2. INTRODUCTION

2.1 Problem Statement

The player will control the car from arrow keys , each player will be given three lives , after hitting an obstacle the player will lose one life and if player cross the obstacle he will earn some points.The player with maximum points wins.

2.2 Literature Survey

2.2.1. **Development of a Car Racing Simulator Game**

Author - Marvin T. Chan

It is a simple car racing game in which the human player races a car against three game-controlled cars in a three-dimensional environment. This game has been designed to include the components and techniques dealing with the software side as well as with VHDL. Whereas, we are going to use the graphics library.

2.2.2. **Pathfinding Car Racing Game in C**

Author - Yoppy Sazaki

In a race car game, the NPC requires pathfinding to be able to walk on the track and avoid obstacles to reach the finish line. Pathfinding method used by NPC (Non playable Character). In this paper also obstacles are present and person with highest score wins, but it uses a pathfinder algorithm to add obstacles.

2.2.3. **Car Racing Simulator Game Using Artificial Intelligence Techniques**

Author - Craig Gelowitz

The game incorporates artificial intelligence (AI) techniques, which enable the cars to be controlled in a manner that mimics natural driving
The game is designed to provide the player real driving experience whereas our project is based on obstacle tackling.

3. DESCRIPTION OF THE PROJECT

3.1 Requirement Analysis

3.1.1 Software Requirements

- Turbo C++
- DOSBOX
- C GRAPHICS library

3.1.2 Hardware Requirements

- Intel Core Pentium and Above
- 2GB RAM and above
- Intel HD graphics

3.2 Modular Description

- THE CAR
- OBSTACLES
- ANIMATIONS
- GAME OVER
- DISPLAY OF SCORES AND LIVES

i. DISPLAY OF SCORE AND LIVES

During the game,the scores and the ‘lives left’ are constantly displayed on the screen. As the game goes on,the scores keeps on getting incremented accordingly. If the game gets over,the final score is displayed. The player gains an extra life for every 10000 points earned.

ii. THE CAR

The car can be controlled using left and right arrow keys. The main aim is to prvent collison from obstacles, on crossing each obstacle the player will earn some points. As the player points will increase, the speed of the car will increase for more difficulty.

iii. OBSTACLES

Obstacles will come across the car path in the game and the player will have to prevent from hitting them. If a player hits the a life is lost , and if a player passes by them he earns points. As the score of player increases the frequency of obstacles will also increase .

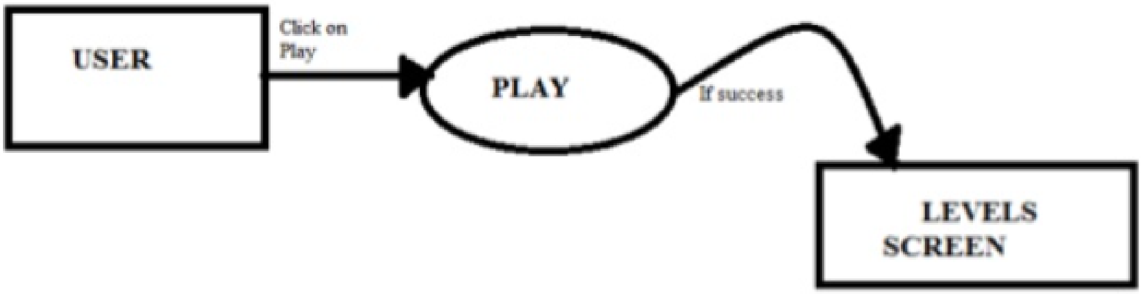
iv. ANIMATIONS

Whenever a level is changed, the car hits an obstacle or game ends proper animations will be displayed

v. GAME OVER

When the player's live gets over the game will get end and final score and highest score will displayed on the screen.

