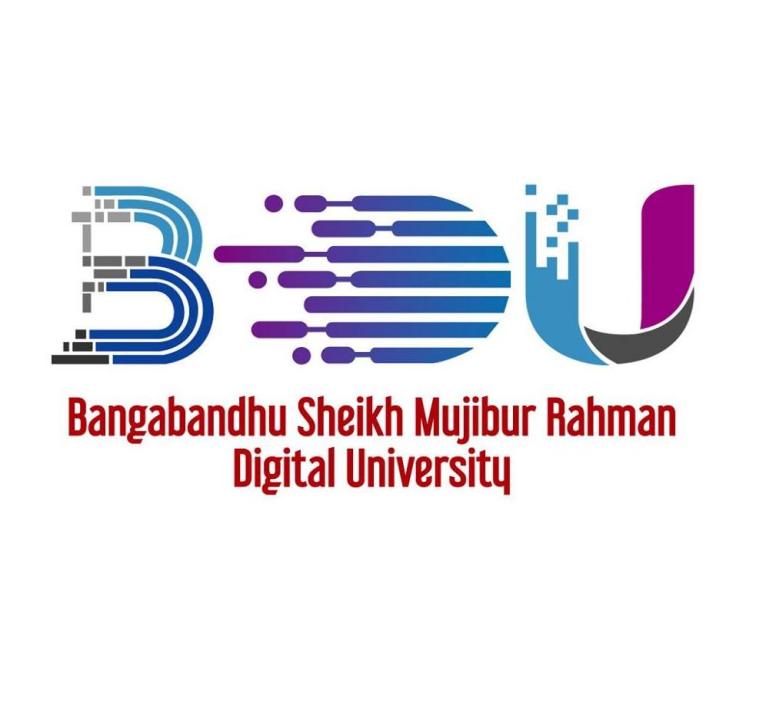
### শিক্ষা নিয়ে গড়বো দেশ তথ্য-প্রযুক্তির বাংলাদেশ

**Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh**



Software Development Project-02

**COURSE NO. PROG 112: Object Oriented Programming Sessional**

**SUBMITTED BY**

Name-Mobashira Mehajabin Arpita(2101008)

Srabonee Paul Tuli(2101015)

Mehrin Farzana(2101013)

Department of IRE

Session :2021-2022

Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

**SUBMITTED TO**

Md.Toukir Ahmed

Lecturer

Department of IRE, BDU

Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

**Date of Submission: 6 November,2023**

**Project title: Not-To-Bump!**

|  |  |
| --- | --- |
| **Name of team members** | **Portion of code in the project** |
| 1. Mobashira Mehajabin Arpita | 294-433 |
| 1. Srabonee Paul Tuli | 41-149 |
| 1. Mehrin Farzana | 153-293 |

**Code:**

//package cargame;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.io.IOException;

import java.util.Random;

import java.util.concurrent.TimeUnit;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.Timer;

// this class is used to build the graphics and the game logic

class CarGame extends JFrame implements KeyListener,ActionListener

{

    private int xpos=300; // x position of the car

    private int ypos=700; // y position of the car

    private ImageIcon car; // car image

    private Timer timer; // timer to update the screen

    Random random=new Random(); // random number generator

    private int num1=400,num2=0,num3=0; // x position of the obstacles

    private int tree1ypos=400,tree2ypos=-200,tree3ypos=-500,tree4ypos=100,tree5ypos=-300,tree6ypos=500; // y position of the obstacles

    private int roadmove=0; // y position of the road

    private int carxpos[]={100,200,300,400,500}; // x position of the car

    private int carypos[]= {-240,-480,-720,-960,-1200}; // y position of the car

    private int cxpos1=0,cxpos2=2,cxpos3=4; // x position of the car

    private int cypos1=random.nextInt(5),cypos2=random.nextInt(5),cypos3=random.nextInt(5); // y position of the car

    int y1pos=carypos[cypos1],y2pos=carypos[cypos2],y3pos=carypos[cypos3]; // y position of the car

    private ImageIcon car1,car2,car3; // car image

    private int score=0,delay=100,speed=90; // score,delay and speed of the game

    private ImageIcon tree1,tree2,tree3; // tree image

    private boolean rightrotate=false,gameover=false,paint=false; // boolean variables to control the game logic and the graphics

