&
(p.y_pos>=b.y_pos && p.y_pos<= b.y_pos+50)){
                              System.out.println("proxpos="+p.x_pos+" proypos="+
p.y_pos + "ballx="+ b.x_pos + "bally="+ b.y_pos);

                              System.out.println("no " + randomNumber);
                              System.out.println(Thread.currentThread().getName());

 if(randomNumber == 0 && Thread.currentThread().getName() == "red"){
                                score = score + 2;
                                System.out.println("yesssssss    redddddd");
                              }
 else if(randomNumber == 1 && Thread.currentThread().getName() == "black"){
                               score = score + 2;
                               System.out.println("yesssssss    black");
                              }
else if(randomNumber == 2 && Thread.currentThread().getName() == "blue"){
                               score = score + 2;
                               System.out.println("yesssssss    blue");
                              }

else if(randomNumber == 3 && Thread.currentThread().getName() == "yellow"){

                                score = score + 2;
                                System.out.println("yesssssss    yelloww");
                              }

else if(randomNumber == 4 && Thread.currentThread().getName() == "orange"){

                                score = score + 2;
                                System.out.println("yesssssss    orangeee");
                              }

else if(randomNumber == 5 && Thread.currentThread().getName() == "green"){

                                score = score + 2;
                                System.out.println("yesssssss    greeeen");
                              }

if(Thread.currentThread().getName() == "gold"){

                                System.out.println("yesssssss    golddddddddddd");
                                score = score + 15;
}
```

```
if(sum == 5 && Thread.currentThread().getName() == "red"){
                score = score - 3;
                System.out.println("yesssssss    reddddd");
     }
    else if(sum == 2 && Thread.currentThread().getName() == "black"){
                score = score - 3;
                System.out.println("yesssssss    black");
     }
    else if(sum == 1 && Thread.currentThread().getName() == "blue"){
               score = score - 3;
               System.out.println("yesssssss    blue");
    }
  else if(sum == 4 && Thread.currentThread().getName() == "yellow"){
                   score = score - 3;
                   System.out.println("yesssssss    yelloww");
    }
 else if(sum == 6 && Thread.currentThread().getName() == "orange"){
                   score = score - 3;
                   System.out.println("yesssssss    orangeee");

   }

  else if(sum == 3 && Thread.currentThread().getName() == "green"){
                   score = score - 3;
                   System.out.println("yesssssss    greeeen");

   }

               ArrayList<Integer> my = new ArrayList<Integer>();
               score = score + 1;
               my.add(score);
               int k = my.get(0);
               System.out.println("score" + score);
               s = new frame(score , k);
               s.paint(randomNumber);
               s.rand(num, num2);
               randomNumber=rand.nextInt((5 - 0) + 1) + 0;
               num = rand.nextInt((3 - 0) + 1) + 0;
               num2 = rand.nextInt((3 - 1) + 1) + 1;
               s = new frame(score , k);
               s.paint(randomNumber);
               s.rand(num, num2);
```

```
                                    p.x_pos=0;
                                    p.y_pos=0;
                                    b.x_pos=1500;
                                    b.y_pos=1500;
                                    Timer timer = new Timer(500, new ActionListener() {
                                      @Override
                                      public void actionPerformed(ActionEvent e) {
                                        b.x_pos=0;
                                         b.y_pos=0;
                                      }
                                    });
                                    timer.setRepeats(false);
                                    timer.setDelay(500);
                                    timer.start();
                                  }
                            }
                      }
                   }

                 }
```

Functions

public void run() :

- This class has only one method which is run because when the thread is created an call run by start() the thread goes to run method which is responsible if the ball is shoot it will check if the ball color is the color which appears on the screen and if it was it will increase the score +3 an if not it will increase the score +1.
- It also checks the mathematical equation result if he hit the ball which contains the number that is the result of the mathematical equation it will make the score -2.
- If he hit the ball with the color appears and at the same time it has the number which is the result of the mathematical equation it will +3 and then -2 so the result will be +1 score.
- It contains the timer that refresh the game every one millisecond.

Code
Class frame

