VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAUM- 590014

A CG Mini-Project Report
On

"CATCH THE BALL"

A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belgaum.*

Submitted by:

ABHISHEK BARNWAL (1DT19CS004)
ABHISHEK MISHRA (1DT19CS005)

Under the Guidance of:

Mrs. Apoorva Busad (Asst. Prof. Dept of CSE)
Mrs. Nivetha Kumaravel (Asst. Prof. Dept of CSE)

Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT
(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI AND APPROVED BY AICTE, NEW DELHI)
Udayapura, Kanakpura Road, Bangalore-560082
2021-22

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI AND APPROVED BY AICTE, NEW DELHI)
Udayapura, Kanakpura Road, Bangalore-560082

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project on Computer Graphics and Visualization (CG) entitled ***"CATCH THE BALL"*** has been successfully carried out by **ABHISHEK BARNWAL (1DT19CS004) and ABHISHEK MISHRA (1DT19CS005)** a bonafide students of **Dayananda Sagar Academy of Technology and Management** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

GUIDE:

Mrs. Apoorva Busad
(Asst. Prof. Dept. of CSE)

Dr. C. NANDINI
(Vice Principal & HOD,
Dept. of CSE)

Examiners:

1:

2:

Signature with Date

ACKNOWLEDGEMENT

It gives us immense pleasure to present before you our project titled “**CATCH THE BALL**”. The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express a sincere thanks to our respected principal **Dr. RAVISHANKAR M**, Principal DSATM for all his support.

We express our deepest gratitude and special thanks to **Dr. C. Nandini**, Vice Principal & H.O.D, Dept. Of Computer Science & Engineering, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project guides, **Ms. Apoorva Busad** (Assistant Professor, Dept of CSE) and **Ms. Nivetha Kumaravel** (Assistant Professor, Dept of CSE).

ABHISHEK BARNWAL (1DT19CS004)
ABHISHEK MISHRA (1DT19CS005)

ABSTRACT

In today’s world advanced technology, interactive computer graphics has become a powerful tool for the production of realistic features. Today’s we find computer graphics used in various areas that include science, engineering, medicine, business, industry, art, entertainment etc. The main reason for effectiveness of the interactive computer graphics is the speed with which the user can understand the displayed information.

OBJECTIVES OF THE PROJECT

- Developing a Catching Ball Game using computer graphics with OpenGL
- Migration from text editor to OpenGL.
- Implementing certain technical concepts like Translation, motion, and use of Idle Function.

The "CATCH THE BALL" game depicts a concept of collecting balls in the basket. It is a single player game. The balls are dropped from the top of the window and the basket is placed at the bottom which can be moved with the help of mouse or touchpad. There are 4 levels in the game, after collecting certain count of balls the level gets incremented and so as the speed of the falling balls. After dropping the certain number of balls, the game gets over and the total score is displayed. Score is directly proportional to the number of balls collected in the basket. The more balls caught in the basket; the more is the score of individual players.

The player is provided with an option to play the game using either MOUSE or Keyboard Buttons (A & D). The movement of mouse is synced with the movement of basket and the movement of balls in the left and right directions are synced with the button ‘A’ and ‘D’ respectively.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

