

Univerzitet u Novom Sadu

Tehnički fakultet „Mihajlo Pupin“

Zrenjanin



Predmet: Projektovanje softvtera

SEMINARSKI RAD

**Monkey Jump**

Predmetni nastavnik: Student:

Doc. dr Eleonora Brtka Dajana Pilipović SI 11/20

Zrenjanin, 2022

# **Sadržaj**

[**I.** **Predmetni cilj** 3](#_Toc114276176)

[**A.** **Sadržaj/struktura predmeta** 3](#_Toc114276177)

[**B.** **Obrazovni cilj** 3](#_Toc114276178)

[**C.** **Ishodi obrazovanja (Stečena znanja)** 3](#_Toc114276179)

[**II.** **Uvod** 4](#_Toc114276180)

[**III.** **Korisne informacije** 4](#_Toc114276181)

[**IV.** **Zaključak** 28](#_Toc114276182)

[**Literatura** 29](#_Toc114276183)

# **Predmetni cilj**

## **Sadržaj/struktura predmeta**

Upoznavanje sa postojećim arhitekturnim stilovima i okvirima. Konceptualni i tehnički dizajn softverskih sistema. Dekompozicija i modularnost softvera. Višeslojna arhitektura softvera. Metode i tehnike konstrukcije. Timski rad i timski razvoj softvera, H-tream programiranje. Testiranje softvera. Pregled alata i tehnika koje se koriste u analizi i ocenjivanju kvaliteta softvera. Integracija softvera, verifikacija i validacija. Dokumentovanje softvera.[1]

## **Obrazovni cilj**

Steći znanja neophodna za kreiranje efikasnog softvera koji uključuje arhitekturu softvera, implementaciju i testiranje. Kreiranje dokumentacije za softverske sisteme.[2]

## **Ishodi obrazovanja (Stečena znanja)**

Osposobljenost za samostalnu konstrukciju softvera koja je bazirana na standarnim procesima implementacije i testiranja. Osposobljenost za kreiranje projektne dokumentacije.[2]

# **Uvod**

Monkey Jump je igrica tipa platformer i rađena je u Java Eclipse. Cilj igrice je da igrač pokupi sve banane unutar nivoa u što kraćem vremenskom periodu.

# **Korisne informacije**

Igrač na početku igrice može da izabere jednog od dva ponuđena karaktera. Cilj je pokupiti sve banane u nivou, u što kraćem vremenskom periodu. Banana u gornjem desnom uglu pored brojača skupljenih banana se ne može pokupiti, jer ona označava šta je potrebno skupiti.

Rad je rađen u tri glavna ekrana:

* Glavni meni u JFrame-u
* Meni posle igre u JDialog-u
* Igrica u JFrame-u

U radu je korišteno delo kompozitora JuliusH, naziv kompozicije je: „String Adagio - Classic String Ballad in Dorian Mood“. Delo je preuzeto u mp3 formatu, pošto klasa AudioSystem podržava wav format, programom „Audacity“ je izvršeno konvertovanje dela iz mp3 u wav format.

Veoma je bitno izvršiti update Jave!

<https://www.oracle.com/java/technologies/downloads/#jdk18-windows>

Slike korištene u radu:



