**University of Dhaka**



**Department of Computer Science and Engineering**

Project Report

“Race – Where It All Begins”

Submitted By –

Pritom Saha

Roll – 56

Submitted To -

Dr.Muahammad Asif Hossain Khan

Associate Professor, Dept. of CSE, DU

Ms. Anna Fariha

Lecturer, Dept. of CSE, DU

# Table Of Contents:

**1.Executive Summary** 3

**2. Objective** 3

**3. The Game** 3

**4. Rules** 4

**5. Patterns** 4

**6. Graphical User Interface** 5-11

**7. Flow Chart of the Game** 12

**8.** **User Defined Functions if the Game** 12-13

**9. Built-In Functions** 14-15

**10. Program Coding**…………..……….…………………………………………………………………16-49

**11. Header Used In The Code :**

**EnemyCars.h :**

**Car\_My.h:** 50-67

**12. Limitations :** 68

**13. Constraints:** 68

**13. Conclusion::** 68

“Race – Where It All Begins”

**1. Executive Summary:**

Race - “Where It All Begins” is a game to be played by the players who wants only better graphics and better artwork. It is a 2D game, we used BGI library to build the game.

There are impressive artworks in different levels. The Players will love these artworks, especially the views beside the road! The graphics of the game isn’t up to the mark, because the utility is very old.

In different levels there are enemy-cars and some bonus enemy cars. If a player touches a car then it will lose scores. It is the kind of game which requires the skill of concentration as the player may require choosing his path wisely to avoid the enemies.

**2. Objective:**

The objective of this project was to create a game using the BGI library and the game chosen for a Car Racing game. The game was implemented in C/C++. The options to be made in the Main Menu were New Game, Instructions, High-Score and Credits. There are Exactly Eight Levels of the game.

**3. The Game:**

This game is played by a Red car. A car that will go on a journey which the player will never forget! A car that can move up, down, left, right. A car that can defy all the odds, all the obstacles that’s been put in front of it.

The enemy cars will want to end the player’s dream run, but the player will try to avoid the enemy cars. If it an enemy car touches the main Car, then the Car blasts and then the car is renewed by some antidote and that’s how the car will lose some scores. There are Eight lives.

So, the player has to be careful! Or the car will never end the journey!

**4. Rules:**

* The Car can only move in the four cardinal directions if there are free spaces in front of it. If the input direction is blocked by enemy-car, then the car will blast and lose points sequentially.
* If the car is hit by an enemy car, then the car will blast and then reincarnate. This will happen 8 times before the game is over!
* Each level will contain 60-80-90 points. With every 60 points the level will change! Going to level 8, then the increase of levels will stop. And the enemy cars will try to make your life a living hell!

**5. Patterns:**

If the player observes carefully he will notice certain patterns in the game. These are as below:

The Car turns into a various color car if it hits the Enemy car.

The Enemy cars will try to stop you from moving forward, you have to be aware of that fact!

The game can be restarted if the player runs out of life, which’s EIGHT.

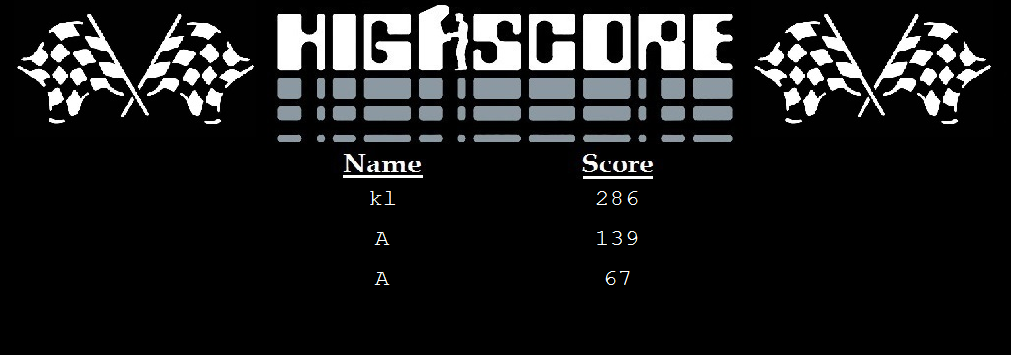
The Player can also know who’s behind him and who’s ahead of him in the High-Score section.

**6. Graphical User Interface :**

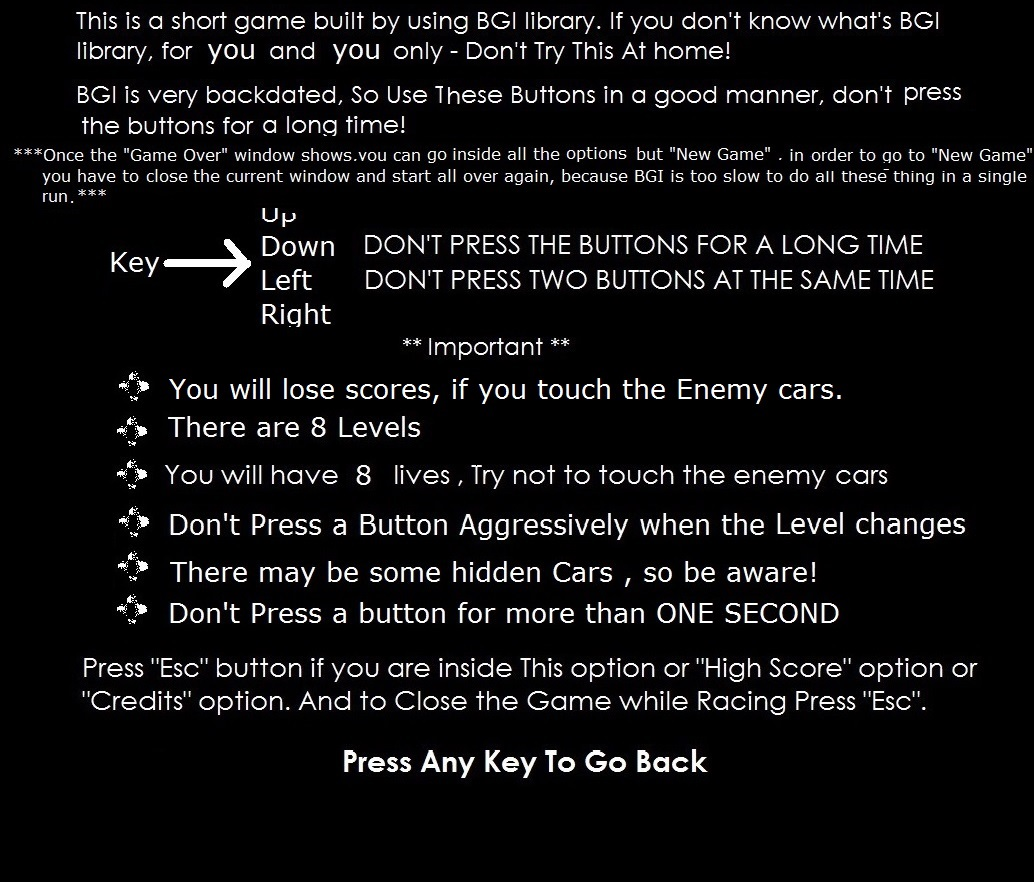
Menu and Front page of the Game :

****

High Score section :

****

Instructions :

****

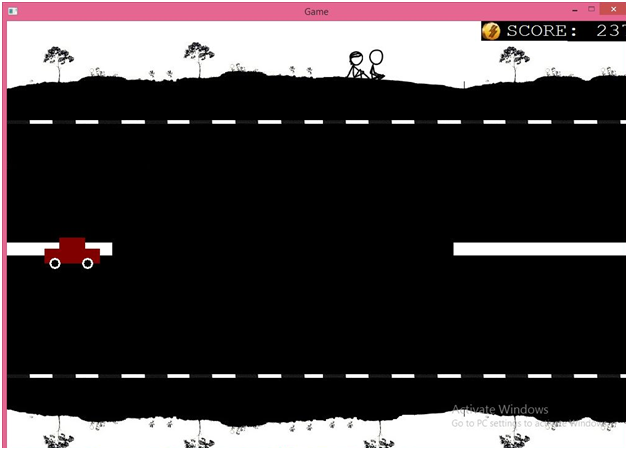
New Game :



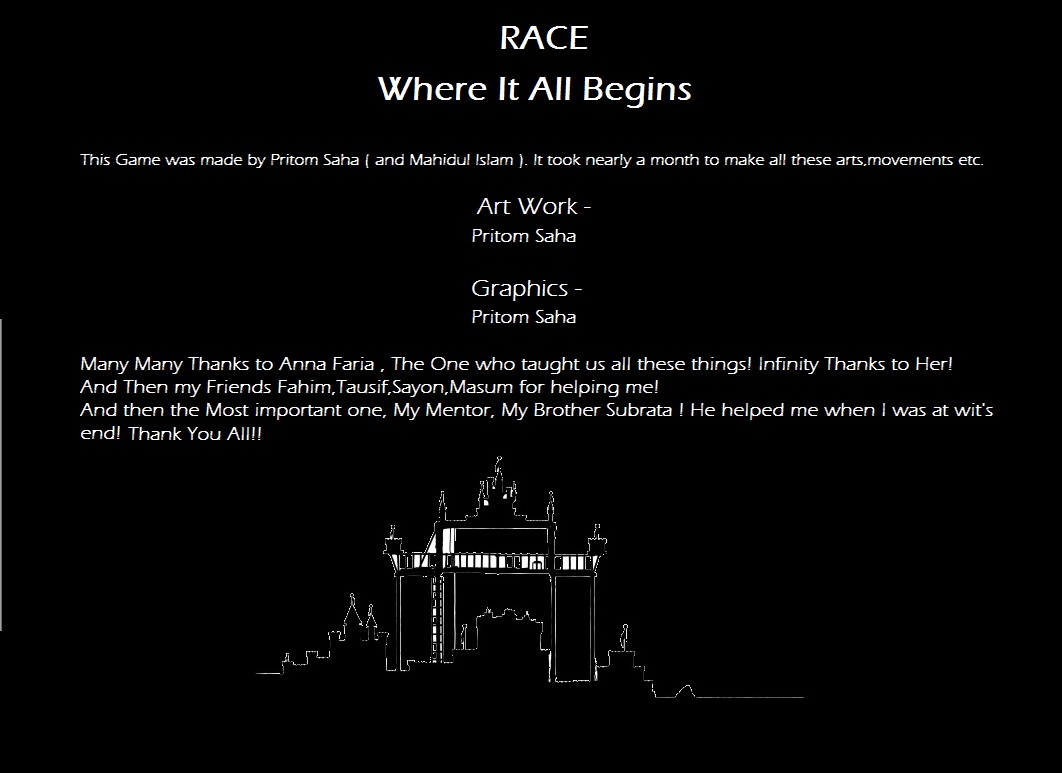
**Level -1 :**



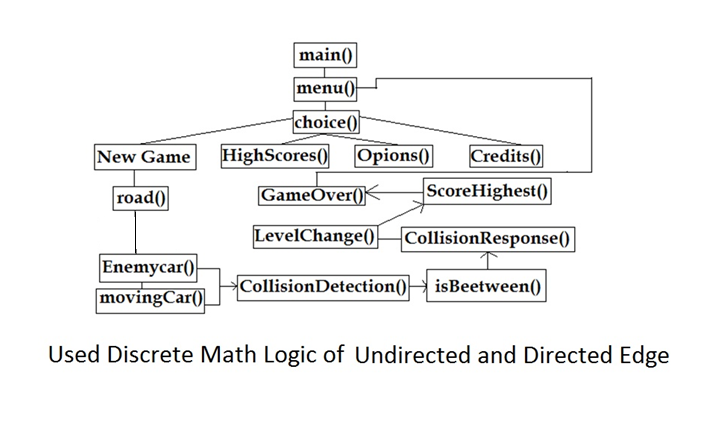
**Level 6 :**

****

Credits :



**7. Flow Chart of the Game –**

****

**8. User Defined Functions if the Game :**

1. int main() – Where it all Begins , The begins here!
2. void createMenu()– The Menu Stars here.
3. void choice1() – New Game Stars.
4. void choice2() – Shows High score.
5. void choice3() – Shows the instructions of the game , mainly how to play the game and the controls of the game.
6. void choice4() – shows you the Credits of the game.
7. void Upper(inta,intb,intc,int d) – shows the Images of the Grass, Sea, Hills of Upper Part of the window.
8. void Lower(inta,intb,intc,int d) – shows the Images of the Grass, Sea, Hills of LowerPart of the window.
9. voidMovingRoad() and void MovingRoad2() – Shows the white part of the road.
10. void Road() – The Pavement of the road.
11. void Score() – shows the Score constantly.
12. voidMovingScore() – The Original Score that changes.
13. void EnemyCarReInCarnated1() and void EnemyCarReInCarnated2() - Re assigns the values of the variables so that they can be used again and again!
14. voidEnemyCar() – Draws The Enemycar.
15. voidsortIfHighest() – Shows the High Score if the new one is higher than the previous ones.
16. voidCollisionDetection() – returns true(1) if the collision between the car and enemy car happens.
17. voidCollisionResponse() – what would happen after the collision , it is created to react to that collision.
18. voidLevelChange() – changes level if the score meets the required need!
19. void test() – Takes UserName Input.
20. void showHighScores() – Shows the High Score if the user wants to see the high scores of the game.
21. void Credits() – Shows the Credits of the game.
22. voidwriteToFile() – Immediately writes the username and score when the game ends.
23. voidgameOver() – Shows an image if the three lives are over.
24. boolisBetween() – detects if the car and the enemy car touches each other.
25. voidMoveTheRoad() – Moves Road.
26. void RoadReInCarnated1() and void RoadReInCarnated2() – Re assigns the values of the road variables.
27. void restart() – Restarts every variable if the game is restarted within the game.
28. voidEnemyCar() – This is where every part of the Enemy Car is drawn!
29. voidmyCar() – This is a Structure , where every part of the car is drawn.
30. voidMovingCar() – This is the main place where the Car moves.
31. void CarrReinCarnation1() and void CarrReinCarnation2() – A place to re assign the values of the car if the game restarts.
32. StructMyStruct{} – A structure used to sort the Points!
33. void EnemycarReincarnated – A function that re assigns the value of the enemy cars when they go off the screen.

