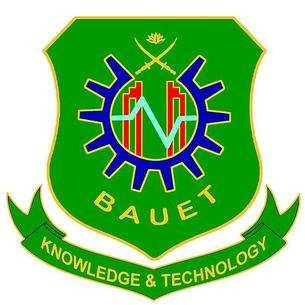
Bangladesh Army University of Engineering & Technology (BAUET)

Qadirabad Cantonment, Natore-6431



**Department of Information and Communication Engineering (ICE)**

Project Report

Obtained Mark Signature & Date

|  |  |
| --- | --- |
| **Course Code:** | ICE 2232 |
| **Course Title:** | Object Oriented Programming |
| **Project:** | A simple Car Game using C++ |

|  |
| --- |
| *Submitted By* |
| **Name: Airin Jahan Akhi** |
| **Student ID: 20105005** |
| **Batch: 9th** |
| **Year: 2nd** |
| **Semester: 2nd** |

|  |
| --- |
| *Submitted To* |
| **Name: Min-Ha-zul Abedin** |
| **Designation: Lecturer** |
| **Dept: ICE** |
|  |
|  |

*Date of Submission:*

**Project Tittle:** A simple Car Game using C++.

**Requirements:** This is a simple Car game. There are three levels in this game, they are:

1. Level-1 (EASY)
2. Level-2 (MEDIUM)
3. Level-3 (HARD)

At the starting of the game, we need to choose a level to play the Car game.

* If we choose 1 then the game will run as Level-1 (EASY)
* If we choose 2 then it will run as Level-2 (MEDIUM).
* If we choose 3 then it will run as Level-3 (HARD)

**Level-1 (EASY)**

When we play the game as Level-1 then we have to choose “1” and press the key “1” then the game will run as EASY level.

In this level enemies will come slower than the Medium and Hard level. So, we have to turn the car Left to Right pressing the key “A” and “J”

In this level if the car touches the enemy one times but score is less than 10 then game will over. But if the car touches the enemy but score is greater than 10 then game will not over but if touches two times game will over.

Here I applied sleep() function and it is sleep(100) for this the enemies will come slowly in this level.

**Level-2 (MEDIUM)**

When we play the game as Level-2 then we have to choose “2” and press the key “2” then the game will run as medium level.

In this level enemies will come faster than the Level-1 and slower than

Level-2. So we have to turn the car Left to Right faster pressing the key “A” and “J”

In this level if the car touches the enemy one times then game will over.

**Level-3 (HARD)**

When we play the game as Level-3 then we have to choose “3” and press the key “3” then the game will run as medium level.

In this level enemies will come fastest than the Level-1 and Level-2. So we have to turn the car Left to Right as fast as possible pressing the key “A” and “J”.

In this level if the car touches the enemy one times then game will over as the level-2.

**Introduction**: Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

***Object:***

This is the basic unit of object-oriented programming. That is both data and function that operate on data are bundled as a unit called an object.

***Class:***

Class is a blueprint for an object. This doesn't actually define any data, but it does define the class name means, an object of the class will consist of and operations can be performed on such an object.

OOP has four basic concepts: 1. **Abstraction**

2. Encapsulation

3. Inheritance

4. Polymorphism

Here are some of the header files and its descriptions used in this game:

|  |  |
| --- | --- |
| Header files | Description |
| #include <iostream> | Is a header file library that lets us work with input and output objects, such as cout,cin etc. |
| #include <conio.h> | Is a header file, stands for “Console Input & Output”. It contains the functions of console input and output |
| #include <dos.h> | dos.h is a header file of C Language. This library has functions that are used for handling interrupts, producing sound, date and time functions, etc. |
| #include<windows.h> | <windows.h> header file is used to access the Win32, in particular includes the library and functions used in the libraries like stdio.h or stdlib.h. And it works for sleep() function also. |
| #include <time.h> | <time.h> is the name of one such header file that deals with time-oriented functions. |
| #define SCREEN\_WIDTH 90 | It defines the width of the game screen |
| #define SCREEN\_HEIGHT 26 | It defines the height of the game screen |
| #define WIN\_WIDTH 70 | Defines the window width |

**using namespace std;** when we run a program to print something, “using namespace std” says if you find something that is not declared in the current scope. computer needs to know the code for the cout, cin functionalities and it needs to know which namespace they are defined.