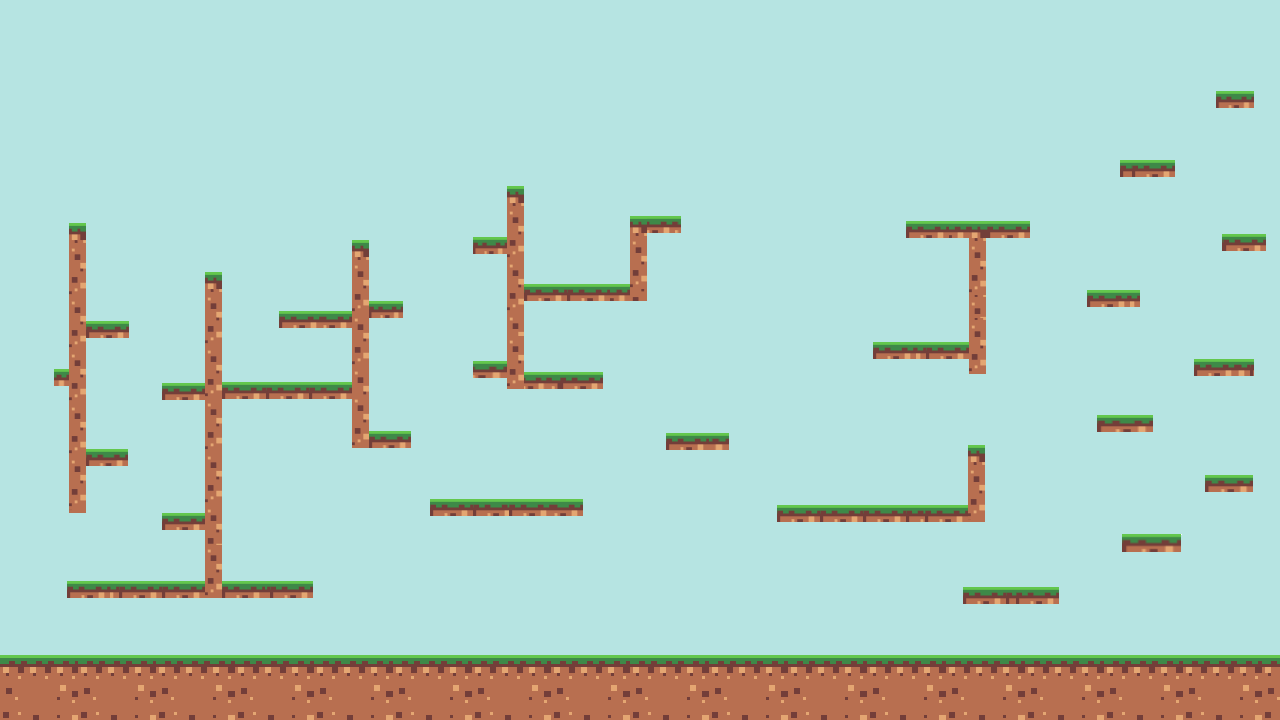
Slika 1. Karakter „Monkey“



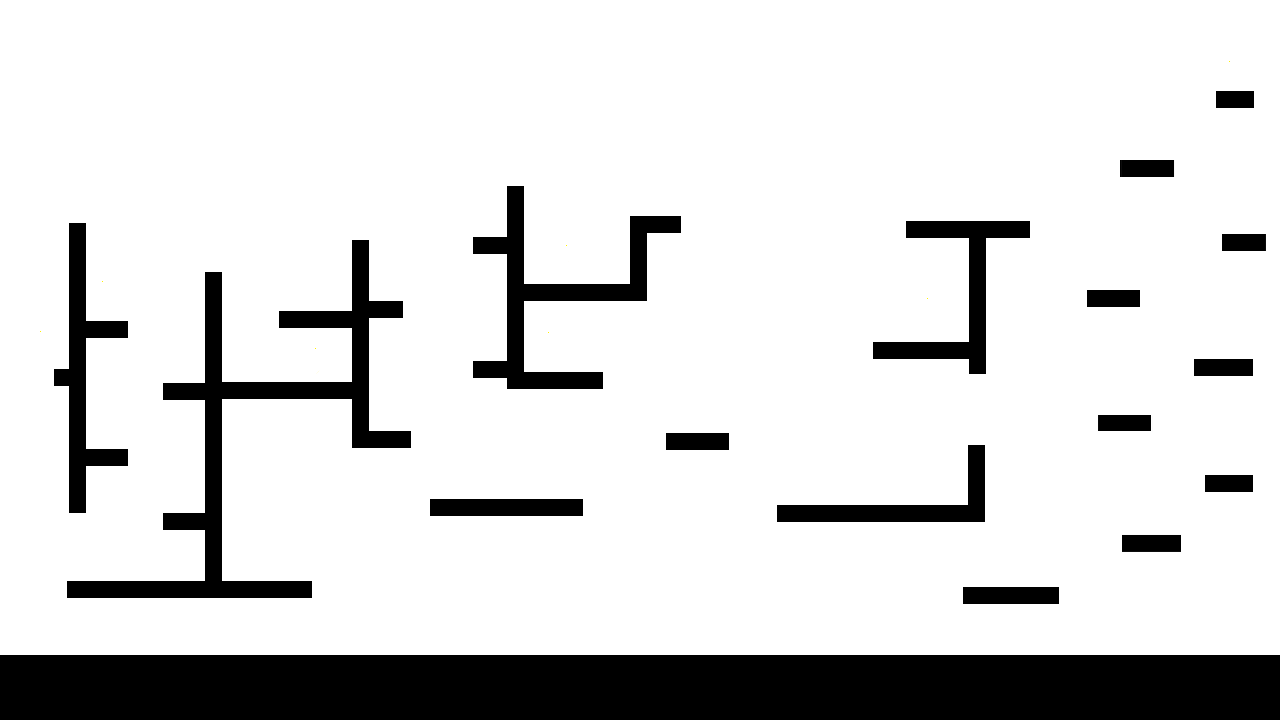
Slika 2. Karakter „Froggy“



Slika 3. Banana



Slika 4. Izgled nivoa



Slika 5. Slika za generisanje platformi u nivou

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.Color;  **import** java.awt.Graphics;  **import** java.awt.image.BufferedImage;  **import** java.io.File;  **import** java.io.IOException;  **import** javax.swing.JLabel;  **import** javax.swing.filechooser.FileSystemView;  **import** org.eclipse.swt.graphics.Image;  **public** **class** GameMethods {    **public** **static** Color[][] UcitajMatricuBoja(BufferedImage slika) **throws** IOException {  **int** visina = slika.getHeight();  **int** sirina = slika.getWidth();  Color[][] colors = **new** Color[visina][sirina];  **for** (**int** x = 0; x < visina; x++) {  **for** (**int** y = 0; y < sirina; y++) {  colors[x][y] = **new** Color(slika.getRGB(y, x), **true**);  }  }  **return** colors;  }    **public** **static** **void** RotirajIgraca(Player igrac)  {  **for** (**int** i = 0; i < igrac.Height; i++)  {  **for** (**int** k = igrac.Width-1, j=0; k>=igrac.Width/2; k--,j++)  {  Color boja;  boja=igrac.matricaBoja[i][j];  igrac.matricaBoja[i][j]=igrac.matricaBoja[i][k];  igrac.matricaBoja[i][k]=boja;  }  }  **for**(**int** i=0;i<igrac.Height;i++)  {  **for**(**int** j=0;j<igrac.Width;j++)  {  Color boja;  boja=igrac.matricaBoja[i][j];  igrac.slika.setRGB(j, i, boja.getRGB());  }  }  }    **public** **static** String VratiDesktop()  {  File home = FileSystemView.*getFileSystemView*().getHomeDirectory();  **return** home.getAbsolutePath();  }    } |