**9. Built-In Functions –**

1. **void bar**(int left, int top, int right, int bottom) - bar draws a filled-in, rectangular, two-dimensional bar. The bar is filled using the current fill pattern and fill color. bar does not outline the bar; to draw an outlined two-dimensional bar, use bar3d with depth equal to 0.

The upper left and lower right corners of the rectangle are given by (left, top) and (right, bottom), respectively. The coordinates refer to pixels.

1. **void cleardevice**(void) - cleardevice erases (that is, fills with the current background color) the entire graphics screen and moves the

CP (current position) to home (0,0).

1. **void delay**(intmillisec) – The delay function is available in the winbgimimplementation of BGI graphics. You do not need to include conio.h; just include graphics.h. The function pauses the computation for the the specified number of milliseconds.
2. **void fillellipse**(int x, int y, intxradius, intyradius) - Draws an ellipse using (x,y) as a center point and xradius and yradius as the horizontal and vertical axes, and fills it with the current fill color and fill pattern.
3. **void circle**(int x, int y, int radius) - circle draws a circle in the current drawing color with its center at (x,y) and the radius given by radius.
4. **int getch**(void) - The getch function is available in the winbgimimplementation of BGI graphics. You do not need to include conio.h; just include graphics.h. The function reads one character from the keyboard and returns its ASCII value (without waiting for a return key). In order to work, the user must click in the graphics window (i.e., the Windows focus must be in the graphics window). For special keys, the getch function first returns ASCII 0. The next call will then return one of these special keys.
5. **void rectangle**(int left, int top, int right, int bottom) - rectangle draws a rectangle in the current line style, thickness, and drawing color. (left,top) is the upper left corner of the rectangle, and (right,bottom) is its lower right corner.
6. **void readimagefile**(const char\* title=NULL,int left=0, int right=0, int right=INT\_MAX, int bottom=INT\_MAX) - The readimagefile function is available in the [winbgim](http://www.cs.colorado.edu/%7Emain/bgi/README.html) implementation of BGI graphics. You do not need to include conio.h; just include graphics.h. The function reads a BMP, GIF, JPG, ICO, EMF or WMF image file and displays it in part of the current active window. The filename may be NULL.
7. **void outtextxy**(int x, int y, char \*textstring) - outtextxy displays a text string in the viewport at the given position (x, y), using the current justification settings and the current font, direction, and size.
8. **void setbkcolor**(int color) - setbkcolor sets the background to the color specified by color. The argument color can be a name or a number as listed below. (These symbolic names are defined in graphics.h.) .
9. **void setcolor**(int color) - setcolor sets the current drawing color to color, which can range from 0 to getmaxcolor. The current drawing color is the value to which pixels are set when lines, and so on are drawn. The drawing colors shown below are available for the CGA and EGA, respectively.

|  |
| --- |
|  |

1. **void setfillstyle**(int pattern, intcolor) - setfillstyle sets the current fill pattern and fill color. To set a user-defined fill pattern, do not give a pattern of 12 (USER\_FILL) to setfillstyle; instead, call setfillpattern.
2. **void settextjustify**(inthoriz, intvert) - Text output after a call to settextjustify is justified around the current position (CP) horizontally and vertically, as specified. The default justification settings are LEFT\_TEXT (for horizontal) and TOP\_TEXT (for vertical). The enumeration text\_just in graphics.h provides names for the horiz and vert settings passed to settextjustify.

|  |
| --- |
|  |

1. **void settextstyle**(int font, int direction, intcharsize) - settextstyle sets the text font, the direction in which text is displayed, and the size of the characters. A call to settextstyle affects all text output by outtext and outtextxy.

|  |
| --- |
|  |

**10. Program Coding:**

#include"graphics.h"

#include<time.h>

#include<stdlib.h>

#include "Car\_My.h"

#include"EnemyCars.h"

#include<vector>

#include<algorithm>

using namespace std;

#define Window\_X 1016

#define Window\_Y 730

#define U\_Grass\_X1 0

#define U\_Grass\_Y1 0

#define U\_Grass\_X2 1016

#define U\_Grass\_Y2 740/3 - 55 - 25 // = 191

#define D\_Grass\_X1 0

#define D\_Grass\_Y1 740 - (740/3 - 55) + 25 // = 549

#define D\_Grass\_X2 1016

#define D\_Grass\_Y2 740

#define Road\_X1 0

#define Road\_Y1 740/3 - 55

#define Road\_X2 1016

#define Road\_Y2 740 - (740/3 - 55)

#define Road\_Moving\_X1 100

#define Road\_Moving\_Y1 360

#define Road\_Moving\_X2 400

#define Road\_Moving\_Y2 380

int road\_moving\_X1=Road\_Moving\_X1;

int road\_moving\_Y1=Road\_Moving\_Y1;

int road\_moving\_X2=Road\_Moving\_X2;

int road\_moving\_Y2=Road\_Moving\_Y2;

int road2\_moving\_X1=Road\_Moving\_X1+500;

int road2\_moving\_Y1=Road\_Moving\_Y1;

int road2\_moving\_X2=Road\_Moving\_X2+500;

int road2\_moving\_Y2=Road\_Moving\_Y2;

#define Sun\_X 1000

#define Sun\_Y 0

#define Sun\_Rad 50

int Enemy\_car\_right\_inner\_x=R\_Inner\_Car\_Wheel\_X+1000; //Enemy Car Vaiables End

int Enemy\_car\_right\_inner\_y=R\_Inner\_Car\_Wheel\_Y-50;

int Enemy\_car\_right\_inner\_rad=R\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car\_right\_inner\_color1=WHITE;

int Enemy\_car\_right\_inner\_color2=WHITE;

#define MAX\_COLLISION\_ALLOWED 8

//for Level(s)

int levelChange=0;

//for Score

int s11=0;

char scc[50];

//for Life

int life = MAX\_COLLISION\_ALLOWED;

char lif[50];

// car Struct

myCar car;

// for menu

int choice;

char userName[100];

//for game ove and then menu

int aks=0;

// level changing scores

#define FIRST\_LEVEL\_SCORE 60

#define SECOND\_LEVEL\_SCORE 120

#define THIRD\_LEVEL\_SCORE 180

#define FOURTH\_LEVEL\_SCORE 240

#define FIFTH\_LEVEL\_SCORE 300

#define SIXTH\_LEVEL\_SCORE 360

#define SEVENTH\_LEVEL\_SCORE 430

// for highscores

FILE \*fout = fopen("highScores.txt","a");

struct MyStruct

{

int key;

char stringValue[100];

bool operator < (const MyStruct& str) const

{

return (str.key < key);

}

};

vector<MyStruct> sortCeleb;

//Declaration

void test();

void showHighScores();

void choice1();

void choice2();

void createMenu();

void sortIfHighest();

void gameOver();

void Credits();

void isBetween();

void INITIALIZE();

void ReinAllEnemy();

//Celebration

int max=-1;

void Upper(int a,int b,int c,int d)

{

readimagefile("palm.jpg",a,b,c,d); //Upper (U\_Grass)

}

void Lower(int a,int b,int c,int d)

{

readimagefile("palmNight.jpg",a,b,c,d);

}

void MovingRoad(int a,int b,int c,int d,int e)

{

setcolor(e);

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,e);

bar(a,b,c,d);

}

void MovingRoad2(int a,int b,int c,int d,int e)

{

setcolor(e);

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,e);

bar(a,b,c,d);

}

void Road()

{

readimagefile("image004.jpg",U\_Grass\_X1,U\_Grass\_Y2-5,U\_Grass\_X2,U\_Grass\_Y2); //Pavement 1

readimagefile("image004.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y1+5); //Pavement 2

}

void RoadReincarnated1()

{

road\_moving\_X1=Road\_Moving\_X1+950;

road\_moving\_Y1=Road\_Moving\_Y1;

road\_moving\_X2=Road\_Moving\_X2+950;

road\_moving\_Y2=Road\_Moving\_Y2;

}

void RoadReincarnated2()

{

road2\_moving\_X1=Road\_Moving\_X1+950;

road2\_moving\_Y1=Road\_Moving\_Y1;

road2\_moving\_X2=Road\_Moving\_X2+950;

road2\_moving\_Y2=Road\_Moving\_Y2;

}

void MoveTheRoad(int speed,int color)

{

MovingRoad(road\_moving\_X1-=speed,road\_moving\_Y1,road\_moving\_X2-=speed,road\_moving\_Y2,color);

MovingRoad2(road2\_moving\_X1-=speed,road2\_moving\_Y1,road2\_moving\_X2-=speed,road2\_moving\_Y2,color);

}

void Enemy\_Above\_Part\_Car(int a,int b,int c,int d,int color)

{

setcolor( color); //(Enemy Above Car)

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,color);

bar(a,b,c,d);

}

void Enemy\_Down\_Part\_Car(int a,int b,int c,int d,int color)

{

setcolor(color); //(Down Car)

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,color);

bar(a,b,c,d);

}

void Enemy\_Left\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color); //Left Wheel

circle(a,b,c);

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Enemy\_Right\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color);

circle(a,b,c); //Right Wheel

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Enemy\_Left\_Inner\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color); //Left Inner Wheel

circle(a,b,c);

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Enemy\_Right\_Inner\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color);

circle(a,b,c); //Right Inner Wheel

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

bool isBetween(int a,int b,int c)

{

return a <= b && b<= c;

}

void EnemyCar(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car\_above\_x1-=Speed,Enemy\_car\_above\_y1,Enemy\_car\_above\_x2-=Speed,Enemy\_car\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car\_down\_x1-=Speed,Enemy\_car\_down\_y1,Enemy\_car\_down\_x2-=Speed,Enemy\_car\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car\_leftwh\_x-=Speed,Enemy\_car\_leftwh\_y,Enemy\_car\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car\_rightwh\_x-=Speed,Enemy\_car\_rightwh\_y,Enemy\_car\_rightwh\_rad,EnemyColors[3]);

Enemy\_Left\_Inner\_Wheel\_Car(Enemy\_car\_left\_inner\_x-=Speed,Enemy\_car\_left\_inner\_y,Enemy\_car\_left\_inner\_rad,EnemyColors[4]);

Enemy\_Right\_Inner\_Wheel\_Car(Enemy\_car\_right\_inner\_x-=Speed,Enemy\_car\_right\_inner\_y,Enemy\_car\_right\_inner\_rad,EnemyColors[5]);

