



Jesenji semestar, 2014/15

Izveštaj o projektu

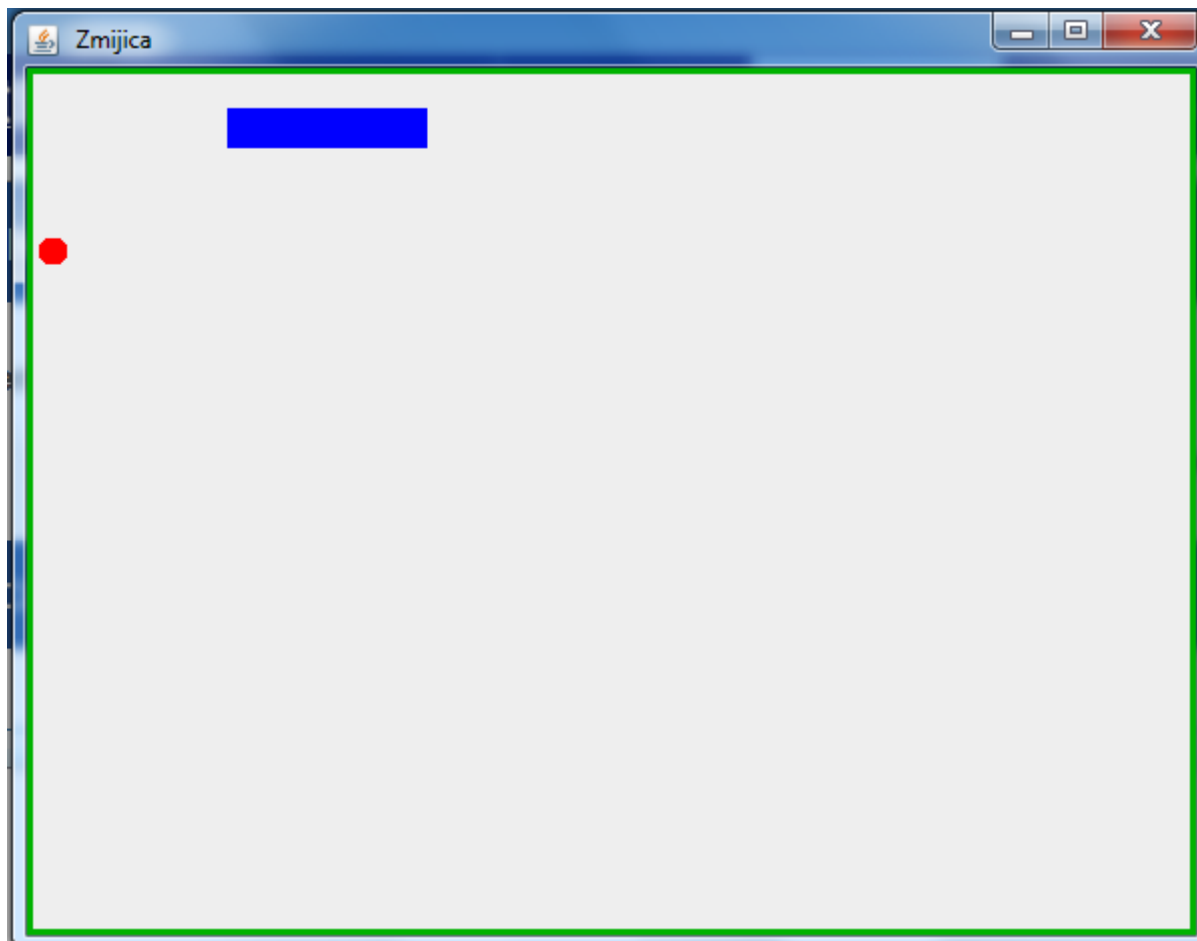
Ime:	Ivan Blagojević
Broj indeksa:	2560
Predmet:	Uvod u objektno orjentisano programiranje
Šifra predmeta:	CS101
Datum:	13.4.2015.

SADRŽAJ:

Uvod.....	3.
Main Frame	4.
Panel	5.
Hrana.....	6.
Vrednosti	7.
Zmijica	7.
Engin	9.
Krajnji ishod	12.

1.UVOD:

Ova igrice pod nazivom Zmijice sastoji se od paketa od kojih se nalazi pet klasa. Zmijica se pomera na strelice. Zmijica jede lopticu. Kada zmija pojede lopticu, loptica se pomoću random metode pojavi na drugom mestu. Klase od kojih se igra sastoji su Main Frame, Hrana, Engin, Panel, Zmijica.