3. 4 Data flow Design and Diagram



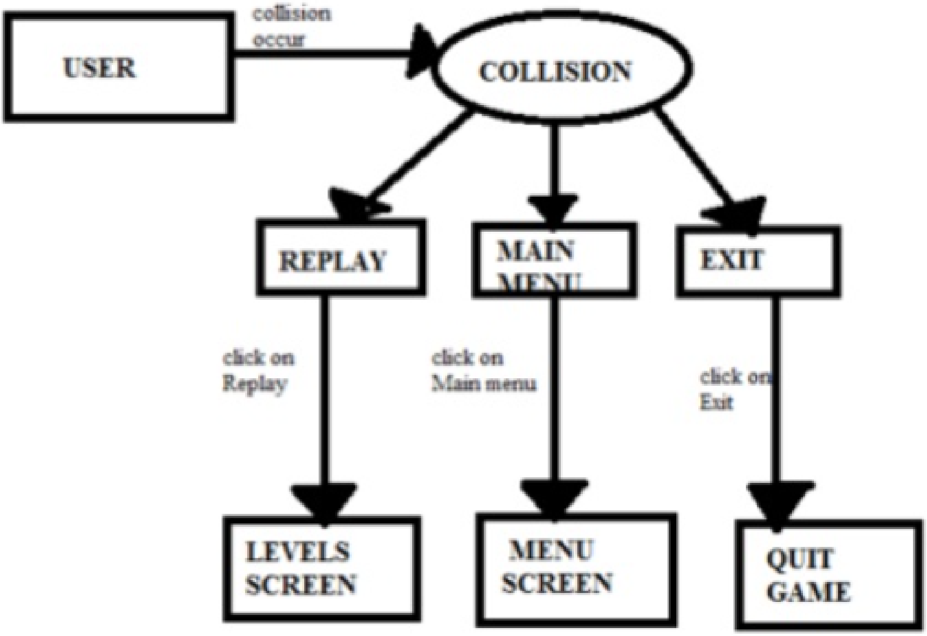


Fig 3.4.1 –DATA FLOW DIAGRAM

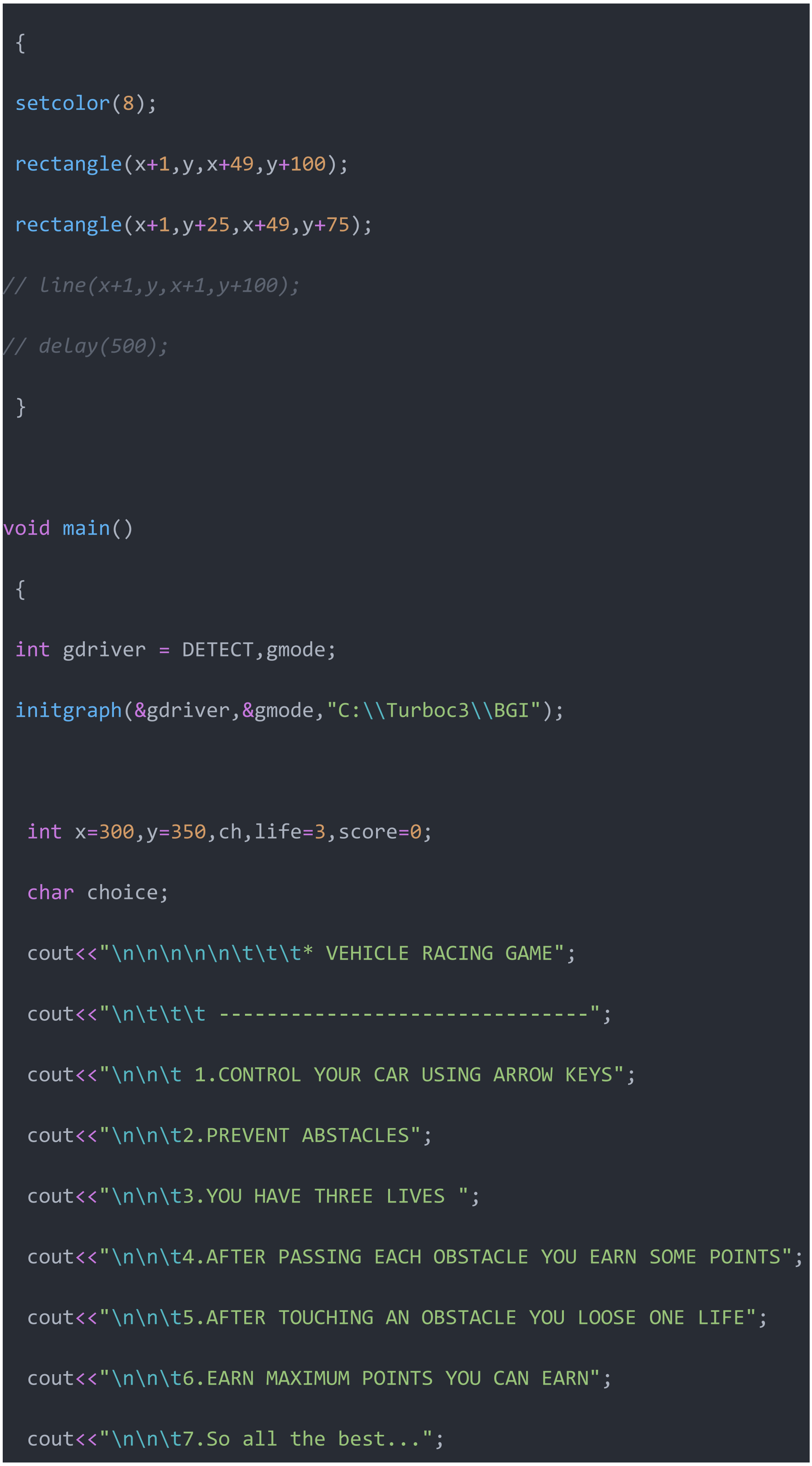
4. IMPLEMENTATION

The car racing game is developed in C++ programming language using graphics library in C++.The game is then tested using Turbo C++ which supports Graphics library. The game can then be converted into an application file(.exe) and can be run on any desired computer.