}

void EnemyCar11(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car11\_above\_x1-=Speed,Enemy\_car11\_above\_y1,Enemy\_car11\_above\_x2-=Speed,Enemy\_car11\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car11\_down\_x1-=Speed,Enemy\_car11\_down\_y1,Enemy\_car11\_down\_x2-=Speed,Enemy\_car11\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car11\_leftwh\_x-=Speed,Enemy\_car11\_leftwh\_y,Enemy\_car11\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car11\_rightwh\_x-=Speed,Enemy\_car11\_rightwh\_y,Enemy\_car11\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar113(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car113\_above\_x1-=Speed,Enemy\_car113\_above\_y1,Enemy\_car113\_above\_x2-=Speed,Enemy\_car113\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car113\_down\_x1-=Speed,Enemy\_car113\_down\_y1,Enemy\_car113\_down\_x2-=Speed,Enemy\_car113\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car113\_leftwh\_x-=Speed,Enemy\_car113\_leftwh\_y,Enemy\_car113\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car113\_rightwh\_x-=Speed,Enemy\_car113\_rightwh\_y,Enemy\_car113\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar114(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car114\_above\_x1-=Speed,Enemy\_car114\_above\_y1,Enemy\_car114\_above\_x2-=Speed,Enemy\_car114\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car114\_down\_x1-=Speed,Enemy\_car114\_down\_y1,Enemy\_car114\_down\_x2-=Speed,Enemy\_car114\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car114\_leftwh\_x-=Speed,Enemy\_car114\_leftwh\_y,Enemy\_car114\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car114\_rightwh\_x-=Speed,Enemy\_car114\_rightwh\_y,Enemy\_car114\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar115(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car115\_above\_x1-=Speed,Enemy\_car115\_above\_y1,Enemy\_car115\_above\_x2-=Speed,Enemy\_car115\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car115\_down\_x1-=Speed,Enemy\_car115\_down\_y1,Enemy\_car115\_down\_x2-=Speed,Enemy\_car115\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car115\_leftwh\_x-=Speed,Enemy\_car115\_leftwh\_y,Enemy\_car115\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car115\_rightwh\_x-=Speed,Enemy\_car115\_rightwh\_y,Enemy\_car115\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar116(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car116\_above\_x1-=Speed,Enemy\_car116\_above\_y1,Enemy\_car116\_above\_x2-=Speed,Enemy\_car116\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car116\_down\_x1-=Speed,Enemy\_car116\_down\_y1,Enemy\_car116\_down\_x2-=Speed,Enemy\_car116\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car116\_leftwh\_x-=Speed,Enemy\_car116\_leftwh\_y,Enemy\_car116\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car116\_rightwh\_x-=Speed,Enemy\_car116\_rightwh\_y,Enemy\_car116\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar117(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car117\_above\_x1-=Speed,Enemy\_car117\_above\_y1,Enemy\_car117\_above\_x2-=Speed,Enemy\_car117\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car117\_down\_x1-=Speed,Enemy\_car117\_down\_y1,Enemy\_car117\_down\_x2-=Speed,Enemy\_car117\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car117\_leftwh\_x-=Speed,Enemy\_car117\_leftwh\_y,Enemy\_car117\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car117\_rightwh\_x-=Speed,Enemy\_car117\_rightwh\_y,Enemy\_car117\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCar118(int EnemyColors[],int Speed)

{

Enemy\_Above\_Part\_Car(Enemy\_car118\_above\_x1-=Speed,Enemy\_car118\_above\_y1,Enemy\_car118\_above\_x2-=Speed,Enemy\_car118\_above\_y2,EnemyColors[0]);

Enemy\_Down\_Part\_Car(Enemy\_car118\_down\_x1-=Speed,Enemy\_car118\_down\_y1,Enemy\_car118\_down\_x2-=Speed,Enemy\_car118\_down\_y2,EnemyColors[1]);

Enemy\_Left\_Wheel\_Car(Enemy\_car118\_leftwh\_x-=Speed,Enemy\_car118\_leftwh\_y,Enemy\_car118\_leftwh\_rad,EnemyColors[2]);

Enemy\_Right\_Wheel\_Car(Enemy\_car118\_rightwh\_x-=Speed,Enemy\_car118\_rightwh\_y,Enemy\_car118\_rightwh\_rad,EnemyColors[3]);

}

void EnemyCarReIncarnated\_1(int y\_speed)

{

Enemy\_car\_above\_x1=Car\_Rectangle\_X1+1000; //Enemy Car

Enemy\_car\_above\_y1=Car\_Rectangle\_Y1-50+y\_speed;

Enemy\_car\_above\_x2=Car\_Rectangle\_X2+1000;

Enemy\_car\_above\_y2=Car\_Rectangle\_Y2-50+y\_speed;

Enemy\_car\_above\_color=RED;

Enemy\_car\_down\_x1=D\_Car\_Rectangle\_X1+1000;

Enemy\_car\_down\_y1=D\_Car\_Rectangle\_Y1-50+y\_speed;

Enemy\_car\_down\_x2=D\_Car\_Rectangle\_X2+1000;

Enemy\_car\_down\_y2=D\_Car\_Rectangle\_Y2-50+y\_speed;

Enemy\_car\_down\_color=RED;

Enemy\_car\_leftwh\_x=L\_Car\_Wheel\_X+1000;

Enemy\_car\_leftwh\_y=L\_Car\_Wheel\_Y-50+y\_speed;

Enemy\_car\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car\_leftwh\_color1=WHITE;

Enemy\_car\_leftwh\_color2=BLACK;

Enemy\_car\_rightwh\_x=R\_Car\_Wheel\_X+1000;

Enemy\_car\_rightwh\_y=R\_Car\_Wheel\_Y-50+y\_speed;

Enemy\_car\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car\_rightwh\_color1=WHITE;

Enemy\_car\_rightwh\_color2=BLACK;

Enemy\_car\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1000;

Enemy\_car\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-50+y\_speed;

Enemy\_car\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car\_left\_inner\_color1=WHITE;

Enemy\_car\_left\_inner\_color2=WHITE;

Enemy\_car\_right\_inner\_x=R\_Inner\_Car\_Wheel\_X+1000; //Enemy Car Vaiables End

Enemy\_car\_right\_inner\_y=R\_Inner\_Car\_Wheel\_Y-50+y\_speed;

Enemy\_car\_right\_inner\_rad=R\_Inner\_Car\_Wheel\_Rad;

Enemy\_car\_right\_inner\_color1=WHITE;

Enemy\_car\_right\_inner\_color2=WHITE;

}

void EnemyCarReIncarnated\_2(int y\_speed)

{

Enemy\_car\_above\_x1=Car\_Rectangle\_X1+1000; //Enemy Car

Enemy\_car\_above\_y1=Car\_Rectangle\_Y1+100+y\_speed;

Enemy\_car\_above\_x2=Car\_Rectangle\_X2+1000;

Enemy\_car\_above\_y2=Car\_Rectangle\_Y2+100+y\_speed;

Enemy\_car\_above\_color=RED;

Enemy\_car\_down\_x1=D\_Car\_Rectangle\_X1+1000;

Enemy\_car\_down\_y1=D\_Car\_Rectangle\_Y1+100+y\_speed;

Enemy\_car\_down\_x2=D\_Car\_Rectangle\_X2+1000;

Enemy\_car\_down\_y2=D\_Car\_Rectangle\_Y2+100+y\_speed;

Enemy\_car\_down\_color=RED;

Enemy\_car\_leftwh\_x=L\_Car\_Wheel\_X+1000;

Enemy\_car\_leftwh\_y=L\_Car\_Wheel\_Y+100+y\_speed;

Enemy\_car\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car\_leftwh\_color1=WHITE;

Enemy\_car\_leftwh\_color2=BLACK;

Enemy\_car\_rightwh\_x=R\_Car\_Wheel\_X+1000;

Enemy\_car\_rightwh\_y=R\_Car\_Wheel\_Y+100+y\_speed;

Enemy\_car\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car\_rightwh\_color1=WHITE;

Enemy\_car\_rightwh\_color2=BLACK;

Enemy\_car\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1000;

Enemy\_car\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+100+y\_speed;

Enemy\_car\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car\_left\_inner\_color1=WHITE;

Enemy\_car\_left\_inner\_color2=WHITE;

Enemy\_car\_right\_inner\_x=R\_Inner\_Car\_Wheel\_X+1000; //Enemy Car Vaiables End

Enemy\_car\_right\_inner\_y=R\_Inner\_Car\_Wheel\_Y+100+y\_speed;

Enemy\_car\_right\_inner\_rad=R\_Inner\_Car\_Wheel\_Rad;

Enemy\_car\_right\_inner\_color1=WHITE;

Enemy\_car\_right\_inner\_color2=WHITE;

}

void sortIfHighest()

{

FILE \*fifH = fopen("highScores.txt","r");

char user[101];

int score,max=-1;

while(fscanf(fifH,"%s %d",user,&score) !=-1)

{

MyStruct myStruct;

myStruct.key = score;

sortCeleb.push\_back(myStruct);

if(max<score)

max=score;

}

sort(sortCeleb.begin(),sortCeleb.end());

printf("max = %d\n\n\n\n",max);

printf("s11 = %d\n\n\n",s11);

if(s11>=max)

{

cleardevice();

readimagefile("new-record-top.jpg",0,0,1016,300);

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,5);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(508,193+120,"ConGratulations!");

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,5);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(508,193+180,userName);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,5);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(508,193+240,"You Are Now The Highest Scorer!");

getch();

delay(3000);

}

}

void Score()

{

setcolor(BLACK);

rectangle(770,0,822,31);

setfillstyle(SOLID\_FILL,BLACK);

bar(770,0,822,31);

readimagefile("coins.jpg",770,0,802,31);

setcolor(BLACK);

rectangle(910,0,1016,32);

setfillstyle(SOLID\_FILL,BLACK);

bar(910,0,1016,32);

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,15);

settextjustify(RIGHT\_TEXT,BOTTOM\_TEXT);

outtextxy(930,32,"SCORE:");

setcolor(BLACK);

rectangle(0,0,300,31);

setfillstyle(SOLID\_FILL,BLACK);

bar(0,0,300,31);

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,15);

settextjustify(RIGHT\_TEXT,BOTTOM\_TEXT);

outtextxy(300,32,"Life Remaining:");

}

void MovingScore()

{

if(Enemy\_car\_down\_x2<=0 )

{

s11+=(10-levelChange);

}

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,15);

settextjustify(RIGHT\_TEXT,BOTTOM\_TEXT);

sprintf(scc,"%d",s11);

outtextxy(1016,32,scc);

sprintf(lif,"%d",life);

outtextxy(340,32,lif);

if(life<10)

{

setcolor(BLACK);

rectangle(300,0,320,31);

setfillstyle(SOLID\_FILL,BLACK);

bar(300,0,320,31);

}

}

bool CollisionDetection()

{

if(Enemy\_car\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection1()

{

if(Enemy\_car11\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car11\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car11\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car11\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car11\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car11\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car11\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car11\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car11\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car11\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car11\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection2()

{

if(Enemy\_car113\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car113\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car113\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car113\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car113\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car113\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car113\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car113\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car113\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car113\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car113\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection3()

{

if(Enemy\_car114\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car114\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car114\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car114\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car114\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car114\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car114\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car114\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car114\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car114\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car114\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection4()

{

if(Enemy\_car115\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car115\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car115\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car115\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car115\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car115\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car115\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car115\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car115\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car115\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car115\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection5()

{

if(Enemy\_car116\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car116\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car116\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car116\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car116\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car116\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car116\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car116\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car116\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car116\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car116\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection6()

{

if(Enemy\_car117\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car117\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car117\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car117\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car117\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car117\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car117\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car117\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car117\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car117\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car117\_down\_x2 )

{

return true;

}

}

}

return false;

}

bool CollisionDetection7()

{

if(Enemy\_car118\_above\_x1 <= car.car\_down\_x2)

{

if(isBetween(car.car\_down\_y1,Enemy\_car118\_above\_y1,car.car\_down\_y2)==true

||isBetween(car.car\_down\_y1,Enemy\_car118\_above\_y2,car.car\_down\_y2)==true)

{

if(car.car\_down\_x1 <= Enemy\_car118\_down\_x2 )

{

return true;

}

}

}

if((Enemy\_car118\_down\_x1 <= car.car\_above\_x2) || (Enemy\_car118\_down\_x1 <= car.car\_down\_x2))

{

if((isBetween(car.car\_above\_y1,Enemy\_car118\_down\_y1,car.car\_above\_y2)==true

|| isBetween(car.car\_above\_y1,Enemy\_car118\_down\_y2,car.car\_above\_y2)==true)

|| (isBetween(car.car\_down\_y1,Enemy\_car118\_down\_y1,car.car\_down\_y2)==true

|| (isBetween(car.car\_above\_y1,Enemy\_car118\_down\_y2,car.car\_above\_y2)==true) ))

{

if(car.car\_down\_x1 <= Enemy\_car118\_down\_x2 )

{

return true;

}

}

}

return false;

}

void gameOver()

{

aks++;

levelChange=0;

sortIfHighest();

readimagefile("Game over.jpg",0,0,1016,730);

delay(4000);

createMenu();

}

void writeToFile()

{

FILE \*fout = fopen("highScores.txt","a");

fprintf(fout,userName);

fprintf(fout," ");

fprintf(fout,"%d\n",s11);

fclose(fout);

}

void CollisionResponse(int count)

{

int i;

s11 -= (10-levelChange-1);

for(i=0;i<=45;i++)

{

car.Carr(i%16);

delay(10);

}

car.Carr(0);

if(car.car\_above\_y2>=382)

{

life--;

car.CarrReinCarnation1();

}

else

{

life--;

car.CarrReinCarnation2();

}

if(count == MAX\_COLLISION\_ALLOWED)

{

char strScore[100];

char userScore [100];

writeToFile();

gameOver();

}

}

void LevelChange()

{

if(levelChange==0)

{

if(s11>=FIRST\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("Checkered20flag.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("402313.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("402314.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==1)