    // constructor to initialize the game

    public CarGame(String title)

    {

        super(title); // call the constructor of the parent class JFrame

        setBounds(300,10,700,700); // set the position and size of the frame

        setVisible(true); // make the frame visible

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // close the frame when the close button is clicked

        setLayout(null); // set the layout of the frame to null

        addKeyListener(this); // add the key listener to the frame

        setFocusable(true); // set the focus to the frame

        setResizable(false); // set the frame to be non resizable

    }

    // this method is used to paint the graphics on the screen

    public void paint(Graphics g)

    {

        g.setColor(new Color(0X82CD47)); // set the color of the grass

        g.fillRect(0, 0, 700, 700);  // draw the grass

        g.setColor(new Color(0X9F8772)); // set the color of the road

        g.fillRect(90,0,10,700); // draw the road

        g.fillRect(600, 0, 10, 700);

        g.fillRect(100, 0, 500, 700);

    // draw the road lines

    if(roadmove==0)

    {

        for(int i=0; i<=700; i+=100)

        {   // draw the road lines

            g.setColor(Color.white);

            g.fillRect(350, i,10, 70); //

        }

        roadmove=1; // set the roadmove to 1

    }

    else if(roadmove==1)

    { // draw the road lines again for the next frame

        for(int i=50; i<=700; i+=100)

        {

            g.setColor(Color.white);

            g.fillRect(350, i,10, 70);

        }

        roadmove=0; // set the roadmove to 0

    }

    try {

        tree1=new ImageIcon(ImageIO.read(getClass().getResource("tree.png"))); // load the tree image

    } catch (IOException e) {

        e.printStackTrace();

    }

    try {

        tree2=new ImageIcon(ImageIO.read(getClass().getResource("tree.png"))); // load the tree image

    } catch (IOException e) {

        e.printStackTrace();

    }

    try {

        tree3=new ImageIcon(ImageIO.read(getClass().getResource("tree.png"))); // load the tree image

    } catch (IOException e) {

        e.printStackTrace();

    }

    tree1.paintIcon(this, g, 0, tree1ypos); // draw the tree image on the screen

    num1=random.nextInt(500); // generate a random number

    tree1ypos+=50; // increment the y position of the tree

    tree2.paintIcon(this, g, 0,tree2ypos ); // draw the tree image on the screen

    tree2ypos+=50; // increment the y position of the tree

    tree3.paintIcon(this,g,0,tree3ypos); // draw the tree image on the screen

    tree3ypos+=50; // increment the y position of the tree

    tree1.paintIcon(this,g,600,tree4ypos);

    tree4ypos+=50;

    tree3.paintIcon(this, g,600,tree5ypos);

    tree5ypos+=50;

    tree2.paintIcon(this, g,600,tree6ypos);

    tree6ypos+=50;

    if(tree1ypos>700)

    { // if the tree goes out of the screen then reset the tree

        num1=random.nextInt(500); // generate a random number

        tree1ypos=-num1; // reset the y position of the tree

    }

    if(tree2ypos>700)

    {

        num1=random.nextInt(500);

        tree2ypos=-num1;

    }

    if(tree3ypos>700)

    {

        num1=random.nextInt(500);

        tree3ypos=-num1;

    }

    if(tree4ypos>700)

    { // if the tree goes out of the screen then reset the tree

        num1=random.nextInt(500);

        tree4ypos=-num1;

    }

    if(tree5ypos>700)

    {

        num1=random.nextInt(500);

        tree5ypos=-num1;

    }

    if(tree6ypos>700)