Slika 1. *Izgled igre*

2.Igra(Main Frame):

```
public class MainFrame extends JApplet implements KeyListener {
    @Override
    public void init() {
        setSize(840,720);
        Container cpane = getContentPane();

        cpane.add(snakePanel);
        addKeyListener(this);
        MainFrame igr = new MainFrame();
        JFrame myFrame = new JFrame(" Zmijica ");
        myFrame.setSize(600,480);
        myFrame.getContentPane().add(igr);
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.addKeyListener(igr);
        myFrame.setLocationRelativeTo(null);

        igr.init();
        igr.start();
        igr.setVisible(true);
        myFrame.setVisible(true);
    }

    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {

        if (e.getKeyCode() == UP) {
            snakePanel.set_Direction(1);
        } else if (e.getKeyCode() == RIGHT) {
            snakePanel.set_Direction(2);
        } else if (e.getKeyCode() == DOWN) {
            snakePanel.set_Direction(3);
        } else if (e.getKeyCode() == LEFT) {
            snakePanel.set_Direction(4);
        }
    }

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

U ovoj klasi dajemo naziv tabli koju smo predhodno dodali, nakon toga toj tabli dajemo opciju da može da se ugasi klikom na "X". Dalje dajemo tabli sirinu i visinu koje smo zapisali u interfejsu Vrednosti. Nakon toga setujemo da se Frame postavi na centar ekrana nakon uključivanja, setujemo da je frame vidljiv i da ne može da se povećava ili smanjuje. Zatim ubacujemo lisene, šta će se desiti kada pritisnemo neku od pozicija (gore, dole, levo, desno).

3.Panel:

U ovoj klasi iscrtavamo sve što je potrebno, određujemo gde će se zmija iscrtati jedni i sve ostalo što je potrebno da bi igrica radila.

```
public class Panel extends JPanel implements Vrednosti{
    public Panel() {
```

```
        setBorder(my_border);
```

```
        mythread.start();
```

```
    }
```

U paint() metodi iscrtali smo liniju po kojoj se zmija kreće i postavljanje boje zmije i loptice.

Metoda:

@Override

```
    public void paintComponent(Graphics g) {
```

```
        super.paintComponent(g);
```

```
        Graphics2D g2D = (Graphics2D) g;
```

```
        g2D.setColor(Color.BLUE);
```

```
        /*
```

```
        g2D.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
```

```
        g2D.setRenderingHint(RenderingHints.KEY_RENDERING, RenderingHints.VALUE_RENDER_QUALITY);*/
```

```
        Parts_X = my_engin.return_Snake_postionsX();
```

```
        Parts_Y = my_engin.return_Snake_postionsY();
```

```
        for (int i = 0; i < my_engin.get_my_snake_lenght(); i++) {
```

```
            g2D.fillRect(Parts_X[i], Parts_Y[i], my_Oval_size, my_Oval_size);
```

```
        }
```

```
        g2D.setColor(Color.RED);
```

```
        g2D.fillOval(my_engin.get_Food_X() + 5, my_engin.get_Food_Y() + 5, my_Oval_size - 5, my_Oval_size - 5);
```

4.Hrana:

U ovoj klasi mi definisemo random metodu , koju koristimo kako bi se hrana pojavila na random mesto. I kada se pojede da se opet pojavi na sledece random mesto.

Metoda:

```
public class Hrana {

    Point2D.Float pozicijaHrane = new Point2D.Float();
    Random generatorHrane = new Random();

    public Hrana() {
        pozicijaHrane.setLocation((((int) ((Math.random() * (400))) % 20) * 20), (((int)
(Math.random() * (90))) % 20) * 20));
        int x = (int) pozicijaHrane.getX();
        int y = (int) pozicijaHrane.getY();

        while (x > 560 || y > 440) {
            pozicijaHrane.setLocation((((int) ((Math.random() * (400))) % 20) * 20), (((int)
(Math.random() * (90))) % 20) * 20));
            x = (int) pozicijaHrane.getX();
            y = (int) pozicijaHrane.getY();
        }

    }

    public void getHrana() {
        pozicijaHrane.setLocation((((int) ((Math.random() * (400))) % 20) * 20), (((int)
(Math.random() * (90))) % 20) * 20));

    }

    public double getX() {
        return pozicijaHrane.getX();
    }

    public double getY() {
        return pozicijaHrane.getY();
    }

    public int get_int_X() {
        return (int) pozicijaHrane.x;
    }

    public int get_int_Y() {
        return (int) pozicijaHrane.y;
    }
}
```

```
}
```

5.Vrednosti:

Interfejs Vrednosti je javina klasa inteface koja ima sve zadate vrednosti tj. Parametre bez kojih igrice ne bi radila.

Interfejs:

```
public interface Vrednosti {  
  
    public Engin my_engin = new Engin();  
    int direction = 2;  
    public final int LEFT = 37;  
    public final int RIGHT = 39;  
    public final int UP = 38;  
    public final int DOWN = 40;  
    public LineBorder my_border = new LineBorder(Color.green.darker(), 3);  
  
}
```

6.Zmijica:

U ovoj javinoj klasi pravimo kako ce zmija izgledati i koliko ce se kordinate pomeriti kada pritisnemo neki od lisenera i proverava gde se nalazi zmija , kako bi postavio hranu na razlicito mesto.

