IV Projekt (Überschritten stehen für neue Klassen)

Frame

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class frame extends JFrame implements ActionListener {

private JButton schliessen;

private JButton einstellung;

private JButton info;

private JButton ende;

public static void main (String[] args){

frame frame= new frame ("Menü");//beschriftung des tabs und erschaffung des menüs

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

frame.setSize(400,400);//gesamt größe des menüs

frame.setLayout(null);

frame.setVisible(true);

}

public frame(String title){

super(title);

schliessen = new JButton("Spiel Starten");//benennung der buttons

schliessen.setBounds(120,40,160,40);//positionierung der buttons im Menü

schliessen.addActionListener(this);

add(schliessen);

einstellung = new JButton("Einstellungen");

einstellung.setBounds(120,120,160,40);

einstellung.addActionListener(this);

add(einstellung);

info = new JButton("Credits");

info.setBounds(120,200,160,40);

info.addActionListener(this);

add(info);

ende = new JButton("Ende");

ende.setBounds(120,260,160,40);

ende.addActionListener(this);

add(ende);

}

public static void fenster() { //öffnung des fensters in dem das Spiel abläuft

JFrame fenster = new JFrame("Game");

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

fenster.setSize(650,350);

fenster.setVisible(true);

fenster.add(new gui());//zugriff auf gui

fenster.setVisible(true);

}

//public static void auswahl() {

// }

@Override

public void actionPerformed(ActionEvent e) {

//zuordnung der funktionen der Buttons

// TODO Auto-generated method stub

if (e.getSource()== schliessen) {

fenster();

}

if (e.getSource() == info ) {

Object[] options = { "OK"};

JOptionPane.showOptionDialog(null,"Programmiert von Tristan und Leon !","Information",JOptionPane.DEFAULT\_OPTION,JOptionPane.PLAIN\_MESSAGE,null,options,options[0]);

}

if (e.getSource()== einstellung) {

//auswahl();

}

if (e.getSource() == ende ) {

System.exit(0);

}

}}

Gui

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import javax.swing.ImageIcon;

import javax.swing.JPanel;

import javax.swing.Timer;

public class gui extends JPanel implements ActionListener{

Timer time;

Image img;

Image img2;

int key ;

int nx ,nx2;

int figur\_y = 230;

int left =0;

int X\_Bild;

int lauf ;

Block block1;

int coin\_score;

int anzahl = 0;

int anzahl2 = 0;

public gui () {

nx=0;

nx2= 690;

key=0;

lauf=0;

setFocusable(true);

//aufrufe von den bildern vom speicher

ImageIcon u = new ImageIcon("D:/Users/trist/eclipse-workspace/IV Projekt/background.jpg");

img = u.getImage();

addKeyListener(new AL());

block1= new Block(250,110,50,50,Color.YELLOW);

Sprung sprung= new Sprung();//zugriff auf die klasse sprung

ImageIcon s= new ImageIcon("D:/Users/trist/eclipse-workspace/IV Projekt/character2.png");

img2 = s.getImage();

time= new Timer(5,this);

time.start();

}

public void actionPerformed(ActionEvent e) {

bewegen();

figur\_y = Sprung.sprungposition;

repaint();

}

public void paint (Graphics g) { //Bild wird eingefügt

super.paint(g);

Graphics2D f2= (Graphics2D) g;

//wieder neu einfügung des bildes nach ende (leider nicht die selbe bild größe wie im toutorial

//und auch bei mehreren rumprobieren mit den werten nicht hinbekommen da nicht erklärt wird für was welcher ist)

if (getXBild() == 510+ (anzahl\*2350)) {

anzahl+=1;

nx=0;

}

if(getXBild() == 1690+(anzahl2\*2350)) {

anzahl2+=1;

nx2=0;

}

if(getXBild()>=510) {

f2.drawImage(img,685-nx,0,null);

}

f2.drawImage(img,685-nx2,0,null);

//einfügen vom charakter

f2.drawImage(img2, left, figur\_y, null);

f2.setColor(block1.getFarbe());

f2.fillRect(block1.getX\_Block()-getXBild(), block1.getY\_Block(),block1.getWidth(),block1.getHight());

//zeichnet das gefüllte quadrat; durch-getXBild verhindern wir das er stehen bleibt

block1.Kolisionsabfrage(block1.getX\_Block()-getXBild(), block1.getY\_Block()+block1.getHight(), left+(74-12), figur\_y);

//überführung der werte

if(block1.coin() == true) {

coin\_score= 1;

