Introduction

The following is an example game written in Java based on the game called ‘CAR RACING’ which has been around since the earliest days of home computing and has re-emerged in recent year in mobile phones. I am starting making this game with the help of rectangles an d assuming them as cars, road and background in the final task I will replace them with the images.So it will look more beautiful.

Acknowledgement

In the first place, I would like to represent my greatest appreciation to my honorable sir Md. Mahmadul Hasan. He gives me plenty of sincere advice related to my final report and the project. Without his constant supports, I cannot complete it in such a short time successfully.

Abstract

This project aims to bring the fun and simplicity of car racing game with some new features. It will include computer controlled intelligent opponents whose aim will be to challenge the score. This project explores a new dimension in the traditional car racing game to make it more interesting and challenging. The simplicity of this game makes it an ideal candidate for a minor project as we can focus on advanced topics like player functionality and implementation of computer controlled intelligent opponents.

Objectives

The main objective of the development of the car racing game is to build a game by using an Object Oriented Analysis and Design Methodology that will enable the programmer to learn the right way of conducting analysis and development of software. The followings are the rest of the objectives for the development of the software.

Objectives of the project:

- The program will be able to read the inputs of the player and react accordingly with good response time.
- The program will be able to react accordingly to the different types of input the player enters during the duration the game is played.
- The program will be able to run with the least amount of system resources.
- The program will be able to run with background layers to provide the player with a more realistic gaming environment.

Programming Environment

We used several open source tools to develop this project:

NetBeans 8.2 IDE

NetBeans is a platform for Java desktop applications and also an integrated development environment (IDE) for developing with not only Java but also many other languages. The NetBeans IDE is written in Java and can run anywhere a Java Virtual Machine (JVM) is installed, including Windows, Mac OS, Linux, and Solaris. A Java Development Kit (JDK) is also required for Java development functionality.

Java Language

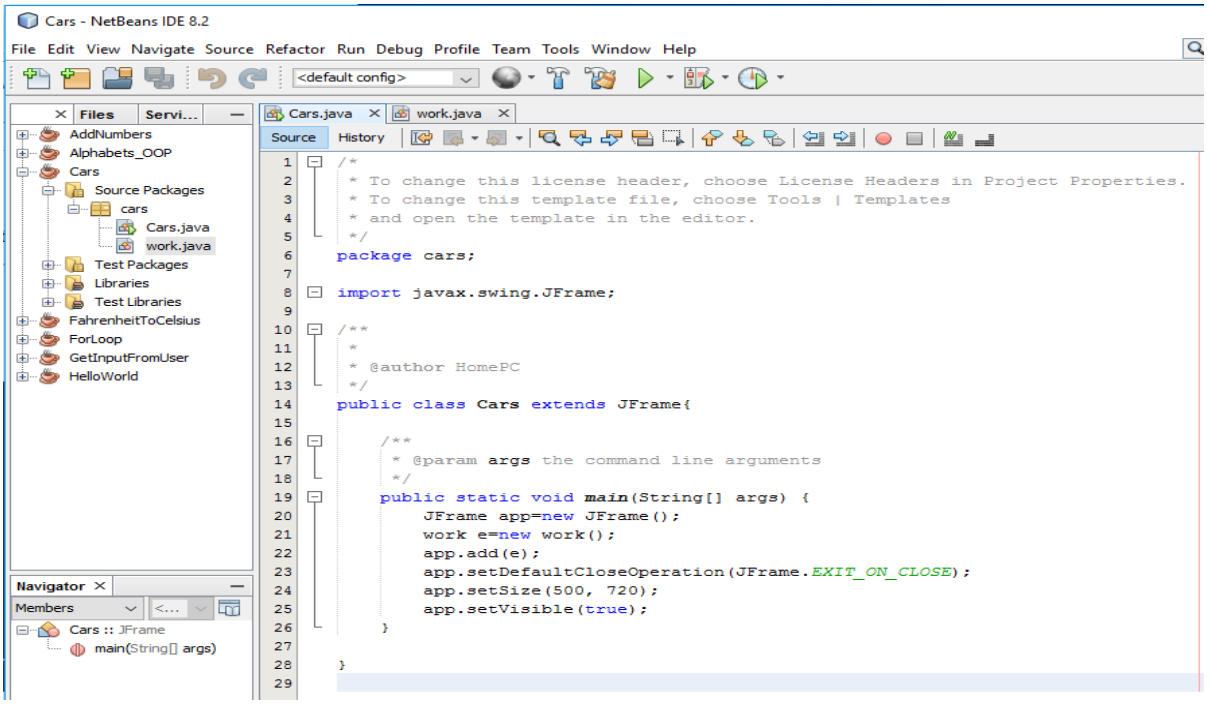
Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Graphics & Image tools

