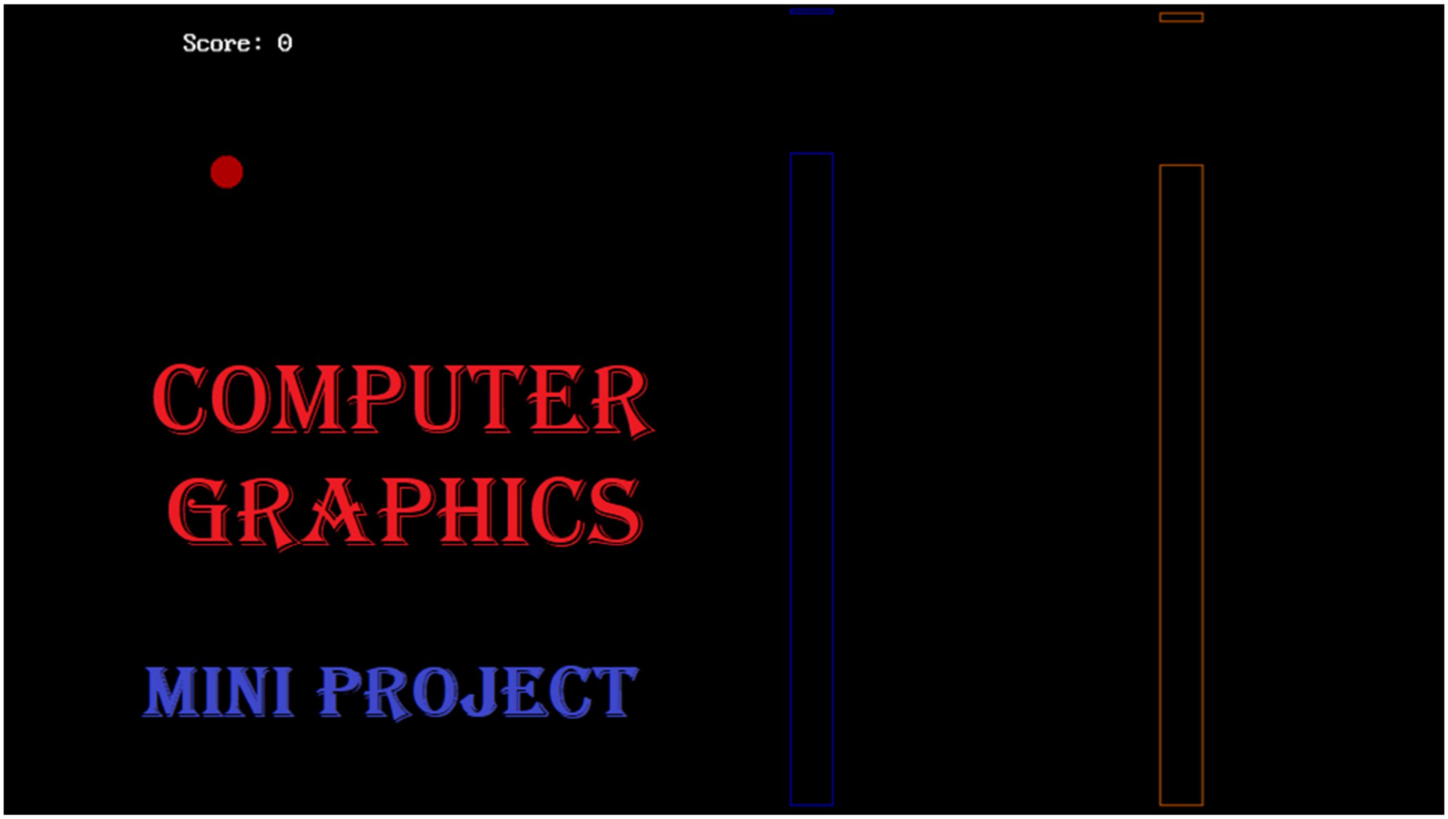


COMPUTER GRAPHICS

MINI PROJECT

TOPIC:-BOUNCING GAME

THE PROJECT IS BASED ON THE CREATION OF A GAME OR AN ANIMATION CLIP USING COMPUTER GRAPHICS IN C++.IN THIS PROJECT I HAVE CREATED A BOUNCIG GAME AND ADDED FEATURES TO IT. THERE IS AN INTRODUCTION GIVEN TO COMPUTER GRAPHICS ALONG WITH THE PROJECT CODE AND THE OUTPUT OF THE PROJECT.