```
package newgame;

 import java.awt.BorderLayout;
 import java.awt.Color;
 import java.awt.Graphics;
 import java.awt.event.KeyEvent;
 import java.awt.event.KeyListener;
 import java.awt.image.BufferedImage;
 import java.io.IOException;
 import java.text.SimpleDateFormat;
 import java.time.LocalTime;
 import java.time.format.DateTimeFormatter;
 import java.util.Date;
 import java.util.Random;
 import java.util.logging.Level;
 import java.util.logging.Logger;
 import javax.imageio.ImageIO;
 import javax.swing.JButton;
 import javax.swing.JFrame;
 import javax.swing.JLabel;
 import javax.swing.JPanel;
 public class frame extends JFrame{
    Ball ball;
    Ball ball_2;
    Ball ball_3;
    Ball ball_4;
    Ball ball_5;
    Ball ball_6;
    Ball gold;
    ship s;
    BufferedImage winner=null;
    BufferedImage loser=null;
    BufferedImage l=null;
    projectile p;
    screen screen;
    screen screen2;
    int arr[];
    private JButton b1;
    private JPanel p1;
    static int score=0;
    static int x=0;
    static int y=0;
    static int k=0;
    static int temp;
    int time;
    boolean flag2 = true;
    boolean flag = true;
    boolean flag3 = true;

    public frame(int score , int k){
            this.score = score;
            this.k=k;
            screen=new screen();
            screen.setBounds(0, 0, 900, 800);
            add(screen);
    }
    public frame(Ball ball,  Ball ball_2,Ball ball_3,  Ball ball_4,Ball ball_5 , Ball
ball_6 , Ball gold , ship s, projectile p){
        super("ball game");
        setSize(800, 700);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
        setLayout(new BorderLayout());
                setVisible(true);
                JPanel p1 = new JPanel();
                JButton button = new JButton("Test");
                 new NewJFrame().setVisible(true);
                this.ball=ball;
                this.ball_2=ball_2;
                this.ball_3=ball_3;
                this.ball_4=ball_4;
                this.ball_5=ball_5;
                this.ball_6=ball_6;
                this.gold=gold;
                this.p = p;
                this.s = s;
                screen=new screen();
                screen.setBounds(0, 0, 900, 800);
                add(screen);
                NewJFrame fo = new NewJFrame();

    }
```

```
public void paint(int temp2){
            temp = temp2 ;
            flag3 = false;
            System.out.println("tempppppppppppp      " +temp);
}
public void rand(int num , int num2){
   x = num;
   y = num2;
}
public class screen extends JLabel {
   protected void paintComponent(Graphics g) {
    super.paintComponent(g);
      g.drawImage(ball.play_ball, (int)ball.x_pos, (int)ball.y_pos, null);
      g.drawImage(ball_2.play_ball, (int)ball_2.x_pos, (int)ball_2.y_pos, null);
      g.drawImage(ball_3.play_ball, (int)ball_3.x_pos, (int)ball_3.y_pos, null);
      g.drawImage(ball_4.play_ball, (int)ball_4.x_pos, (int)ball_4.y_pos, null);
      g.drawImage(ball_5.play_ball, (int)ball_5.x_pos, (int)ball_5.y_pos, null);
      g.drawImage(ball_6.play_ball, (int)ball_6.x_pos, (int)ball_6.y_pos, null);

      g.drawImage(p.proo,(int)s.x_pos+40, (int)p.y_pos, null);
      g.drawImage(s.shipo, (int)s.x_pos, (int)s.y_pos, null);
      g.drawLine(0, 550 ,900 ,550);
      g.drawRect(10, 20, 120, 100);
      g.drawRect(630, 20, 140, 100);
      g.drawRect(300, 20, 140, 100);
      g.drawString(x + " " + " + " + " " + y  , 350 , 70);

      String str[] = {"Red" , "Black" , "Blue" , "Yellow" , "Orange" , "Green"};

      if(temp == 0 ){
         g.drawString("Colour: " + str[0] , 650, 70);
         flag3 = false;
      }
      else if(temp == 1){
         g.drawString("Colour: " + str[1] , 650, 70);
         flag3 = false;
      }
      else if(temp == 2){
         g.drawString("Colour: " + str[2] , 650, 70);
         flag3 = false;
      }
```