Here are some of the data types and its descriptions used in this game:

|  |  |
| --- | --- |
| Data type | Description |
| int | Integer occupies 2 bytes (16 bits) of memory to store an integer value. |
| char | It occupies a memory size of 1 byte. C++ Char only stores single character. |
| bool | It manage and write conditional statements using a boolean value, rather than an int. |

Here are some of the functions and descriptions used in this game:

|  |  |
| --- | --- |
| Functions | Descriptions |
| drawBorder() | This function is used to draw a line from a point (x1,y1) to point (x2,y2) |
| gotoxy() | This function places the cursor at the desired location on the screen. |
| sleep () | Sleep () function suspends the execution of the program for a specified period of time. This time period is specified as an argument to the sleep () function |
| void function\_name() | This void function means pointer to something left unspecified |
| void genEnemy(int ind) | This function is used for declare enemy location |
| void drawEnemy(int ind) | This function is used for enemy shape |
| void eraseEnemy(int ind) | This function is used for enemy deletion |
| void resetEnemy(int ind) | This function is used for new enemy coming |
| void drawCar() | This function is used for car drawing |
| collision() | This function is used for collision between car and enemy |
| void gameover() | This is used for over the game |
| System() | This is used to invoke an operating system command from a C/C++ program. |
| void updateScore() | This is used for update the score |
| void fast() | This is used for another level of the game |
| void play() | This is used for ply the game in first level |
| void fastest() | Is used for last level of the game |
| getch() | This function takes in a single character from the standard input ( stdin ), and returns an integer. |

***Loop:*** In computer programming, a loop is a sequence of instructions that is repeated until a certain condition is reached. An operation is done, such as getting an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number.

* **For loop:** In computer programming, a loop is a sequence of instructions that is repeated until a certain condition is reached. An operation is done, such as getting an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number.

**\_\_**In this game I used for loop for the screen height to increase the game screen height continuously.

* **While loop:** In while loop, condition is evaluated first and if it returns true then the statements inside while loop execute, this happens repeatedly until the condition returns false. When condition returns false, the control comes out of loop and jumps to the next statement in the program after while loop.

**­­\_\_**In this game I used while loop for continue the game with car control.

***Break;*** The break in C or C++ is a loop control statement which is used to terminate the loop. As soon as the break statement is encountered from within a loop, the loop iterations stops there and control returns from the loop immediately to the first statement after the loop.

***Return 0;*** The return statement returns the flow of the execution to the function from where it is called.

**Code:**

#include <iostream>

#include <conio.h>

#include <dos.h>

#include <windows.h>

#include <time.h>

#define SCREEN\_WIDTH 90

#define SCREEN\_HEIGHT 26

#define WIN\_WIDTH 70

using namespace std;

HANDLE console = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD CursorPosition;

int enemy\_y[3], enemy\_x[3], enemyF[3];

char mycar[4][4] = { ' ','±','±',' ','±','±','±','±',

' ','±','±',' ','±','±','±','±'

};

int carPos=WIN\_WIDTH/2;

int score=0;

void gotoxy(int x, int y)

{

CursorPosition.X=x;

CursorPosition.Y=y;

SetConsoleCursorPosition(console, CursorPosition);

}

void setcursor(bool visible, DWORD size)

{

if(size==0)

{

size=20;

}

CONSOLE\_CURSOR\_INFO lpCursor;

lpCursor.bVisible = visible;

lpCursor.dwSize = size;

SetConsoleCursorInfo(console, &lpCursor);

}

void drawBorder()

{

for(int i = 0; i < SCREEN\_HEIGHT; i++)

{

for(int j = 0; j < 17; j++)

{

gotoxy(0 + j, i);

cout << "±";

gotoxy(WIN\_WIDTH - j, i);

cout << "±";

}

}

for(int i = 0; i < SCREEN\_HEIGHT; i++)

{

gotoxy(SCREEN\_WIDTH,i);

cout << "±";

}}

void genEnemy(int ind)

{

enemy\_x[ind] = 17 + rand() % (33);

}

void drawEnemy(int ind)

{

if( enemyF[ind] == true)

{

gotoxy(enemy\_x[ind], enemy\_y[ind]);

cout << "\*\*\*\*";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 1);

cout << " \*\* ";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 2);

cout << "\*\*\*\*";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 3);

cout << " \*\* ";

}}

void eraseEnemy(int ind)

{

if( enemyF[ind]==true)

