***The Game Of Minion***

This is an adventure game. There Are total 15 classes in this project. For our project, we have used **GUI**, **filesystem** and **OOP.**

|  |
| --- |
| Main |
| -TGOM:JFrame |
| +Main( )  -initialize( ) : void  +getMain( ) : JFrame  +setMain(JFrame) :void  +getTGOM( ) : JFrame  +setTGOM(JFrame) :void |

|  |
| --- |
| HighScore |
| -highframe : JFrame  -dname : JTextField |
| +HighScore( )  -initialize( ) : void  +getHighframe( ) : JFrame  +setHighframe(JFrame) : void |

|  |
| --- |
| Help |
| -hpframe : JFrame |
| -initialize( ) : void  +getHpframe( ) ; JFrame  +setHpframe(JFrame) : void |

|  |
| --- |
| Credits |
| -cframe : JFrame |
| +Credits( )  -initialize( ) : void  +getCframe( ) : JFrame  +setCframe(JFrame) : void |

|  |
| --- |
| GameWindow |
| -gameScreen : GameScreen |

|  |
| --- |
| Animation |
| -list : List<BufferedImage>  -deltatime : long  -currentFrame : int  -previousTime : long |
| +Animation(int)  +updateFrame( ) :void  +addFrame(BufferedImage) : void  +getFrame( ) : BufferedImage |

|  |
| --- |
| GameScreen |
| -START\_GAME\_STATE : int {read-only}  -GAME\_PLAYING\_STATE : int {read-only}  -GAME\_OVER\_STATE : int {read-only}  -mainCharacter : MainCharacter  -enemiesManager : EnamiesManager  -clouds : Clouds  -thread ; Thread  -isKeyPressed : Boolean  -gameState : int  -backgroundImage : Image |
| +GameScreen( )  +startGame( ) : void  +gameUpdate( ) : void  +paint(Graphics) : void  +run( ) : void  +keyPressed(KeyEvent) : void  +keyeleased(KeyEvent) ; void  +keyTyped(KeyEvent) : void  +resetGame( ) : void |

|  |
| --- |
| Clouds |
| -listCloud : List<ImageCloud>  -cloud : BufferedImage  -mainCharacter : MainCharacter |
| +Clouds(Main Character)  +update( ) : void  +draw(Graphics) : void  +ImageCLoud |

|  |
| --- |
| EnemiesManager |
| -Snow : BufferedImage  -Castle : BufferedImage  -rand : Random  -enemies : List<Enemy>  -mainCharacter : MainCharacter |
| +EnemiesManager(MainCharacter)  +update( ) : void  +draw(Graphics) : void  +createEnemy( ) : Enemy  +isCillision( ) : Boolean  +reset( ) : void |

|  |
| --- |
| EntryName |
| +EntryName :String {read-only}  -EnNameframe : JFrame  -uname : JTextField |
| +EntryName( )  -initialize( ) : void  +getEnNameframe( ) : JFrame  +setEnNameframe(JFrame) |

|  |
| --- |
| MainCharacter |
| +LAND\_POSY : int {read-only}  +GRAVITY : float {read-only}  -NORMAL\_RUN : int {read-only}  -JUMPING : int {read-only}  -DEATH : int {read-only}  -posY : float  -posX : float  -speedX : float  -speedY : float  -rectBound : Rectangle  +score : int  +hignscore : int  -state : int  -normalRunAnim : Animation  -jumping : BufferedImage -deathImage : BufferedImage |
| +MainCharacter( )  +getSpeedX( ) : float  +setSpeedX(int) : void  +draw(Graphics) : void  +update( ) : void  +jump( ) : void  +getBound( ) : Rectangle  +dead(boolean) : void  +reset( ) : void  +upScore( ) : void |

|  |
| --- |
| *Enemy* |
| +*update( ) : void*  *+draw(Graphics) : void*  *+getBound( ) : Rectangle*  *+isOutOfScreen( ) :boolean* |

|  |
| --- |
| Object |
| +YLAND : int {read-only}  -posX : int  -width : int  -height : int  -image : BufferedImage  -mainCharacter : MainCharacter  -rectBound : Rectangle |
| +Object(MainCharacter,int,int,int,BufferedImage)  +update( ) : void  +draw(Graphics) : void  +getBound( ) : Rectangle  +isOutOfScreen( ) : boolean |

|  |
| --- |
| Welcome |
| +Welname : String  -welframe : JFrame |
| +Welcome(String)  -initialize( ) : void  +getwelframe( ) : JFrame  +setwelframe(JFrame) : void  +keyPressed(KeyEvent) : void  +keyeleased(KeyEvent) ; void  +keyTyped(KeyEvent) : void |

|  |
| --- |
| Resource |
| +getResourceImage(String) : BufferedImage |

Main Class:

package TGOM;

import java.awt.EventQueue;

import java.awt.Image;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import java.awt.Toolkit;

import javax.swing.JLabel;

import java.awt.Color;

import java.awt.SystemColor;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Main {

private JFrame TGOM;

public static void main(String[] args){

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Main window = new Main();

window.getTGOM().setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public Main() {

initialize();