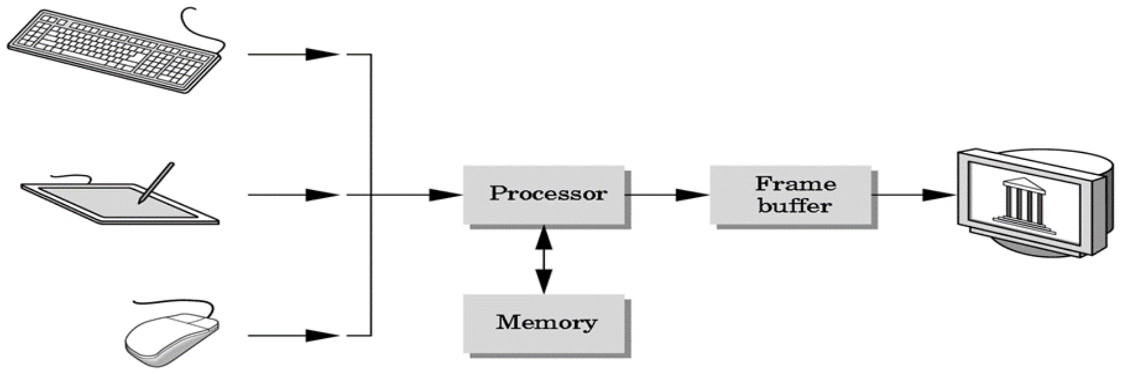
1.1 About Computer Graphics

The Computer Graphics is one of the most effective and commonly used methods to communicate the processed information to the user. It displays the information in the form of graphics objects such as pictures, charts, graphs and diagram instead of simple text.

In computer graphics, pictures or graphics objects are presented as a collection of discrete picture elements called **pixels**. The pixel is the smallest addressable screen element Computer graphics today is largely interactive: The user controls the contents structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen.

Computer Graphics relies on an internal model of the scene, that is, mathematical representation suitable for graphical computations. The model describes the 3D shapes, layout and materials of the scene. This 3D representation then has to be projected to compute a 2D image from a given viewpoint, this is rendering step. Rendering involves projecting the objects, handling visibility (which parts of objects are hidden) and computing their appearance and lighting interactions. Finally, for animated sequence, the motion of objects has to be specified.

Computer graphics today is largely interactive: The user controls the contents' structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen.



The image processing can be classified as

- Image enhancement.
- Pattern detection and recognition.
- Pattern detection and recognition.

The image enhancement deals with the improvement in the image quality by eliminating noise or by increasing image contrast. Pattern detection and recognition deals with the detection and clarification of standard patterns
And finding deviations from these patterns. The optical character recognition (OCR) technology is a practical example for pattern detection & recognition. Scene analysis deals with the recognition and reconstruction of 3D model of scene from several 2D images.

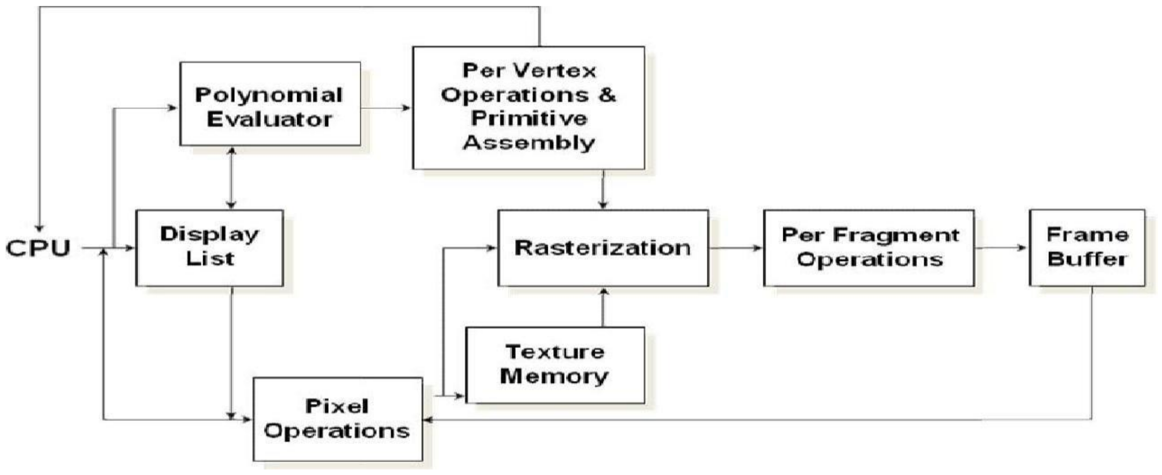
1.2 About OpenGL

OpenGL (**O**pen **G**raphics **L**ibrary) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc.
OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as the OpenGL state machine.