{

gotoxy(enemy\_x[ind], enemy\_y[ind]);

cout << " ";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 1);

cout << " ";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 2);

cout << " ";

gotoxy(enemy\_x[ind], enemy\_y[ind] + 3);

cout << " ";

}}

void resetEnemy(int ind)

{

eraseEnemy(ind);

enemy\_y[ind] = 1;

genEnemy(ind);

}

void drawCar()

{

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

{

for(int j = 0; j < 4; j++)

{

gotoxy(j + carPos, i + 22);

cout << mycar[i][j];

} }}

void eraseCar()

{

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

{

for(int j = 0; j < 4; j++)

{

gotoxy(j + carPos, i+22);

cout << " ";

} }}

int collision()

{

if( enemy\_y[0] + 4 >= 23)

{

if(enemy\_x[0] + 4 - carPos >= 0 && enemy\_x[0] + 4 - carPos < 9)

{

return 1;

} }

return 0;

}

void gameover()

{

system("cls");

cout << endl;

cout << "\t\t--------------------------------" << endl;

cout << "\t\t-------- OPpPs Game Over -------" << endl;

cout << "\t\t--------------------------------" << endl << endl;

cout << "\t\tPress any key to go back to menu : ";

getch();

}

void updateScore()

{

gotoxy(WIN\_WIDTH + 7, 5);

cout << "Score: " << score << endl;

}

void fast()

{

carPos = -1 + WIN\_WIDTH / 2;

score=0;

enemyF[0]=1;

enemyF[1]=0;

enemy\_y[0]=enemy\_y[1]=1;

system("cls");

drawBorder();

updateScore();

genEnemy(0);

genEnemy(1);

gotoxy(WIN\_WIDTH + 7, 2);

cout << "Car Game";

gotoxy(WIN\_WIDTH + 6, 4);

cout << "----------";

gotoxy(WIN\_WIDTH + 6, 6);

cout << "----------";

gotoxy(WIN\_WIDTH + 7, 12);

cout << "Control ";

gotoxy(WIN\_WIDTH + 7, 13);

cout << "-------- ";

gotoxy(WIN\_WIDTH + 2, 14);

cout << " A Key - Left";

gotoxy(WIN\_WIDTH + 2, 15);

cout << " J Key - Right";

gotoxy(18, 5);

cout << "Press any key to start: ";

getch();

gotoxy(18, 5);

cout << " ";

while(1)

{

if(kbhit())

{

char ch = getch();

if(ch=='a' || ch=='A')

{

if(carPos>18)

{

carPos -= 4;

}

}

if(ch=='j' || ch=='J')

{

if(carPos < 50)

{

carPos += 4;

}

}

if(ch==27)

{

break;

}}

drawCar();

drawEnemy(0);

drawEnemy(1);

if(collision() == 1)

{

gameover();

return;

}

Sleep(25);

eraseCar();

eraseEnemy(0);

eraseEnemy(1);

if(enemy\_y[0] == 10)

{

if(enemyF[1] == 0)

{

enemyF[1] = 1;

}

}

if(enemyF[0] == 1)

{

enemy\_y[0] += 1;

}

if(enemyF[1] == 1)

{

enemy\_y[1] += 1;

}

if(enemy\_y[0] > SCREEN\_HEIGHT-4)

{

resetEnemy(0);

score++;

updateScore();

}

if(enemy\_y[1] > SCREEN\_HEIGHT-4)

{

resetEnemy(1);

score++;

updateScore();

} }}

void play()

{

carPos = -1 + WIN\_WIDTH / 2;

score=0;

enemyF[0]=1;

enemyF[1]=0;

enemy\_y[0]=enemy\_y[1]=1;

system("cls");

drawBorder();

updateScore();

genEnemy(0);

genEnemy(1);

gotoxy(WIN\_WIDTH + 7, 2);

cout << "Car Game";

gotoxy(WIN\_WIDTH + 6, 4);

cout << "----------";

gotoxy(WIN\_WIDTH + 6, 6);

cout << "----------";

gotoxy(WIN\_WIDTH + 7, 12);

cout << "Control ";

gotoxy(WIN\_WIDTH + 7, 13);

cout << "-------- ";

gotoxy(WIN\_WIDTH + 2, 14);

cout << " A Key - Left";

gotoxy(WIN\_WIDTH + 2, 15);

cout << " J Key - Right";

gotoxy(18, 5);

cout << "Press any key to start: ";

getch();

gotoxy(18, 5);

cout << " ";

while(1)