}

private void initialize() {

setTGOM(new JFrame());

getTGOM().getContentPane().setBackground(new Color(100, 149, 237));

getTGOM().setBackground(Color.WHITE);

getTGOM().setTitle("The Game Of Minion");

getTGOM().setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\LogMin.jpg"));

getTGOM().setBounds(300, 80, 800, 600);

getTGOM().setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel label = new JLabel("");

label.setBounds(141, 218, 230, 260);

Image img=new ImageIcon(this.getClass().getResource("/Front1.png")).getImage();

getTGOM().getContentPane().setLayout(null);

label.setIcon(new ImageIcon(img));

getTGOM().getContentPane().add(label);

JLabel lblNewLabel = new JLabel("The Game Of Minion");

lblNewLabel.setBounds(169, 100, 459, 61);

lblNewLabel.setFont(new Font("BankGothic Md BT", Font.BOLD, 40));

getTGOM().getContentPane().add(lblNewLabel);

JButton btnNewButton = new JButton("Start");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

EntryName EnName=new EntryName();

getTGOM().setVisible(false);

EnName.getEnNameframe().setVisible(true);

}

});

btnNewButton.setBounds(471, 249, 179, 33);

btnNewButton.setForeground(SystemColor.window);

btnNewButton.setFont(new Font("BankGothic Lt BT", Font.BOLD, 18));

btnNewButton.setBackground(SystemColor.desktop);

getTGOM().getContentPane().add(btnNewButton);

JButton btnScore = new JButton("High Score");

btnScore.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

HighScore hs=new HighScore();

getTGOM().setVisible(false);

hs.getHighframe().setVisible(true);

}

});

btnScore.setBounds(471, 295, 179, 33);

btnScore.setForeground(SystemColor.window);

btnScore.setFont(new Font("BankGothic Lt BT", Font.BOLD, 18));

btnScore.setBackground(SystemColor.desktop);

getTGOM().getContentPane().add(btnScore);

JButton btnHelp = new JButton("Help");

btnHelp.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Help hp=new Help();

getTGOM().setVisible(false);

hp.getHpframe().setVisible(true);

}

});

btnHelp.setBounds(471, 339, 179, 33);

btnHelp.setForeground(SystemColor.window);

btnHelp.setFont(new Font("BankGothic Lt BT", Font.BOLD, 18));

btnHelp.setBackground(SystemColor.desktop);

getTGOM().getContentPane().add(btnHelp);

JButton btnExit = new JButton("Credits");

btnExit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Credits c=new Credits();

TGOM.setVisible(false);

c.getCframe().setVisible(true);

}

});

btnExit.setBounds(471, 383, 179, 33);

btnExit.setForeground(SystemColor.window);

btnExit.setFont(new Font("BankGothic Lt BT", Font.BOLD, 18));

btnExit.setBackground(SystemColor.desktop);

getTGOM().getContentPane().add(btnExit);

JLabel label\_1 = new JLabel("");

label\_1.setBounds(424, 249, 37, 33);

Image imgPlay=new ImageIcon(this.getClass().getResource("/Play.png")).getImage();

label\_1.setIcon(new ImageIcon(imgPlay));

getTGOM().getContentPane().add(label\_1);

JLabel label\_2 = new JLabel("");

label\_2.setBounds(424, 295, 30, 33);

Image imgTrophy=new ImageIcon(this.getClass().getResource("/Trophy-icon.png")).getImage();

label\_2.setIcon(new ImageIcon(imgTrophy));

getTGOM().getContentPane().add(label\_2);

JLabel label\_3 = new JLabel("");

label\_3.setBounds(422, 339, 39, 33);

Image imgQues=new ImageIcon(this.getClass().getResource("/Question2.png")).getImage();

label\_3.setIcon(new ImageIcon(imgQues));

getTGOM().getContentPane().add(label\_3);

JButton btnExit\_1 = new JButton("Exit");

btnExit\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

btnExit\_1.setBounds(471, 427, 179, 33);

btnExit\_1.setForeground(Color.WHITE);

btnExit\_1.setFont(new Font("BankGothic Lt BT", Font.BOLD, 18));

btnExit\_1.setBackground(Color.BLACK);

getTGOM().getContentPane().add(btnExit\_1);

JLabel label\_4 = new JLabel("");

label\_4.setBounds(422, 427, 39, 33);

Image imgSetting=new ImageIcon(this.getClass().getResource("/Settings1.png")).getImage();

label\_4.setIcon(new ImageIcon(imgSetting));

getTGOM().getContentPane().add(label\_4);

JLabel label\_5 = new JLabel("");

label\_5.setBounds(422, 383, 39, 33);

Image imgCredits=new ImageIcon(this.getClass().getResource("/Credits.png")).getImage();

label\_5.setIcon(new ImageIcon(imgCredits));

getTGOM().getContentPane().add(label\_5);

