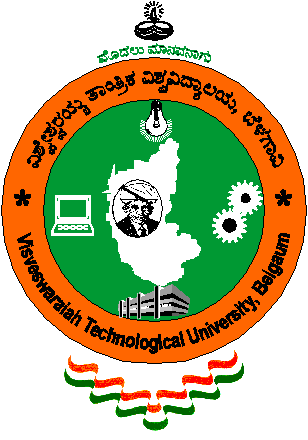
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELGAUM - 590014**



##### **A Report on**

**“Arcade Game Championship”**

##### **Submitted in the partial fulfillment for Computer Graphics Laboratory: An Open Ended Problem.**

##### 

##### **Submitted By:**

**NAME (USN)**

**SUMANTH M 1SI16CS115**

**SIDDESH M G 1SI16CS104**

**Under the guidance of:**

**Mrs. Prithvi M G Mr. Raghavendra M**

**Assistant Profrssor Assistant Professor**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103**

**(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Recognized by AICTE and Accredited by NBA, New Delhi)**

**2018-2019**

**ACKNOWLEDGEMENT**

A successful project is a fruitful culmination of efforts by many people, few directly involved and others who encourage and extend their support from the background, in silence. We avail this opportunity to express our gratitude to all those who have contribute their valuable assistance and support in making our project a success.

We express our deep sense of gratitude, indebtedness and sincere salutations to His Holiness **Dr. Sree Sree Sree Shivakumara Swamigalu**, President, Sree Siddaganga Education Society, for being a constant source of inspiration.

We express our gratitude to **Dr. K P Shivananda**, Principal, Siddaganga Institute of Technology and **Dr. R Sumathi**, Professor and Head of Department, Computer Science and Engineering, for fostering an excellent academic environment.

We are thankful, immensely and remain indebted to **Prof. Prithvi M G** and **Prof. Raghavendra M,** Assistant Professor Department of Computer Science and Engineering, for being our guide, who in spite of their busy schedule was a remarkable and constant source of motivation to us. His guidance, encouragement and constructive suggestions during difficult phases of the project gave us impetus to carry out the task. He indeed was the driving force during the entire course of successful completion of project.

**TABLE OF CONTENTS**

1. ABSRTRACT………………………………………………….04
2. INTRODUCTION……………………………………………..05
3. REQUIREMENT SPECIFICATION………………………….06
4. IMPLEMENTATION…………………………………………07
5. RESULTS………………………………………………………
6. CONCLUSION………………………………………………...

**ABSTRACT**

The computer games are the most popular thing among the youths. The Arcade Racing games are mostly addictive as they are never ending and they can never be completed. This induces a tendency in the player to play again and again to beat his/her own best score.

The game can be played by people of all the ages as it is simple and amusing to play. The game with its impressive game play will entertain the people and acts as a stress buster at time.

The game considered here is an arcade car game which has a car that is running on a straight main road with other cars also in the road moving in same direction. We can control the car using the keyboard key such as arrow keys. As the car moves a particular distance the score increases. The score continuously keeps on increasing until the game completes. The game over situation is that the player clashes/crashes the car with the other car that is moving along with it in the same direction.

**INTRODUCTION**

This is an arcade game with unique gaming experience. It consists of the basic computer graphics concepts for building the game where there will be car continuously moving on the highway which will face many hurdles on its way.

There is no practical end to the game where the player wins. The game is designed such a way that he has to defeat his own previous record and win. Thus the game is called the arcade game.

There will be many other cars in the highway the player must play in such a way that he should not collide with any cars. As long he is not colliding the points will be increasing continuously for a frequent amount of time.

But once the car collides then the game ends and the current score is displayed on the screen.If that score is greater than all previous score then it will be considered as the highest score for the particular player.

**REQUIREMENT AND SPECIFICATION**

**REQUIREMENT**

**OpenGL libraries**

The OpenGL specification describes an abstract [API](https://en.wikipedia.org/wiki/Application_programming_interface) for drawing 2D and 3D graphics. Although it is possible for the API to be implemented entirely in software, it is designed to be implemented mostly or entirely [in hardware](https://en.wikipedia.org/wiki/Hardware_acceleration).