These graphics development tools were extensively used for the development of user Interface components. The illustrations presented in this report have also been prepared using these open source tools.

Interface

Step 1: Open Cars Project in NetBeans.



Step 2: Run Cars.java program.

After Run, Here shown the interface and user play the game. Yellow car is user car and white car is randomly computer controlled. User can use Arrow key. If

you use left arrow key then car goes too left side in road. If you use right arrow key then car goes to right side in road. If yellow car crushed in White car then game is over. Otherwise after 10 cars white car speed is increased.

Source Code

Cars.java (Main class)

```
package cars;

import javax.swing.JFrame;



public class Cars extends JFrame{



    public static void main(String[] args) {
        JFrame app=new JFrame();
        work e=new work();
        app.add(e);
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.setSize(500, 720);
        app.setVisible(true);
    }

}
```

work.java

```
package cars;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Rectangle;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.JPanel;
import javax.swing.Timer;


public class work extends JPanel implements ActionListener,KeyListener{

    private int space;
    private int width=80;
```

```
    private int height=70;
    private int speed;
    private int WIDTH=500;
    private int HEIGHT=650;
    private int move=20;
    private int count=1;
    private Rectangle car;
    private ArrayList <Rectangle> ocars;
    private ArrayList <Rectangle> line;
    private Random rand;
    BufferedImage bg;
    BufferedImage user;
    BufferedImage op1;
    BufferedImage op2;
    BufferedImage road;
    Boolean linef=true;
    Timer t;

 public work(){

        try {
            user=ImageIO.read(new
File("C:\\Users\\ASUS\\Documents\\NetBeansProjects\\Cars\\image\\user1.png"
));

            op1=ImageIO.read(new
File("C:\\Users\\ASUS\\Documents\\NetBeansProjects\\Cars\\image\\op1.png"))
;
```

```
            op2=ImageIO.read(new
File("C:\\Users\\ASUS\\Documents\\NetBeansProjects\\Cars\\image\\op3.png"))
;


            bg=ImageIO.read(new
File("C:\\Users\\ASUS\\Documents\\NetBeansProjects\\Cars\\image\\bg.png"));


            road=ImageIO.read(new
File("C:\\Users\\ASUS\\Documents\\NetBeansProjects\\Cars\\image\\road.png")
);
        }


catch (IOException ex) {
            Logger.getLogger(work.class.getName()).log(Level.SEVERE, null, ex);
        }


        t=new Timer(10,this);
        rand=new Random();
        ocars=new ArrayList<Rectangle>();
        line=new ArrayList<Rectangle>();
        car=new Rectangle(WIDTH/2-90,HEIGHT-100,width,height);
        System.out.println(car.y);
        space=300;
        speed=1;
        addKeyListener(this);
        setFocusable(true);
        addocars(true);
        addocars(true);
        addocars(true);
```

```
    addocars(true);
    addlines(true);
    addlines(true);
    addlines(true);
    addlines(true);
    addlines(true);
    addlines(true);
    addlines(true);
    addlines(true);
    t.start();
}

public void addlines(Boolean first){

    int x=WIDTH/2-2;
    int y=700;
    int width=4;
    int height=100;
    int sp=130;

    if(first){
        line.add(new Rectangle(x,y-(line.size()*sp),width,height));
    }
      else{
        line.add(new Rectangle(x,line.get(line.size()-1).y-sp,width,height));
    }
}
```

```
    public void addocars(boolean first){

        int positionx=rand.nextInt()%2;
        int x=0;
        int y=0;
        int Width=width;
        int Height=height;

          if(positionx==0){
            x=WIDTH/2-90;
        }
          else{
            x=WIDTH/2+10;
        }

          if(first){
            ocars.add(new Rectangle(x,y-100-(ocars.size()*space),Width,Height));
        }
          else{