JLabel lblNewLabel\_1 = new JLabel("");

lblNewLabel\_1.setBounds(65, 82, 94, 97);

Image imgfront=new ImageIcon(this.getClass().getResource("/FrontS.png")).getImage();

lblNewLabel\_1.setIcon(new ImageIcon(imgfront));

TGOM.getContentPane().add(lblNewLabel\_1);

JLabel label\_6 = new JLabel("");

label\_6.setBounds(642, 82, 94, 97);

Image imgfront2=new ImageIcon(this.getClass().getResource("/FrontS.png")).getImage();

label\_6.setIcon(new ImageIcon(imgfront2));

TGOM.getContentPane().add(label\_6);

}

public JFrame getMain() {

return getTGOM();

}

public void setMain(JFrame m) {

setTGOM(m);

}

public JFrame getTGOM() {

return TGOM;

}

public void setTGOM(JFrame tGOM) {

TGOM = tGOM;

}

}

HighScore Class:

package TGOM;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.SystemColor;

import java.awt.Toolkit;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class HighScore extends Main {

private JFrame highframe;

private JTextField dname;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

HighScore window = new HighScore();

window.getHighframe().setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public HighScore() {

initialize();

}

private void initialize() {

highframe = new JFrame();

highframe.setTitle("The Game Of Minion");

highframe.setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\LogMin.jpg"));

highframe.getContentPane().setBackground(new Color(0, 206, 209));

highframe.getContentPane().setLayout(null);

JLabel lblName = new JLabel("Name ");

lblName.setBounds(284, 93, 99, 37);

lblName.setFont(new Font("BankGothic Md BT", Font.BOLD, 23));

lblName.setHorizontalAlignment(SwingConstants.CENTER);

highframe.getContentPane().add(lblName);

JLabel lblScore = new JLabel("Score");

lblScore.setBounds(437, 93, 93, 37);

lblScore.setHorizontalAlignment(SwingConstants.CENTER);

lblScore.setFont(new Font("BankGothic Md BT", Font.BOLD, 23));

highframe.getContentPane().add(lblScore);

dname = new JTextField();

dname.setToolTipText("Enter Name");

dname.setBounds(309, 465, 170, 20);

highframe.getContentPane().add(dname);

dname.setColumns(10);

JLabel lblDeleteUserName = new JLabel("Delete User Name");

lblDeleteUserName.setForeground(SystemColor.desktop);

lblDeleteUserName.setHorizontalAlignment(SwingConstants.CENTER);

lblDeleteUserName.setFont(new Font("BankGothic Lt BT", Font.BOLD, 14));

lblDeleteUserName.setBounds(322, 440, 141, 14);

highframe.getContentPane().add(lblDeleteUserName);

JButton btnDone = new JButton("Done");

btnDone.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String name = dname.getText().toString();

try {

File f1= new File("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt");

File f2= new File("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User1.txt");

FileReader fr = new FileReader(f1);

FileWriter fw = new FileWriter(f2,true);

BufferedReader br = new BufferedReader(fr);

BufferedWriter bw=new BufferedWriter(fw);

String line, fuser, fsc;

while ((line = br.readLine()) != null) {

fuser = line.split(" ")[0];

fsc = line.split(" ")[1];

if (fuser.equals(name)) {

}

else

{

fw.write(line+"\n");

}

}

fr.close();

fw.close();

f1.delete();

File rf =new File("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt");

f2.renameTo(rf);

}

catch (Exception ep) {

ep.printStackTrace();

}

Main main=new Main();

highframe.setVisible(false);

main.getMain().setVisible(true);

}

});

btnDone.setFont(new Font("BankGothic Lt BT", Font.BOLD, 14));

btnDone.setBorderPainted(false);

btnDone.setForeground(Color.WHITE);

btnDone.setBackground(Color.BLACK);

btnDone.setBounds(345, 496, 89, 23);

highframe.getContentPane().add(btnDone);

JButton btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Main main=new Main();

highframe.setVisible(false);

main.getMain().setVisible(true);

}

});

btnBack.setForeground(Color.WHITE);

btnBack.setFont(new Font("BankGothic Lt BT", Font.BOLD, 14));

btnBack.setBorderPainted(false);

btnBack.setBackground(Color.BLACK);

btnBack.setBounds(685, 527, 89, 23);

highframe.getContentPane().add(btnBack);

highframe.setTitle("The Game Of Minion");

highframe.setBounds(300, 80, 800, 600);

getHighframe().setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

try

{

File file=new File("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt");

FileReader fr=new FileReader(file);

BufferedReader br=new BufferedReader(fr);

StringBuffer sb=new StringBuffer();

String line;

int x=300;

int y=0;

while((line=br.readLine())!=null)

{

JLabel lblNewLabel = new JLabel(line+"\n");

lblNewLabel.setBounds(x, y, 531, 336);

lblNewLabel.setFont(new Font("BankGothic Md BT", Font.BOLD, 24));

y+=40;

highframe.getContentPane().add(lblNewLabel);

}

fr.close();

}

catch(IOException e)