{

if(kbhit())

{

char ch = getch();

if(ch=='a' || ch=='A')

{

if(carPos>18)

{

carPos -= 4;

} }

if(ch=='j' || ch=='J')

{

if(carPos < 50)

{

carPos += 4;

} }

if(ch==27)

{

break;

}}

drawCar();

drawEnemy(0);

drawEnemy(1);

if(collision() == 1)

{

gameover();

return;

}

Sleep(100);

eraseCar();

eraseEnemy(0);

eraseEnemy(1);

if(enemy\_y[0] == 10)

{

if(enemyF[1] == 0)

{

enemyF[1] = 1;

} }

if(enemyF[0] == 1)

{

enemy\_y[0] += 1;

}

if(enemyF[1] == 1)

{

enemy\_y[1] += 1;

}

if(enemy\_y[0] > SCREEN\_HEIGHT-4)

{

resetEnemy(0);

score++;

updateScore();

}

if(enemy\_y[1] > SCREEN\_HEIGHT-4)

{

resetEnemy(1);

score++;

updateScore();

} }}

void fastest()

{

carPos = -1 + WIN\_WIDTH / 2;

score=0;

enemyF[0]=1;

enemyF[1]=0;

enemy\_y[0]=enemy\_y[1]=1;

system("cls");

drawBorder();

updateScore();

genEnemy(0);

genEnemy(1);

gotoxy(WIN\_WIDTH + 7, 2);

cout << "Car Game";

gotoxy(WIN\_WIDTH + 6, 4);

cout << "----------";

gotoxy(WIN\_WIDTH + 6, 6);

cout << "----------";

gotoxy(WIN\_WIDTH + 7, 12);

cout << "Control ";

gotoxy(WIN\_WIDTH + 7, 13);

cout << "-------- ";

gotoxy(WIN\_WIDTH + 2, 14);

cout << " A Key - Left";

gotoxy(WIN\_WIDTH + 2, 15);

cout << " J Key - Right";

gotoxy(18, 5);

cout << "Press any key to start: ";

getch();

gotoxy(18, 5);

cout << " ";

while(1)

{

if(kbhit())

{

char ch = getch();

if(ch=='a' || ch=='A')

{

if(carPos>18)

{

carPos -= 4;

} }

if(ch=='j' || ch=='J')

{

if(carPos < 50)

{

carPos += 4;

} }

if(ch==27)

{

break;

}}

drawCar();

drawEnemy(0);

drawEnemy(1);

if(collision() == 1)

{

gameover();

return;

}

Sleep(5);

eraseCar();

eraseEnemy(0);

eraseEnemy(1);

if(enemy\_y[0] == 10)

{

if(enemyF[1] == 0)

{

enemyF[1] = 1;

} }

if(enemyF[0] == 1)

{

enemy\_y[0] += 1;

}

if(enemyF[1] == 1)

{

enemy\_y[1] += 1;

}

if(enemy\_y[0] > SCREEN\_HEIGHT-4)

{

resetEnemy(0);

score++;

updateScore();

}

if(enemy\_y[1] > SCREEN\_HEIGHT-4)

{

resetEnemy(1);

score++;

updateScore();

} }}

int main()

{

setcursor(0, 0);

srand((unsigned)time(NULL));

do

{

system("cls");

gotoxy(10, 5);

cout << " --------------------------------- ";

gotoxy(10, 6);

cout << " | Welcome to Airin's game | ";

gotoxy(10, 7);

cout << " ----------------------------------";

gotoxy(10, 9);

cout << "1. Level-1 (EASY)";

gotoxy(10, 10);

cout << "2. Level-2 (MEDIUM)";

gotoxy(10, 11);

cout << "3. Level-3 (HARD)";

gotoxy(10, 13);

cout << "Select code : ";

char op = getche();

if(op=='1') play();

else if( op=='2') fast();

else if( op=='3') fastest();

}

while(1);

return 0;