Tabela 1. GameMethods.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.BorderLayout;  **import** java.awt.EventQueue;  **import** java.awt.image.BufferedImage;  **import** java.io.IOException;  **import** java.util.Timer;  **import** java.util.TimerTask;  **import** javax.swing.JFrame;  **import** javax.swing.JLabel;  **import** javax.swing.JPanel;  **import** javax.swing.border.EmptyBorder;  **import** javax.imageio.ImageIO;  **import** javax.swing.ImageIcon;  **import** javax.swing.JButton;  **import** java.awt.Color;  **import** java.awt.event.KeyAdapter;  **import** java.awt.event.KeyEvent;  **import** java.awt.event.KeyListener;  **import** javax.swing.SwingConstants;  **import** java.awt.Font;  **import** java.awt.Toolkit;  **public** **class** GameWindowFrame **extends** JFrame **implements** KeyListener {  GlavniMeni glavniMeni;  String ime;  String karakter;    **int** SekundeOdPocetka=0;  **int** MiliSekunde=0;  JLabel lblTime;    //Prozor  **private** JPanel contentPane;  BufferedImage img;  BufferedImage imgObjekti;  BufferedImage pozadina;  JLabel lbl;  **int** SirinaPozadine=1280;  **int** VisinaPozadine=720;  **int**[][] matricaIgre = **new** **int**[720][1280];    //Banane  Banana[] banane;  **int** brojBanana=0;  JLabel[] lbBanane;  **int** BananaRadius=12;  **int** brojPokupljenihBanana=0;    //Vrednosti za matricu igre  **int** Platforma=3;  **int** Banana=2;  **int** Igrac=1;  **int** Nista=0;    //Gravitacija  **int** BaseGravitacija=3;  **int** Gravitacija=3;    //Igrac  Player igrac;  JLabel lbSlikaIgraca;    **private** JLabel lblBananaCountIcon;  JLabel lblBananaCount;  /\*\*  \* Launch the application.  \*/  /\*\*  \* Create the frame.  \* **@throws** IOException  \*/    Timer timerVreme=**new** Timer();  TimerTask vreme=**new** TimerTask() {  **public** **void** run() {  **int** minute=SekundeOdPocetka/60;  String sMinute=String.*valueOf*(minute);  **if**(minute<10)  {  sMinute="0"+sMinute;  }  **int** seconds=SekundeOdPocetka%60;  String sSeconds=String.*valueOf*(seconds);  **if**(seconds<10)  {  sSeconds="0"+sSeconds;  }      lblTime.setText(sMinute+":"+sSeconds);  System.***out***.println("Proslo je "+ SekundeOdPocetka++ +" s");  }  };      Timer timer=**new** Timer();  TimerTask updateFrame=**new** TimerTask(){  **public** **void** run() {  //ovde ide kod za updatovanje frame, poziv pomeranja  // GameMethods.NacrtajIgraca(img, igrac);      PomeriIgraca();  VertikalnoPomeranjeKaraktera();  ProveriDaLiSamNaBanani();  // igrac.OkreniKaraktera();  lbSlikaIgraca.setIcon(**new** ImageIcon(igrac.slika));  lbSlikaIgraca.setLocation(igrac.xCoordinate, igrac.yCoordinate);    // System.out.print(igrac.kretanje);    }    };    **public** **void** Start() {  timer.schedule(updateFrame, 0, 16);  timerVreme.schedule(vreme, 0, 1000);  }      **public** GameWindowFrame(String Ime,String Karakter,GlavniMeni meni) **throws** IOException {  setIconImage(Toolkit.*getDefaultToolkit*().getImage(GameWindowFrame.**class**.getResource("/Slike/Banana.png")));  setTitle("Monkey Jump");  setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  setBounds(0, 0,1296, 759);  contentPane = **new** JPanel();  **this**.addKeyListener(**this**);  **this**.setResizable(**false**);  contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(**null**);  setLocationRelativeTo(**null**);  lblTime = **new** JLabel("15:20");  lblTime.setFont(**new** Font("Tahoma", Font.***PLAIN***, 43));  lblTime.setHorizontalAlignment(SwingConstants.***CENTER***);  lblTime.setForeground(Color.***WHITE***);  lblTime.setBounds(566, 39, 138, 44);  contentPane.add(lblTime);  ime=Ime;  karakter=Karakter;  **this**.glavniMeni = meni;    //Igrac  igrac=**new** Player(Karakter,600,600,5);  ImageIcon igracSlika= **new** ImageIcon(igrac.slika);    lblBananaCount = **new** JLabel("0/7");  lblBananaCount.setHorizontalAlignment(SwingConstants.***CENTER***);  lblBananaCount.setForeground(Color.***YELLOW***);  lblBananaCount.setFont(**new** Font("Tahoma", Font.***PLAIN***, 36));  lblBananaCount.setBounds(850, 39, 80, 44);  contentPane.add(lblBananaCount);    lblBananaCountIcon = **new** JLabel();  lblBananaCountIcon.setHorizontalAlignment(SwingConstants.***CENTER***);  lblBananaCountIcon.setForeground(Color.***YELLOW***);  lblBananaCountIcon.setFont(**new** Font("Tahoma", Font.***PLAIN***, 36));  lblBananaCountIcon.setBounds(900, 20, 90, 90);  BufferedImage slikaBanane;  slikaBanane = ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/Banana.png"));  ImageIcon bananaIcon=**new** ImageIcon(slikaBanane);  lblBananaCountIcon.setIcon(bananaIcon);  contentPane.add(lblBananaCountIcon);        lbSlikaIgraca=**new** JLabel(igracSlika,JLabel.***CENTER***);  lbSlikaIgraca.setBounds(igrac.xCoordinate, igrac.yCoordinate, igrac.Height+10, igrac.Width);        // lbSlikaIgraca.setIcon(igracSlika);  contentPane.add(lbSlikaIgraca);  UcitajSlikuPozadine("Background");  NapraviPozadinu();  GenerisiMatricuIgre("MatrixBackground");  //Banane  DobiBrojBanana();  banane=**new** Banana[brojBanana];  GenerisiBanane();    lbBanane=**new** JLabel[brojBanana];  **for**(**int** i=0;i<brojBanana;i++)  {  lbBanane[i]=**new** JLabel();  ImageIcon icon=**new** ImageIcon(banane[i].slika);  lbBanane[i].setIcon(icon);  lbBanane[i].setBounds(banane[i].xCoordinate, banane[i].yCoordinate, banane[i].Width, banane[i].Height);  contentPane.add(lbBanane[i]);  }      //Pozadina  lbl=**new** JLabel();  lbl.setBounds(0, 0, SirinaPozadine, VisinaPozadine);  contentPane.add(lbl);  // UcitajSlikuPozadine("Background");    PostaviPozadinu();  GenerisiOkvirPozadine(5);  Start();  }  **public** **void** ProveriDaLiSamNaBanani() {  **int** aktivnaBananaIndex=-1;  **for**(**int** i=0;i<igrac.Height;i++)  {  **for**(**int** j=0;j<igrac.Width;j++)  {  **if**(matricaIgre[igrac.yCoordinate+i][igrac.xCoordinate+j]==Banana)  {  **for**(**int** k=0;k<brojBanana;k++)  {  **if**(Math.*abs*(banane[k].xCoordinate-igrac.xCoordinate)<2\*BananaRadius && Math.*abs*(banane[k].yCoordinate-igrac.yCoordinate)<2\*BananaRadius)  {  aktivnaBananaIndex=k;  }  }    }  }  }  **if**(aktivnaBananaIndex!