{

if(s11>=SECOND\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("l3.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("canvas-particle-2.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("canvas-particle-3.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==2)

{

if(s11>=THIRD\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("asdd.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("1 layer.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("eletree-grass.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==3)

{

if(s11>=FOURTH\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("5hhj.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("cartoon-land-4.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("cartoon-land-5.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==4)

{

if(s11>=FIFTH\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("65gf.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("time-animated.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("time-animated2.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==5)

{

if(s11>=SIXTH\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("7ghg.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("grassLAnd.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("grassLAnd11.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

if(levelChange==6)

{

if(s11>=SEVENTH\_LEVEL\_SCORE)

{

ReinAllEnemy();

cleardevice();

readimagefile("8.jpg",0,0,1016,730);

getch();

delay(2000);

cleardevice();

readimagefile("1last.jpg",U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2); //Upper (U\_Grass)

readimagefile("last.jpg",D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Score();

Road();

levelChange++;

}

}

}

void test()

{

cleardevice();

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,2);

settextjustify(LEFT\_TEXT,TOP\_TEXT);

outtextxy(85,50,"Please Don't Use BackSpace or Escape Button While Writting Your Name...");

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,3);

settextjustify(LEFT\_TEXT,TOP\_TEXT);

outtextxy(250,320,"Please Enter Your Name : ");

int i = 0;

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,3);

while(true)

{

char ch = getch();

if(ch == 13)

{

break;

}

if(ch == 8)

{

userName[--i] = '\0';

}

else

{

userName[i++] = ch;

userName[i] = '\0';

outtextxy(624,320,userName);

}

}

userName[i++] = ' ';

userName[i] = '\0';

}

char user[101];

int score;

void showHighScores()

{

vector<MyStruct> vHighScores;

FILE \*fin = fopen("highScores.txt","r");

int row = 193;

char destBuf[100];

while(fscanf(fin,"%s %d",user,&score) !=-1)

{

printf("%s\t%d\n",user,score);

MyStruct myStruct;

myStruct.key = score;

strcpy(myStruct.stringValue,user);

vHighScores.push\_back(myStruct);

}

if(choice==2)

{

sort(vHighScores.begin(), vHighScores.end());

int i;

cleardevice();

readimagefile("highscore1.jpg",0,0,1016,192);

for(i=0;i<vHighScores.size();i++)

{

MyStruct tempStruct = vHighScores[i];

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,3);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(385,row+12,tempStruct.stringValue);

setcolor(WHITE);

settextstyle(COMPLEX\_FONT,HORIZ\_DIR,3);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(620,row+12,itoa(tempStruct.key,destBuf,10));

row += 40;

}

getch();

fclose(fin);

cleardevice();

}

delay(100);

}

int getRandomColorCode()

{

srand(time(NULL));

return (1 + (rand() % 15));

}

void pushRandomColorsIntoArray(int \*colorArray)

{

int i;

for(i=1;i<6;i++){

colorArray[i] = getRandomColorCode();

}

}

void ReinAllEnemy()

{

Enemy\_car113\_above\_x1=Car\_Rectangle\_X1+1280; //Enemy Car

Enemy\_car113\_above\_y1=Car\_Rectangle\_Y1 -40-50;

Enemy\_car113\_above\_x2=Car\_Rectangle\_X2+1280;

Enemy\_car113\_above\_y2=Car\_Rectangle\_Y2 -40-50;

Enemy\_car113\_above\_color=RED;

Enemy\_car113\_down\_x1=D\_Car\_Rectangle\_X1+1280;

Enemy\_car113\_down\_y1=D\_Car\_Rectangle\_Y1-40-50;

Enemy\_car113\_down\_x2=D\_Car\_Rectangle\_X2+1280;

Enemy\_car113\_down\_y2=D\_Car\_Rectangle\_Y2-40-50;

Enemy\_car113\_down\_color=RED;

Enemy\_car113\_leftwh\_x=L\_Car\_Wheel\_X+1280;

Enemy\_car113\_leftwh\_y=L\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car113\_leftwh\_color1=WHITE;

Enemy\_car113\_leftwh\_color2=BLACK;

Enemy\_car113\_rightwh\_x=R\_Car\_Wheel\_X+1280;

Enemy\_car113\_rightwh\_y=R\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car113\_rightwh\_color1=WHITE;

Enemy\_car113\_rightwh\_color2=BLACK;

Enemy\_car113\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1280;

Enemy\_car113\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car113\_left\_inner\_color1=WHITE;

Enemy\_car113\_left\_inner\_color2=WHITE;

Enemy\_car115\_above\_x1=Car\_Rectangle\_X1+1400; //Enemy Car

Enemy\_car115\_above\_y1=Car\_Rectangle\_Y1-40-50+330;

Enemy\_car115\_above\_x2=Car\_Rectangle\_X2+1400;

Enemy\_car115\_above\_y2=Car\_Rectangle\_Y2-40-50+330;

Enemy\_car115\_above\_color=RED;

Enemy\_car115\_down\_x1=D\_Car\_Rectangle\_X1+1400;

Enemy\_car115\_down\_y1=D\_Car\_Rectangle\_Y1-40-50+330;

Enemy\_car115\_down\_x2=D\_Car\_Rectangle\_X2+1400;

Enemy\_car115\_down\_y2=D\_Car\_Rectangle\_Y2-40-50+330;

Enemy\_car115\_down\_color=RED;

Enemy\_car115\_leftwh\_x=L\_Car\_Wheel\_X+1400;

Enemy\_car115\_leftwh\_y=L\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car115\_leftwh\_color1=WHITE;

Enemy\_car115\_leftwh\_color2=BLACK;

Enemy\_car115\_rightwh\_x=R\_Car\_Wheel\_X+1400;

Enemy\_car115\_rightwh\_y=R\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car115\_rightwh\_color1=WHITE;

Enemy\_car115\_rightwh\_color2=BLACK;

Enemy\_car115\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1400;

Enemy\_car115\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car115\_left\_inner\_color1=WHITE;

Enemy\_car115\_left\_inner\_color2=WHITE;

int inc1 = getRandomSign() \* getRandomPositionWithinRange(1,100);

Enemy\_car117\_above\_x1=Car\_Rectangle\_X1+1200; //Enemy Car

Enemy\_car117\_above\_y1=Car\_Rectangle\_Y1+160+inc1;

Enemy\_car117\_above\_x2=Car\_Rectangle\_X2+1200;

Enemy\_car117\_above\_y2=Car\_Rectangle\_Y2+160+inc1;

Enemy\_car117\_above\_color=RED;

Enemy\_car117\_down\_x1=D\_Car\_Rectangle\_X1+1200;

Enemy\_car117\_down\_y1=D\_Car\_Rectangle\_Y1+160+inc1;

Enemy\_car117\_down\_x2=D\_Car\_Rectangle\_X2+1200;

Enemy\_car117\_down\_y2=D\_Car\_Rectangle\_Y2+160+inc1;

Enemy\_car117\_down\_color=RED;

Enemy\_car117\_leftwh\_x=L\_Car\_Wheel\_X+1200;

Enemy\_car117\_leftwh\_y=L\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car117\_leftwh\_color1=WHITE;

Enemy\_car117\_leftwh\_color2=BLACK;

Enemy\_car117\_rightwh\_x=R\_Car\_Wheel\_X+1200;

Enemy\_car117\_rightwh\_y=R\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car117\_rightwh\_color1=WHITE;

Enemy\_car117\_rightwh\_color2=BLACK;

Enemy\_car117\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1200;

Enemy\_car117\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car117\_left\_inner\_color1=WHITE;

Enemy\_car117\_left\_inner\_color2=WHITE;

int inc2 = getRandomSign() \* getRandomPositionWithinRange(1,45);

Enemy\_car118\_above\_x1=Car\_Rectangle\_X1+1600; //Enemy Car

Enemy\_car118\_above\_y1=Car\_Rectangle\_Y1+160+inc2;

Enemy\_car118\_above\_x2=Car\_Rectangle\_X2+1600;

Enemy\_car118\_above\_y2=Car\_Rectangle\_Y2+160+inc2;

Enemy\_car118\_above\_color=RED;

Enemy\_car118\_down\_x1=D\_Car\_Rectangle\_X1+1600;

Enemy\_car118\_down\_y1=D\_Car\_Rectangle\_Y1+160+inc2;

Enemy\_car118\_down\_x2=D\_Car\_Rectangle\_X2+1600;

Enemy\_car118\_down\_y2=D\_Car\_Rectangle\_Y2+160+inc2;

Enemy\_car118\_down\_color=RED;

Enemy\_car118\_leftwh\_x=L\_Car\_Wheel\_X+1600;

Enemy\_car118\_leftwh\_y=L\_Car\_Wheel\_Y+160+inc2;

Enemy\_car118\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car118\_leftwh\_color1=WHITE;

Enemy\_car118\_leftwh\_color2=BLACK;

Enemy\_car118\_rightwh\_x=R\_Car\_Wheel\_X+1600;

Enemy\_car118\_rightwh\_y=R\_Car\_Wheel\_Y+160+inc2;

Enemy\_car118\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car118\_rightwh\_color1=WHITE;

Enemy\_car118\_rightwh\_color2=BLACK;

Enemy\_car118\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1600;

Enemy\_car118\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160+inc2;

Enemy\_car118\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car118\_left\_inner\_color1=WHITE;

Enemy\_car118\_left\_inner\_color2=WHITE;

Enemy\_car116\_above\_x1=Car\_Rectangle\_X1+1300; //Enemy Car

Enemy\_car116\_above\_y1=Car\_Rectangle\_Y1+160-140;

Enemy\_car116\_above\_x2=Car\_Rectangle\_X2+1300;

Enemy\_car116\_above\_y2=Car\_Rectangle\_Y2+160-140;

Enemy\_car116\_above\_color=RED;

Enemy\_car116\_down\_x1=D\_Car\_Rectangle\_X1+1300;

Enemy\_car116\_down\_y1=D\_Car\_Rectangle\_Y1+160-140;

Enemy\_car116\_down\_x2=D\_Car\_Rectangle\_X2+1300;

Enemy\_car116\_down\_y2=D\_Car\_Rectangle\_Y2+160-140;

Enemy\_car116\_down\_color=RED;

Enemy\_car116\_leftwh\_x=L\_Car\_Wheel\_X+1300;

Enemy\_car116\_leftwh\_y=L\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car116\_leftwh\_color1=WHITE;

Enemy\_car116\_leftwh\_color2=BLACK;

Enemy\_car116\_rightwh\_x=R\_Car\_Wheel\_X+1300;

Enemy\_car116\_rightwh\_y=R\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car116\_rightwh\_color1=WHITE;

Enemy\_car116\_rightwh\_color2=BLACK;

Enemy\_car116\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1300;

Enemy\_car116\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car116\_left\_inner\_color1=WHITE;

Enemy\_car116\_left\_inner\_color2=WHITE;

Enemy\_car114\_above\_x1=Car\_Rectangle\_X1+1340; //Enemy Car

Enemy\_car114\_above\_y1=Car\_Rectangle\_Y1-40+100;

Enemy\_car114\_above\_x2=Car\_Rectangle\_X2+1340;

Enemy\_car114\_above\_y2=Car\_Rectangle\_Y2-40+100;

Enemy\_car114\_above\_color=RED;

Enemy\_car114\_down\_x1=D\_Car\_Rectangle\_X1+1340;

Enemy\_car114\_down\_y1=D\_Car\_Rectangle\_Y1-40+100;

Enemy\_car114\_down\_x2=D\_Car\_Rectangle\_X2+1340;

Enemy\_car114\_down\_y2=D\_Car\_Rectangle\_Y2-40+100;

Enemy\_car114\_down\_color=RED;

Enemy\_car114\_leftwh\_x=L\_Car\_Wheel\_X+1340;

Enemy\_car114\_leftwh\_y=L\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car114\_leftwh\_color1=WHITE;

Enemy\_car114\_leftwh\_color2=BLACK;

Enemy\_car114\_rightwh\_x=R\_Car\_Wheel\_X+1340;

Enemy\_car114\_rightwh\_y=R\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car114\_rightwh\_color1=WHITE;

Enemy\_car114\_rightwh\_color2=BLACK;

Enemy\_car114\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1340;

Enemy\_car114\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car114\_left\_inner\_color1=WHITE;

Enemy\_car114\_left\_inner\_color2=WHITE;

Enemy\_car11\_above\_x1=Car\_Rectangle\_X1+1100; //Enemy Car

Enemy\_car11\_above\_y1=Car\_Rectangle\_Y1+160;

Enemy\_car11\_above\_x2=Car\_Rectangle\_X2+1100;

Enemy\_car11\_above\_y2=Car\_Rectangle\_Y2+160;

Enemy\_car11\_above\_color=RED;

Enemy\_car11\_down\_x1=D\_Car\_Rectangle\_X1+1100;

Enemy\_car11\_down\_y1=D\_Car\_Rectangle\_Y1+160;