{

e.printStackTrace();

}

}

public JFrame getHighframe() {

return highframe;

}

public void setHighframe(JFrame highframe) {

this.highframe = highframe;

}

}

Help Class:

package TGOM;

import java.awt.EventQueue;

import java.awt.Toolkit;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import java.awt.Color;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Image;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Help extends Main {

private JFrame hpframe;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Help window = new Help();

window.getHpframe().setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public Help() {

initialize();

}

private void initialize() {

setHpframe(new JFrame());

getHpframe().setTitle("The Game Of Minion");

getHpframe().setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\Desktop\\minion.jpg"));

getHpframe().setBounds(300, 80, 800, 600);

getHpframe().setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public JFrame getHpframe() {

return hpframe;

}

public void setHpframe(JFrame hpframe) {

this.hpframe = hpframe;

hpframe.getContentPane().setBackground(new Color(64, 224, 208));

hpframe.getContentPane().setLayout(null);

JLabel lblUseSpaceBar = new JLabel("Use \"Space\" To Avoid Snow Man");

lblUseSpaceBar.setFont(new Font("BankGothic Md BT", Font.BOLD, 23));

lblUseSpaceBar.setBounds(199, 172, 414, 14);

hpframe.getContentPane().add(lblUseSpaceBar);

JLabel lblUsedoubleSpace = new JLabel("Use \"Double Space\" To Avoid Castle");

lblUsedoubleSpace.setFont(new Font("BankGothic Md BT", Font.BOLD, 23));

lblUsedoubleSpace.setBounds(173, 295, 473, 14);

hpframe.getContentPane().add(lblUsedoubleSpace);

JLabel label = new JLabel("");

label.setBounds(367, 210, 67, 58);

Image imgPlay=new ImageIcon(this.getClass().getResource("/SnowHP.png")).getImage();

label.setIcon(new ImageIcon(imgPlay));

hpframe.getContentPane().add(label);

JLabel label\_1 = new JLabel("");

label\_1.setBounds(341, 329, 118, 94);

Image imgPlay2=new ImageIcon(this.getClass().getResource("/CastleHP.png")).getImage();

label\_1.setIcon(new ImageIcon(imgPlay2));

hpframe.getContentPane().add(label\_1);

JButton btnBack = new JButton("Back");

btnBack.setForeground(new Color(255, 255, 255));

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Main m=new Main();

hpframe.setVisible(false);

m.getTGOM().setVisible(true);

}

});

btnBack.setBackground(new Color(0, 0, 0));

btnBack.setFont(new Font("BankGothic Md BT", Font.BOLD, 16));

btnBack.setBounds(354, 516, 89, 23);

hpframe.getContentPane().add(btnBack);

}

}

Credits Class:

package TGOM;

import java.awt.EventQueue;

import java.awt.Toolkit;

import javax.swing.JFrame;

import java.awt.Color;

import javax.swing.JLabel;

import java.awt.Font;

import javax.swing.SwingConstants;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Credits extends Main {

private JFrame cframe;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Credits window = new Credits();

window.getCframe().setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public Credits() {

initialize();

}

private void initialize() {

setCframe(new JFrame());

getCframe().setTitle("The Game Of Minion");

getCframe().setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\Desktop\\minion.jpg"));

getCframe().setBounds(300, 80, 800, 600);

getCframe().setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public JFrame getCframe() {

return cframe;

}

public void setCframe(JFrame cframe) {

this.cframe = cframe;

cframe.getContentPane().setBackground(new Color(0, 191, 255));

cframe.getContentPane().setLayout(null);

JLabel lblKaziR = new JLabel("Kazi Ramisa Rifa");

lblKaziR.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

lblKaziR.setBounds(307, 115, 191, 25);

cframe.getContentPane().add(lblKaziR);

JLabel label = new JLabel("1911019642");

label.setHorizontalAlignment(SwingConstants.CENTER);

label.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

label.setBounds(307, 151, 191, 25);

cframe.getContentPane().add(label);

JLabel lblFarihaNurHasin = new JLabel("Fariha Nur Hasin");

lblFarihaNurHasin.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

lblFarihaNurHasin.setBounds(307, 223, 202, 25);

cframe.getContentPane().add(lblFarihaNurHasin);

JLabel label\_3 = new JLabel("1912148642");

label\_3.setHorizontalAlignment(SwingConstants.CENTER);

label\_3.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

label\_3.setBounds(307, 259, 191, 25);

cframe.getContentPane().add(label\_3);

JLabel lblFaridaYeasminMim = new JLabel("Farida Yeasmin Mim");

lblFaridaYeasminMim.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

lblFaridaYeasminMim.setBounds(294, 331, 220, 25);

cframe.getContentPane().add(lblFaridaYeasminMim);

JLabel label\_6 = new JLabel("1911653642");

label\_6.setHorizontalAlignment(SwingConstants.CENTER);

label\_6.setFont(new Font("BankGothic Md BT", Font.BOLD, 20));

label\_6.setBounds(307, 367, 191, 25);

cframe.getContentPane().add(label\_6);