=-1)  **try** {  PokupiBananu(aktivnaBananaIndex);  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }    **public** **void** PokupiBananu(**int** IndexBanane) **throws** IOException  {  brojPokupljenihBanana++;  lblBananaCount.setText(brojPokupljenihBanana+"/"+brojBanana);  lbBanane[IndexBanane].setVisible(**false**);  **for**(**int** i=0;i<banane[IndexBanane].Height+2\*BananaRadius;i++)  {  **for**(**int** j=0;j<banane[IndexBanane].Width+2\*BananaRadius;j++)  {  **if**(matricaIgre[banane[IndexBanane].yCoordinate-BananaRadius+i][banane[IndexBanane].xCoordinate-BananaRadius+j]==Banana) {  matricaIgre[banane[IndexBanane].yCoordinate-BananaRadius+i][banane[IndexBanane].xCoordinate-BananaRadius+j]=Nista;    }      }  }  banane[IndexBanane].Pokupljena=**true**;  **if**(brojPokupljenihBanana==brojBanana)  {  timerVreme.cancel();  timerVreme.purge();  **try** {  MeniPosleIgre meni=**new** MeniPosleIgre(ime,lblTime.getText(),karakter,glavniMeni);  meni.~~show~~();  **this**.dispose();  }  **catch**(Exception ex)  {  ex.printStackTrace();  }  }  }      **public** **void** UcitajSlikuPozadine(String nazivSlike) **throws** IOException {    pozadina = ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/"+ nazivSlike +".png"));    }  **public** **void** NapraviPozadinu()  {    img = **new** BufferedImage(1280, 720,BufferedImage.***TYPE\_INT\_RGB***);    **for**(**int** i=0;i<img.getHeight();i++)  **for**(**int** j=0;j<img.getWidth();j++)  {    img.setRGB(j, i, pozadina.getRGB(j, i));  }    }    **public** **void** GenerisiMatricuIgre(String nazivSlike) **throws** IOException  {  imgObjekti = ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/"+ nazivSlike +".png"));  **for**(**int** i=0;i<VisinaPozadine;i++)  {  **for**(**int** j=0;j<SirinaPozadine;j++)  {  //System.out.println(imgObjekti.getRGB(j, i));  **if**(imgObjekti.getRGB(j, i)==-16777216)  {  matricaIgre[i][j]=Platforma;  }    }  }  }    **public** **void** GenerisiOkvirPozadine(**int** Debljina)  {  **int** visina=img.getHeight();  **int** sirina=img.getWidth();  **for**(**int** i=0; i<visina; i++)  {  **for**(**int** j=0; j<sirina; j++)  {  **if**(i<Debljina || i>visina-Debljina || j<Debljina || j>sirina-Debljina)  {  img.setRGB(j, i, Color.***GREEN***.getRGB());  matricaIgre[i][j]=3;  }  }  }  }  **public** **void** DobiBrojBanana()  {    **for**(**int** i=0; i<VisinaPozadine; i++)  {  **for**(**int** j=0; j<SirinaPozadine; j++)  {  //-4690096 -1792398  **if**(imgObjekti.getRGB(j, i)==-3584)  {  brojBanana++;  }  //System.out.println(imgObjekti.getRGB(j,i));  }  }  }    **public** **void** GenerisiBanane() **throws** IOException  {    **int** tempBanane=0;  **for**(**int** i=0; i<VisinaPozadine; i++)  {  **for**(**int** j=0; j<SirinaPozadine; j++)  {  //-4690096 -1792398  **if**(imgObjekti.getRGB(j, i)==-3584)  {  **for**(**int** n=i-BananaRadius;n<i+BananaRadius;n++)  {  **for**(**int** m=j-BananaRadius;m<j+BananaRadius;m++)  matricaIgre[n][m]=2;  }    banane[tempBanane]=**new** Banana(j,i);  tempBanane++;  }  //System.out.println(imgObjekti.getRGB(j,i));  }  }  }  **public** **void** PostaviIgraca()  {  ImageIcon icon=**new** ImageIcon(igrac.slika);  lbSlikaIgraca.setIcon(icon);  }    **public** **void** PostaviPozadinu()  {  ImageIcon icon=**new** ImageIcon(img);  lbl.setIcon(icon);  }    @Override  **public** **void** keyTyped(KeyEvent e) {  // **TODO** Auto-generated method stub    }  @Override  **public** **void** keyPressed(KeyEvent e) {  // **TODO** Auto-generated method stub  **if**(e.getKeyChar()=='A' || e.getKeyChar()=='a' || e.getKeyCode()==e.***VK\_LEFT***)  {  **if**(igrac.kretanje>-1)  {  igrac.kretanje-=1;  //igrac.GledaDesno=false;  }  }  **else** **if**(e.getKeyChar()=='D' || e.getKeyChar()=='d' || e.getKeyCode()==e.***VK\_RIGHT***)  {  **if**(igrac.kretanje<1)  {  igrac.kretanje+=1;  // igrac.GledaDesno=true;  }    }  **else** **if**((e.getKeyCode()==e.***VK\_SPACE*** || e.getKeyCode()==e.***VK\_UP*** || e.getKeyCode()==e.***VK\_W***) && igrac.mozeDaSkace==**true**)  {  igrac.mozeDaSkace=**false**;  igrac.jacinaSkoka=17;  }  **else** **if**(e.getKeyCode()==e.***VK\_ESCAPE***)  {  glavniMeni.setVisible(**true**);  **this**.dispose();  }  }  @Override  **public** **void** keyReleased(KeyEvent e) {  // **TODO** Auto-generated method stub  **if**(e.getKeyChar()=='A' || e.getKeyChar()=='a' || e.getKeyCode()==e.***VK\_LEFT***)  {  **if**(igrac.kretanje<1)  igrac.kretanje+=1;  }  **else** **if**(e.getKeyChar()=='D' || e.getKeyChar()=='d' || e.getKeyCode()==e.***VK\_RIGHT***)  {  **if**(igrac.kretanje>-1)  igrac.kretanje-=1;  }  }      **public** **void** PomeriIgraca()  {  **if**(igrac.kretanje>0)  {  //Pixel po pixel pomeranje, za precizniju detekciju  **for**(**int** i=1;i<=igrac.brzina;i++)  { //Pixel po pixel levo od karaktera ali za svaki pixel u koloni visine  **boolean** mozeDaSePomeri=**true**;  **for**(**int** k=0;k<igrac.Height;k++)  {    **if**(matricaIgre[igrac.yCoordinate+k][igrac.xCoordinate+igrac.Width+1]==Platforma)  mozeDaSePomeri=**false**;  }  **if**(mozeDaSePomeri==**true**)  igrac.xCoordinate+=1;  }      }    **else** **if**(igrac.kretanje<0)  {  **for**(**int** i=1;i<=igrac.brzina;i++)  { //Pixel po pixel levo od karaktera ali za svaki pixel u koloni visine  **boolean** mozeDaSePomeri=**true**;  **for**(**int** k=0;k<igrac.Height;k++)  {    **if**(matricaIgre[igrac.yCoordinate+k][igrac.xCoordinate-1]==Platforma)  mozeDaSePomeri=**false**;  }  **if**(mozeDaSePomeri==**true**)  igrac.xCoordinate-=1;  }    }    igrac.OkreniKaraktera();  }    **public** **void** VertikalnoPomeranjeKaraktera()  {  **if**(Gravitacija>=igrac.jacinaSkoka)  {  **for**(**int** i=1;i<=Gravitacija-igrac.jacinaSkoka;i++)  {  **boolean** mozeDaPadne=**true**;  **for**(**int** k=0;k<igrac.Width;k++)  {    **if**(matricaIgre[igrac.yCoordinate+igrac.Height+1][igrac.xCoordinate+k]==Platforma)  {  mozeDaPadne=**false**;  igrac.mozeDaSkace=**true**;  Gravitacija=BaseGravitacija;  }  **else**  {  //igrac.mozeDaSkace=false;  }  }  **if**(mozeDaPadne==**true**)  igrac.yCoordinate=igrac.yCoordinate+1;    }  Gravitacija+=1;  }  **else** **if**(igrac.jacinaSkoka>Gravitacija)  {  **for**(**int** i=1;i<=igrac.jacinaSkoka-Gravitacija;i++)  {  **boolean** mozeDaIdeGore=**true**;  **for**(**int** k=0;k<igrac.Width;k++)  {    **if**(matricaIgre[igrac.yCoordinate-1][igrac.xCoordinate+k]==Platforma)  {  mozeDaIdeGore=**false**;  igrac.jacinaSkoka=0;  }    }  **if**(mozeDaIdeGore==**true**)  igrac.yCoordinate=igrac.yCoordinate-1;  }  igrac.jacinaSkoka--;  }    }  } |