Enemy\_car11\_down\_x2=D\_Car\_Rectangle\_X2+1100;

Enemy\_car11\_down\_y2=D\_Car\_Rectangle\_Y2+160;

Enemy\_car11\_down\_color=RED;

Enemy\_car11\_leftwh\_x=L\_Car\_Wheel\_X+1100;

Enemy\_car11\_leftwh\_y=L\_Car\_Wheel\_Y+160;

Enemy\_car11\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car11\_leftwh\_color1=WHITE;

Enemy\_car11\_leftwh\_color2=BLACK;

Enemy\_car11\_rightwh\_x=R\_Car\_Wheel\_X+1100;

Enemy\_car11\_rightwh\_y=R\_Car\_Wheel\_Y+160;

Enemy\_car11\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car11\_rightwh\_color1=WHITE;

Enemy\_car11\_rightwh\_color2=BLACK;

Enemy\_car11\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1100;

Enemy\_car11\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160;

Enemy\_car11\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car11\_left\_inner\_color1=WHITE;

Enemy\_car11\_left\_inner\_color2=WHITE;

}

void choice1()

{

int count=0,i=0,a=2,b=2,co=0;

if(choice==1 && aks==0)

{

aks++;

test();

readimagefile("checkered-flag-vector.jpg",0,0,1016,730);

readimagefile("ready.jpg",425,2,590,80);

delay(1000);

readimagefile("1.jpg",490,80,520,150);

delay(1000);

readimagefile("2.jpg",490,160,520,230);

delay(1000);

readimagefile("3.jpg",490,240,520,340);

delay(1000);

readimagefile("4.jpg",0,0,1016,730);

delay(1000);

getch();

cleardevice();

Upper(U\_Grass\_X1,U\_Grass\_Y1,U\_Grass\_X2,U\_Grass\_Y2);

Lower(D\_Grass\_X1,D\_Grass\_Y1,D\_Grass\_X2,D\_Grass\_Y2);

Road();

Score();

int colors[]={RED,RED,WHITE,WHITE,BLACK,BLACK};

int clearColors[]={BLACK,BLACK,BLACK,BLACK,BLACK,BLACK};

int testColors1[]={5,5,15,15,15,15};

int testColors2[]={15,15,15,15,9,9};

int testColors3[]={14,14,15,15,15,15};

int testColors4[]={6,6,6,6,15,15};

int testColors5[]={10,10,15,15,15,15};

int testColors6[]={12,12,15,15,15,15};

s11=0;

Score();

MovingRoad(road\_moving\_X1,road\_moving\_Y1,road\_moving\_X2,road\_moving\_Y2,15);

MovingRoad2(road2\_moving\_X1,road2\_moving\_Y1,road2\_moving\_X2,road2\_moving\_Y2,15);

int jkk=0;

while(true)

{

jkk++;

printf("J = %d\n",jkk);

MovingScore();

MoveTheRoad(0,0);

MoveTheRoad(6,15);

if(road\_moving\_X2 < 0)

{

RoadReincarnated1();

}

if(road2\_moving\_X2 < 0)

{

RoadReincarnated2();

}

car.MovingCar(colors,0,1);

int EnemyColors[]={a,b,YELLOW,YELLOW,YELLOW,YELLOW};

if(s11>=525)

{

writeToFile();

gameOver();

delay(4000);

cleardevice();

createMenu();

}

if(Enemy\_car\_down\_x2 > -1)

{

EnemyCar(clearColors,0);

EnemyCar(EnemyColors,8+(levelChange));

}

else

{

srand(time(NULL));

(a+=rand())%16+1;

(b+=rand())%16+1;

srand(time(NULL));

if( (count++ +rand()) % 2 == 0)

{

if(Enemy\_car\_leftwh\_y>=540)

{

i=0;

}

srand(time(NULL));

EnemyCarReIncarnated\_2(i + rand()%140);

}

else

{

if(Enemy\_car\_leftwh\_y>=320)

{

i=0;

}

srand(time(NULL));

EnemyCarReIncarnated\_1(i + rand()%56);

}

}

delay(20);

if(levelChange==0 || levelChange==1 || levelChange==2

|| levelChange==3 || levelChange==4 || levelChange==5

|| levelChange==6)

{

LevelChange();

}

bool flag = CollisionDetection();

bool flag1 = CollisionDetection1();

bool flag2 = CollisionDetection2();

bool flag3 = CollisionDetection3();

bool flag4 = CollisionDetection4();

bool flag5 = CollisionDetection5();

bool conflag6 = CollisionDetection6();

bool conflag7 = CollisionDetection7();

if(flag == true || flag1 == true || flag2 == true

|| flag3 == true || flag4 == true || flag5 == true

|| conflag6 == true || conflag7 == true)

{

co++;

CollisionResponse(co);

if(co==MAX\_COLLISION\_ALLOWED)

{

co=0;

return;

}

}

if(Enemy\_car118\_down\_x2<0)

{

pushRandomColorsIntoArray(testColors2);

con2EnemyCarReIncarnated\_2();

}

if(Enemy\_car117\_down\_x2<0)

{

pushRandomColorsIntoArray(testColors1);

con1EnemyCarReIncarnated\_2();

}

if(Enemy\_car116\_down\_x2<0)

{

pushRandomColorsIntoArray(testColors3);

con1EnemyCarReIncarnated\_2();

}

if(Enemy\_car115\_down\_x2<0)

{

pushRandomColorsIntoArray(testColors4);

EnemyCar115ReIncarnated\_2();

}

if(Enemy\_car113\_down\_x2<0)

{

pushRandomColorsIntoArray(testColors5);

EnemyCar113ReIncarnated\_2();

}

if(levelChange==0 || levelChange==1)

{

EnemyCar118(clearColors,0);

EnemyCar118(testColors3,10+(levelChange));

EnemyCar117(clearColors,0);

EnemyCar117(testColors6,10+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors2,9+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors4,12+(levelChange));

}

if(levelChange==2)

{

EnemyCar113(clearColors,0);

EnemyCar113(testColors4,9+(levelChange));

EnemyCar11(clearColors,0);

EnemyCar11(testColors1,7+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors6,11+(levelChange));

EnemyCar116(clearColors,0);

EnemyCar116(testColors3,10+(levelChange));

}

if(levelChange==3)

{

EnemyCar118(clearColors,0);

EnemyCar118(testColors3,10+(levelChange));

EnemyCar117(clearColors,0);

EnemyCar117(testColors6,9+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors2,9+(levelChange));

EnemyCar11(clearColors,0);

EnemyCar11(testColors1,7+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors4,8+(levelChange));

}

if(levelChange==4)

{

EnemyCar116(clearColors,0);

EnemyCar116(testColors3,8+(levelChange));

EnemyCar11(clearColors,0);

EnemyCar11(testColors2,8+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors6,10+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors1,9+(levelChange));

}

if(levelChange==5)

{

EnemyCar117(clearColors,0);

EnemyCar117(testColors6,11+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors2,12+(levelChange));

EnemyCar114(clearColors,0);

EnemyCar114(testColors3,6+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors1,8+(levelChange));

}

if(levelChange==6)

{

EnemyCar11(clearColors,0);

EnemyCar11(testColors1,6+(levelChange));

EnemyCar118(clearColors,0);

EnemyCar118(testColors3,13+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors2,9+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors1,16+(levelChange));

EnemyCar117(clearColors,0);

EnemyCar117(testColors6,7+(levelChange));

}

if(levelChange==7)

{

EnemyCar116(clearColors,0);

EnemyCar116(testColors6,12+(levelChange));

EnemyCar118(clearColors,0);

EnemyCar118(testColors3,13+(levelChange));

EnemyCar117(clearColors,0);

EnemyCar117(testColors6,15+(levelChange));

EnemyCar11(clearColors,0);

EnemyCar11(testColors1,18+(levelChange));

EnemyCar115(clearColors,0);

EnemyCar115(testColors4,11+(levelChange));

EnemyCar113(clearColors,0);

EnemyCar113(testColors2,17+(levelChange));

}

if(kbhit())

{

int input\_key=getch();

if(input\_key == 27)

{

exit(0);

}

if(input\_key==72&&car.car\_above\_y1>=180)

{

count=1;

car.MovingCar(clearColors,0,count);

car.MovingCar(colors,10,count);

}

if(input\_key==75 && car.car\_down\_x1>=0)

{

count=2;

car.MovingCar(clearColors,0,count);

car.MovingCar(colors,10,count);

}

if(input\_key==77&&car.car\_down\_x2<=1010)

{

count=3;

car.MovingCar(clearColors,0,count);

car.MovingCar(colors,10,count);

}

if(input\_key==80 && car.car\_leftwh\_y<=550)

{

count=4;

car.MovingCar(clearColors,0,count);

car.MovingCar(colors,10,count);

}

}

}

}

}

void choice2()

{

cleardevice();

showHighScores();

}

void createMenu()

{

int x,y;

readimagefile("COVVER.jpg",0,0,1016,730);

clearmouseclick(WM\_LBUTTONDOWN);

while(!ismouseclick(WM\_LBUTTONDOWN))

{

delay(100);

}

getmouseclick(WM\_LBUTTONDOWN,x,y);

if((x>=432&&x<=577) && (y>=341 && y<=387))

{

choice=1;

choice1();

cleardevice();

}

else if((x>=428&&x<=582) && (y>=431 && y<=474))

{

choice=2;

choice2();

cleardevice();

}

else if((x>=431&&x<=581) && (y>=517 && y<=561))

{

choice=3;

cleardevice();

readimagefile("UntitledAQ.jpg",0,0,1016,730);

getch();

cleardevice();

}

else if((x>=431&&x<=582) && (y>=599 && y<=634))

{

choice=4;

cleardevice();

Credits();

}

else

{

createMenu();

}

}

void Credits()

{

readimagefile("Untidqtled.jpg",0,0,1016,730);

getch();

cleardevice();

}

void INITIALIZE()

{

initwindow(Window\_X,Window\_Y,"Game");

readimagefile("fooo.jpg",0,0,1016,730);

getch();

cleardevice();

}

int main( )

{

INITIALIZE();

while(true)

{

createMenu();

cleardevice();

}

system("pause");

return 0;

}

**11. Header Used In The Code :**

**EnemyCars.h :**

#include<stdio.h>

#include"graphics.h"

#define Car\_Rectangle\_X1 85 //Above Car

#define Car\_Rectangle\_Y1 262

#define Car\_Rectangle\_X2 150-17-7

#define Car\_Rectangle\_Y2 295-10-4

int car\_above\_x1=Car\_Rectangle\_X1; //Variable For The Car

int car\_above\_y1=Car\_Rectangle\_Y1;

int car\_above\_x2=Car\_Rectangle\_X2;

int car\_above\_y2=Car\_Rectangle\_Y2;

int car\_above\_color=RED;

#define D\_Car\_Rectangle\_X1 61 //Down Car

#define D\_Car\_Rectangle\_Y1 305-20-5

#define D\_Car\_Rectangle\_X2 235-55-20-12

#define D\_Car\_Rectangle\_Y2 345-25-10-7

int car\_down\_x1=D\_Car\_Rectangle\_X1; //Variable For The Car

int car\_down\_y1=D\_Car\_Rectangle\_Y1;

int car\_down\_x2=D\_Car\_Rectangle\_X2;

int car\_down\_y2=D\_Car\_Rectangle\_Y2;

int car\_down\_color=RED;

#define L\_Car\_Wheel\_X 78 //Left Wheel

#define L\_Car\_Wheel\_Y 352-36-12

#define L\_Car\_Wheel\_Rad 9

int car\_leftwh\_x=L\_Car\_Wheel\_X; //Variable For The Car

int car\_leftwh\_y=L\_Car\_Wheel\_Y;

int car\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int car\_leftwh\_color1=WHITE;

int car\_leftwh\_color2=BLACK;

#define R\_Car\_Wheel\_X 155-16-8 //Right Wheel

#define R\_Car\_Wheel\_Y 352-36-12

#define R\_Car\_Wheel\_Rad 9

int getRandomPositionWithinRange(int,int);

int car\_rightwh\_x=R\_Car\_Wheel\_X; //Variable For The Car

int car\_rightwh\_y=R\_Car\_Wheel\_Y;

int car\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int car\_rightwh\_color1=WHITE;

int car\_rightwh\_color2=BLACK;

#define L\_Inner\_Car\_Wheel\_X 78

#define L\_Inner\_Car\_Wheel\_Y 352-36-12

#define L\_Inner\_Car\_Wheel\_Rad 6

int car\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X; //Variable For The Car

int car\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y;

int car\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int car\_left\_inner\_color1=WHITE;

int car\_left\_inner\_color2=WHITE;