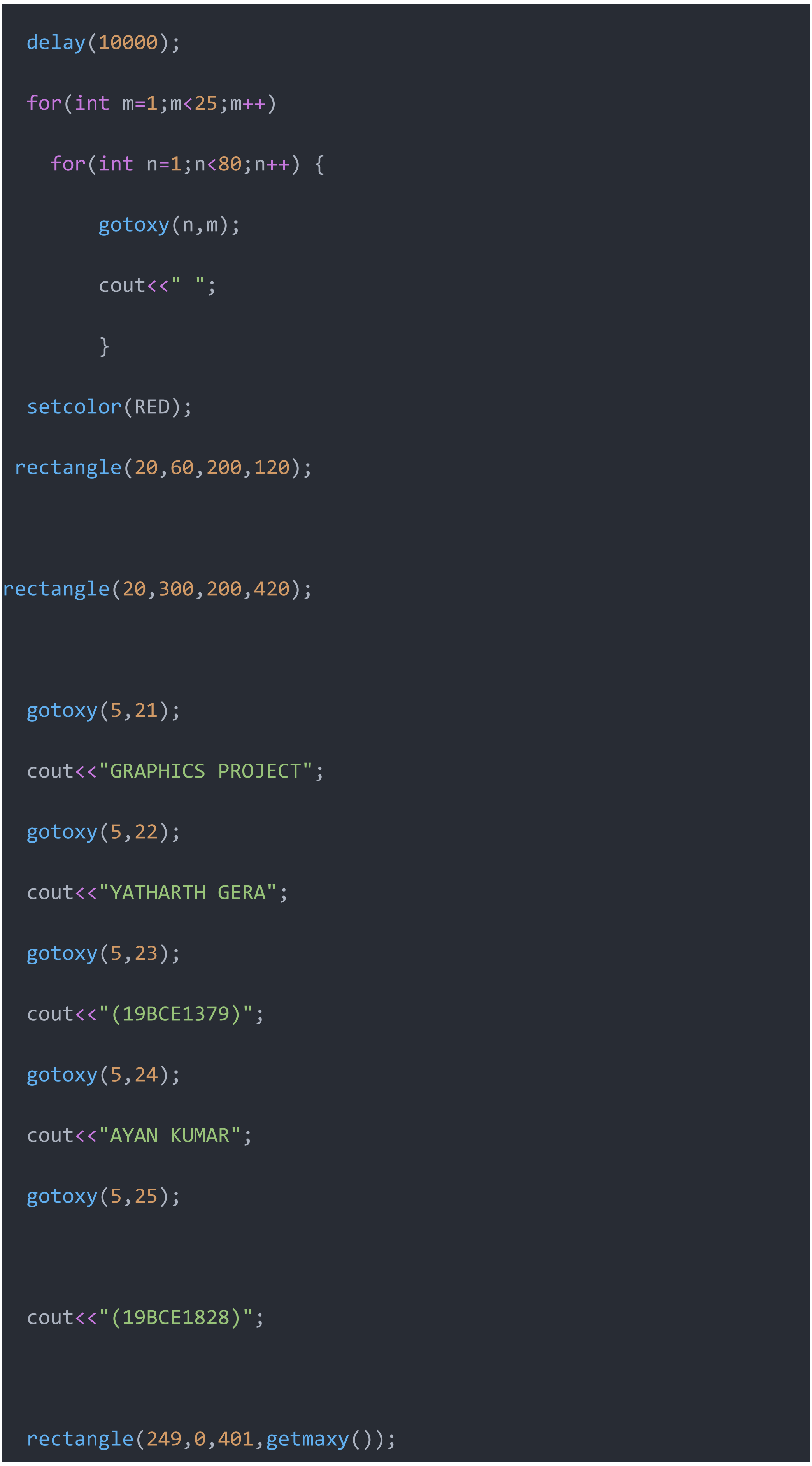
- The car can be turned using left and right arrow keys.
- To pause the game press escape button.
- To exit pause menu , press escape button again.
- To exit the game click on exit button on pause menu.