The API is defined as a set of [functions](https://en.wikipedia.org/wiki/Subroutine) which may be called by the client program, alongside a set of [named integer constants](https://en.wikipedia.org/wiki/Enumerated_type) (for example, the constant GL\_TEXTURE\_2D, which corresponds to the [decimal](https://en.wikipedia.org/wiki/Decimal) number 3553). Although the function definitions are superficially similar to those of the programming language [C](https://en.wikipedia.org/wiki/C_(programming_language)), they are language-independent. As such, OpenGL has many [language bindings](https://en.wikipedia.org/wiki/Language_binding), some of the most noteworthy being the [JavaScript](https://en.wikipedia.org/wiki/JavaScript) binding [WebGL](https://en.wikipedia.org/wiki/WebGL) (API, based on [OpenGL ES 2.0](https://en.wikipedia.org/wiki/OpenGL_ES_2.0), for 3D rendering from within a [web browser](https://en.wikipedia.org/wiki/Web_browser)); the C bindings [WGL](https://en.wikipedia.org/wiki/WGL_(software)), [GLX](https://en.wikipedia.org/wiki/GLX) and [CGL](https://en.wikipedia.org/wiki/Core_OpenGL); the C binding provided by [iOS](https://en.wikipedia.org/wiki/IOS); and the [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and C bindings provided by [Android](https://en.wikipedia.org/wiki/Android_(operating_system)).

In addition to being language-independent, OpenGL is also cross-platform. The specification says nothing on the subject of obtaining, and managing an OpenGL context, leaving this as a detail of the underlying [windowing system](https://en.wikipedia.org/wiki/Windowing_system). For the same reason, OpenGL is purely concerned with rendering, providing no APIs related to input, audio, or windowing.

**Text Editor(Code Blocks)**

Code::Blocks is a [free](https://en.wikipedia.org/wiki/Free_software), [open-source](https://en.wikipedia.org/wiki/Open-source_software) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) that supports multiple [compilers](https://en.wikipedia.org/wiki/Compilers) including [GCC](https://en.wikipedia.org/wiki/GNU_Compiler_Collection), [Clang](https://en.wikipedia.org/wiki/Clang) and [Visual C++](https://en.wikipedia.org/wiki/Microsoft_Visual_C++). It is developed in [C++](https://en.wikipedia.org/wiki/C++) using [wxWidgets](https://en.wikipedia.org/wiki/WxWidgets) as the [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C++), and [Fortran](https://en.wikipedia.org/wiki/Fortran). It has a custom [build system](https://en.wikipedia.org/wiki/Build_automation) and optional [Make](https://en.wikipedia.org/wiki/Make_(software)) support.