#define R\_Inner\_Car\_Wheel\_X 155-16-8

#define R\_Inner\_Car\_Wheel\_Y 352-36-12

#define R\_Inner\_Car\_Wheel\_Rad 6

int car\_right\_inner\_x=R\_Inner\_Car\_Wheel\_X; //Variable For The Car

int car\_right\_inner\_y=R\_Inner\_Car\_Wheel\_Y;

int car\_right\_inner\_rad=R\_Inner\_Car\_Wheel\_Rad;

int car\_right\_inner\_color1=WHITE;

int car\_right\_inner\_color2=WHITE;

// Enemy Car(s)

int Enemy\_car\_above\_x1=Car\_Rectangle\_X1+1000; //Enemy Car

int Enemy\_car\_above\_y1=Car\_Rectangle\_Y1-50;

int Enemy\_car\_above\_x2=Car\_Rectangle\_X2+1000;

int Enemy\_car\_above\_y2=Car\_Rectangle\_Y2-50;

int Enemy\_car\_above\_color=RED;

int Enemy\_car\_down\_x1=D\_Car\_Rectangle\_X1+1000;

int Enemy\_car\_down\_y1=D\_Car\_Rectangle\_Y1-50;

int Enemy\_car\_down\_x2=D\_Car\_Rectangle\_X2+1000;

int Enemy\_car\_down\_y2=D\_Car\_Rectangle\_Y2-50;

int Enemy\_car\_down\_color=RED;

int Enemy\_car\_leftwh\_x=L\_Car\_Wheel\_X+1000;

int Enemy\_car\_leftwh\_y=L\_Car\_Wheel\_Y-50;

int Enemy\_car\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car\_leftwh\_color1=WHITE;

int Enemy\_car\_leftwh\_color2=BLACK;

int Enemy\_car\_rightwh\_x=R\_Car\_Wheel\_X+1000;

int Enemy\_car\_rightwh\_y=R\_Car\_Wheel\_Y-50;

int Enemy\_car\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car\_rightwh\_color1=WHITE;

int Enemy\_car\_rightwh\_color2=BLACK;

int Enemy\_car\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1000;

int Enemy\_car\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-50;

int Enemy\_car\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car\_left\_inner\_color1=WHITE;

int Enemy\_car\_left\_inner\_color2=WHITE;

//Enemy Car 2

int Enemy\_car11\_above\_x1=Car\_Rectangle\_X1+1200; //Enemy Car

int Enemy\_car11\_above\_y1=Car\_Rectangle\_Y1+160;

int Enemy\_car11\_above\_x2=Car\_Rectangle\_X2+1200;

int Enemy\_car11\_above\_y2=Car\_Rectangle\_Y2+160;

int Enemy\_car11\_above\_color=RED;

int Enemy\_car11\_down\_x1=D\_Car\_Rectangle\_X1+1200;

int Enemy\_car11\_down\_y1=D\_Car\_Rectangle\_Y1+160;

int Enemy\_car11\_down\_x2=D\_Car\_Rectangle\_X2+1200;

int Enemy\_car11\_down\_y2=D\_Car\_Rectangle\_Y2+160;

int Enemy\_car11\_down\_color=RED;

int Enemy\_car11\_leftwh\_x=L\_Car\_Wheel\_X+1200;

int Enemy\_car11\_leftwh\_y=L\_Car\_Wheel\_Y+160;

int Enemy\_car11\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car11\_leftwh\_color1=WHITE;

int Enemy\_car11\_leftwh\_color2=BLACK;

int Enemy\_car11\_rightwh\_x=R\_Car\_Wheel\_X+1200;

int Enemy\_car11\_rightwh\_y=R\_Car\_Wheel\_Y+160;

int Enemy\_car11\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car11\_rightwh\_color1=WHITE;

int Enemy\_car11\_rightwh\_color2=BLACK;

int Enemy\_car11\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1200;

int Enemy\_car11\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160;

int Enemy\_car11\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car11\_left\_inner\_color1=WHITE;

int Enemy\_car11\_left\_inner\_color2=WHITE;

//Enemy Car 3

int Enemy\_car113\_above\_x1=Car\_Rectangle\_X1+1280; //Enemy Car

int Enemy\_car113\_above\_y1=Car\_Rectangle\_Y1-40-50;

int Enemy\_car113\_above\_x2=Car\_Rectangle\_X2+1280;

int Enemy\_car113\_above\_y2=Car\_Rectangle\_Y2-40-50;

int Enemy\_car113\_above\_color=RED;

int Enemy\_car113\_down\_x1=D\_Car\_Rectangle\_X1+1280;

int Enemy\_car113\_down\_y1=D\_Car\_Rectangle\_Y1-40-50;

int Enemy\_car113\_down\_x2=D\_Car\_Rectangle\_X2+1280;

int Enemy\_car113\_down\_y2=D\_Car\_Rectangle\_Y2-40-50;

int Enemy\_car113\_down\_color=RED;

int Enemy\_car113\_leftwh\_x=L\_Car\_Wheel\_X+1280;

int Enemy\_car113\_leftwh\_y=L\_Car\_Wheel\_Y-40-50;

int Enemy\_car113\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car113\_leftwh\_color1=WHITE;

int Enemy\_car113\_leftwh\_color2=BLACK;

int Enemy\_car113\_rightwh\_x=R\_Car\_Wheel\_X+1280;

int Enemy\_car113\_rightwh\_y=R\_Car\_Wheel\_Y-40-50;

int Enemy\_car113\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car113\_rightwh\_color1=WHITE;

int Enemy\_car113\_rightwh\_color2=BLACK;

int Enemy\_car113\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1280;

int Enemy\_car113\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50;

int Enemy\_car113\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car113\_left\_inner\_color1=WHITE;

int Enemy\_car113\_left\_inner\_color2=WHITE;

//Enemy Car 4

int Enemy\_car114\_above\_x1=Car\_Rectangle\_X1+1360; //Enemy Car

int Enemy\_car114\_above\_y1=Car\_Rectangle\_Y1-40+100;

int Enemy\_car114\_above\_x2=Car\_Rectangle\_X2+1360;

int Enemy\_car114\_above\_y2=Car\_Rectangle\_Y2-40+100;

int Enemy\_car114\_above\_color=RED;

int Enemy\_car114\_down\_x1=D\_Car\_Rectangle\_X1+1360;

int Enemy\_car114\_down\_y1=D\_Car\_Rectangle\_Y1-40+100;

int Enemy\_car114\_down\_x2=D\_Car\_Rectangle\_X2+1360;

int Enemy\_car114\_down\_y2=D\_Car\_Rectangle\_Y2-40+100;

int Enemy\_car114\_down\_color=RED;

int Enemy\_car114\_leftwh\_x=L\_Car\_Wheel\_X+1360;

int Enemy\_car114\_leftwh\_y=L\_Car\_Wheel\_Y-40+100;

int Enemy\_car114\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car114\_leftwh\_color1=WHITE;

int Enemy\_car114\_leftwh\_color2=BLACK;

int Enemy\_car114\_rightwh\_x=R\_Car\_Wheel\_X+1360;

int Enemy\_car114\_rightwh\_y=R\_Car\_Wheel\_Y-40+100;

int Enemy\_car114\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car114\_rightwh\_color1=WHITE;

int Enemy\_car114\_rightwh\_color2=BLACK;

int Enemy\_car114\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1360;

int Enemy\_car114\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40+100;

int Enemy\_car114\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car114\_left\_inner\_color1=WHITE;

int Enemy\_car114\_left\_inner\_color2=WHITE;

//Enemy Car 5

int Enemy\_car115\_above\_x1=Car\_Rectangle\_X1+1430; //Enemy Car

int Enemy\_car115\_above\_y1=Car\_Rectangle\_Y1-40-50+330;

int Enemy\_car115\_above\_x2=Car\_Rectangle\_X2+1430;

int Enemy\_car115\_above\_y2=Car\_Rectangle\_Y2-40-50+330;

int Enemy\_car115\_above\_color=RED;

int Enemy\_car115\_down\_x1=D\_Car\_Rectangle\_X1+1430;

int Enemy\_car115\_down\_y1=D\_Car\_Rectangle\_Y1-40-50+330;

int Enemy\_car115\_down\_x2=D\_Car\_Rectangle\_X2+1430;

int Enemy\_car115\_down\_y2=D\_Car\_Rectangle\_Y2-40-50+330;

int Enemy\_car115\_down\_color=RED;

int Enemy\_car115\_leftwh\_x=L\_Car\_Wheel\_X+1430;

int Enemy\_car115\_leftwh\_y=L\_Car\_Wheel\_Y-40-50+330;

int Enemy\_car115\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car115\_leftwh\_color1=WHITE;

int Enemy\_car115\_leftwh\_color2=BLACK;

int Enemy\_car115\_rightwh\_x=R\_Car\_Wheel\_X+1430;

int Enemy\_car115\_rightwh\_y=R\_Car\_Wheel\_Y-40-50+330;

int Enemy\_car115\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car115\_rightwh\_color1=WHITE;

int Enemy\_car115\_rightwh\_color2=BLACK;

int Enemy\_car115\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1430;

int Enemy\_car115\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50+330;

int Enemy\_car115\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car115\_left\_inner\_color1=WHITE;

int Enemy\_car115\_left\_inner\_color2=WHITE;

//Enemy Car 6

int Enemy\_car116\_above\_x1=Car\_Rectangle\_X1+1390; //Enemy Car

int Enemy\_car116\_above\_y1=Car\_Rectangle\_Y1+160-140;

int Enemy\_car116\_above\_x2=Car\_Rectangle\_X2+1390;

int Enemy\_car116\_above\_y2=Car\_Rectangle\_Y2+160-140;

int Enemy\_car116\_above\_color=RED;

int Enemy\_car116\_down\_x1=D\_Car\_Rectangle\_X1+1390;

int Enemy\_car116\_down\_y1=D\_Car\_Rectangle\_Y1+160-140;

int Enemy\_car116\_down\_x2=D\_Car\_Rectangle\_X2+1390;

int Enemy\_car116\_down\_y2=D\_Car\_Rectangle\_Y2+160-140;

int Enemy\_car116\_down\_color=RED;

int Enemy\_car116\_leftwh\_x=L\_Car\_Wheel\_X+1390;

int Enemy\_car116\_leftwh\_y=L\_Car\_Wheel\_Y+160-140;

int Enemy\_car116\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car116\_leftwh\_color1=WHITE;

int Enemy\_car116\_leftwh\_color2=BLACK;

int Enemy\_car116\_rightwh\_x=R\_Car\_Wheel\_X+1390;

int Enemy\_car116\_rightwh\_y=R\_Car\_Wheel\_Y+160-140;

int Enemy\_car116\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car116\_rightwh\_color1=WHITE;

int Enemy\_car116\_rightwh\_color2=BLACK;

int Enemy\_car116\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1390;

int Enemy\_car116\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160-140;

int Enemy\_car116\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car116\_left\_inner\_color1=WHITE;

int Enemy\_car116\_left\_inner\_color2=WHITE;

// Constant Enemy Cars 1

int Enemy\_car117\_above\_x1=Car\_Rectangle\_X1+1220; //Enemy Car

int Enemy\_car117\_above\_y1=Car\_Rectangle\_Y1+160-105;

int Enemy\_car117\_above\_x2=Car\_Rectangle\_X2+1220;

int Enemy\_car117\_above\_y2=Car\_Rectangle\_Y2+160-105;

int Enemy\_car117\_above\_color=RED;

int Enemy\_car117\_down\_x1=D\_Car\_Rectangle\_X1+1220;

int Enemy\_car117\_down\_y1=D\_Car\_Rectangle\_Y1+160-105;

int Enemy\_car117\_down\_x2=D\_Car\_Rectangle\_X2+1220;

int Enemy\_car117\_down\_y2=D\_Car\_Rectangle\_Y2+160-105;

int Enemy\_car117\_down\_color=RED;

int Enemy\_car117\_leftwh\_x=L\_Car\_Wheel\_X+1220;

int Enemy\_car117\_leftwh\_y=L\_Car\_Wheel\_Y+160-105;

int Enemy\_car117\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car117\_leftwh\_color1=WHITE;

int Enemy\_car117\_leftwh\_color2=BLACK;

int Enemy\_car117\_rightwh\_x=R\_Car\_Wheel\_X+1220;

int Enemy\_car117\_rightwh\_y=R\_Car\_Wheel\_Y+160-105;

int Enemy\_car117\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car117\_rightwh\_color1=WHITE;

int Enemy\_car117\_rightwh\_color2=BLACK;

int Enemy\_car117\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1220;

int Enemy\_car117\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160-105;

int Enemy\_car117\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car117\_left\_inner\_color1=WHITE;

int Enemy\_car117\_left\_inner\_color2=WHITE;