JButton btnNewButton = new JButton("Back");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Main m=new Main();

cframe.setVisible(false);

m.getTGOM().setVisible(true);

}

});

btnNewButton.setBackground(new Color(0, 0, 0));

btnNewButton.setForeground(new Color(255, 255, 255));

btnNewButton.setFont(new Font("BankGothic Md BT", Font.BOLD, 16));

btnNewButton.setBounds(685, 527, 89, 23);

cframe.getContentPane().add(btnNewButton);

}

}

EntryName Class:

package TGOM;

import java.awt.EventQueue;

import java.awt.SystemColor;

import java.awt.Toolkit;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.UIManager;

import java.awt.event.ActionListener;

import java.io.FileWriter;

import java.io.IOException;

import java.awt.event.ActionEvent;

import java.awt.Color;

public class EntryName extends Main {

public static final String EntryName = null;

private JFrame EnNameframe;

private JTextField uname;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

EntryName window = new EntryName();

window.getEnNameframe().setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public EntryName() {

initialize();

}

private void initialize() {

setEnNameframe(new JFrame());

getEnNameframe().getContentPane().setBackground(new Color(30, 144, 255));

EnNameframe.getContentPane().setLayout(null);

JLabel lblEnterYourName = new JLabel("Enter Your Name");

lblEnterYourName.setFont(new Font("BankGothic Md BT", Font.BOLD, 26));

lblEnterYourName.setBounds(264, 179, 271, 58);

EnNameframe.getContentPane().add(lblEnterYourName);

uname = new JTextField();

uname.setToolTipText("Enter Your Name");

uname.setFont(new Font("BankGothic Lt BT", Font.BOLD, 14));

uname.setBounds(264, 246, 272, 20);

EnNameframe.getContentPane().add(uname);

uname.setColumns(10);

JButton btnEnter = new JButton("OK");

btnEnter.setFont(new Font("BankGothic Md BT", Font.BOLD, 18));

btnEnter.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

String path = "C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt";

FileWriter myWriter = new FileWriter("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt",true);

myWriter.write(uname.getText()+" ");

myWriter.close();

}

catch (IOException ep) {

System.out.println("ERROR 404!");

ep.printStackTrace();

}

String wname = uname.getText().toString();

Welcome wel=new Welcome(wname);

EnNameframe.setVisible(false);

wel.getwelframe().setVisible(true);

}

});

btnEnter.setForeground(SystemColor.textText);

btnEnter.setBackground(Color.WHITE);

btnEnter.setBounds(340, 287, 111, 23);

EnNameframe.getContentPane().add(btnEnter);

JButton btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Main main=new Main();

EnNameframe.setVisible(false);

main.getMain().setVisible(true);

}

});

btnBack.setForeground(Color.BLACK);

btnBack.setFont(new Font("BankGothic Md BT", Font.BOLD, 18));

btnBack.setBackground(Color.WHITE);

btnBack.setBounds(663, 521, 111, 29);

EnNameframe.getContentPane().add(btnBack);

getEnNameframe().setIconImage(Toolkit.getDefaultToolkit().getImage("/LogMin.jpg"));

getEnNameframe().setTitle("The Game Of Minion");

getEnNameframe().setBounds(300, 80, 800, 600);

getEnNameframe().setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public JFrame getEnNameframe() {

return EnNameframe;

}

public void setEnNameframe(JFrame enNameframe) {

EnNameframe = enNameframe;

}

}

Welcome Class:

package TGOM;

import java.awt.\*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Welcome implements KeyListener {

public String Welname;

private JFrame welframe;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Welcome window = new Welcome("NULL");

window.welframe.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public Welcome(String wname) {

Welname=wname;

initialize();

}

private void initialize() {

welframe = new JFrame();

welframe.getContentPane().setBackground(new Color(0, 191, 255));

getwelframe().setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\LogMin.jpg"));

getwelframe().setTitle("The Game Of Minion");

welframe.setBounds(300, 80, 800, 600);

welframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

welframe.getContentPane().setLayout(null);

JLabel lblWe = new JLabel(Welname);

lblWe.setHorizontalAlignment(SwingConstants.CENTER);

lblWe.setForeground(Color.WHITE);

lblWe.setFont(new Font("BankGothic Md BT", Font.BOLD, 24));

lblWe.setBounds(182, 201, 445, 45);

welframe.getContentPane().add(lblWe);

JLabel lblWelcome = new JLabel("Welcome");

lblWelcome.setForeground(Color.WHITE);

lblWelcome.setFont(new Font("BankGothic Md BT", Font.BOLD, 30));

lblWelcome.setBounds(327, 157, 151, 45);

welframe.getContentPane().add(lblWelcome);

JLabel lblPressEnterTo = new JLabel("Press The Button To Continue...");

lblPressEnterTo.setHorizontalAlignment(SwingConstants.CENTER);

lblPressEnterTo.setForeground(Color.WHITE);

lblPressEnterTo.setFont(new Font("BankGothic Md BT", Font.BOLD, 18));

lblPressEnterTo.setBounds(216, 267, 383, 14);

welframe.getContentPane().add(lblPressEnterTo);