COMPLETE CODE:

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
#include<process.h>
#include<stdlib.h>

void enemycar(int x, int y)
 {
  setcolor(15);
  rectangle(x+1,y,x+49,y+100);
  rectangle(x+1,y+25,x+49,y+75);
  setfillstyle( SOLID_FILL,YELLOW);
  floodfill((x+x+50)/2,(y+y+100)/2,15);
  setfillstyle(1,YELLOW);
  floodfill((x+x+50)/2,(y+y+40)/2,15);
  floodfill((x+x+50)/2,(y+y+160)/2,15);

  }

void mycar(int x, int y)
 {
  setcolor(15);
  rectangle(x+1,y,x+49,y+100);
  rectangle(x+1,y+25,x+49,y+75);
  setfillstyle(5,RED);
  floodfill((x+x+50)/2,(y+y+100)/2,15);
  setfillstyle(2,BLUE);
  floodfill((x+x+50)/2,(y+y+40)/2,15);
  floodfill((x+x+50)/2,(y+y+160)/2,15);
  }



void myclear(int x,int y)
 {
 setcolor(8);
 rectangle(x+1,y,x+49,y+100);
 rectangle(x+1,y+25,x+49,y+75);
 setfillstyle(SOLID_FILL,8);
 floodfill((x+x+50)/2,(y+y+100)/2,8);
 floodfill((x+x+50)/2,(y+y+40)/2,8);
 floodfill((x+x+50)/2,(y+y+160)/2,8);
 }


 void enemyclear(int x,int y)
```



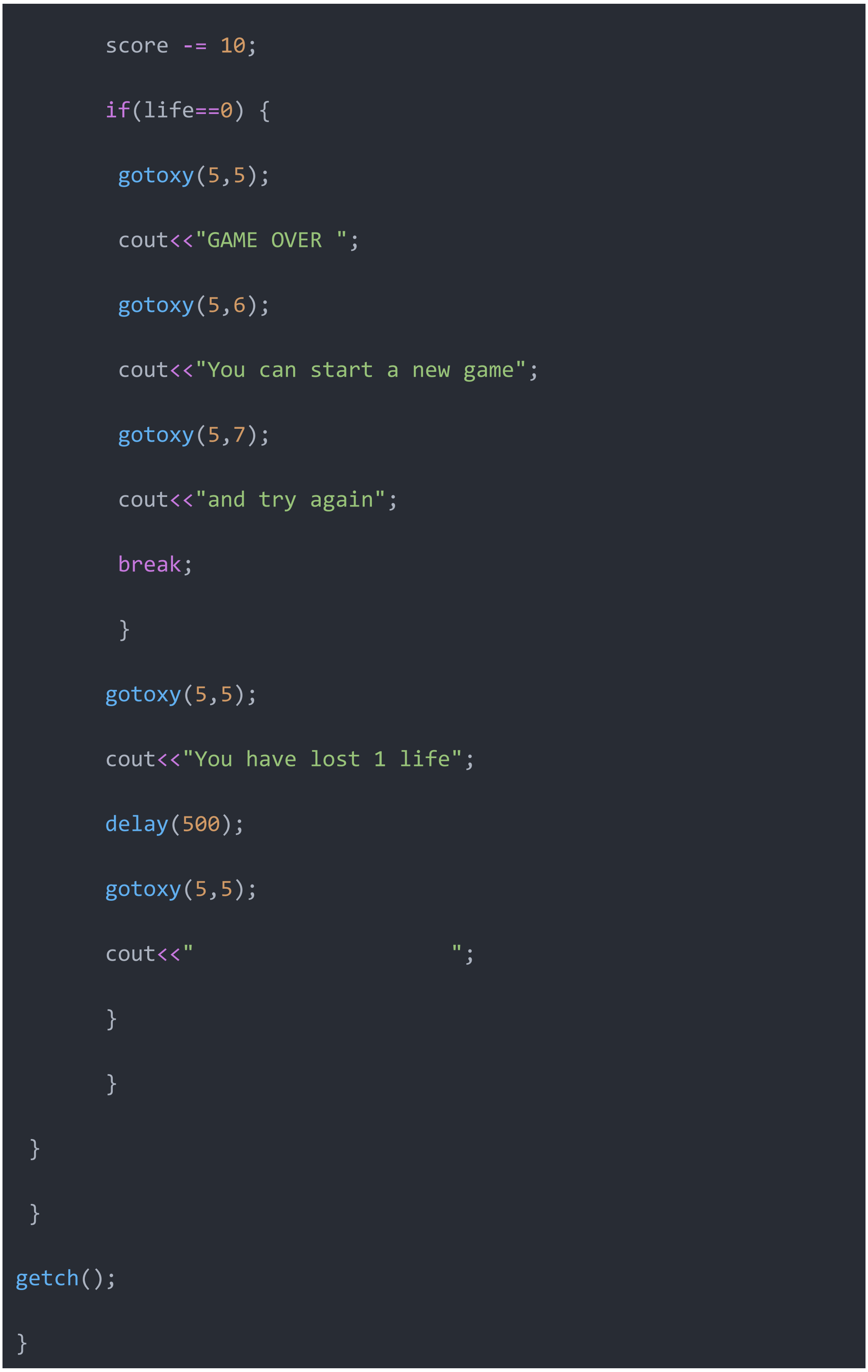
```
 {
 setcolor(8);
 rectangle(x+1,y,x+49,y+100);
 rectangle(x+1,y+25,x+49,y+75);
// line(x+1,y,x+1,y+100);
// delay(500);
 }