1.3 Features of OpenGL

- Geometric Primitives allow you to construct mathematical descriptions of objects.
- Viewing and Modelling permits arranging objects in a 3-dimensional scene, move our camera around space and select the desired ad vantage point for viewing the scene to be rendered.
- Materials lighting OpenGL provides commands to compute the color of any point given the properties of the material and the sources of light in the room.
- Transformations: rotation, scaling, translations, perspectives in 3D, etc. page

1.4 OpenGL Architecture



OpenGL Architecture

This is the most important diagram you will see today, representing the flow of graphical information, as it is processed from CPU to the frame buffer.
There are two pipelines of data flow. The upper pipeline is for geometric, vertex-based primitives. The lower pipeline is for pixel-based, image primitives. Texturing combines the two types of primitives together.

1.5 Overview of the Project

- Computer graphics involves the designing of objects in different forms which are regular and irregular in shape. The mini project named “CATCH THE BALL” is single player game where the player collects the ball in the basket.
- The balls are dropped from the top of the window and the basket is placed at the bottom which can be moved with the help of mouse or touchpad.
- There are 4 levels in the game. After each level, the speed of balls is increased.
- After dropping 20 balls, the game gets over and the total score is displayed.
- The more balls collected in the basket; the more is the score.
- Player can set the high score every time he/she plays.

CHAPTER 2

REQUIREMENTS

We have a wide range of options of languages. From these options we can choose appropriate platform/ tools and languages for development of the project. Some of these are as follows: -

Programming Languages: - In programming language we have C++.

Compiler: - GNU GCC compiler/C++ compiler

2.1 SOFTWARE REQUIREMENTS:

Operating system	:	Windows 2000 or later
Language Tools	:	OpenGL
Documentation Tool	:	Visual C++ with OpenGL functions

2.2 HARDWARE SPECIFICATIONS

Processor	:	Intel Pentium or more
Ram	:	128 MB or more
Cache	:	512 KB
Hard disk	:	16 GB hard disk recommended
Keyboard	:	Standard 101 key keyboard

CHAPTER 3

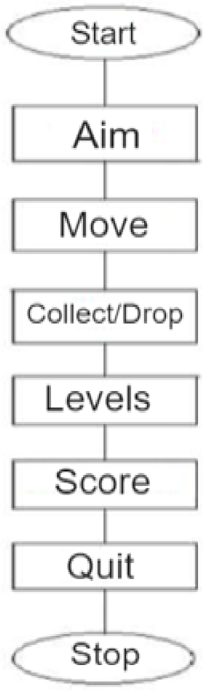
DESIGN

Design is the planning that lays the basics for the making of all objects or systems. This chapter involves designing of various aspects and different stages of project. When program is made to execute, the output window is displayed first. The flow of operation from output window is shown in figure.

3.1 Initialization

Initialize the interaction with the windows. Initialize the display mode, double buffer and depth buffer. Initialize the various call back functions for dropping and collecting balls and moving basket, for mouse interface. Initialize the window position and size and create the window to display the output.

3.2 Flowchart



Flowchart for catching ball program

CHAPTER 4

IMPLEMENTATION

4.1 OpenGL Functions

The various functions used in implementing this project are:

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

  This function is used to initialize glut.

- **glutInitDisplayMode()**

  This function is used to initialize a display mode for the screen. It can choose one of the following constant values as it's parameters:

  - GLUT_SINGLE
  - GLUT_RGB
  - GLUT_DOUBLE

- **glutInitWindowPosition()**

  This function is used to set the position of the top left corner of the window. It takes in 2 integer values as input parameters which are the coordinates specified for the position for the top left corner of the window.

- **glutInitWindowSize()**

  This function is used to specify the size of the created window, inside which the scene will be generated. It takes in two integer parameters, the width and height of the window.

- **glutCreateWindow()**

  This function creates the window with the specified dimensions and position and assigns the string parameter passed to it as the name of the window.

- **glutMainLoop()**

  This function is called at most once in a GLUT program. Once called, this routine will never return. It will call as necessary, any callbacks required for the program.