    { // if the tree goes out of the screen then reset the tree

        num1=random.nextInt(500);

        tree6ypos=-num1;

    }

        // load image for car

        try {

            car=new ImageIcon(ImageIO.read(getClass().getResource("gamecar3.png"))); // load the car image

        } catch (IOException e) {

            e.printStackTrace();

        }

        // car=new ImageIcon("gamecar1.png");

        car.paintIcon(this,g,xpos,ypos); // draw the car image on the screen

        ypos-=40;

        if(ypos<500)

        {

        ypos=500;

        }

        // load the opponent image for car

        try {

            car1=new ImageIcon(ImageIO.read(getClass().getResource("gamecar1.png")));

        } catch (IOException e) {

                e.printStackTrace();

        }

        // load the opponent image for car

        try {

            car2=new ImageIcon(ImageIO.read(getClass().getResource("gamecar2.png")));

        } catch (IOException e) {

                e.printStackTrace();

        }

        // load the opponent image for car

        try {

            car3=new ImageIcon(ImageIO.read(getClass().getResource("gamecar4.png")));

        } catch (IOException e) {

                e.printStackTrace();

        }

        car1.paintIcon(this, g, carxpos[cxpos1], y1pos); // draw the opponent car image on the screen

        car2.paintIcon(this, g, carxpos[cxpos2], y2pos);

        car3.paintIcon(this, g, carxpos[cxpos3], y3pos);

        y1pos+=50; // increment the y position of the opponent car

        y2pos+=50;

        y3pos+=50;

        if(y1pos>700)

        { // if the opponent car goes out of the screen then reset the opponent car

            cxpos1=random.nextInt(5); // generate a random number

            cypos1=random.nextInt(5);

            y1pos=carypos[cypos1]; // reset the y position of the opponent car

        }

        if(y2pos>700)

        { // if the opponent car goes out of the screen then reset the opponent car

            cxpos2++;

            if(cxpos2>4)

            {

                cxpos2=0;

            }

            cxpos2=random.nextInt(5);

            cypos2=random.nextInt(5);

            y2pos=carypos[cypos2];

        }

        if(y3pos>700)

        {

            cxpos3++;

            if(cxpos3>4)

            {

                cxpos3=0;

            }

            cxpos3=random.nextInt(5);

            cypos3=random.nextInt(5);

            y3pos=carypos[cypos3];

        }

        if(cxpos1==cxpos2 && cypos1>-100 && cypos2>-100)

        {

            cxpos1-=1;

            if(cxpos1<0)

            {

                cxpos1+=2;

            }

        }

        if(cxpos1==cxpos3&& cypos1>-100 && cypos3>-100)

        {

            cxpos3-=1;

            if(cxpos3<0)

            {

                cxpos3+=2;

            }

        }

        if(cxpos2==cxpos3&& cypos3>-100 && cypos2>-100)

        {

            cxpos2-=1;

            if(cxpos2<0)

            {

                cxpos2+=2;

            }

        }

        if(cxpos1<2 && cxpos2<2 && cxpos3<2)

        {

            if(cxpos1==0 && cxpos2==0 && cxpos3==1)

            {

                cxpos3++;

                cxpos2++;

            }

            else if(cxpos1==0 && cxpos2==1 && cxpos3==0)

            {

                cxpos3++;

                cxpos2++;

            }

            else if(cxpos1==1 && cxpos2==0 && cxpos3==0)

            {

                cxpos1++;

                cxpos2++;

            }

        }

        // if the opponent car hits the player car then reset the game

        if(y1pos<ypos && y1pos+175>ypos && carxpos[cxpos1]==xpos)

        {

        gameover=true;

        }

        if(y2pos<ypos && y2pos+175>ypos && carxpos[cxpos2]==xpos)

        {

        gameover=true;

        }

        if(y3pos<ypos  && y3pos+175>ypos && carxpos[cxpos3]==xpos)

        {

        gameover=true;