KAUSTUBH SHRIKANT KABRA

SE COMPUTER SHIFT I

20

PROBLEM STATEMENT:

DESIGN AND IMPLEMENT A GAME/ ANIMATION CLIP/ GRAPHICS EDITOR USING OPEN SOURCE GRAPHICS LIBRARY. MAKE USE OF MAXIMUM FEATURES OF OBJECT ORIENTED PROGRAMMING.

INTRODUCTION

1.1 COMPUTER GRAPHICS

1.1.1 WHAT IS COMPUTER GRAPHICS?

Computer graphics is a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Although the term often refers to the study of three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing.

1.1.2 SUB-FIELDS IN COMPUTER GRAPHICS.

A broad classification of major subfields in computer graphics might be:

1. *Geometry: ways to represent and process surfaces*
2. *Animation: ways to represent and manipulate motion*
3. *Rendering: algorithms to reproduce light transport*
4. *Imaging: image acquisition or image editing*

1.1.3 APPLICATION OF COMPUTER GRAPHICS.

Applications of computer graphics include:

- *Print design*
- *Digital art*
- *Special effects*
- *Video games*
- *Visual effects*

ABOUT THE PROJECT

2.1 DESCRIPTION

The project is based on the concepts of computer graphics used in C++.
The main objective of the project is to demonstrate a moving object, in this case, a ball using the concepts of computer graphics and object oriented programming.

2.2 CONCEPTS USED

2.2.1 HEADER FILES

The different types of header files used are:

- <graphics.h>: It is used to include all the graphics methods and function. Example: line(), rectangle(), etc.
- <conio.h>: It is used for console input and output.
- <dos.h>: It is used to include the delay() function. The delay function helps to stop the execution of the program for the given time.
- <stdlib.h>: It is used for dynamic memory allocation, interger arithmetics, conversion,etc.
- <stdio.h>: It is also used for standard input and output operations.

CODE

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main()
{
  Play_Again:
  int gdriver = DETECT, gmode, errorcode;
  int midx,midy,maxx,maxy,ball_x,ball_y,key,speed,bar_upper,bar_lower,gap,score,fail,i;
  int bar_x[3],bar_y[3],color[3],bar_i[3],pattern[3];
  float ball_vlct;
  initgraph(&gdriver,&gmode,"C:\\TurboC3\\BGI");
  fail=0;
  speed=0;
  score=0;
  key=0;
  gap=85;
  ball_vlct=0.80;
  midx=getmaxx()/2;
  midy=getmaxy()/2;
  maxx=getmaxx();
  maxy=getmaxy();
  ball_x=25;
  ball_y=midy;
  bar_upper=5;
  bar_lower=maxy-5;
  bar_x[0]=maxx-10;
  bar_x[1]=bar_x[0]+219;      //draws bar at a distance of 219px
  bar_x[2]=bar_x[1]+219;
  for(i=0;i<=2;i++)           //bars having random height and color
  {
   bar_y[i]=random(maxy-150);
   color[i]=random(15)+1;
   bar_i[i]=0;
   pattern[i]=random(13);
  }

  do
  {
   if(kbhit())               //check for spacebar hit
   {
    if((key=getch())==' '||(key=getch())=='  ')
    {
        ball_vlct=0.6;
        speed=speed+ball_vlct-50;   //ball goes upper by 55px
  }
}

if(fail==1)               //changes color of all bars to red if fail
{
 for(i=0;i<=2;i++)
      color[i]=4;
}

setcolor(BLACK);
setfillstyle(1,20);
fillellipse(ball_x,ball_y+speed,10,10); //draws ball
for(i=0;i<=2;i++)                    //draws bar with random color
{
  setcolor(color[i]);
  //setfillstyle(pattern[i],color[i]);
  rectangle(bar_x[i]-bar_i[i],bar_y[i]+gap,bar_x[i]+25-bar_i[i],bar_lower);
  rectangle(bar_x[i]-bar_i[i],bar_y[i],bar_x[i]+25-bar_i[i],bar_upper);
}
if(fail==1)
{
  delay(1000);         //exit if fail
  break;
}

gotoxy(1,2);
printf("Score: %d",score);
for(i=0;i<=2;i++)          //check for fail or not
{
  if(bar_x[i]-bar_i[i]<=25)
  {
       if(!(ball_y+speed>=bar_y[i]&&ball_y+speed<=bar_y[i]+85))
         fail=1;
       else
         fail=0;
  }
}
//checks for ball doesn't go out of the screen
if(ball_y+speed>maxy-20)
{
  ball_vlct=0.0;
  speed=0;
  ball_y=maxy-27;
}
if(ball_y+speed<20)
{
  speed=0;
  ball_y=25;
}
//increments the values
ball_vlct=ball_vlct+0.81;
speed=speed+ball_vlct;
```