- **glBegin()**

  This function is used in drawing the basic primitives like lines, points, polygons and quads. It takes in a constant value which specifies the primitive to be drawn. Some of the parameters it can have are-

  ▪ GL_POLYGON
  ▪ GL_LINES CG Mini-project – Archery Game
  ▪ GL_LINE_LOOP
  ▪ GL_QUADS
  ▪ GL_QUAD_STRIP

- **glClear()**

  This function is used to clear the contents of the buffer passed as the parameters to this function, onto the screen.

  It takes a constant value as a parameter like:

  ▪ GL_CLEAR_BUFFER_BIT
  ▪ GL_DEPTH_BUFFER_BIT

- **glClearColor()**

  This function takes in four floating values:

  ▪ Red
  ▪ Green
  ▪ Blue
  ▪ Alpha

  Based on RGB values, the colour is decided. The Alpha value gives the degree of transparency of the colour. The values for RGBA will range from 0.0 to 1.0.

- **glEnd()**

  This function works with the glBegin() function. It marks the end of all the vertices to be used for drawing the primitive.

CHAPTER 5

SOURCE CODE

```
#include<stdlib.h>
#include<stdio.h>
#include<GL/glut.h>
#include<math.h>
#include<string.h>
#include<process.h>

int balls_caught=0,missed_balls=0,level_count=1,points=0;
int p=0;                    // To increment the points based on color
int d=0;                    // For color selection
int ball_x,ball_y=10;       // To compute the x and y co-ordinates of ball
int basket_x,basket_y;      // To compute the x and y co-ordinates of basket
int a=500,b=500;            // To set the default screen size
int s=0;                    // To display menu options
int dropped_balls=0;        // To calculate the number of balls dropped
int speed_1=1,speed_2=2,speed_3=3,speed_4=4;
int red_ball=0,white_ball=0;
int blue_ball=0,yellow_ball=0,green_ball=0;
int w=48,h=48,t=10,e=9,g=12;

void myinit();
void start_screen(int,int);
void ball();
void basket(int,int);
void print_score();
void ball_start();
void color();
void score();
void display(void);
void basket_set(int,int);
void myReshape(int,int);
void keys(unsigned char,int,int);
void menu(int);
int main(int,char**);

void myinit()
{
   glViewport(0,0,a,b);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0,(GLdouble)a,0,(GLdouble)b);
```

```
    glMatrixMode(GL_MODELVIEW);
    glutPostRedisplay();
}

// Drawing the initial screen with with the title
void start_screen(int i,int j)
{
    int k;
    char cat[50]="CATCH THE BALL";

    glColor3f(0,1,0);
    glRasterPos2i(140,180);
    for(k=0;k<50;k++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,cat[k]);

    glFlush();
    glColor3f(1,1,0);
    i+=g+w;
    j-=g+h;

    glBegin(GL_LINE_LOOP);
        glVertex2f(i,j);
        glVertex2f(i+w,j);
        glVertex2f(i+w,j+h/2);
        glVertex2f(i+w/2-t/2,j+h/2);
        glVertex2f(i+w/2-t/2,j+h/2-3*t/4);
        glVertex2f(i+w-t,j+h/2-3*t/4);
        glVertex2f(i+w-t,j+t);
        glVertex2f(i+t,j+t);
        glVertex2f(i+t,j+h-t);
        glVertex2f(i+w,j+h-t);
        glVertex2f(i+w,j+h);
        glVertex2f(i,j+h);
    glEnd();
    glFlush();

    i+=g+w;
    glBegin(GL_LINE_LOOP);
        glVertex2f(i,j);
        glVertex2f(i+t,j);
        glVertex2f(i+w/2-3*t/4,j+h/2-t);
        glVertex2f(i+w/2+3*t/4,j+h/2-t);
        glVertex2f(i+w-t,j);