        }

        if(ypos<y1pos && ypos+175>y1pos && carxpos[cxpos1]==xpos)

        {

        gameover=true;

        }

        if(ypos<y2pos && ypos+175>y2pos && carxpos[cxpos2]==xpos)

        {

        gameover=true;

        }

        if(ypos<y3pos  && ypos+175>y3pos && carxpos[cxpos3]==xpos)

        {

        gameover=true;

        }

        //score

        g.setColor(Color.red);

        g.fillRect(120,35,220,50);

        g.setColor(Color.black);

        g.fillRect(125,40, 210, 40);

        g.setColor(Color.red);

        g.fillRect(385,35,180,50);

        g.setColor(Color.black);

        g.fillRect(390,40, 170, 40);

        g.setColor(Color.white);

        g.setFont(new Font("MV Boli",Font.BOLD,30));

        g.drawString("Score : "+score, 130, 67);

        g.drawString(speed+" Km/h", 400, 67);

        score++; // increment the score

        speed++; // increment the speed

        if(speed>140)

        { // if the speed is greater than 140 then reset the speed

            speed=240-delay;

        }

        if(score%50==0)

        { // if the score is divisible by 50 then increase the delay

            delay-=10;

            if(delay<60)

            {

                delay=60; // set the delay to 60

            }

        }

        //delay

        try

        {

            TimeUnit.MILLISECONDS.sleep(delay); // delay the game

        }

        catch (InterruptedException e) {

            e.printStackTrace();

        }

        if(y1pos<ypos && y1pos+175>ypos && carxpos[cxpos1]==xpos)

        {

        gameover=true;

        }

        if(y2pos<ypos && y2pos+175>ypos && carxpos[cxpos2]==xpos)

        {

        gameover=true;

        }

        if(y3pos<ypos  && y3pos+175>ypos && carxpos[cxpos3]==xpos)

        {

        gameover=true;

        }

        if(gameover)

        {

        g.setColor(Color.gray);

        g.fillRect(120, 210, 460, 200);

        g.setColor(Color.DARK\_GRAY);

        g.fillRect(130, 220, 440, 180);

        g.setFont(new Font("MV Boli",Font.BOLD,50));

        g.setColor(Color.red);

        g.drawString("Game Over !",210, 270);

        g.setColor(Color.white);

        g.setFont(new Font("MV Boli",Font.BOLD,30));

        g.drawString("Press Enter to Restart", 190, 340);

        if(!paint)

        {

            repaint();

            paint=true;

        }

        }

        else

        {

        repaint();

        }

    }

    public static void main(String args[])

    {

        CarGame c=new CarGame("Car Racing Game");

    }

    @Override

    public void keyPressed(KeyEvent e) {

        if(e.getKeyCode()==KeyEvent.VK\_LEFT && !gameover)

        { // if the left key is pressed then move the car to the left

            xpos-=100;

            if(xpos<100)

            {

                xpos=100; // set the car to the left most position

            }

        }

        if(e.getKeyCode()==KeyEvent.VK\_RIGHT&&!gameover)

        {   // if the right key is pressed then move the car to the right

            xpos+=100;

            if(xpos>500)

            {

                xpos=500; // if the car is at the right most position then don't move it

            }

        }

        if(e.getKeyCode()==KeyEvent.VK\_ENTER && gameover)

        { // if the game is over and the enter key is pressed then restart the game

            gameover=false;

            paint=false;

            cxpos1=0;

            cxpos2=2;

            cxpos3=4;

            cypos1=random.nextInt(5); // randomize the position of the opponent cars

            cypos2=random.nextInt(5);

            cypos3=random.nextInt(5);

            y1pos=carypos[cypos1]; // set the position of the opponent cars

            y2pos=carypos[cypos2];

            y3pos=carypos[cypos3];

            speed=90; // set the speed to 90

            score=0; // set the score to 0

            delay=100; // set the delay to 100

            xpos=300; // set the position of the player car to the center

            ypos=700;    // set the position of the player car to the bottom

        }

    }

    @Override

    public void keyReleased(KeyEvent arg0) {

        // TODO Auto-generated method stub

    }

    @Override

    public void keyTyped(KeyEvent e) {

        if(e.getKeyChar()=='a'&&!gameover)