Tabela 2. GameWindowFrame.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.BorderLayout;  **import** java.awt.EventQueue;  **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **import** javax.swing.JScrollPane;  **import** javax.swing.border.EmptyBorder;  **import** java.awt.Color;  **import** java.awt.Dimension;  **import** javax.swing.JButton;  **import** javax.swing.JTextField;  **import** javax.swing.JLabel;  **import** javax.swing.JTable;  **import** javax.swing.SwingConstants;  **import** java.awt.Font;  **import** javax.swing.JToggleButton;  **import** javax.swing.JComboBox;  **import** javax.swing.DefaultComboBoxModel;  **import** java.awt.event.ActionListener;  **import** java.io.File;  **import** java.io.IOException;  **import** java.io.InputStream;  **import** java.awt.event.ActionEvent;  **import** javax.swing.JSlider;  **import** java.awt.Toolkit;  **import** javax.swing.border.LineBorder;  **import** javax.swing.table.AbstractTableModel;  **import** javax.swing.table.JTableHeader;  **import** javax.swing.table.TableColumn;  **import** javax.swing.table.TableColumnModel;  **import** javax.swing.table.TableModel;  **import** javax.sound.sampled.Clip;  **import** javax.sound.sampled.FloatControl;  **import** org.eclipse.osgi.container.Module.Settings;  **import** javax.sound.sampled.\*;  **import** java.net.MalformedURLException;  **import** java.net.URL;  **import** javax.swing.event.ChangeListener;  **import** javax.swing.event.ChangeEvent;  **import** java.awt.FlowLayout;  **public** **class** GlavniMeni **extends** JFrame {  **private** JPanel contentPane;  **private** JTextField txtIme;  **private** JTable tableScores;  **private** JButton btnZapocni;  **private** Clip clip;  **private** **int** vol;  **private** JPanel panel;    /\*\*  \* Launch the application.  \*/  **public** **static** **void** main(String[] args) {  EventQueue.*invokeLater*(**new** Runnable() {  **public** **void** run() {  **try** {  GlavniMeni frame = **new** GlavniMeni();  frame.setVisible(**true**);  } **catch** (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\*  \* Create the frame.  \*/  **public** GlavniMeni() **throws** Exception {  setTitle("Monkey Jump");  setIconImage(Toolkit.*getDefaultToolkit*().getImage(GlavniMeni.**class**.getResource("/Slike/Banana.png")));  GlavniMeni meni=**this**;  setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  setBounds(100, 100, 720, 480);  contentPane = **new** JPanel();  contentPane.setBackground(**new** Color(228, 209, 146));  contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(**null**);  setLocationRelativeTo(**null**);    JButton btnExit = **new** JButton("Exit");  btnExit.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  System.*exit*(***DISPOSE\_ON\_CLOSE***);  }  });  btnExit.setBounds(568, 390, 114, 31);  contentPane.add(btnExit);    txtIme = **new** JTextField();  txtIme.setBounds(500, 110, 114, 20);  contentPane.add(txtIme);  txtIme.setColumns(10);    JLabel lbl\_Ime = **new** JLabel("Ime");  lbl\_Ime.setFont(**new** Font("Tahoma", Font.***PLAIN***, 16));  lbl\_Ime.setHorizontalAlignment(SwingConstants.***TRAILING***);  lbl\_Ime.setBounds(426, 110, 46, 20);  contentPane.add(lbl\_Ime);    panel = **new** JPanel();    FlowLayout flowLayout = (FlowLayout) panel.getLayout();  flowLayout.setAlignment(FlowLayout.***LEFT***);  flowLayout.setVgap(0);  flowLayout.setHgap(0);  panel.setBackground(**new** Color(228, 209, 146));  panel.setBounds(70, 110, 229, 272);  contentPane.add(panel);    PopuniTabelu();    JComboBox cmbKarakter = **new** JComboBox();  cmbKarakter.setModel(**new** DefaultComboBoxModel(**new** String[] {"Monkey", "Froggy"}));  cmbKarakter.setBounds(500, 180, 116, 22);  contentPane.add(cmbKarakter);    btnZapocni = **new** JButton("Započni Igru");  btnZapocni.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  String Ime=txtIme.getText();  **if**(txtIme.getText().isEmpty())  Ime="Guest";  **try** {  GameWindowFrame igra= **new** GameWindowFrame(Ime,cmbKarakter.getSelectedItem().toString(),meni);  igra.~~show~~();  meni.setVisible(**false**);  } **catch** (IOException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }    }  });  btnZapocni.setBounds(500, 320, 114, 31);  contentPane.add(btnZapocni);    JSlider slider = **new** JSlider(-30, 0, 0);  slider.setPaintTicks(**true**);  slider.addChangeListener(**new** ChangeListener() {  **public** **void** stateChanged(ChangeEvent e) {  vol=(slider.getValue());  **if**(vol==-30) vol=-80;  PromenaZvuka();  }  });  slider.setBorder(**new** LineBorder(Color.***LIGHT\_GRAY***, 1, **true**));  slider.setForeground(Color.***BLACK***);  slider.setMinorTickSpacing(1);  slider.setMajorTickSpacing(1);  slider.setSnapToTicks(**true**);  slider.setBounds(500, 250, 182, 40);  slider.setFont(**new** Font("Tahoma", Font.***PLAIN***, 9));  contentPane.add(slider);  Muzika();    JLabel lbNaslov = **new** JLabel("Monkey Jump");  lbNaslov.setForeground(**new** Color(184, 134, 11));  lbNaslov.setHorizontalAlignment(SwingConstants.***CENTER***);  lbNaslov.setFont(**new** Font("Sylfaen", Font.***BOLD*** | Font.***ITALIC***, 40));  lbNaslov.setBounds(206, 11, 279, 109);  contentPane.add(lbNaslov);    JLabel lbl\_Volume = **new** JLabel("Volume");  lbl\_Volume.setBounds(410, 250, 60, 39);  lbl\_Volume.setHorizontalAlignment(SwingConstants.***TRAILING***);  lbl\_Volume.setFont(**new** Font("Tahoma", Font.***PLAIN***, 16));  contentPane.add(lbl\_Volume);    JLabel lbl\_Karakter = **new** JLabel("Karakter");  lbl\_Karakter.setHorizontalAlignment(SwingConstants.***TRAILING***);  lbl\_Karakter.setFont(**new** Font("Tahoma", Font.***PLAIN***, 16));  lbl\_Karakter.setBounds(400, 180, 72, 22);  contentPane.add(lbl\_Karakter);  }    **public** **void** PopuniTabelu()  {  panel.removeAll();  PripremaTabele model = **new** PripremaTabele();  JTable table=**new** JTable();  table.setModel(model);  table.setBounds(0, 0, 150, 300);  table.setPreferredScrollableViewportSize(**new** Dimension(150,250));    JTableHeader thead= table.getTableHeader();  TableColumnModel tcm = table.getColumnModel();  TableColumn tc = tcm.getColumn(0);  JScrollPane scrollpane = **new** JScrollPane(table);  scrollpane.setBounds(0,0,150,300);  tc.setHeaderValue("Ime");  tc=tcm.getColumn(1);  tc.setHeaderValue("Vreme");  panel.add(scrollpane);  }    **void** Muzika() **throws** Exception {  URL url=getClass().getResource("/Muzika/StringAdagio-ClassicStringBalladInDorianMood.wav");  AudioInputStream muzika = AudioSystem.*getAudioInputStream*(url);  clip=AudioSystem.*getClip*();  clip.open(muzika);  clip.setFramePosition(0);  clip.start();  clip.loop(Clip.***LOOP\_CONTINUOUSLY***);  }    **void** PromenaZvuka() {  FloatControl gainControl = (FloatControl) clip.getControl(FloatControl.Type.***MASTER\_GAIN***);  gainControl.setValue(vol);  }    } |