```
            else if(temp == 3){
                g.drawString("Colour: " + str[3] , 650, 70);
                flag3 = false;
            }
            else if(temp == 4){
                g.drawString("Colour: " + str[4] , 650, 70);
                flag3 = false;
            }
            else if(temp == 5){
                g.drawString("Colour: " + str[5] , 650, 70);
                flag3 = false;
            }
            if(score < 0 || k < 0){
                score=0;
                k=0;
            }
            time = time + 1;
           if((time > 0 && time < 500) ||(time > 2000 && time < 2500) ||(time >
4000 && time < 4500)  ){
                 g.drawImage(gold.play_ball, (int)gold.x_pos, (int)gold.y_pos, null);
            }

            //score =score + 1;
            g.drawString("Score: " + score, 30, 60);
            if(score >= 100 || flag == false){
              if(score > 100){
                  score = 100;
              }
             try {

winner=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStream("
newgame/win.jpg"));

             } catch (IOException ex) {
                 Logger.getLogger(frame.class.getName()).log(Level.SEVERE, null, ex);
             }

             g.drawImage(winner, (int)-100, (int)0, null);
             g.drawString("Score: " + score , 30, 60);
             time = 0;
             NewClass.isGame = false;
             score=k;
             flag = false;
            }
            else if(score < 100 && time > 6000 || flag2 == false){
             try {

loser=ImageIO.read(Newgame.class.getClassLoader().getResourceAsStream("ne
wgame/over.jpg"));

             } catch (IOException ex) {
                 Logger.getLogger(frame.class.getName()).log(Level.SEVERE, null, ex);
             }
                //g.drawImage(l, (int)-100, (int)0, null);
                g.drawImage(loser, (int)-100, (int)0, null);
                g.drawString("Score: " + score , 30, 60);
               NewClass.isGame = false;
               time = 0;
               score = k;
               flag2 = false;
              again n =  new again();
            }

            String dateAsText = new SimpleDateFormat("mm:ss.SS")
                          .format(new Date(time * 10L));
            g.drawString("Time: " + dateAsText, 30, 100);

        }
    }

     public class screen2 extends JLabel {
        protected void paintComponent2(Graphics g) {
         super.paintComponent(g);
            g.drawString("Score: " + score, 30, 60);
        }
    }
}
```

Functions

- It is to draw the balls, the ships and the rectangle frames and it also takes random numbers from the class NewClass to choose
  which color to be printed on the screen and takes also the two random numbers form class NewClass to print the mathematical equation on the screen.
- It also has the timer of the golden ball it appears in three different intervals.
- It has the conditions for the winning and losing and appearing the winning and losing images.
- It contains the score and timer calculator.

Code

Class Ball

```
package newgame;

import java.awt.Dimension;
import java.awt.image.BufferedImage;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JPanel;
public class Ball extends JPanel {
    float x_pos;
    float y_pos;
    float x_speed;
    float y_speed;
    BufferedImage play_ball;

    public Ball( BufferedImage play_ball,float x_pos, float y_pos,float
x_speed,float y_speed) {
        this.x_pos = x_pos;
        this.play_ball=play_ball;
        this.y_pos = y_pos;
        this.x_speed=x_speed;
        this.y_speed=y_speed;

    }
    public void update()
    {
        x_pos=x_pos+x_speed;
        y_pos=y_pos+y_speed;
       if (x_pos<0)
           x_speed *=-1;
       if(x_pos+play_ball.getWidth()>800)
             x_speed *=-1;
        if (y_pos<0)
           y_speed *=-1;
       if(y_pos+play_ball.getHeight()>550)
             y_speed *=-1;
    }
}
```

Functions

- To set the position of the ball and update the position every time the ball is being shoot.
- To set the speed of the ball and update the position every time the ball is being shoot.

Code
Class ship

```
package newgame;
import java.awt.Dimension;
import java.awt.image.BufferedImage;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JPanel;
public class ship extends JPanel {
   float x_pos;
   float y_pos;
   BufferedImage shipo;
   public ship(){
   }
   public ship(float x_pos, float y_pos,  BufferedImage shipo) {
      this.x_pos = x_pos;
      this.y_pos = y_pos;
      this.shipo = shipo;
   }
   public void updateright(){
      x_pos=x_pos+50;
   }
   public void updateleft(){
      x_pos= x_pos-50;
   }
}
```

<u>Functions</u>

- To set the position of the ship and update the position every time the program is being refreshed.
- **<u>public void updateright():</u>**
  It updates the moving position to right by +50 to the original x-axis position to move right.
- **<u>public void updateleft():</u>**
  It updates the moving position to left by -50 to the original x-axis position to move left.

<u>Code</u>
<u>Class createShip</u>