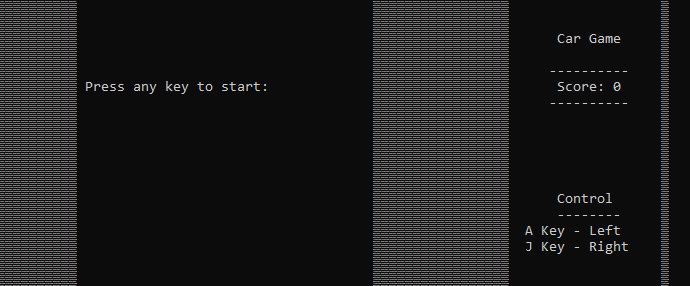
}

**Output:**

Starting the game:



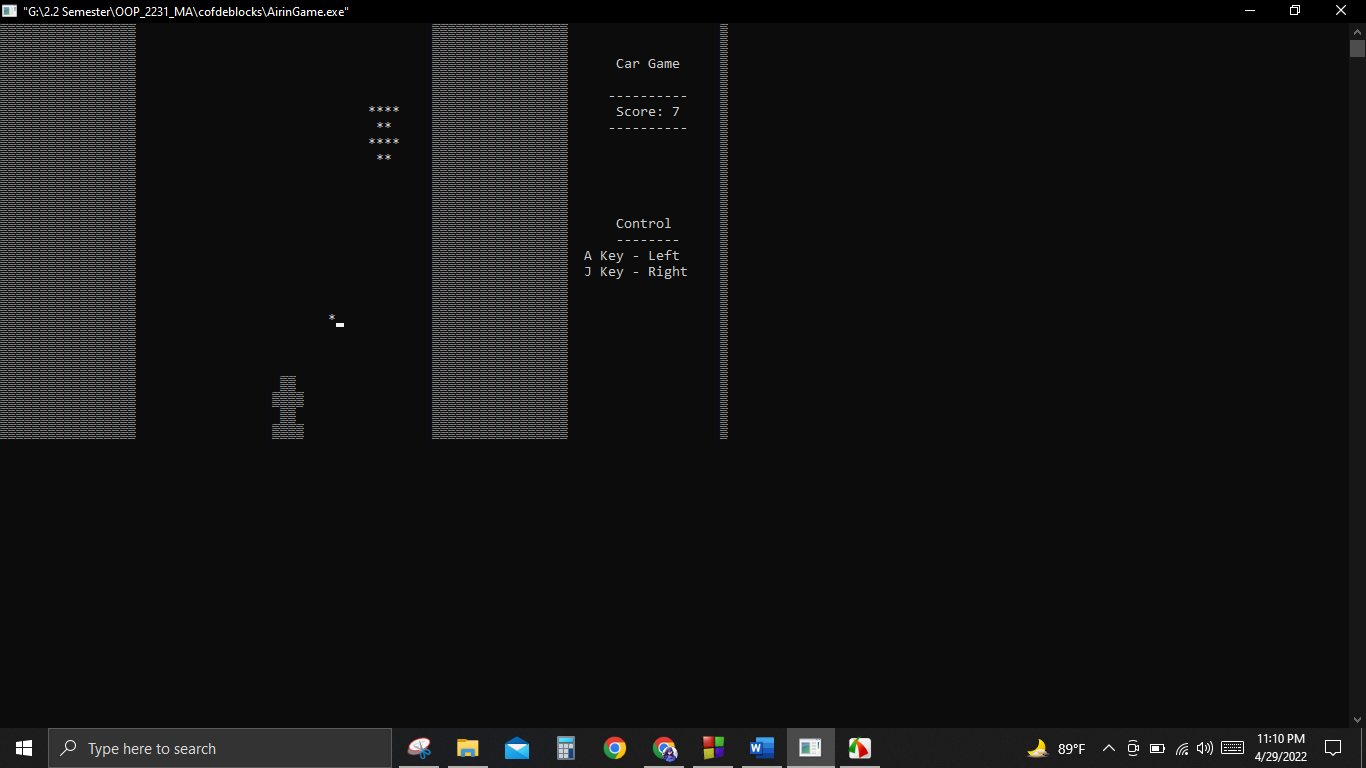
Choose any level-the level interface:



**Level-1 (EASY):**



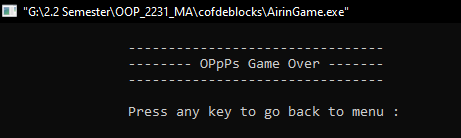
Level-2(MEDIUM):



Level-3(HARD):

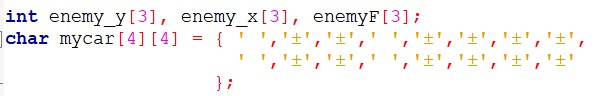


Game over interface:



**Discussion:**

I am trying to make a racing game using C or C++. Then I developed this car racing game using C++. It is easy to console. So in this game I used some header files which are needed, used some data type and some function, also used loop like for loop, while loop etc. and some algorithm.