Tabela 3. GlavniMeni.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.BorderLayout;  **import** java.awt.FlowLayout;  **import** javax.swing.JButton;  **import** javax.swing.JDialog;  **import** javax.swing.JPanel;  **import** javax.swing.border.EmptyBorder;  **import** java.awt.Color;  **import** javax.swing.JLabel;  **import** java.awt.Font;  **import** javax.swing.SwingConstants;  **import** java.awt.event.ActionListener;  **import** java.io.BufferedReader;  **import** java.io.BufferedWriter;  **import** java.io.FileReader;  **import** java.io.FileWriter;  **import** java.io.IOException;  **import** java.awt.event.ActionEvent;  **import** java.awt.Toolkit;  **public** **class** MeniPosleIgre **extends** JDialog {  **private** GlavniMeni glavniMeni;  **private** **final** JPanel contentPanel = **new** JPanel();  **private** String vreme;  **private** String ime;  **private** String karakter;  MeniPosleIgre meni;  /\*\*  \* Launch the application.  \*/    /\*\*  \* Create the dialog.  \* **@throws** IOException  \*/  **public** **void** initComponents() {  {  meni=**this**;  JButton okButton = **new** JButton("Igraj Ponovo");  okButton.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  GameWindowFrame igra;  **try** {  igra = **new** GameWindowFrame(ime,karakter,glavniMeni);  igra.~~show~~();  meni.dispose();  } **catch** (IOException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }    }  });  okButton.setBounds(148, 171, 141, 23);  contentPanel.add(okButton);  okButton.setActionCommand("OK");  getRootPane().setDefaultButton(okButton);  }  {  JButton cancelButton = **new** JButton("Nazad U Meni");  cancelButton.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  glavniMeni.PopuniTabelu();  glavniMeni.setVisible(**true**);  meni.dispose();    }  });  cancelButton.setBounds(148, 214, 141, 23);  contentPanel.add(cancelButton);  cancelButton.setActionCommand("Cancel");  }    JButton btnSacuvajRez = **new** JButton("Sacuvaj Rezultat");    btnSacuvajRez.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  **try** {    BufferedReader reader= **new** BufferedReader(**new** FileReader(GameMethods.*VratiDesktop*()+"/Vremena.txt"));  **int** numberOfLines=1;  String line=reader.readLine();    **while**(line!=**null**)  {  numberOfLines++;  line=reader.readLine();  }    reader.close();  **if**(numberOfLines==2) {    FileWriter fWstream = **new** FileWriter(GameMethods.*VratiDesktop*()+"/Vremena.txt");  fWstream.write("Ime|Vreme"+System.*lineSeparator*());  fWstream.write(ime+"|"+vreme+System.*lineSeparator*());  fWstream.close();  }  **else** {  reader=**new** BufferedReader(**new** FileReader(GameMethods.*VratiDesktop*()+"/Vremena.txt"));  String[] lines=**new** String[numberOfLines];  line=reader.readLine();  **int** i=0;  line=reader.readLine();  **while**(line!=**null**)  {  lines[i]=line;  line=reader.readLine();  i++;  }    String[] vremena=vreme.split(":");    **int** timeInMiliSeconds=Integer.*parseInt*(vremena[0])\*60 + Integer.*parseInt*(vremena[1]);    **boolean** napisan=**false**;  String vremeTeksta;  String imenaTeksta;  String[] vremenaTeksta;  **int** timeInMiliSecondsTeksta=0;  FileWriter fWstream = **new** FileWriter(GameMethods.*VratiDesktop*()+"/Vremena.txt");  fWstream.write("Ime|Vreme"+System.*lineSeparator*());  **for**(**int** k=0;k<numberOfLines-2;k++)  {    String[] vremeIzTeksta=lines[k].split("[|]");  imenaTeksta=vremeIzTeksta[0].trim();    vremeTeksta=vremeIzTeksta[1].trim();    vremenaTeksta=vremeTeksta.split(":");    timeInMiliSecondsTeksta=Integer.*parseInt*(vremenaTeksta[0])\*60 + Integer.*parseInt*(vremenaTeksta[1]);    System.***out***.println("Vreme za "+imenaTeksta+" je: " +timeInMiliSecondsTeksta+ " a vreme igraca je: " +timeInMiliSeconds);    **if**(timeInMiliSeconds<=timeInMiliSecondsTeksta && napisan==**false**)  {  fWstream.write(ime+"|"+vreme+System.*lineSeparator*());  fWstream.write(imenaTeksta+"|"+vremeTeksta+System.*lineSeparator*());  napisan=**true**;  }  **else**  {  fWstream.write(imenaTeksta+"|"+vremeTeksta+System.*lineSeparator*());  }    }  **if**(napisan==**false**)  fWstream.write(ime+"|"+vreme+System.*lineSeparator*());  fWstream.close();      }    btnSacuvajRez.setEnabled(**false**) ;  }  **catch**(Exception ex)  {  ex.printStackTrace();  }  }  });  btnSacuvajRez.setBounds(148, 127, 141, 23);  contentPanel.add(btnSacuvajRez);  btnSacuvajRez.setBounds(148, 127, 141, 23);  contentPanel.add(btnSacuvajRez);      {  JLabel lbVreme = **new** JLabel("Vaše vreme je:");  lbVreme.setForeground(Color.***BLACK***);  lbVreme.setHorizontalAlignment(SwingConstants.***CENTER***);  lbVreme.setFont(**new** Font("Tahoma", Font.***PLAIN***, 36));  lbVreme.setBounds(67, 11, 302, 44);  contentPanel.add(lbVreme);  }  {  JLabel lblVreme = **new** JLabel("vreme");  lblVreme.setHorizontalAlignment(SwingConstants.***CENTER***);  lblVreme.setForeground(Color.***BLACK***);  lblVreme.setFont(**new** Font("Tahoma", Font.***PLAIN***, 36));  lblVreme.setBounds(67, 56, 302, 44);  lblVreme.setText(vreme);  contentPanel.add(lblVreme);  }  }    **public** MeniPosleIgre(String ImeIgraca,String Vreme,String Karakter,GlavniMeni meni) **throws** IOException {  setBackground(**new** Color(255, 218, 185));  setIconImage(Toolkit.*getDefaultToolkit*().getImage(MeniPosleIgre.**class**.getResource("/Slike/Banana.png")));  setTitle("Monkey Jump");  setBounds(100, 100, 450, 300);  getContentPane().setLayout(**null**);  contentPanel.setBackground(**new** Color(255, 218, 185));  contentPanel.setBounds(0, 0, 434, 261);  contentPanel.setBorder(**new** EmptyBorder(5, 5, 5, 5));  getContentPane().add(contentPanel);  setLocationRelativeTo(**null**);  contentPanel.setLayout(**null**);  ime=ImeIgraca;  karakter=Karakter;  **this**.vreme=Vreme;  **this**.glavniMeni = meni;  initComponents();  }  } |