```
    for(i=0;i<=2;i++)
      bar_i[i]=bar_i[i]+5;

    for(i=0;i<=2;i++)
    {
      if(bar_x[i]-bar_i[i]+25<=0) //restart the bar values again
      {
          bar_i[i]=0;
          bar_x[i]=maxx;
          bar_y[i]=random(maxy-150);
          color[i]=random(15)+1;
          pattern[i]=random(13);
          score++;
      }
    }
    delay(50);
    cleardevice();
  }while(1);
  closegraph();
  clrscr();

  char name[12];
  gotoxy(25,10);
  textcolor(3);
  cprintf("Enter your name: ");
  gets(name);
  gotoxy(25,11);
  textcolor(RED);
  cputs(name);
  cprintf(" your score is %d",score);
  gotoxy(26,13);
  textcolor(RED+BLINK);
  cprintf("Play Again(y/n)");
  char play=getch();
  if(play=='y')
  {
    goto Play_Again;    //restart the game
  }
  return 0;
}
```

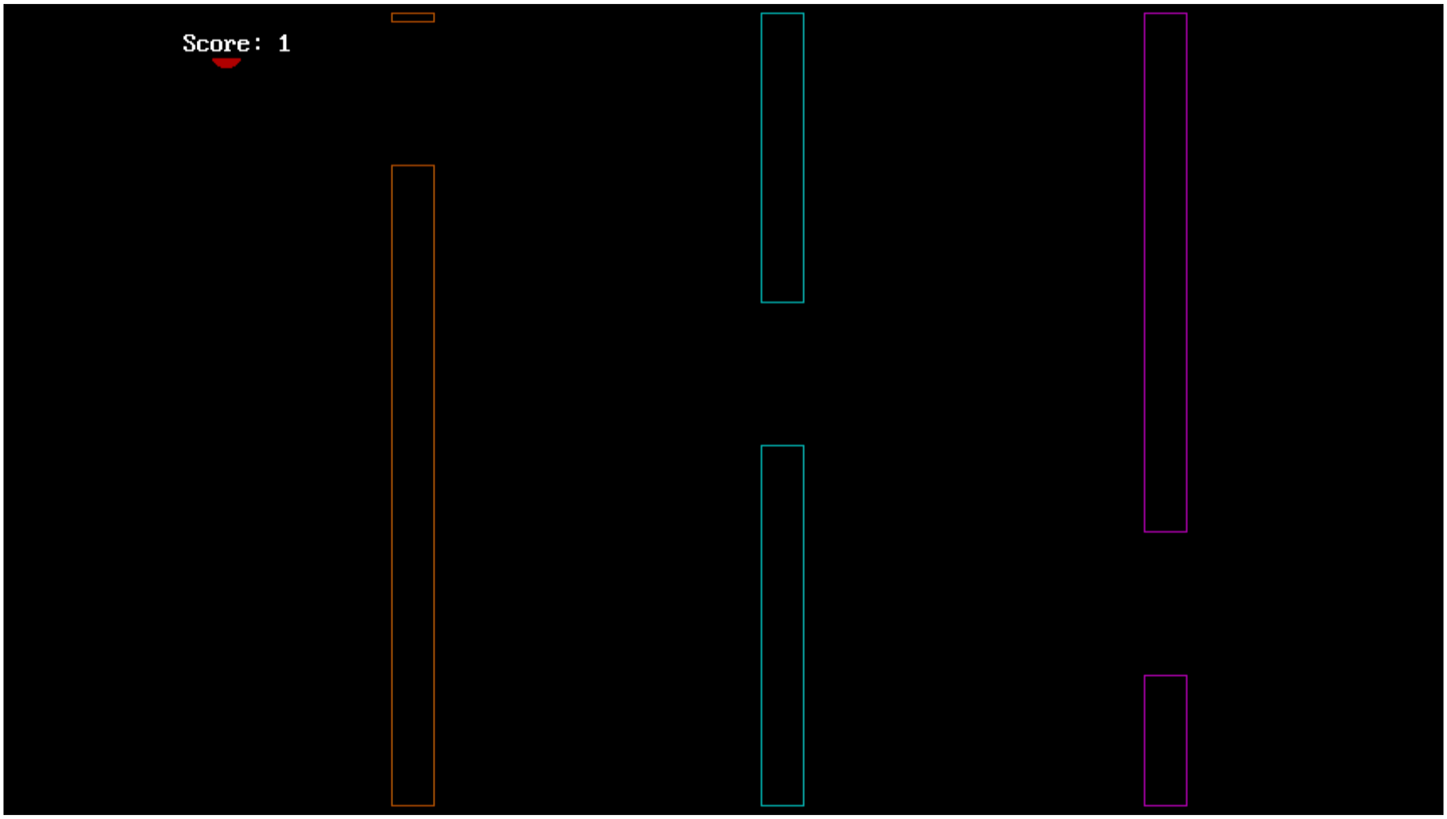
OUTPUT

3.1 DESCRIPTION

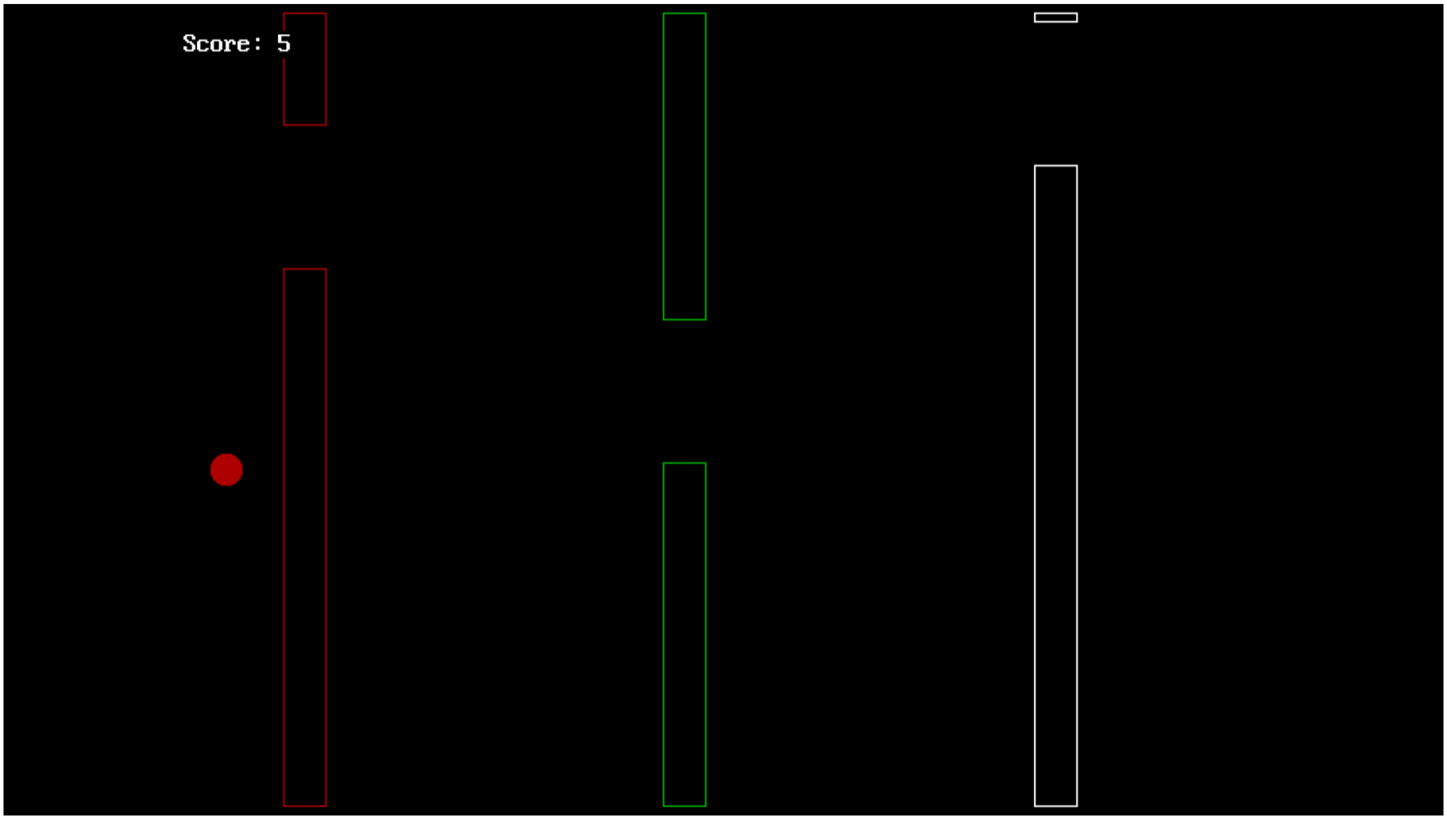