        { // if the key pressed is 'a' then move the car left

            xpos-=100; // decrement the xpos by 100

        }

        if(e.getKeyChar()=='s'&&!gameover)

        { // if the key pressed is 's' then move the car right

            xpos+=100; // increment the xpos by 100

        }

        repaint();

    }

    @Override

    public void actionPerformed(ActionEvent arg0) {}

}

**Images:**

****

Image name: gamecar1.png

****

Image name: gamecar2.png

****

Image name: gamecar3.png

****

Image name:gamecar4.png

****

Image name: tree.png

**Output:**

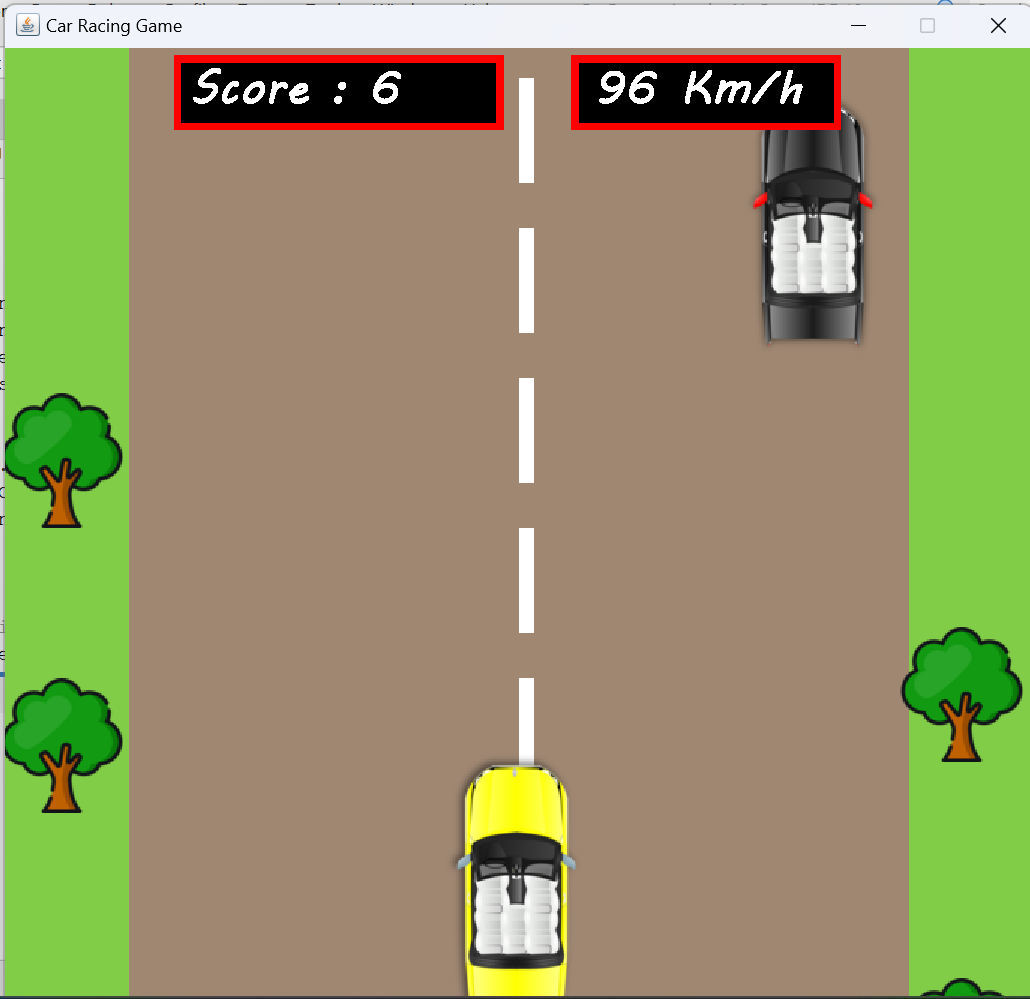


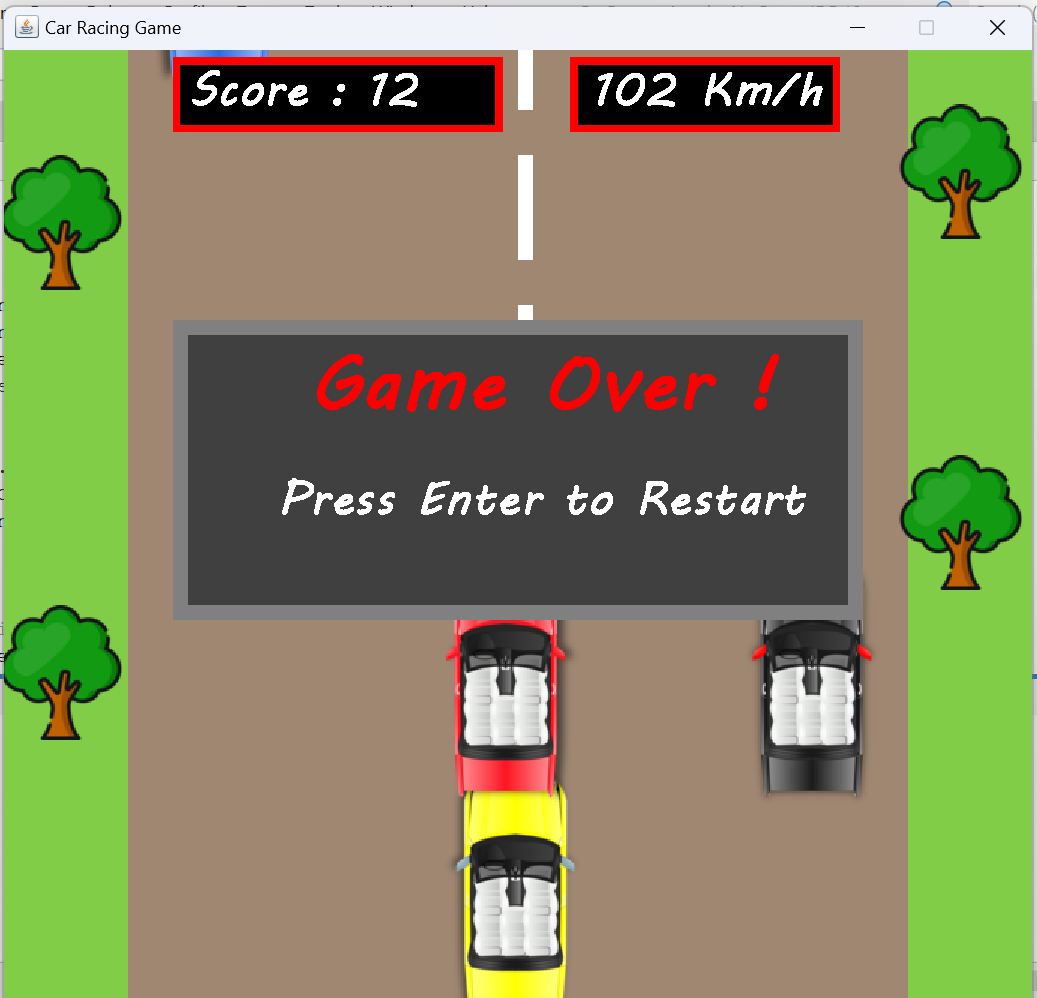
Fig1: Output window

Fig2: Output window

GitHub URL: [Visit our GitHub Repo.](https://github.com/MelyFaj/java-project/tree/main)