Tabela 4. MeniPosleIgre.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.Color;  **import** java.awt.Toolkit;  **import** java.awt.image.BufferedImage;  **import** java.io.File;  **import** java.io.IOException;  **import** javax.imageio.ImageIO;  **import** javax.print.DocFlavor.URL;  **public** **class** Player {  **int** xCoordinate;  **int** yCoordinate;  **int** Width=0;  **int** Height=0;  BufferedImage slika;  **int** animProgress;  Color[][] matricaBoja;        **boolean** gledaDesno;  **boolean** daLiSeKrece;  **boolean** mozeDaSkace;    **int** brzina;  **int** visinaSkoka;  **int** jacinaSkoka;  **int** kretanje;    **public** Player(String nazivSlike,**int** x,**int** y,**int** Brzina) **throws** IOException  {  UcitajSlikuKaraktera(nazivSlike);  getHeight();  getWidth();  matricaBoja=GameMethods.*UcitajMatricuBoja*(slika);  xCoordinate=x;  yCoordinate=y;  gledaDesno=**true**;  daLiSeKrece=**false**;  **this**.brzina=Brzina;  kretanje=0;  getHeight();  getWidth();  mozeDaSkace=**false**;  jacinaSkoka=0;  visinaSkoka=60;  }  **public** **void** getHeight()  {  Height=slika.getHeight();  }  **public** **void** getWidth()  {  Width=slika.getWidth();  }      **public** **void** UcitajSlikuKaraktera(String nazivSlike) **throws** IOException  {  slika=ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/"+ nazivSlike +".png"));  }    **public** **void** OkreniKaraktera()  {  **if**(kretanje>0)  {  **if**(gledaDesno==**false**)  {  GameMethods.*RotirajIgraca*(**this**);  gledaDesno=**true**;  }    }  **else** **if**(kretanje<0)  {  **if**(gledaDesno==**true**)  {  GameMethods.*RotirajIgraca*(**this**);  gledaDesno=**false**;  }  }  }  } |