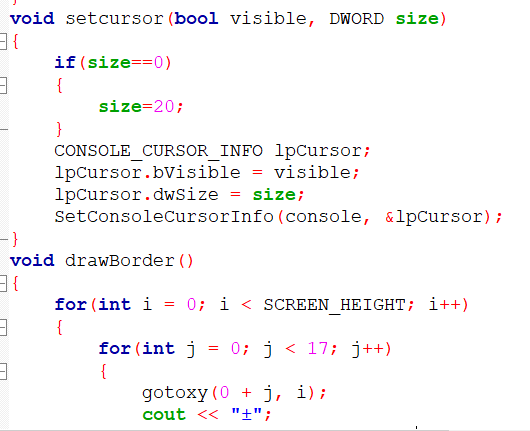
This is the car of the game. Here 2D or two-dimensional array is used to create the car shape.





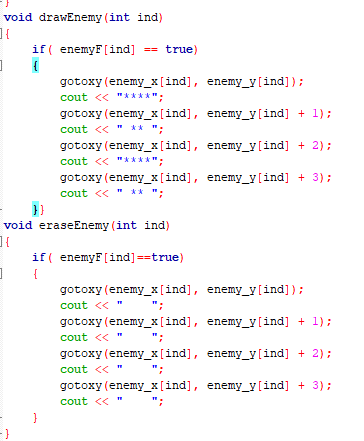


Here sleep(25) means the enemy will come faster than sleep (100) and slower than sleep(5).

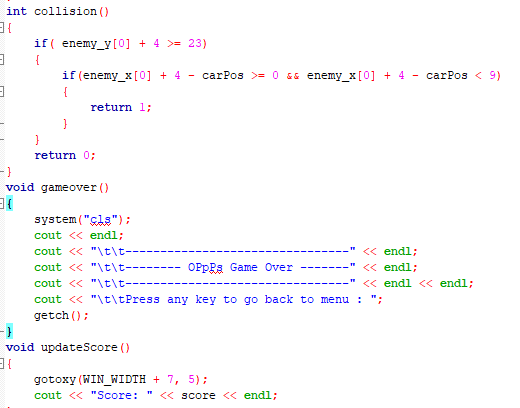


Here void function is used. This void function means pointer to something left unspecified. We used void setcursor() function that sets the cursor image to the specified cursor. This cursor image is displayed when the contains method for this component returns true for the current cursor location, and this Component is visible bool data type, displayable, and enabled.

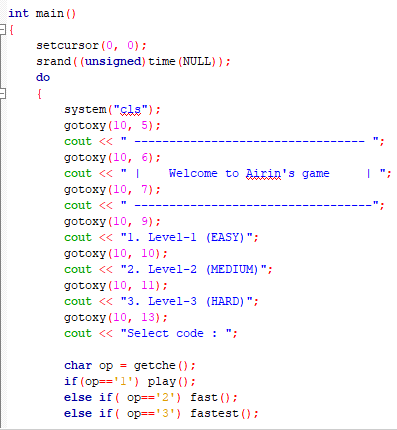
Here we used for loop that increases the screen height continuously that is 0 to 16.



This is the game enemy. Which is printed as ‘x’. gotoxy() function is used to place the cursor x at the desire location. void eraseEnemy() function is declared for delete the enemy and then comes the new enemy for the function void drawEnemy().



Here collision() function is used for the collision between car and enemy then the game will over for the gameover() function.



Then we did cout for the output of the game in the main function and welcome message is printed. Here we give the condition of level-1 for press 1, level 2 for press 2 and level 3 for press 3.

**Conclusion:**

We have learned from this experiment C++ allows to create games for Windows, Mac, Linux, Android, and iOS, among other platforms. C++ has nearly become the default for embedded code for this reason. It has had the greatest impact on the creation of game engines. Unity, Unreal, and Frost byte were all created using the C++ programming languages. This car game is simple and fun to play making it available to anyone. So, it was a great experience of my first game development with facing lot of errors and finally developed the Car Game.