// Constant Enemy Cars 2

int Enemy\_car118\_above\_x1=Car\_Rectangle\_X1+1540; //Enemy Car

int Enemy\_car118\_above\_y1=Car\_Rectangle\_Y1+160-56;

int Enemy\_car118\_above\_x2=Car\_Rectangle\_X2+1540;

int Enemy\_car118\_above\_y2=Car\_Rectangle\_Y2+160-56;

int Enemy\_car118\_above\_color=RED;

int Enemy\_car118\_down\_x1=D\_Car\_Rectangle\_X1+1540;

int Enemy\_car118\_down\_y1=D\_Car\_Rectangle\_Y1+160-56;

int Enemy\_car118\_down\_x2=D\_Car\_Rectangle\_X2+1540;

int Enemy\_car118\_down\_y2=D\_Car\_Rectangle\_Y2+160-56;

int Enemy\_car118\_down\_color=RED;

int Enemy\_car118\_leftwh\_x=L\_Car\_Wheel\_X+1540;

int Enemy\_car118\_leftwh\_y=L\_Car\_Wheel\_Y+160-56;

int Enemy\_car118\_leftwh\_rad=L\_Car\_Wheel\_Rad;

int Enemy\_car118\_leftwh\_color1=WHITE;

int Enemy\_car118\_leftwh\_color2=BLACK;

int Enemy\_car118\_rightwh\_x=R\_Car\_Wheel\_X+1540;

int Enemy\_car118\_rightwh\_y=R\_Car\_Wheel\_Y+160-56;

int Enemy\_car118\_rightwh\_rad=R\_Car\_Wheel\_Rad;

int Enemy\_car118\_rightwh\_color1=WHITE;

int Enemy\_car118\_rightwh\_color2=BLACK;

int Enemy\_car118\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1540;

int Enemy\_car118\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160-56;

int Enemy\_car118\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

int Enemy\_car118\_left\_inner\_color1=WHITE;

int Enemy\_car118\_left\_inner\_color2=WHITE;

int getRandomPositionWithinRange(int a,int b){

srand(time(NULL));

return a + rand()% (b-a);

}

int getRandomSign(){

srand(time(NULL));

if(rand()%2 == 0)

return 1;

return -1;

}

void con1EnemyCarReIncarnated\_2()

{

int inc1 = getRandomSign() \* getRandomPositionWithinRange(1,100);

Enemy\_car117\_above\_x1=Car\_Rectangle\_X1+1200; //Enemy Car

Enemy\_car117\_above\_y1=Car\_Rectangle\_Y1+160+inc1;

Enemy\_car117\_above\_x2=Car\_Rectangle\_X2+1200;

Enemy\_car117\_above\_y2=Car\_Rectangle\_Y2+160+inc1;

Enemy\_car117\_above\_color=RED;

Enemy\_car117\_down\_x1=D\_Car\_Rectangle\_X1+1200;

Enemy\_car117\_down\_y1=D\_Car\_Rectangle\_Y1+160+inc1;

Enemy\_car117\_down\_x2=D\_Car\_Rectangle\_X2+1200;

Enemy\_car117\_down\_y2=D\_Car\_Rectangle\_Y2+160+inc1;

Enemy\_car117\_down\_color=RED;

Enemy\_car117\_leftwh\_x=L\_Car\_Wheel\_X+1200;

Enemy\_car117\_leftwh\_y=L\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car117\_leftwh\_color1=WHITE;

Enemy\_car117\_leftwh\_color2=BLACK;

Enemy\_car117\_rightwh\_x=R\_Car\_Wheel\_X+1200;

Enemy\_car117\_rightwh\_y=R\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car117\_rightwh\_color1=WHITE;

Enemy\_car117\_rightwh\_color2=BLACK;

Enemy\_car117\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1200;

Enemy\_car117\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160+inc1;

Enemy\_car117\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car117\_left\_inner\_color1=WHITE;

Enemy\_car117\_left\_inner\_color2=WHITE;

}

void con2EnemyCarReIncarnated\_2()

{

int inc1 = getRandomSign() \* getRandomPositionWithinRange(1,45);

Enemy\_car118\_above\_x1=Car\_Rectangle\_X1+1600; //Enemy Car

Enemy\_car118\_above\_y1=Car\_Rectangle\_Y1+160+inc1;

Enemy\_car118\_above\_x2=Car\_Rectangle\_X2+1600;

Enemy\_car118\_above\_y2=Car\_Rectangle\_Y2+160+inc1;

Enemy\_car118\_above\_color=RED;

Enemy\_car118\_down\_x1=D\_Car\_Rectangle\_X1+1600;

Enemy\_car118\_down\_y1=D\_Car\_Rectangle\_Y1+160+inc1;

Enemy\_car118\_down\_x2=D\_Car\_Rectangle\_X2+1600;

Enemy\_car118\_down\_y2=D\_Car\_Rectangle\_Y2+160+inc1;

Enemy\_car118\_down\_color=RED;

Enemy\_car118\_leftwh\_x=L\_Car\_Wheel\_X+1600;

Enemy\_car118\_leftwh\_y=L\_Car\_Wheel\_Y+160+inc1;

Enemy\_car118\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car118\_leftwh\_color1=WHITE;

Enemy\_car118\_leftwh\_color2=BLACK;

Enemy\_car118\_rightwh\_x=R\_Car\_Wheel\_X+1600;

Enemy\_car118\_rightwh\_y=R\_Car\_Wheel\_Y+160+inc1;

Enemy\_car118\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car118\_rightwh\_color1=WHITE;

Enemy\_car118\_rightwh\_color2=BLACK;

Enemy\_car118\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1600;

Enemy\_car118\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160+inc1;

Enemy\_car118\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car118\_left\_inner\_color1=WHITE;

Enemy\_car118\_left\_inner\_color2=WHITE;

}

void EnemyCar115ReIncarnated\_2()

{

Enemy\_car115\_above\_x1=Car\_Rectangle\_X1+1400; //Enemy Car

Enemy\_car115\_above\_y1=Car\_Rectangle\_Y1-40-50+330;

Enemy\_car115\_above\_x2=Car\_Rectangle\_X2+1400;

Enemy\_car115\_above\_y2=Car\_Rectangle\_Y2-40-50+330;

Enemy\_car115\_above\_color=RED;

Enemy\_car115\_down\_x1=D\_Car\_Rectangle\_X1+1400;

Enemy\_car115\_down\_y1=D\_Car\_Rectangle\_Y1-40-50+330;

Enemy\_car115\_down\_x2=D\_Car\_Rectangle\_X2+1400;

Enemy\_car115\_down\_y2=D\_Car\_Rectangle\_Y2-40-50+330;

Enemy\_car115\_down\_color=RED;

Enemy\_car115\_leftwh\_x=L\_Car\_Wheel\_X+1400;

Enemy\_car115\_leftwh\_y=L\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car115\_leftwh\_color1=WHITE;

Enemy\_car115\_leftwh\_color2=BLACK;

Enemy\_car115\_rightwh\_x=R\_Car\_Wheel\_X+1400;

Enemy\_car115\_rightwh\_y=R\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car115\_rightwh\_color1=WHITE;

Enemy\_car115\_rightwh\_color2=BLACK;

Enemy\_car115\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1400;

Enemy\_car115\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50+330;

Enemy\_car115\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car115\_left\_inner\_color1=WHITE;

Enemy\_car115\_left\_inner\_color2=WHITE;

}

void EnemyCar113ReIncarnated\_2()

{

Enemy\_car113\_above\_x1=Car\_Rectangle\_X1+1280; //Enemy Car

Enemy\_car113\_above\_y1=Car\_Rectangle\_Y1 -40-50;

Enemy\_car113\_above\_x2=Car\_Rectangle\_X2+1280;

Enemy\_car113\_above\_y2=Car\_Rectangle\_Y2 -40-50;

Enemy\_car113\_above\_color=RED;

Enemy\_car113\_down\_x1=D\_Car\_Rectangle\_X1+1280;

Enemy\_car113\_down\_y1=D\_Car\_Rectangle\_Y1-40-50;

Enemy\_car113\_down\_x2=D\_Car\_Rectangle\_X2+1280;

Enemy\_car113\_down\_y2=D\_Car\_Rectangle\_Y2-40-50;

Enemy\_car113\_down\_color=RED;

Enemy\_car113\_leftwh\_x=L\_Car\_Wheel\_X+1280;

Enemy\_car113\_leftwh\_y=L\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car113\_leftwh\_color1=WHITE;

Enemy\_car113\_leftwh\_color2=BLACK;

Enemy\_car113\_rightwh\_x=R\_Car\_Wheel\_X+1280;

Enemy\_car113\_rightwh\_y=R\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car113\_rightwh\_color1=WHITE;

Enemy\_car113\_rightwh\_color2=BLACK;

Enemy\_car113\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1280;

Enemy\_car113\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40-50;

Enemy\_car113\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car113\_left\_inner\_color1=WHITE;

Enemy\_car113\_left\_inner\_color2=WHITE;

}

void EnemyCar11ReIncarnated\_2()

{

Enemy\_car11\_above\_x1=Car\_Rectangle\_X1+1200; //Enemy Car

Enemy\_car11\_above\_y1=Car\_Rectangle\_Y1+160;

Enemy\_car11\_above\_x2=Car\_Rectangle\_X2+1200;

Enemy\_car11\_above\_y2=Car\_Rectangle\_Y2+160;

Enemy\_car11\_above\_color=RED;

Enemy\_car11\_down\_x1=D\_Car\_Rectangle\_X1+1200;

Enemy\_car11\_down\_y1=D\_Car\_Rectangle\_Y1+160;

Enemy\_car11\_down\_x2=D\_Car\_Rectangle\_X2+1200;

Enemy\_car11\_down\_y2=D\_Car\_Rectangle\_Y2+160;

Enemy\_car11\_down\_color=RED;

Enemy\_car11\_leftwh\_x=L\_Car\_Wheel\_X+1200;

Enemy\_car11\_leftwh\_y=L\_Car\_Wheel\_Y+160;

Enemy\_car11\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car11\_leftwh\_color1=WHITE;

Enemy\_car11\_leftwh\_color2=BLACK;

Enemy\_car11\_rightwh\_x=R\_Car\_Wheel\_X+1200;

Enemy\_car11\_rightwh\_y=R\_Car\_Wheel\_Y+160;

Enemy\_car11\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car11\_rightwh\_color1=WHITE;

Enemy\_car11\_rightwh\_color2=BLACK;

Enemy\_car11\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1200;

Enemy\_car11\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160;

Enemy\_car11\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car11\_left\_inner\_color1=WHITE;

Enemy\_car11\_left\_inner\_color2=WHITE;

}

void EnemyCar114ReIncarnated\_2()

{

Enemy\_car114\_above\_x1=Car\_Rectangle\_X1+1360; //Enemy Car

Enemy\_car114\_above\_y1=Car\_Rectangle\_Y1-40+100;

Enemy\_car114\_above\_x2=Car\_Rectangle\_X2+1360;

Enemy\_car114\_above\_y2=Car\_Rectangle\_Y2-40+100;

Enemy\_car114\_above\_color=RED;

Enemy\_car114\_down\_x1=D\_Car\_Rectangle\_X1+1360;

Enemy\_car114\_down\_y1=D\_Car\_Rectangle\_Y1-40+100;

Enemy\_car114\_down\_x2=D\_Car\_Rectangle\_X2+1360;

Enemy\_car114\_down\_y2=D\_Car\_Rectangle\_Y2-40+100;

Enemy\_car114\_down\_color=RED;

Enemy\_car114\_leftwh\_x=L\_Car\_Wheel\_X+1360;

Enemy\_car114\_leftwh\_y=L\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car114\_leftwh\_color1=WHITE;

Enemy\_car114\_leftwh\_color2=BLACK;

Enemy\_car114\_rightwh\_x=R\_Car\_Wheel\_X+1360;

Enemy\_car114\_rightwh\_y=R\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car114\_rightwh\_color1=WHITE;

Enemy\_car114\_rightwh\_color2=BLACK;

Enemy\_car114\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1360;

Enemy\_car114\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y-40+100;

Enemy\_car114\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car114\_left\_inner\_color1=WHITE;

Enemy\_car114\_left\_inner\_color2=WHITE;

}

void EnemyCar116ReIncarnated\_2()

{

Enemy\_car116\_above\_x1=Car\_Rectangle\_X1+1390; //Enemy Car

Enemy\_car116\_above\_y1=Car\_Rectangle\_Y1+160-140;

Enemy\_car116\_above\_x2=Car\_Rectangle\_X2+1390;

Enemy\_car116\_above\_y2=Car\_Rectangle\_Y2+160-140;

Enemy\_car116\_above\_color=RED;

Enemy\_car116\_down\_x1=D\_Car\_Rectangle\_X1+1390;

Enemy\_car116\_down\_y1=D\_Car\_Rectangle\_Y1+160-140;

Enemy\_car116\_down\_x2=D\_Car\_Rectangle\_X2+1390;

Enemy\_car116\_down\_y2=D\_Car\_Rectangle\_Y2+160-140;

Enemy\_car116\_down\_color=RED;

Enemy\_car116\_leftwh\_x=L\_Car\_Wheel\_X+1390;

Enemy\_car116\_leftwh\_y=L\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_leftwh\_rad=L\_Car\_Wheel\_Rad;