}//wenn der block getroffen wird

f2.drawString("Score :"+coin\_score, 10, 15);

//erschaffen vom score board

}

private int getXBild() {

// TODO Auto-generated method stub

return X\_Bild;

}

//Bild possition wird verändert

public void bewegen() {

// einstellen das der charackter nur bis zum linken ende vom bildschirm laufen kann

if(lauf != -5) {

if( left+lauf<=75) {

left+= lauf;

}else {

X\_Bild += lauf;

nx += lauf;

nx2 += lauf;

}

}else {

if(left+lauf>0) {

left+=lauf;

}

}

}

private class AL extends KeyAdapter{

//Al = ActionListener

public AL() {

}

public void keyPressed(KeyEvent e) {

//was passiert wenn ich die linke oder die rechte pfeil taste drücke

key= e.getKeyCode();

if (key == KeyEvent.VK\_LEFT) {

lauf= -5;

}

if (key == KeyEvent.VK\_RIGHT) {

lauf= +5;

}

if (key == KeyEvent.VK\_ESCAPE) {

System.exit(0);

}

if( key == KeyEvent.VK\_SPACE) {

Sprung();

}

}

public void keyReleased(KeyEvent e) {

// was passiert wenn ich die tasten nicht mehr berühre

key= e.getKeyCode();

if(key==KeyEvent.VK\_LEFT|| key==KeyEvent.VK\_RIGHT) {

lauf = 0;

}

}

}

public void Sprung() {

Sprung SprungAnimation = new Sprung();

SprungAnimation.start();

}

}

Sprung

**public** **class** Sprung **extends** Thread {

**static** **boolean** *fertig* = **true**;

**static** **boolean** *hochpunkt*=**false**;

**int** sprunghöhe = 80;//max sprunghöhe

**static** **int** *ursprungY* = 230;//laufhöhe

**static** **int** *sprungposition* = *ursprungY*;//abfrage wo er sich gerade befindet

**public** Sprung() {

}

**public** **void** run() {

*fertig* = **false**;

**int** verzögerung =3;

**while**(*fertig* == **false**) {

Sprung();

**try** {

Thread.*sleep*(verzögerung);

//durch thread wird gewährleistet das er beim springen nicht stoppt sondern weiter läuft

}**catch**(Exception e) {

}

}

*hochpunkt*=**false**;

}

**public** **void** Sprung(){

//hier wird der sprung selbst gemacht , so dass er eine schrittweise kurve macht

//bei java wird von oben nach unten gezählt deswegen für hoch - und runter +

**if**(*hochpunkt* == **false**) {

*sprungposition*--;

}

**if**(*sprungposition* == (*ursprungY*-sprunghöhe)) {

*hochpunkt*=**true**;

}

**if**(*hochpunkt*== **true** && *sprungposition* <= *ursprungY* ) {

*sprungposition* ++;

**if** (*sprungposition* == *ursprungY*) {

*fertig*= **true**;

}

}

}

}

Block

**import** java.awt.Color;

**public** **class** Block {

**final** **int** X , Y;//für Koordinaten des Blocks

**final** **int** WIDTH, HIGHT;//für weite höhe des blocks

**boolean** coin = **false**;

**final** Color COLOR;//für die Farbe des Blocks

**public** Block(**int** p\_x, **int** p\_y, **int** p\_width, **int** p\_higth, Color p\_color ) {

X= p\_x;

Y= p\_y;

WIDTH= p\_width;

HIGHT = p\_higth;

COLOR =p\_color;

}

**public** **boolean** Kolisionsabfrage(**int** x\_block, **int** y\_block, **int** x\_char, **int** y\_char) {

//also um zu bestimmen wann der char den block trifft o. überhaupt treffen kann

//Bedingung x von anfang cha > als anfang block und x ende char < ende block

**if**(x\_char >= x\_block && y\_char>= y\_block && y\_char<= y\_block+HIGHT) {

**if**(x\_char<= x\_block+WIDTH){

coin = **true**;

**return** **true**;

}**else** {

**if**(x\_char-40<= x\_block+WIDTH) {

coin = **true**;

**return** **true**;

}

}

}**else** {

**return** **false**;

}

**return** **false**;

}

**public** **int** getX\_Block() {

**return** X;

}

**public** **int** getY\_Block() {

**return** Y;

}

**public** **int** getWidth() {

**return** WIDTH;

}

**public** **int** getHight() {

**return** HIGHT;

}

**public** Color getFarbe() {

**return** COLOR;

}

**public** **boolean** coin() {

**return** coin;

}

}