void main()
 {
 int gdriver = DETECT,gmode;
 initgraph(&gdriver,&gmode,"C:\\Turboc3\\BGI");

  int x=300,y=350,ch,life=3,score=0;
  char choice;
  cout<<"\n\n\n\n\n\t\t\t* VEHICLE RACING GAME";
  cout<<"\n\t\t\t -------------------------------";
  cout<<"\n\n\t 1.CONTROL YOUR CAR USING ARROW KEYS";
  cout<<"\n\n\t2.PREVENT ABSTACLES";
  cout<<"\n\n\t3.YOU HAVE THREE LIVES ";
  cout<<"\n\n\t4.AFTER PASSING EACH OBSTACLE YOU EARN SOME POINTS";
  cout<<"\n\n\t5.AFTER TOUCHING AN OBSTACLE YOU LOOSE ONE LIFE";
  cout<<"\n\n\t6.EARN MAXIMUM POINTS YOU CAN EARN";
  cout<<"\n\n\t7.So all the best...";
```



```
  delay(10000);
  for(int m=1;m<25;m++)
    for(int n=1;n<80;n++) {
        gotoxy(n,m);
        cout<<" ";
        }
  setcolor(RED);
 rectangle(20,60,200,120);

rectangle(20,300,200,420);

  gotoxy(5,21);
  cout<<"GRAPHICS PROJECT";
  gotoxy(5,22);
  cout<<"YATHARTH GERA";
  gotoxy(5,23);
  cout<<"(19BCE1379)";
  gotoxy(5,24);
  cout<<"AYAN KUMAR";
  gotoxy(5,25);

  cout<<"(19BCE1828)";

  rectangle(249,0,401,getmaxy());
```

```
setfillstyle(SOLID_FILL,8);
floodfill(325,getmaxy()/2,RED);
setcolor(RED);
rectangle(20,200,200,250);
gotoxy(5,15);
cout<<"Press <Esc> to Exit";
for(int level=1;(level<=5)&&(life>0);level++){
  if(level==1){
    gotoxy(5,5);
    cout<<"Your War Starts Now";
    gotoxy(5,7);
    cout<<"All the best";
    delay(500);
    gotoxy(5,5);
    cout<<"                    ";
    gotoxy(5,7);
    cout<<"            ";
    }
   else {
       gotoxy(5,5);
       cout<<"Next level.";
       delay(500);
       gotoxy(5,5);
       cout<<"           ";
       }

for(int i=0;(i<15)&&(life>0);i++) {

  if((level==5)&&(i==14)){

     gotoxy(5,5);

     cout<<"You have won";

     gotoxy(5,6);

     cout<<"Wanna continue <y/n>";

     choice = getch();

     if ((choice=='y')||(choice=='Y'))

      main();

     else

      exit(0);

     }

  setcolor(RED);

  rectangle(420,125,600,175);

  gotoxy(55,10);

  cout<<"Level = "<<level;

  rectangle(420,250,600,300);

  gotoxy(55,18);

  cout<<"Lives = "<<life;

  rectangle(420,350,600,400);

  gotoxy(55,24);

  cout<<"Score = "<<score;

  int accident=0;
  int y1=1,x1=250+((rand()%3)*50);

  int y2=1,x2=250+((rand()%3)*50);

  score += 10;

  while(y1<getmaxy()-1)

  {



      enemyclear(x1,y1);

      enemyclear(x2,y2);

      y1++;

      y2++;

      if(accident == 1)

          break;

      enemycar(x1,y1);

      enemycar(x2,y2);

      mycar(x,y);

      delay(5-level);

      if(kbhit())