            ocars.add(new Rectangle(x,ocars.get(ocars.size()-1).y-
300,Width,Height));
        }
    }

public void paintComponent(Graphics g){

        super.paintComponents(g);
        g.drawImage(bg, 0, 0, null);
```

```
    g.drawImage(road, WIDTH/2-125,0,null);
    g.setColor(Color.WHITE);

    for(Rectangle rect:line){
        g.fillRect(rect.x, rect.y, rect.width, rect.height);
    }

    g.drawImage(user, car.x, car.y, null);

    g.setColor(Color.MAGENTA);

    for(Rectangle rect:ocars){

        if(rand.nextInt()%1==0){
            g.drawImage(op1, rect.x, rect.y, null);
        }
         else{
            g.drawImage(op2, rect.x, rect.y,null);
        }

    }


}
```

```
public void actionPerformed(ActionEvent e) {
    Rectangle rect;
    count++;

    for(int i=0;i<ocars.size();i++){
       rect=ocars.get(i);

       if(count%10000==0){

          if(move<100){
             speed++;
             move+=100;
          }

       }
       rect.y+=speed;
    }

    //cars crashinng with oponents

    for(Rectangle r:ocars){

       if(r.intersects(car)){
          car.y=r.y+height;
       }

    }
```

```
for(int i=0;i<ocars.size();i++){
    rect=ocars.get(i);

    if(rect.y+rect.height>HEIGHT){
        ocars.remove(rect);
        addocars(false);
    }

}
for(int i=0;i<line.size();i++){
    rect=line.get(i);

    if(count%100000==0){
        speed++;
    }

    rect.y+=speed;
}

for(int i=0;i<line.size();i++){
    rect=line.get(i);

    if(rect.y>HEIGHT){
        line.remove(rect);
        addlines(false);
    }
}
```

```
        repaint();
    }

    //moveing

    public void moveup(){

        if(car.y-move<0){
            System.out.println("\b");
        }
          else{
            car.y-=move;
        }
    }

public void movedown(){

        if(car.y+move+car.height>HEIGHT-1){
            System.out.println("\b");
        }
          else{
            car.y+=move;
        }
    }
```

```
public void moveleft(){

    if(car.x-move<WIDTH/2-90){
        System.out.println("\b");
    }
      else{
        car.x-=move;
    }
  }

 public void moveright(){

    if(car.x+move>WIDTH/2+10){
        System.out.println("\b");
    }
      else{
        car.x+=move;
    }
  }

  @Override
  public void keyTyped(KeyEvent e) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
  }

  @Override
  public void keyPressed(KeyEvent e) {
```

```
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public void keyReleased(KeyEvent e) {
        int key=e.getKeyCode();
        switch(key){
            case KeyEvent.VK_UP:
                moveup();
                break;
            case KeyEvent.VK_DOWN:
                movedown();
                break;
            case KeyEvent.VK_LEFT:
                moveleft();
                break;
            case KeyEvent.VK_RIGHT:
                moveright();
               break;
            default:
                break ;
        }
    }

}
```

Future Enhancements

- Level Extension
- Improve Graphical Representation
- Introduce new game features
- Introduce new environment and scenes

Conclusion

A software project means a lot of experience. In this section I summarize the experience gained by project development “Car Racing”. This project has sharpened our concept of Game engine, animation and the software-hardware interface. This project not only tested our technical skills but also our temperament. There were times that we almost lost hope but we recovered through constant concentration and hard work.

References:

1. **https://www.youtube.com/playlist?list=PLNOVFoZQ-AVWrStGwkc5IryBgsXh3O6bf**
2. **http://www.java-gaming.org/topics/my-java-games-with-source-code-tetris-mineweeper-tron-etc/26759/view.html**
3. **https://gist.github.com/antimatter15/822345**
4. **https://www.slideshare.net/NadiaIIT/final-project-report-of-a-game?from_action=save**
5. **Programming with Java A Premier – E.Balagurusamy**