The output consists of a ball moving on a on screen with a obstracles between it. The various obtracles are coloured. The obstracles include wide rectangular strips and in ahesd game some more obstracles come in to play. The image of the output are shown below with the different different position in game.

3.2 CODE OUTPUT

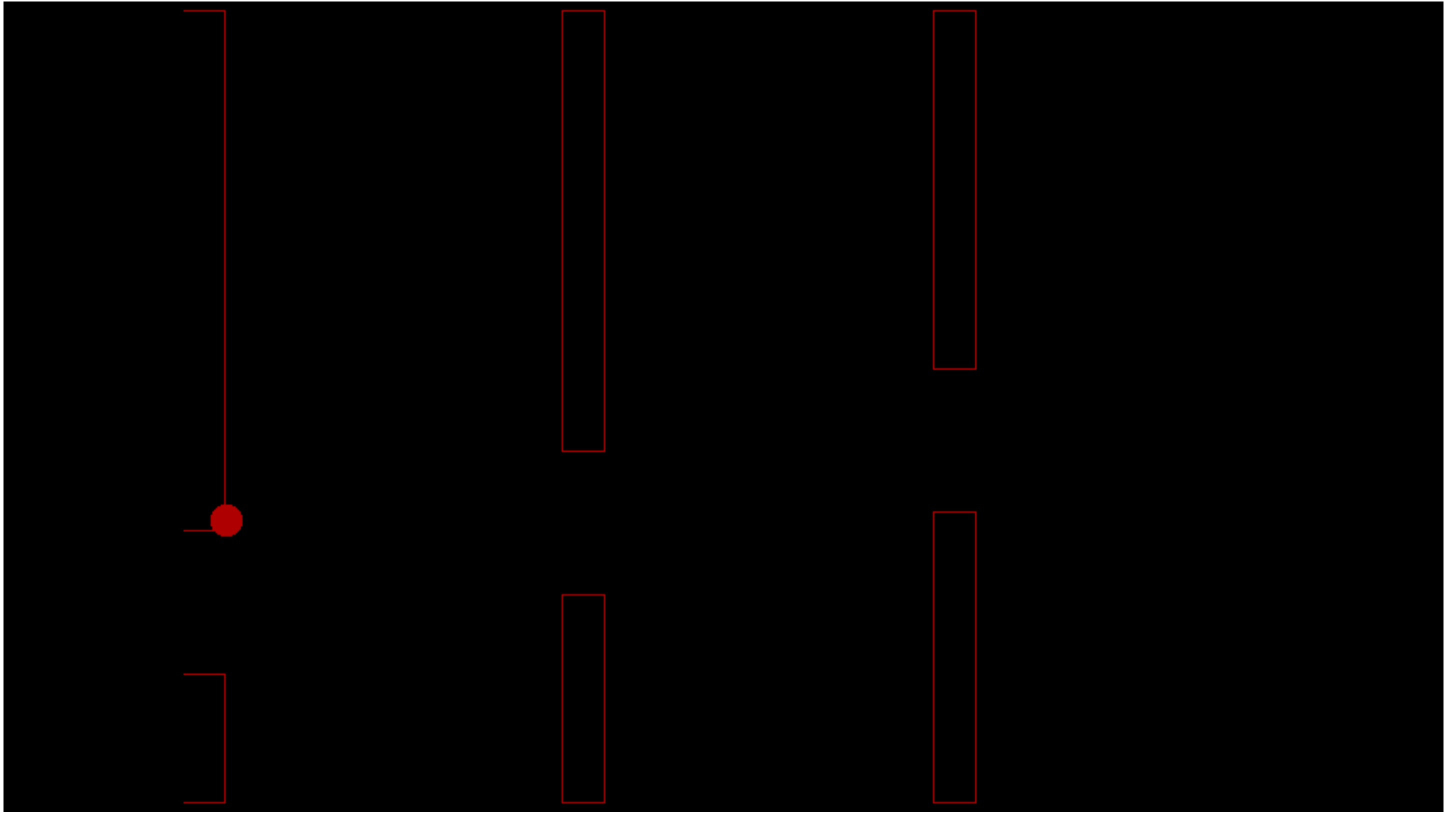
The output of the above code is attached below.

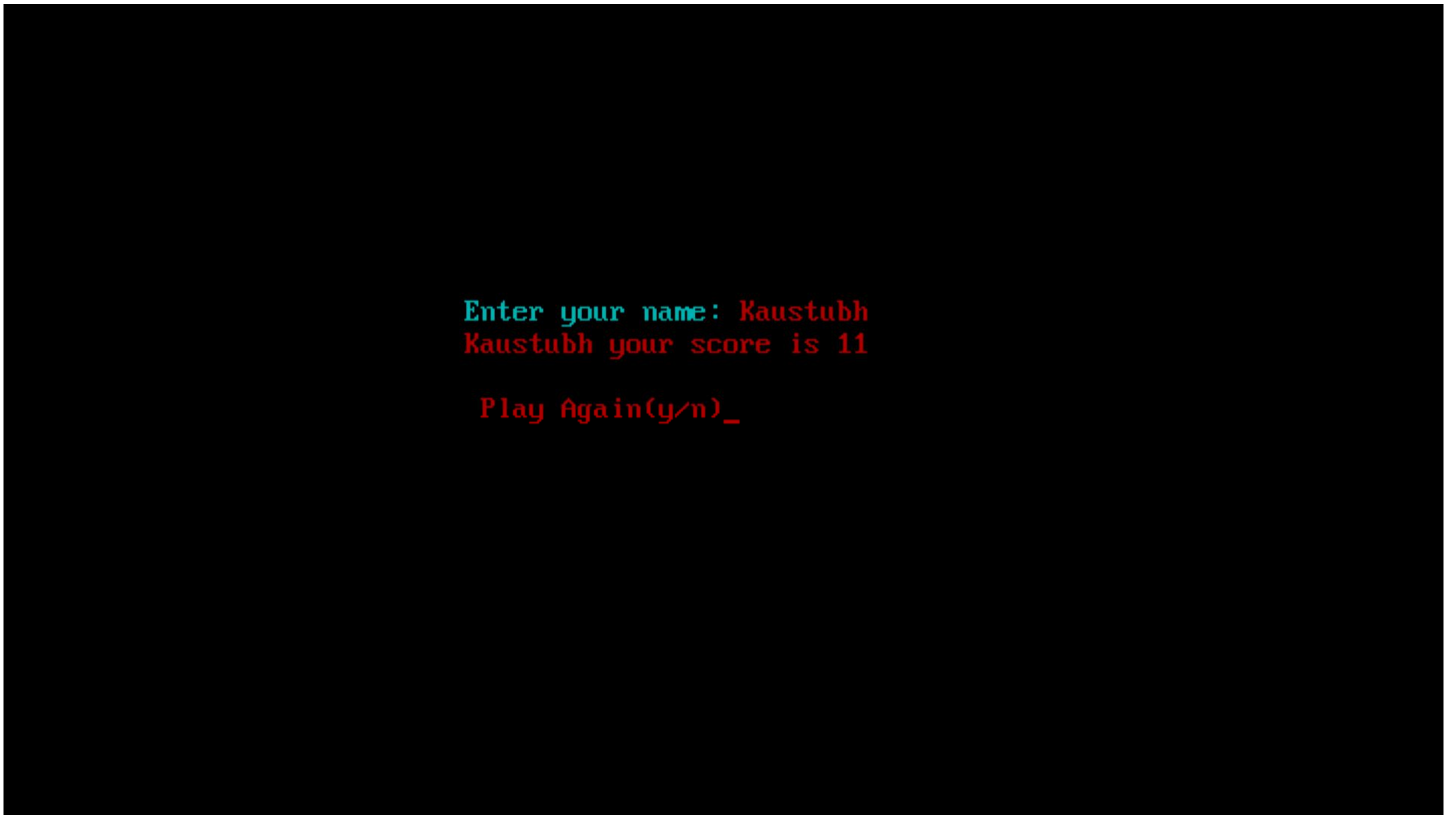


Score: 1



Score: 5





```
Enter your name: Kaustubh
Kaustubh your score is 11

 Play Again(y/n)_
```

REFERENCES

https://www.angryanimator.com/

https://www.youtube.com/watch?v=xADFPM_yviU&feature=youtu.be

CONCLUSION

Thus we have implemented a bouncing ball using the concepts of computer graphics and object oriented programming in C++. We have also added colour to the obstracles along with the other features.