```
package newgame;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.Timer;
import static newgame.NewClass.randomNumber;
import java.util.Random;
public class createShip implements Runnable  {
   private final Object lock = new Object();
   ship p;
   frame f;
    boolean flag = true;
    int randomNumber;
    Random rand = new Random();
   public createShip(ship p,  frame f) {
      this.p = p;
      this.f = f;
   }
public void run(){
   try{
         synchronized(lock){
             f.addKeyListener(new KeyListener() {
             @Override
             public synchronized void keyPressed(KeyEvent e) {
             int key = e.getKeyCode();
                  if (key == 39 && p.x_pos<700) {
                     p.updateright();
                  }
                  if(key == 37 && p.x_pos>0){
                     p.updateleft();
                  }
             }
             @Override
             public synchronized void keyTyped(KeyEvent e) {
             }
```

```
 @Override
              public synchronized void keyReleased(KeyEvent e) {
              }
           });}

        }
        catch (NullPointerException e){
                        System.out.println(e);
                    }
}
}
```

Functions

- It is responsible for moving the ship left and right by clicking on right button to move right and left button to move left.
- This run method has a synchronized lock to lock on the run method as when we clicked on right button only the ship will be moved to the right and no one will access the run method.
- This run method has a synchronized lock to lock on the run method as when we clicked on left button only the ship will be moved to the left and no one will access the run method.

Code
Class projectile

```
package newgame;
import java.awt.image.BufferedImage;
import javax.swing.JPanel;
public class projectile extends JPanel {
   float x_pos;
   float y_pos;
   BufferedImage proo;
   ship ships;

   public projectile(float x_pos, float y_pos, BufferedImage proo, ship ships) {
      this.x_pos = x_pos;
      this.y_pos = y_pos;
      this.proo = proo;
      this.ships = ships;
   }
   public void update(){
      y_pos = y_pos - 10; // sor3it al tal2a
      x_pos = ships.x_pos;
   }
}
```

Functions

- It is responsible to set the position in x-axis and y-axis of the projectile (The small ball) and makes it in the fixed position from the ship.
- **public void update():**
  It is responsible to set and updates the speed of the projectile (The small ball) and makes it in the relative from the ship.

Code
Class createprojectile

```
package newgame;

import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.Graphics;
import java.util.logging.Level;
import java.util.logging.Logger;
public class createPro implements Runnable {
   projectile p;
   frame f;
   public createPro(projectile p, frame f) {
      this.p = p;
      this.f = f;
   }
private final Object lock = new Object();
public void run(){
   while(p.y_pos>0){
           try{
                   p.update();
                   try{
                        f.screen.repaint();
                    }
                    catch (NullPointerException e){
                          System.out.println(e);
                    }
                       Thread.sleep(15);
                    }
                    catch (InterruptedException ex){
                    Logger.getLogger(Ball.class.getName()).log(Level.SEVERE, null,ex);
                    }
   }
   p.y_pos=620;
   }}
```

Functions

- It updates the projectile's position to move upwards until the end of the frame, if it reaches the end of the frame it terminates.

Code

Class NewJFrame

```
 package newgame;
public class NewJFrame extends javax.swing.JFrame {
   public NewJFrame() {
      initComponents();
   }
   @SuppressWarnings("unchecked")
   private void initComponents() {
      jButton1 = new javax.swing.JButton();
      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
      jButton1.setText("How to play?");
      jButton1.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
         }
      });
      javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 63,
Short.MAX_VALUE)
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addComponent(jButton1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
      );
      pack();
   }
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
       new how().setVisible(true);
       dispose();
   }
   public static void main(String args[]) {
      try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        }
          java.awt.EventQueue.invokeLater(new Runnable() {
           public void run() {
               new NewJFrame();
               //new NewJFrame().setResizable(false);
            }
        });
    }
    private javax.swing.JButton jButton1;
}
```

Functions

- It is responsible for making the window how to play that appears in the first time when you run the game this window contains only one button.

<u>Code</u>
<u>Class how</u>

Functions

- It is responsible for making the window how to play that appears in the first time when you run the game this window contains the 10 steps of how to play this game.