JButton btnNewButton = new JButton("");

btnNewButton.setBorderPainted(false);

Image imgPlay=new ImageIcon(this.getClass().getResource("/PlayWel.png")).getImage();

btnNewButton.setIcon(new ImageIcon(imgPlay));

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

GameWindow gw=new GameWindow();

welframe.setVisible(false);

gw.setVisible(true);

}

});

btnNewButton.setBackground(new Color(0, 191, 255));

btnNewButton.setFont(new Font("BankGothic Lt BT", Font.BOLD, 14));

btnNewButton.setBounds(359, 292, 92, 73);

welframe.getContentPane().add(btnNewButton);

}

public JFrame getwelframe() {

return welframe;

}

public void setwelframe(JFrame welf) {

welframe = welf;

}

public void keyReleased(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_SPACE) {

System.exit(0);

}

}

public void keyTyped(KeyEvent e) {

// TODO Auto-generated method stub

}

public void keyPressed(KeyEvent e) {

// TODO Auto-generated method stub

}

}

GameWindow Class:

package TGOM;

import java.awt.Color;

import java.awt.Image;

import java.awt.SystemColor;

import java.awt.Toolkit;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

public class GameWindow extends JFrame {

public static final int SCREEN\_WIDTH = 800;

private GameScreen gameScreen;

public GameWindow() {

super("The Game Of Minion");

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\LogMin.jpg"));

setBounds(300, 80, 800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

gameScreen = new GameScreen();

addKeyListener(gameScreen);

add(gameScreen);

startGame();

}

public void startGame() {

setVisible(true);

gameScreen.startGame();

}

public static void main(String args[]) {

GameWindow gw=new GameWindow();

}

}

GameScreen Class:

package TGOM;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.SystemColor;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import TGOM.Clouds;

import TGOM.EnemiesManager;

import TGOM.MainCharacter;

import TGOM.Resource;

public class GameScreen extends JPanel implements Runnable, KeyListener {

private static final int START\_GAME\_STATE=0;

private static final int GAME\_PLAYING\_STATE=1;

private static final int GAME\_OVER\_STATE=2;

private MainCharacter mainCharacter;

private EnemiesManager enemiesManager;

private Clouds clouds;

private Thread thread;

private boolean isKeyPressed;

private int gameState = START\_GAME\_STATE;

private Image backgroundImage;

public GameScreen() {

try {

backgroundImage= ImageIO.read(new File("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Back.png"));

} catch (IOException e) {

e.printStackTrace();

}

mainCharacter = new MainCharacter();

mainCharacter.setSpeedX(4);

enemiesManager = new EnemiesManager(mainCharacter);

clouds = new Clouds(mainCharacter);

}

public void startGame() {

thread = new Thread(this);

thread.start();

}

public void gameUpdate() {

if (gameState == GAME\_PLAYING\_STATE) {

clouds.update();

mainCharacter.update();

enemiesManager.update();

if (enemiesManager.isCollision()) {

gameState = GAME\_OVER\_STATE;

mainCharacter.dead(true);

}

}

}

public void paint(Graphics g) {

super.paint(g);

g.drawImage(backgroundImage, 0, 0, null);

switch (gameState) {

case START\_GAME\_STATE:

mainCharacter.draw(g);

break;

case GAME\_PLAYING\_STATE:

case GAME\_OVER\_STATE:

clouds.draw(g);

enemiesManager.draw(g);

mainCharacter.draw(g);

g.setColor(Color.BLACK);

g.setFont(new Font("BankGothic Md BT",1,20));

g.drawString("HIGH SCORE: " +mainCharacter.highscore, 540, 25);

g.drawString("SCORE: " +mainCharacter.score, 540, 42);

if (gameState==GAME\_OVER\_STATE) {

mainCharacter.score=0;

g.setFont(new Font("BankGothic Md BT",1,40));

g.drawString("GAME OVER",248,200);

g.setFont(new Font("BankGothic Md BT",1,30));

g.drawString("Press Space To Continue",160,250);

g.setFont(new Font("BankGothic Md BT",1,20));

g.drawString("Press E To Exit Or Press B For Back Menu",142,300);

}

break;

}

}

public void run() {

int fps=100;

long msPerFrame=1000\*1000000/fps;

long lastTime=0;

long elapsed;

int msSleep;

int nanoSleep;

long endProcessGame;

long lag=0;

while (true) {

gameUpdate();

repaint();

endProcessGame=System.nanoTime();

elapsed=(lastTime+msPerFrame-System.nanoTime());

msSleep=(int)(elapsed/1000000);

nanoSleep=(int)(elapsed%1000000);

if (msSleep<=0) {

lastTime = System.nanoTime();

continue;

}

try {

Thread.sleep(msSleep, nanoSleep);

} catch (InterruptedException e) {

e.printStackTrace();