      {

          mycar(x,y);

          ch=getch();

          switch(ch) {

          case 27: exit(0);

          break;

          case 75:myclear(x,y);
```

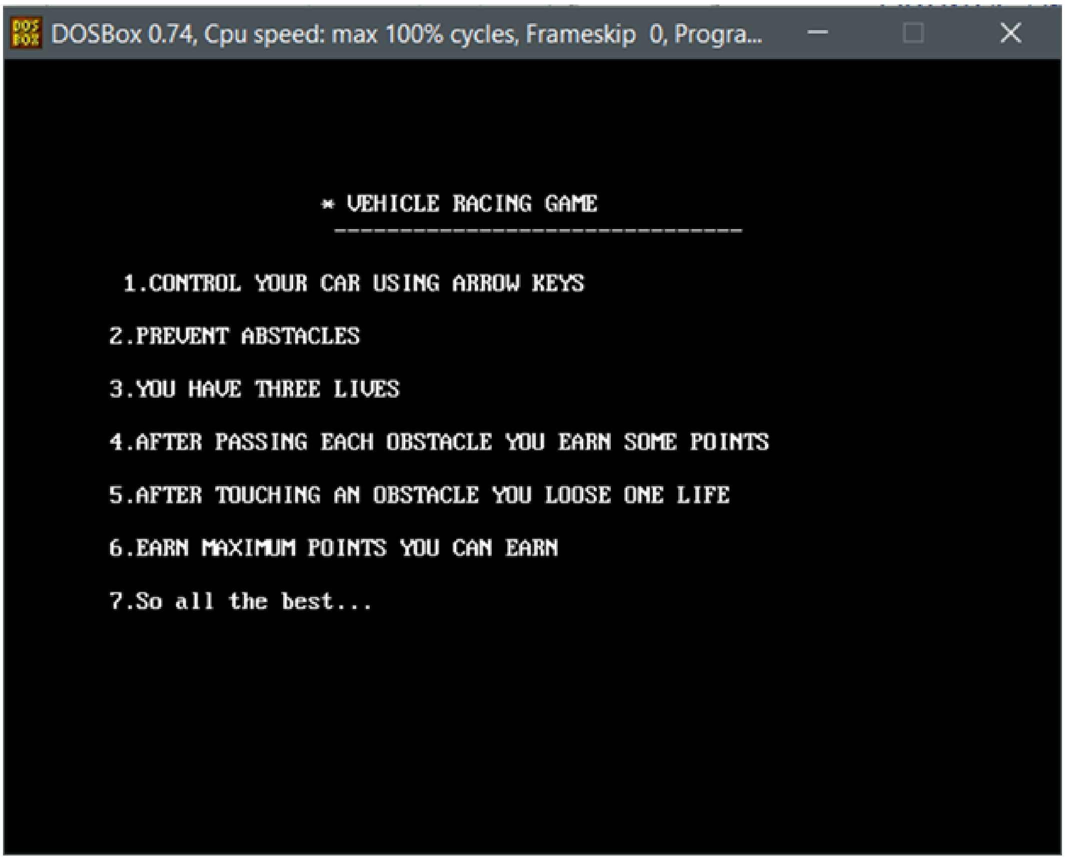
```
                if(x>250)
                    x=x-50;
                    if((x==x1)||(x==x2))
                        if( ( ((y-y1)<100)&&((y-y1)>0) ) || ( ((y-y2)<100)&&((y-y2)>0) ) )
                        {
                            accident=1;
                            x=x+50;
                            mycar(x,y);
                            goto Next1;
                        }
                 mycar(x,y);
            break;
            case 77:myclear(x,y);
                if(x<350)
                    x=x+50;
                    if((x==x1)||(x==x2))
                        if( ( ((y-y1)<100)&&((y-y1)>0) ) || ( ((y-y2)<100)&&((y-y2)>0) ) )
                        {
                            accident=1;
                            x=x-50;
                            mycar(x,y);
                            goto Next1;
                        }
                mycar(x,y);
            break;
            case 72:myclear(x,y);
                if(y>0)
                      y=y-5;
                mycar(x,y);
             break;
             case 80:myclear(x,y);
                if(y<getmaxy()-105)
                    y=y+5;
                mycar(x,y);
             break;
            }
        }
        if((x==x1)||(x==x2))
            if( ( ((y-y1)<100)&&((y-y1)>0) ) || (
((y-y2)<100)&&((y-y2)>0) ) )
                accident=1;