```
public class Zmijica {  
  
    private int headX = 20;  
    private int headY = 20;  
    private int[] PostionsX = new int[300];  
    private int[] PostionsY = new int[300];  
    private int snake_length = 5;  
    private final int Oval_size = 20;  
  
    public Zmijica() {  
        for (int i = 0; i < snake_length; i++) {  
            PostionsX[i] = 20;  
            PostionsY[i] = 20;  
        }  
    }  
  
    public void pomeriZmijicu() {  
        for (int i = 0; i < snake_length; i++) {  
            PostionsX[snake_length - i] = PostionsX[snake_length - (i + 1)];  
            PostionsY[snake_length - i] = PostionsY[snake_length - (i + 1)];  
        }  
    }  
}
```

```
}

PostionsX[0] = headX;
PostionsY[0] = headY;

}

public void put_Head_postion(int x, int y) {
    headX += x;
    headY += y;
}

public boolean proveraZaKrajIgre() {
    for (int i = 1; i < snake_length; i++) {
        if ((PostionsX[i] == headX) && (PostionsY[i] == headY)) {

            return true;
        }
    }
    if (headX < 20 || headX > 540 || headY < 20 || headY > 400) {

        return true;
    }

    return false;
}

public boolean compare_with_foodPostion(int x, int y) {

    for (int i = 0; i < snake_length; i++) {
        if (x == PostionsX[i] && y == PostionsY[i]) {
            return true;
        }
    }

    return false;
}

public synchronized void povecajDuzinu() {
    snake_length++;
}

public int getDuzinu() {
    return snake_length;
}
```



```
public int getOval_Size() {
    return Oval_size;
}

public int[] get_All_postionsX() {
    return PostionsX;
}

public int[] get_All_postionsY() {
    return PostionsY;
}

public int getHeadX() {
    return PostionsX[0];
}

public int getHeadY() {
    return PostionsY[0];
}
}
```

7.Engin:

U ovoj klasi mi imamo sta se desava kada zmija pojede hranu , i sta se desava kada udari u zid. I vidimo da se ispisuje rezultat kada se završi igrice preko JOptionPane.

```
public class Engin {
    private int smer = 2;
    private boolean krajIgre;
    private boolean Moving_vertical = false;
    private Zmijica zmijica = new Zmijica();
    private final Hrana hranaZaZmijicu = new Hrana();
    private int step_to_move = 20;
    private int rezultat;

    public Engin() {

    }

    public void kretanje(int newDirection, boolean food_eaten) {

        if (!krajIgre) {

            switch (newDirection) {
                case 1:
                    if (smer != 3) {
```

```
        zmijica.put_Head_postion(0, -step_to_move);
        smer = 1;
    } else {
        move_as_last(smer);
    }
    break;//UP

case 2:
    if (smer != 4) {
        zmijica.put_Head_postion(step_to_move, 0);
        smer = 2;
    } else {
        move_as_last(smer);
    }
    break;//RIGHT
case 3:
    if (smer != 1) {
        zmijica.put_Head_postion(0, step_to_move);
        smer = 3;
    } else {
        move_as_last(smer);
    }
    break;//DOWN

case 4:
    if (smer != 2) {
        zmijica.put_Head_postion(-step_to_move, 0);
        smer = 4;
    } else {
        move_as_last(smer);
    }

    break;//LEFT
}

if (food_eaten) {
    zmijica.povecajDuzinu();
}
zmijica.pomeriZmijicu();

if ((zmijica.getHeadX() == hranaZaZmijicu.get_int_X()) && (zmijica.getHeadY() ==
hranaZaZmijicu.get_int_Y())) {
    hranaZaZmijicu.getHrana();
    rezultat += 20;

    zmijica.povecajDuzinu();
```

```
        while (check_food_place()) {
            hranaZaZmijicu.getHrana();
        }

    }

    krajlgre = zmijica.proveraZaKrajlgre();

}
if (krajlgre) {
    JOptionPane.showConfirmDialog(null, "Score = " + Integer.toString(rezultat), "Your Score",
JOptionPane.DEFAULT_OPTION);
    // Thread.yield();
}

}

public int[] return_Snake_postionsX() {
    return zmijica.get_All_postionsX();
}

public int[] return_Snake_postionsY() {
    return zmijica.get_All_postionsY();
}

public int get_Oval_size() {
    return zmijica.getOval_Size();
}

public int get_my_snake_lenght() {
    return zmijica.getDuzinu();
}

public void set_step_length() {
    step_to_move = 5;
}

public int get_Food_X() {
    return (int) hranaZaZmijicu.getX();
}

public int get_Food_Y() {
    return (int) hranaZaZmijicu.getY();
}

public boolean check_food_place() {
```

```
int xfood = (int) hranaZaZmijicu.getX();  
int yfood = (int) hranaZaZmijicu.getY();  
if (xfood > 560 || yfood > 440) {  
    return true;  
} else {  
    return zmijica.compare_with_foodPostion(xfood, yfood);  
}  
  
}  
  
public boolean get_Game_over() {  
    return krajIgre;  
}  
  
}
```

8.Krajnji ishod igrice