}

lastTime = System.nanoTime();

}

}

public void keyPressed(KeyEvent e) {

if (!isKeyPressed) {

isKeyPressed = true;

switch (gameState) {

case START\_GAME\_STATE:

if (e.getKeyCode() == KeyEvent.VK\_SPACE) {

gameState = GAME\_PLAYING\_STATE;

}

break;

case GAME\_PLAYING\_STATE:

if (e.getKeyCode() == KeyEvent.VK\_SPACE) {

mainCharacter.jump();

}

break;

case GAME\_OVER\_STATE:

if (e.getKeyCode() == KeyEvent.VK\_SPACE) {

gameState = GAME\_PLAYING\_STATE;

resetGame();

}

else if (e.getKeyCode() == KeyEvent.VK\_E) {

try {

String path = "C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt";

FileWriter myWriter = new FileWriter("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt",true);

myWriter.write(mainCharacter.highscore+" "+"\n");

myWriter.close();

System.exit(0);

}

catch (IOException ep) {

ep.printStackTrace();

}

}

else if (e.getKeyCode() == KeyEvent.VK\_B) {

try {

String path = "C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt";

FileWriter myWriter = new FileWriter("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\User.txt",true);

myWriter.write(mainCharacter.highscore+" "+"\n");

myWriter.close();

Main main=new Main();

setVisible(false);

main.getMain().setVisible(true);

}

catch (IOException ep) {

ep.printStackTrace();

}

}

break;

}

}

}

public void keyReleased(KeyEvent e) {

isKeyPressed=false;

}

public void keyTyped(KeyEvent e) {

}

public void resetGame() {

enemiesManager.reset();

mainCharacter.dead(false);

mainCharacter.reset();

}

}

Enemy Class:

package TGOM;

import java.awt.Graphics;

import java.awt.Rectangle;

public abstract class Enemy {

public abstract void update();

public abstract void draw(Graphics g);

public abstract Rectangle getBound();

public abstract boolean isOutOfScreen();

}

EnemiesManager Class:

package TGOM;

import java.awt.Graphics;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import TGOM.Resource;

public class EnemiesManager {

private BufferedImage Snow;

private BufferedImage Castle;

private Random rand;

private List<Enemy> enemies;

private MainCharacter mainCharacter;

public EnemiesManager(MainCharacter mainCharacter) {

rand = new Random();

Snow = Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Snow.png");

Castle = Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Castle.png");

enemies = new ArrayList<Enemy>();

this.mainCharacter = mainCharacter;

enemies.add(createEnemy());

}

public void update() {

for(Enemy e : enemies) {

e.update();

}

Enemy enemy = enemies.get(0);

if(enemy.isOutOfScreen()) {

mainCharacter.upScore();

enemies.clear();

enemies.add(createEnemy());

}

}

public void draw(Graphics g) {

for(Enemy e : enemies) {

e.draw(g);

}

}

public Enemy createEnemy() {

int type=rand.nextInt(2);

if(type==0) {

return new Object(mainCharacter,800,Snow.getWidth(),Snow.getHeight(),Snow);

}

else {

return new Object(mainCharacter,800,Castle.getWidth(),Castle.getHeight(),Castle);

}

}

public boolean isCollision() {

for(Enemy e : enemies) {

if (mainCharacter.getBound().intersects(e.getBound())) {

return true;

}

}

return false;

}

public void reset() {

enemies.clear();

enemies.add(createEnemy());

}

}

Main Character Class:

package TGOM;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.Rectangle;

import java.awt.image.BufferedImage;

import java.net.MalformedURLException;

import java.net.URL;

import javax.swing.ImageIcon;

import TGOM.Animation;

import TGOM.Resource;

public class MainCharacter {

public static final int LAND\_POSY = 425;

public static final float GRAVITY = 0.4f;

private static final int NORMAL\_RUN = 0;

private static final int JUMPING = 1;

private static final int DEATH = 2;

private float posY;

private float posX;

private float speedX;

private float speedY;

private Rectangle rectBound;

public int score = 0;

public int highscore=0;

private int state = NORMAL\_RUN;

private Animation normalRunAnim;

private BufferedImage jumping;

private BufferedImage deathImage;

public MainCharacter() {

posX = 50;

posY = LAND\_POSY;

rectBound = new Rectangle();

normalRunAnim = new Animation(150);

normalRunAnim.addFrame(Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Minion1.png"));

jumping = Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Minion2.png");

deathImage = Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Evil.png");

}

public float getSpeedX()

{

return speedX;

}

public void setSpeedX(int speedX)

{

this.speedX = speedX;

}

public void draw(Graphics g) {

switch(state) {

case NORMAL\_RUN:

g.drawImage(normalRunAnim.getFrame(), (int) posX, (int) posY, null);

break;

case JUMPING:

g.drawImage(jumping, (int) posX, (int) posY, null);

break;

case DEATH:

g.drawImage(deathImage, (int) posX, (int) posY, null);

break;