      Next1:
      if(accident==1){
        life =life-1;
```



```
       score -= 10;
       if(life==0) {
        gotoxy(5,5);
        cout<<"GAME OVER ";
        gotoxy(5,6);
        cout<<"You can start a new game";
        gotoxy(5,7);
        cout<<"and try again";
        break;
        }
       gotoxy(5,5);
       cout<<"You have lost 1 life";
       delay(500);
       gotoxy(5,5);
       cout<<"                    ";
       }
       }
 }
 }
getch();
}
```

4.3 SCREENSHOTS AND EXPLANATION

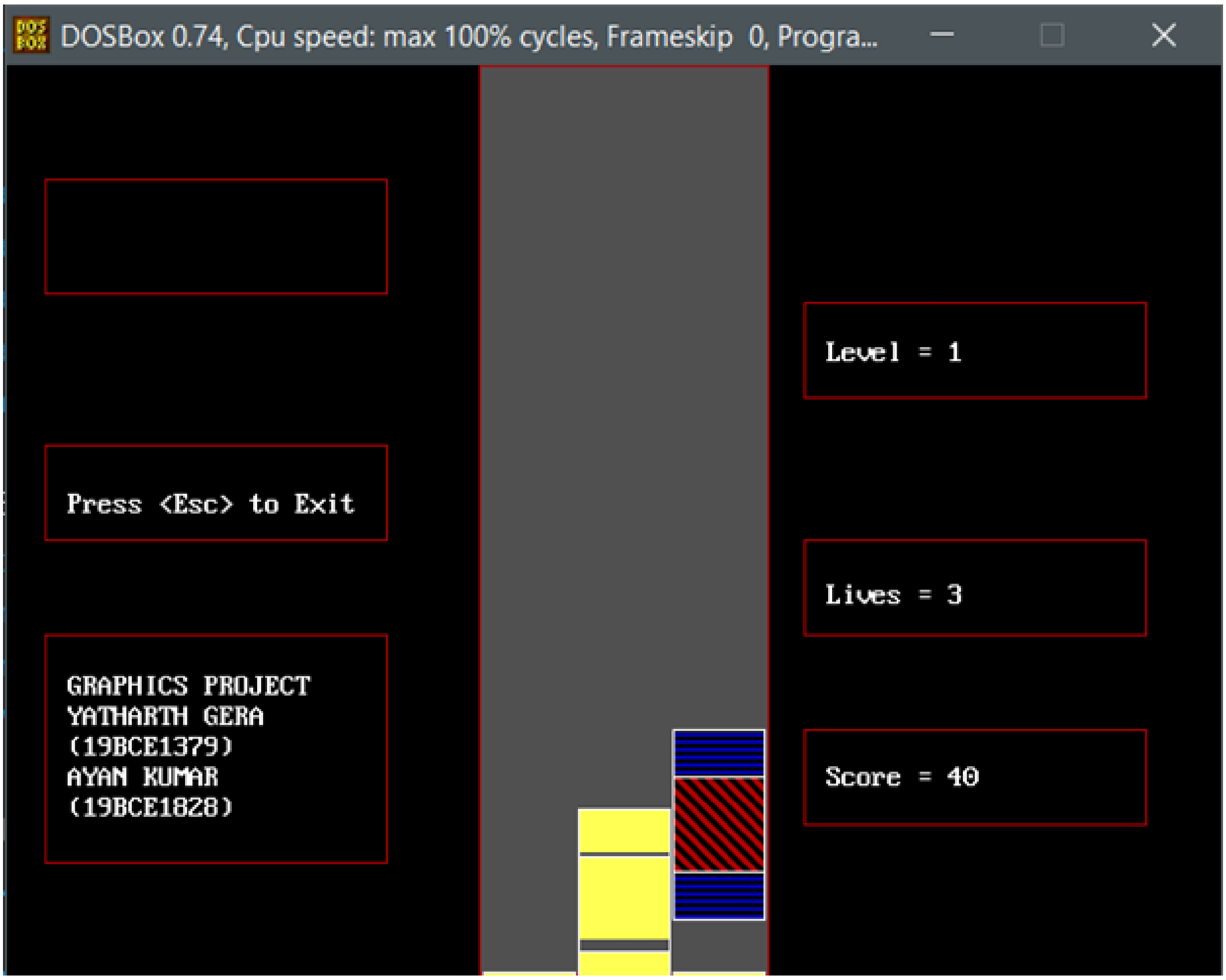
4.3.1 HOME SCREEN WITH INSTRUCTIONS



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
                * VEHICLE RACING GAME
                 -------------------------------

  1.CONTROL YOUR CAR USING ARROW KEYS

 2.PREVENT ABSTACLES

 3.YOU HAVE THREE LIVES

 4.AFTER PASSING EACH OBSTACLE YOU EARN SOME POINTS

 5.AFTER TOUCHING AN OBSTACLE YOU LOOSE ONE LIFE

 6.EARN MAXIMUM POINTS YOU CAN EARN

 7.So all the best...
```