Tabela 5. Player.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.image.BufferedImage;  **import** java.io.File;  **import** java.io.IOException;  **import** javax.imageio.ImageIO;  **public** **class** Banana {      **public** **int** xCoordinate;  **public** **int** yCoordinate;  **public** BufferedImage slika;  **public** **boolean** Pokupljena;  **public** **int** Width;  **public** **int** Height;    **public** Banana(**int** x,**int** y) **throws** IOException  {  xCoordinate=x;  yCoordinate=y;  UcitajSlikuBanane();  Width=slika.getWidth();  Height=slika.getHeight();  Pokupljena=**false**;    }    **public** **void** UcitajSlikuBanane() **throws** IOException {  slika = ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/Banana.png"));  }  } |

Tabela 6. Banana.java

|  |
| --- |
| **package** pilipovicd;  **import** java.awt.image.BufferedImage;  **import** java.io.File;  **import** java.io.IOException;  **import** javax.imageio.ImageIO;  **public** **class** Banana {  **public** **int** xCoordinate;  **public** **int** yCoordinate;  **public** BufferedImage slika;  **public** **boolean** Pokupljena;  **public** **int** Width;  **public** **int** Height;    **public** Banana(**int** x,**int** y) **throws** IOException  {  xCoordinate=x;  yCoordinate=y;  UcitajSlikuBanane();  Width=slika.getWidth();  Height=slika.getHeight();  Pokupljena=**false**;    }    **public** **void** UcitajSlikuBanane() **throws** IOException {  slika = ImageIO.*read*(Player.**class**.getResourceAsStream("/Slike/Banana.png"));  }  } |

Tabela 7. PripremaTabele.java

# **Zaključak**

Igrica je takvog tipa da podstiče igrača da je završi u što kraćem vremenskom periodu, i time navodi igrača da ponovo odigra igricu u cilju ostvarivanja boljeg rezultata.

# **Literatura**

1. <http://www.tfzr.rs/Predmet/Projektovanje%20softvera/--2>
2. <http://www.tfzr.rs/Predmet/Projektovanje%20softvera/--1>
3. <https://pixabay.com/de/music/klassisches-streichquartett-string-adagio-classic-string-ballad-in-dorian-mood-2075/>