Enemy\_car116\_leftwh\_color1=WHITE;

Enemy\_car116\_leftwh\_color2=BLACK;

Enemy\_car116\_rightwh\_x=R\_Car\_Wheel\_X+1390;

Enemy\_car116\_rightwh\_y=R\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_rightwh\_rad=R\_Car\_Wheel\_Rad;

Enemy\_car116\_rightwh\_color1=WHITE;

Enemy\_car116\_rightwh\_color2=BLACK;

Enemy\_car116\_left\_inner\_x=L\_Inner\_Car\_Wheel\_X+1390;

Enemy\_car116\_left\_inner\_y=L\_Inner\_Car\_Wheel\_Y+160-140;

Enemy\_car116\_left\_inner\_rad=L\_Inner\_Car\_Wheel\_Rad;

Enemy\_car116\_left\_inner\_color1=WHITE;

Enemy\_car116\_left\_inner\_color2=WHITE;

}

**Car\_My.h :**

#include"graphics.h"

typedef struct myCar

{

int car\_above\_x1;

int car\_above\_y1;

int car\_above\_x2;

int car\_above\_y2;

int car\_above\_color;

int car\_down\_x1;

int car\_down\_y1;

int car\_down\_x2;

int car\_down\_y2;

int car\_down\_color;

int car\_leftwh\_x;

int car\_leftwh\_y;

int car\_leftwh\_rad;

int car\_leftwh\_color1;

int car\_leftwh\_color2;

int car\_rightwh\_x;

int car\_rightwh\_y;

int car\_rightwh\_rad;

int car\_rightwh\_color1;

int car\_rightwh\_color2;

int car\_left\_inner\_x;

int car\_left\_inner\_y;

int car\_left\_inner\_rad;

int car\_left\_inner\_color1;

int car\_left\_inner\_color2;

int car\_right\_inner\_x;

int car\_right\_inner\_y;

int car\_right\_inner\_rad;

int car\_right\_inner\_color1;

int car\_right\_inner\_color2;

myCar()

{

car\_above\_x1 = 85;

car\_above\_y1 = 262;

car\_above\_x2 = 150-17-7;

car\_above\_y2 = 295-10-4;

car\_above\_color = 1;

car\_down\_x1 = 61;

car\_down\_y1 = 305-20-5;

car\_down\_x2 = 235-55-20-10;

car\_down\_y2 = 345-25-10-7;

car\_down\_color = 1;

car\_leftwh\_x = 78;

car\_leftwh\_y = 352-36-12;

car\_leftwh\_rad = 9;

car\_leftwh\_color1 = 0;

car\_leftwh\_color2 = 15;

car\_rightwh\_x = 155-16-8;

car\_rightwh\_y = 352-36-12;

car\_rightwh\_rad = 9;

car\_rightwh\_color1 = 12;

car\_rightwh\_color2 = 12;

car\_left\_inner\_x = 78;

car\_left\_inner\_y = 352-36-12;

car\_left\_inner\_rad = 6;

car\_left\_inner\_color1 = 12 ;

car\_left\_inner\_color2 = 12 ;

car\_right\_inner\_x = 155-16-8;

car\_right\_inner\_y = 352-36-12;

car\_right\_inner\_rad = 6;

car\_right\_inner\_color1 = 12 ;

car\_right\_inner\_color2 = 12 ;

}

void CarrReinCarnation1()

{

car\_above\_x1 = 85+240;

car\_above\_y1 = 262-40;

car\_above\_x2 = 150-17-7+240;

car\_above\_y2 = 295-10-4-40;

car\_above\_color = 1;

car\_down\_x1 = 61+240;

car\_down\_y1 = 305-20-5-40;

car\_down\_x2 = 235-55-20-10+240;

car\_down\_y2 = 345-25-10-7-40;

car\_down\_color = 1;

car\_leftwh\_x = 78+240;

car\_leftwh\_y = 352-36-12-40;

car\_leftwh\_rad = 9;

car\_leftwh\_color1 = 0;

car\_leftwh\_color2 = 15;

car\_rightwh\_x = 155-16-8+240;

car\_rightwh\_y = 352-36-12-40;

car\_rightwh\_rad = 9;

car\_rightwh\_color1 = 12;

car\_rightwh\_color2 = 12;

car\_left\_inner\_x = 78+240;

car\_left\_inner\_y = 352-36-12-40;

car\_left\_inner\_rad = 6;

car\_left\_inner\_color1 = 12 ;

car\_left\_inner\_color2 = 12 ;

car\_right\_inner\_x = 155-16-8+240;

car\_right\_inner\_y = 352-36-12-40;

car\_right\_inner\_rad = 6;

car\_right\_inner\_color1 = 12 ;

car\_right\_inner\_color2 = 12 ;

}

void CarrReinCarnation2()

{

car\_above\_x1 = 85+130;

car\_above\_y1 = 262+145+40;

car\_above\_x2 = 150-17-7+130;

car\_above\_y2 = 295-10-4+145+40;

car\_above\_color = 1;

car\_down\_x1 = 61+130;

car\_down\_y1 = 305-20-5+145+40;

car\_down\_x2 = 235-55-20-10+130;

car\_down\_y2 = 345-25-10-7+145+40;

car\_down\_color = 1;

car\_leftwh\_x = 78+130;

car\_leftwh\_y = 352-36-12+145+40;

car\_leftwh\_rad = 9;

car\_leftwh\_color1 = 0;

car\_leftwh\_color2 = 15;

car\_rightwh\_x = 155-16-8+130;

car\_rightwh\_y = 352-36-12+145+40;

car\_rightwh\_rad = 9;

car\_rightwh\_color1 = 12;

car\_rightwh\_color2 = 12;

car\_left\_inner\_x = 78+130;

car\_left\_inner\_y = 352-36-12+145+40;

car\_left\_inner\_rad = 6;

car\_left\_inner\_color1 = 12 ;

car\_left\_inner\_color2 = 12 ;

car\_right\_inner\_x = 155-16-8+130;

car\_right\_inner\_y = 352-36-12+145+40;

car\_right\_inner\_rad = 6;

car\_right\_inner\_color1 = 12 ;

car\_right\_inner\_color2 = 12 ;

}

void Carr(int color)

{

Above\_Part\_Car(car\_above\_x1,car\_above\_y1,car\_above\_x2,car\_above\_y2,color);

Down\_Part\_Car(car\_down\_x1,car\_down\_y1,car\_down\_x2,car\_down\_y2,color);

Left\_Wheel\_Car(car\_leftwh\_x,car\_leftwh\_y,car\_leftwh\_rad,color);

Right\_Wheel\_Car(car\_rightwh\_x,car\_rightwh\_y,car\_rightwh\_rad,color);

Left\_Inner\_Wheel\_Car(car\_left\_inner\_x,car\_left\_inner\_y,car\_left\_inner\_rad,color);

Right\_Inner\_Wheel\_Car(car\_right\_inner\_x,car\_right\_inner\_y,car\_right\_inner\_rad,color);

}

void Above\_Part\_Car(int a,int b,int c,int d,int color)

{

setcolor( color); //(Above Car)

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,color);

bar(a,b,c,d);

}

void Down\_Part\_Car(int a,int b,int c,int d,int color)

{

setcolor(color); //(Down Car)

rectangle(a,b,c,d);

setfillstyle(SOLID\_FILL,color);

bar(a,b,c,d);

}

void Left\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color); //Left Wheel

circle(a,b,c);

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Right\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color);

circle(a,b,c); //Right Wheel

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Left\_Inner\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color); //Left Inner Wheel

circle(a,b,c);

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void Right\_Inner\_Wheel\_Car(int a,int b,int c,int color)

{

setcolor(color);

circle(a,b,c); //Right Inner Wheel

setfillstyle(SOLID\_FILL,color);

fillellipse(a,b,c,c);

}

void MovingCar(int colors[],int Speed,int count)

{

if(count==1)

{

Above\_Part\_Car(car\_above\_x1,car\_above\_y1-=Speed,car\_above\_x2,car\_above\_y2-=Speed,colors[0]);

Down\_Part\_Car(car\_down\_x1,car\_down\_y1-=Speed,car\_down\_x2,car\_down\_y2-=Speed,colors[1]);

Left\_Wheel\_Car(car\_leftwh\_x,car\_leftwh\_y-=Speed,car\_leftwh\_rad,colors[2]);

Right\_Wheel\_Car(car\_rightwh\_x,car\_rightwh\_y-=Speed,car\_rightwh\_rad,colors[3]);

Left\_Inner\_Wheel\_Car(car\_left\_inner\_x,car\_left\_inner\_y-=Speed,car\_left\_inner\_rad,colors[4]);

Right\_Inner\_Wheel\_Car(car\_right\_inner\_x,car\_right\_inner\_y-=Speed,car\_right\_inner\_rad,colors[5]);

}

if(count==2)

{

Above\_Part\_Car(car\_above\_x1-=Speed,car\_above\_y1,car\_above\_x2-=Speed,car\_above\_y2,colors[0]);

Down\_Part\_Car(car\_down\_x1-=Speed,car\_down\_y1,car\_down\_x2-=Speed,car\_down\_y2,colors[1]);

Left\_Wheel\_Car(car\_leftwh\_x-=Speed,car\_leftwh\_y,car\_leftwh\_rad,colors[2]);

Right\_Wheel\_Car(car\_rightwh\_x-=Speed,car\_rightwh\_y,car\_rightwh\_rad,colors[3]);

Left\_Inner\_Wheel\_Car(car\_left\_inner\_x-=Speed,car\_left\_inner\_y,car\_left\_inner\_rad,colors[4]);

Right\_Inner\_Wheel\_Car(car\_right\_inner\_x-=Speed,car\_right\_inner\_y,car\_right\_inner\_rad,colors[5]);

}

if(count==3)

{

Above\_Part\_Car(car\_above\_x1+=Speed,car\_above\_y1,car\_above\_x2+=Speed,car\_above\_y2,colors[0]);

Down\_Part\_Car(car\_down\_x1+=Speed,car\_down\_y1,car\_down\_x2+=Speed,car\_down\_y2,colors[1]);

Left\_Wheel\_Car(car\_leftwh\_x+=Speed,car\_leftwh\_y,car\_leftwh\_rad,colors[2]);

Right\_Wheel\_Car(car\_rightwh\_x+=Speed,car\_rightwh\_y,car\_rightwh\_rad,colors[3]);

Left\_Inner\_Wheel\_Car(car\_left\_inner\_x+=Speed,car\_left\_inner\_y,car\_left\_inner\_rad,colors[4]);

Right\_Inner\_Wheel\_Car(car\_right\_inner\_x+=Speed,car\_right\_inner\_y,car\_right\_inner\_rad,colors[5]);

}

if(count==4)

{

Above\_Part\_Car(car\_above\_x1,car\_above\_y1+=Speed,car\_above\_x2,car\_above\_y2+=Speed,colors[0]);

Down\_Part\_Car(car\_down\_x1,car\_down\_y1+=Speed,car\_down\_x2,car\_down\_y2+=Speed,colors[1]);

Left\_Wheel\_Car(car\_leftwh\_x,car\_leftwh\_y+=Speed,car\_leftwh\_rad,colors[2]);

Right\_Wheel\_Car(car\_rightwh\_x,car\_rightwh\_y+=Speed,car\_rightwh\_rad,colors[3]);

Left\_Inner\_Wheel\_Car(car\_left\_inner\_x,car\_left\_inner\_y+=Speed,car\_left\_inner\_rad,colors[4]);

Right\_Inner\_Wheel\_Car(car\_right\_inner\_x,car\_right\_inner\_y+=Speed,car\_right\_inner\_rad,colors[5]);

}

delay(0);

}

};

**12. Limitations :**

It’s very hard to debug a code while doing these kind of works. It may take someone hours to find a small bug in the code. So bugs are bad for our lives!

BGI has limits, after some time the project may not work like before. It may become slower!

The speed of the enemy car increases in levels. The higher the level the more the speed, but it blinks too much!

**13. Constraints:**

This game was created using the BGI library which is severely backdated and has many drawbacks. The performance of the game varies from machine to machine. It was hard to figure out what to do and what not to!

These functions are slow. It was hard to figure out what they do! But <http://www.cs.colorado.edu/~main/bgi/doc/> , this website helped us a lot! It was hard to sort the Scores with Username but with the help of the Internet, I was able to overcome it!

**14. Conclusion:**

To ensure that this program works on all Windows devices, the .exe file was built using Release Mode on Visual Studio 2010. The game was coded to make optimal use of the BGI functions and to negate most of its drawbacks. The idea was to implement a small Racing game and although not entirely similar, Race – “Where It All Begins” has the lights of Enemy car and ghosts! Which gives a taste of2D (actually 1D) gaming experiences!

I would like to thank our honorable teachers – Dr. Md. Asif Hossain Khan and our beloved Anna Faria ma’am for their continuous suggestions and care!