}

}

public void update() {

normalRunAnim.updateFrame();

if(posY >= LAND\_POSY) {

posY = LAND\_POSY;

state = NORMAL\_RUN;

} else

{

speedY += GRAVITY;

posY += speedY;

}

}

public void jump()

{

speedY = -10.7f;

posY += speedY;

state = JUMPING;

}

public Rectangle getBound() {

rectBound = new Rectangle();

rectBound.x = (int) posX;

rectBound.y = (int) posY;

rectBound.width = normalRunAnim.getFrame().getWidth();

rectBound.height = normalRunAnim.getFrame().getHeight();

return rectBound;

}

public void dead(boolean isDeath) {

if(isDeath) {

state = DEATH;

} else {

state = NORMAL\_RUN;

}

}

public void reset() {

posY = LAND\_POSY;

}

public void upScore() {

score += 20;

if(score>highscore)

highscore=score;

}

}

Object Class:

package TGOM;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Rectangle;

import java.awt.image.BufferedImage;

public class Object extends Enemy {

public static final int YLAND = 500;

private int posX;

private int width;

private int height;

private BufferedImage image;

private MainCharacter mainCharacter;

private Rectangle rectBound;

public Object(MainCharacter mainCharacter, int posX, int width, int height, BufferedImage image)

{

this.posX=posX;

this.width=width;

this.height=height;

this.image=image;

this.mainCharacter=mainCharacter;

rectBound=new Rectangle();

}

public void update()

{

posX-=mainCharacter.getSpeedX();

}

public void draw(Graphics g)

{

g.drawImage(image,posX,YLAND-image.getHeight(),null);

g.setColor(Color.red);

}

public Rectangle getBound() {

rectBound=new Rectangle();

rectBound.x=posX;

rectBound.y =500-image.getHeight();

rectBound.width =width;

rectBound.height= height;

return rectBound;

}

public boolean isOutOfScreen() {

if(posX<-image.getWidth()) {

return true;

}

return false;

}

}

Clouds Class:

package TGOM;

import java.awt.Graphics;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import TGOM.GameWindow;

import TGOM.Resource;

public class Clouds {

private List<ImageCloud> listCloud;

private BufferedImage cloud;

private MainCharacter mainCharacter;

public Clouds(MainCharacter mainCharacter) {

this.mainCharacter = mainCharacter;

cloud = Resource.getResouceImage("C:\\Users\\user\\eclipse-workspace\\The Game Of Minion\\Img\\Cloud.png");

listCloud = new ArrayList<ImageCloud>();

ImageCloud imageCloud=new ImageCloud();

imageCloud.posX=0;

imageCloud.posY=30;

listCloud.add(imageCloud);

imageCloud=new ImageCloud();

imageCloud.posX=150;

imageCloud.posY=40;

listCloud.add(imageCloud);

imageCloud=new ImageCloud();

imageCloud.posX=300;

imageCloud.posY=50;

listCloud.add(imageCloud);

imageCloud=new ImageCloud();

imageCloud.posX=450;

imageCloud.posY=20;

listCloud.add(imageCloud);

imageCloud=new ImageCloud();

imageCloud.posX=600;

imageCloud.posY=60;

listCloud.add(imageCloud);

imageCloud=new ImageCloud();

imageCloud.posX=750;

imageCloud.posY=50;

listCloud.add(imageCloud);

}

public void update()

{

Iterator<ImageCloud> itr= listCloud.iterator();

ImageCloud firstElement= itr.next();

firstElement.posX-= mainCharacter.getSpeedX()/2;

while(itr.hasNext())

{

ImageCloud element= itr.next();

element.posX-= mainCharacter.getSpeedX()/2;

}

if(firstElement.posX<-cloud.getWidth())

{

listCloud.remove(firstElement);

firstElement.posX = GameWindow.SCREEN\_WIDTH;

listCloud.add(firstElement);

}

}

public void draw(Graphics g) {

for(ImageCloud imgLand : listCloud) {

g.drawImage(cloud, (int) imgLand.posX, imgLand.posY, null);

}

}

public class ImageCloud {

float posX;

int posY;

}

}

Animation Class:

package TGOM;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.List;

public class Animation {

private List<BufferedImage> list;

private long deltaTime;

private int currentFrame = 0;

private long previousTime;

public Animation(int deltaTime)

{

this.deltaTime = deltaTime;

list = new ArrayList<BufferedImage>();

previousTime = 0;

}

public void updateFrame()

{

if (System.currentTimeMillis()-previousTime>=deltaTime) {

currentFrame++;

if (currentFrame>=list.size()) {

currentFrame=0;

}

previousTime = System.currentTimeMillis();

}

}

public void addFrame(BufferedImage image)

{

list.add(image);

}

public BufferedImage getFrame()

{

return list.get(currentFrame);

}

}

Resource Class:

package TGOM;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

public class Resource {

public static BufferedImage getResouceImage(String path) {

BufferedImage img = null;

try {

img = ImageIO.read(new File(path));

} catch (IOException e) {

e.printStackTrace();

}

return img;

}

}