**IMPLEMENTATION**

**#include<glut.h>**

**#include<iostream>//for strlen**

**#include<stdlib.h>**

**int i,q;**

**int score = 0;//for score counting**

**int screen = 0;**

**bool collide = false;//check if car collide to make game over**

**char buffer[10];**

**int vehicleX = 200, vehicleY = 70;**

**int ovehicleX[4], ovehicleY[4];**

**int divx = 250, divy = 4, movd;**

**void drawText(char ch[],int xpos, int ypos)//draw the text for score and game over**

**{**

**int numofchar = strlen(ch);**

**glLoadIdentity ();**

**glRasterPos2f( xpos , ypos);**

**for (i = 0; i <= numofchar - 1; i++)**

**{**

**glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, ch[i]);//font used here, may use other font also**

**}**

**}**

**void drawTextNum(char ch[],int numtext,int xpos, int ypos)//counting the score**

**{**

**int len;**

**int k;**

**k = 0;**

**len = numtext - strlen (ch);**

**glLoadIdentity ();**

**glRasterPos2f( xpos , ypos);**

**for (i = 0; i <=numtext - 1; i++)**

**{**

**if ( i < len )**

**glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,'0');**

**else**

**{**

**glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, ch[k]);**

**k++;**

**}**

**}**

**}**

**void ovpos()**

**{**

**glClearColor(0,0,1,0);**

**for(i = 0; i < 4; i++)**

**{**

**if(rand() % 2 == 0)**

**{**

**ovehicleX[i] = 200;**

**}**

**else**

**{**

**ovehicleX[i] = 300;**

**}**

**ovehicleY[i] = 1000 - i \* 160;**

**}**

**}**

**void drawRoad()**

**{**

**glBegin(GL\_QUADS);**

**glColor3f(0.5,0.5,0.5);**

**glVertex2f(250 - 100, 500);**

**glVertex2f(250 - 100, 0);**

**glVertex2f(250 + 100, 0);**

**glVertex2f(250 + 100, 500);**

**glEnd();**

**}**

**void drawDivider()//black patch drawn in middle of road**

**{**

**glLoadIdentity();**

**glTranslatef(0, movd, 0);**

**for(i = 1; i <= 10; i++)**

**{**

**glColor3f(0, 0, 0);**

**glBegin(GL\_QUADS);**

**glVertex2f(divx - 5, divy \* 15 \* i + 18);**

**glVertex2f(divx - 5, divy \* 15 \* i - 18);**

**glVertex2f(divx + 5, divy \* 15 \* i - 18);**

**glVertex2f(divx + 5, divy \* 15 \* i + 18);**

**glEnd();**

**}**

**glLoadIdentity();**

**}**

**void drawVehicle()//car for racing**

**{**

**glPointSize(10.0);**

**glBegin(GL\_POINTS);//tire**

**glColor3f(0,0,0);**

**glVertex2f(vehicleX - 25, vehicleY + 16);**

**glVertex2f(vehicleX + 25, vehicleY + 16);**

**glVertex2f(vehicleX - 25, vehicleY - 16);**

**glVertex2f(vehicleX + 25, vehicleY - 16);**

**glEnd();**

**glBegin(GL\_QUADS);**

**glColor3f(1,0,0);//middle body**

**glVertex2f(vehicleX - 25, vehicleY + 20);**

**glVertex2f(vehicleX - 25, vehicleY - 20);**

**glVertex2f(vehicleX + 25, vehicleY - 20);**

**glVertex2f(vehicleX + 25, vehicleY + 20);**

**glEnd();**

**glBegin(GL\_QUADS);//up body**

**glColor3f(0,0,1);**

**glVertex2f(vehicleX - 23, vehicleY + 20);**

**glVertex2f(vehicleX - 19, vehicleY + 40);**

**glVertex2f(vehicleX + 19, vehicleY + 40);**

**glVertex2f(vehicleX + 23, vehicleY + 20);**

**glEnd();**

**glBegin(GL\_QUADS);//down body**

**glColor3f(0,0,1);**

**glVertex2f(vehicleX - 23, vehicleY - 20);**

**glVertex2f(vehicleX - 19, vehicleY - 35);**

**glVertex2f(vehicleX + 19, vehicleY - 35);**

**glVertex2f(vehicleX + 23, vehicleY - 20);**

**glEnd();**

**}**

**void drawOVehicle()//other cars**

**{**

**for(i = 0; i < 4; i++)**

**{**

**glPointSize(10.0);**

**glBegin(GL\_POINTS);//tire**

**glColor3f(0,0,0);**

**glVertex2f(ovehicleX[i] - 25, ovehicleY[i] + 16);**

**glVertex2f(ovehicleX[i] + 25, ovehicleY[i] + 16);**

**glVertex2f(ovehicleX[i] - 25, ovehicleY[i] - 16);**

**glVertex2f(ovehicleX[i] + 25, ovehicleY[i] - 16);**

**glEnd();**

**glBegin(GL\_QUADS);**

**glColor3f(0.99609, 0.83984, 0);//middle body**

**glVertex2f(ovehicleX[i] - 25, ovehicleY[i] + 20);**

**glVertex2f(ovehicleX[i] - 25, ovehicleY[i] - 20);**

**glVertex2f(ovehicleX[i] + 25, ovehicleY[i] - 20);**

**glVertex2f(ovehicleX[i] + 25, ovehicleY[i] + 20);**

**glEnd();**