4.3.2 GAME STARTED WITH LEVEL :1



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

Press <Esc> to Exit

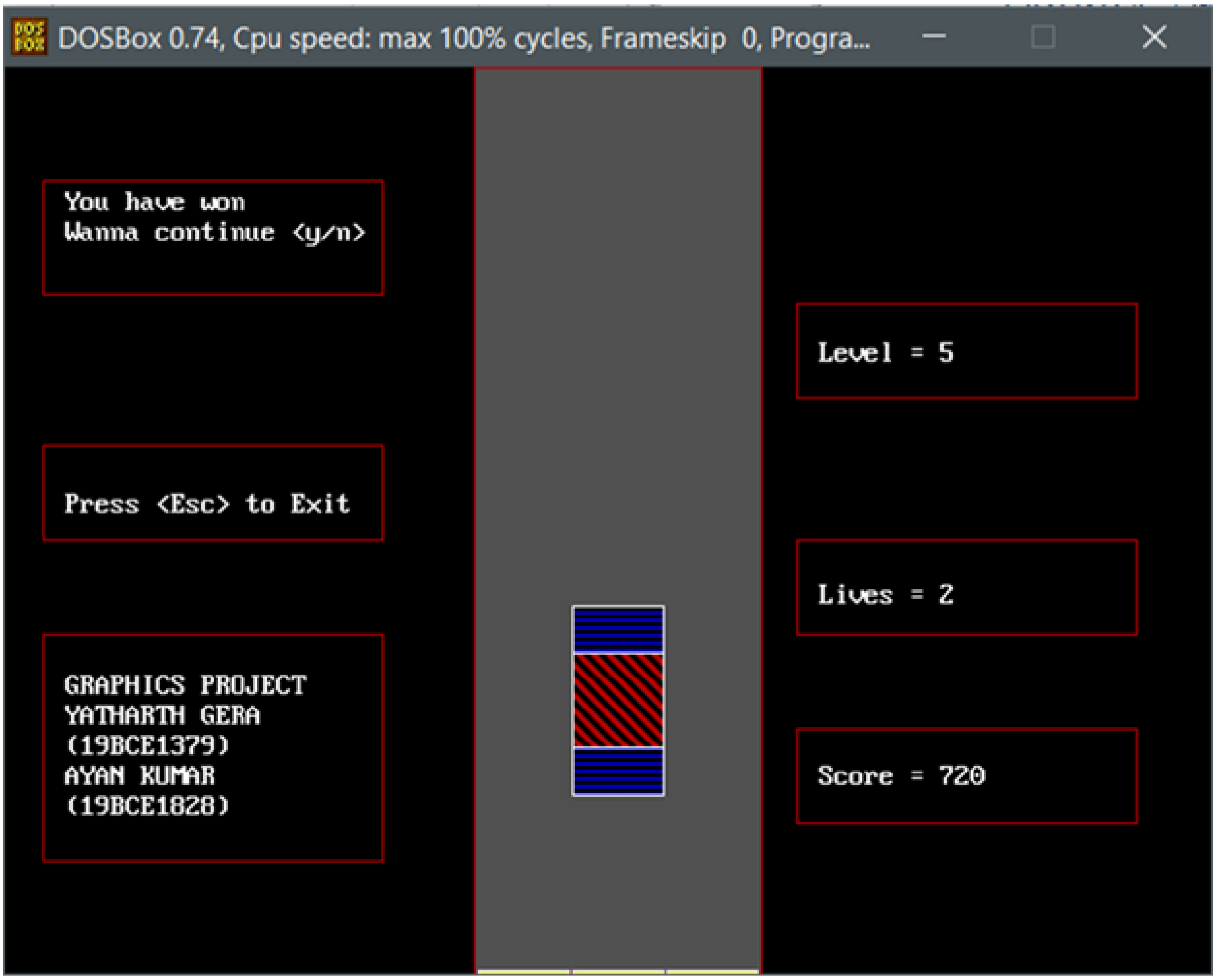
GRAPHICS PROJECT
YATHARTH GERA
(19BCE1379)
AYAN KUMAR
(19BCE1828)

Level = 1

Lives = 3

Score = 40

4.3.3 MESSAGE DISPLAYED AFTER WINNING THE GAME



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
You have won
Wanna continue <y/n>

Press <Esc> to Exit

GRAPHICS PROJECT
YATHARTH GERA
(19BCE1379)
AYAN KUMAR
(19BCE1828)

Level = 5

Lives = 2

Score = 720
```

6. CONCLUSIONS

1.The player controls the car and prevents it from hitting obstacles.

2..After all the lives of a player are exhausted the game ends and the score is displayed.

3.We will implement the game and all the features using c and graphics library.

7. REFERENCES

1. https://www.researchgate.net/publication/323063200_Pathfinding_car_racing_game_using_dynamic_pathfinding_algorithm_and_algorithm_A

2. https://core.ac.uk/download/pdf/188806317.pdf

3. https://www.hindawi.com/journals/ijcgt/2015/839721/