**glBegin(GL\_QUADS);//up body**

**glColor3f(0,1,0);**

**glVertex2f(ovehicleX[i] - 23, ovehicleY[i] + 20);**

**glVertex2f(ovehicleX[i] - 19, ovehicleY[i] + 40);**

**glVertex2f(ovehicleX[i] + 19, ovehicleY[i] + 40);**

**glVertex2f(ovehicleX[i] + 23, ovehicleY[i] + 20);**

**glEnd();**

**glBegin(GL\_QUADS);//down body**

**glColor3f(0,1,0);**

**glVertex2f(ovehicleX[i] - 23, ovehicleY[i] - 20);**

**glVertex2f(ovehicleX[i] - 19, ovehicleY[i] - 35);**

**glVertex2f(ovehicleX[i] + 19, ovehicleY[i] - 35);**

**glVertex2f(ovehicleX[i] + 23, ovehicleY[i] - 20);**

**glEnd();**

**ovehicleY[i] = ovehicleY[i] - 8;**

**if(ovehicleY[i] > vehicleY - 25 - 25 && ovehicleY[i] < vehicleY + 25 + 25 && ovehicleX[i] == vehicleX)**

**{**

**collide = true;**

**}**

**if(ovehicleY[i] < -25)**

**{**

**if(rand() % 2 == 0)**

**{**

**ovehicleX[i] = 200;**

**}**

**else**

**{**

**ovehicleX[i] = 300;**

**}**

**ovehicleY[i] = 600;**

**}**

**}**

**}**

**void Specialkey(int key, int x, int y)//allow to use navigation key for movement of car**

**{**

**switch(key)**

**{**

**case GLUT\_KEY\_UP:**

**for(i = 0; i <4; i++)**

**{**

**ovehicleY[i] = ovehicleY[i] - 10;**

**}**

**movd = movd - 20;**

**break;**

**case GLUT\_KEY\_DOWN:**

**for(i = 0; i <4; i++)**

**{**

**ovehicleY[i] = ovehicleY[i] + 10;**

**}**

**movd = movd + 20;**

**break;**

**case GLUT\_KEY\_LEFT:vehicleX = 200;**

**break;**

**case GLUT\_KEY\_RIGHT:vehicleX = 300;**

**break;**

**}**

**glutPostRedisplay();**

**}**

**void Normalkey(unsigned char key, int x, int y)**

**{**

**switch(key)**

**{**

**/\*case '1':if(screen == 0)**

**screen=1;**

**break;**

**case '2':if(screen == 1)**

**screen=2;**

**break;\*/**

**case 27:exit(0);**

**}**

**}**

**void init()**

**{**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**gluOrtho2D(0, 500, 0, 500);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

**void display()**

**{**

**/\*if(screen == 0)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(1,1,1);**

**drawText("WELCOME", 170, 290);**

**drawText("PLAYER", 170, 250);**

**drawText("Press 1 to continue", 170, 150);**

**glutSwapBuffers();**

**}**

**else if(s == 1)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(1,1,1);**

**drawText("Use w to accelerate", 170, 290);**

**drawText("Use a to move left", 170, 250);**

**drawText("Use d to move right", 170, 210);**

**drawText("Press 2 to continue", 170, 150);**

**glutSwapBuffers();**

**}**

**else**

**{\*/**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**drawRoad();**

**drawDivider();**

**drawVehicle();**

**drawOVehicle();**

**movd = movd - 16;**

**if(movd < - 60)**

**{**

**movd = 0;**

**}**

**score = score + 1;**

**glColor3f(1,1,1);**

**drawText("Score:", 360,455);**

**itoa (score, buffer, 10);**

**drawTextNum(buffer, 6, 420,455);**

**glutSwapBuffers();**

**for(q = 0; q<= 10000000; q++){;}**

**if(collide == true)**

**{**

**glColor3f(0,0,0);**

**drawText("Game Over", 200,250);**

**glutSwapBuffers();**

**getchar();**

**//exit(0);**

**}**

**//}**

**}**

**void main(int argc, char \*\*argv)**

**{**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_RGB|GLUT\_DOUBLE);**

**glutInitWindowPosition(100,100);**

**glutInitWindowSize(800,500);**

**glutCreateWindow("2D Car Racing game");**

**ovpos();**

**init();**

**glutDisplayFunc(display);**

**glutSpecialFunc(Specialkey);**

**glutKeyboardFunc(Normalkey);**

**glutIdleFunc(display);**

**glutMainLoop();**

**}**

**RESULT**

**CONCLUSION**

This is a specialised game designed for the entertainment purpose.It is very user friendly and has an interactive game play this helps the players to play the game easily.

The game is an arcade game so there is no practical end to the game. The player can play this game as many times he want and can achieve the target.

The hurdles makes the game more interesting and challenging and attracts the players to play the game more often.