        glVertex2f(i+w,j);
        glVertex2f(i+w/2+t/2,j+h);
        glVertex2f(i+w/2-t/2,j+h);
```

```
glEnd();

glBegin(GL_LINE_LOOP);
   glVertex2f(i+w/2,j+h-t);
   glVertex2f(i+w/2-t/2,j+h/2);
   glVertex2f(i+w/2+t/2,j+h/2);
glEnd();

i+=g+w;
glBegin(GL_LINE_LOOP);
   glVertex2f(i,j);
   glVertex2f(i+t,j);
   glVertex2f(i+t,j+h-2*t);
   glVertex2f(i+w/2,j+t/2);
   glVertex2f(i+w-t,j+h-2*t);
   glVertex2f(i+w-t,j);
   glVertex2f(i+w,j);
   glVertex2f(i+w,j+h);
   glVertex2f(i+w-t,j+h);
   glVertex2f(i+w/2,j+h/2);
   glVertex2f(i+t,j+h);
   glVertex2f(i,j+h);
glEnd();

i+=g+w;

glBegin(GL_LINE_LOOP);
   glVertex2f(i,j);
   glVertex2f(i+w,j);
   glVertex2f(i+w,j+t);
   glVertex2f(i+t,j+t);
   glVertex2f(i+t,j+t+e);
   glVertex2f(i+t+2*e,j+t+e);
   glVertex2f(i+t+2*e,j+2*t+e);
   glVertex2f(i+t,j+2*t+e);
   glVertex2f(i+t,j+2*t+2*e);
   glVertex2f(i+w,j+2*t+2*e);
   glVertex2f(i+w,j+h);
   glVertex2f(i,j+h);
glEnd();
glFlush();



glColor3f(1,1,1);
ball_x=375;
```

```
    ball_y=370;

    ball();
    basket(330,0);
}
// Drawing an ball
void ball()
{
    float x,y,z;
    float t;
    glBegin(GL_POLYGON);
    for(t=0;t<=360;t+=1)
    {
        x = ball_x+16*(cos(t));
        y = ball_y+16*(sin(t));
        z = 0;
        glVertex3f(x,y,z);
    }
    glEnd();

}

// Drawing the basket
void basket(int i,int j)
{
    j=10;
    if(i>=a-60)i=a-60;

    glColor3f(1.0,0.0,0.0);
    glBegin(GL_QUADS);
        glVertex2f(0.0+i,50.0+j);
        glVertex2f(10.0+i,10.0+j);
        glVertex2f(50.0+i,10.0+j);
        glVertex2f(60.0+i,50.0+j);
    glEnd();
}

//Displaying the score. This function is called when the game is over
void print_score()
{
    printf("\nLevel reached: %d\n\n",level_count);
    printf("\nNo. of balls dropped    = %d \n",dropped_balls);
    printf("\nNo. of balls caught      = %d\n",balls_caught);
    printf("\nNo. of balls missed      = %d\n",missed_balls);
    printf("\nWhite balls   = %d\tpoints gained   = %d\n",white_ball,white_ball);
    printf("\nRed balls  = %d\tpoints gained   = %d\n",red_ball,red_ball*5);
    printf("\nGreen balls   = %d\tpoints gained   = %d\n",green_ball,green_ball*10);
    printf("\nBlue balls    = %d\tpoints gained   = %d\n",blue_ball,blue_ball*15);
    printf("\nYellow balls  = %d\tpoints gained   = %d\n\n",yellow_ball,yellow_ball*100);

    printf("\n\n\nYour score = %d\n\n",points);

}
//init_score initialises score counters after a game is over.
void init_score()
{
    balls_caught=0,missed_balls=0,level_count=1;
    dropped_balls=0;
    missed_balls=0;
    white_ball=0;
    red_ball=0;
    green_ball=0;
    blue_ball=0;
    yellow_ball=0;
    points=0;
    s=0;
}


// All the three ball generators start dropping the balls
void ball_start()
{
    ball_y=800;
    if(missed_balls<10)
    {
        dropped_balls++;
        switch(rand()%9)
        {
            case 0:ball_x=105;d=rand()%5; break;
            case 1:ball_x=245;d=rand()%5; break;
            case 2:ball_x=380;d=rand()%5; break;
            case 5:ball_x=105;d=rand()%5; break;
            case 3:ball_x=245;d=rand()%5; break;
            case 4:ball_x=380;d=rand()%5; break;
            case 7:ball_x=105;d=rand()%5; break;
            case 6:ball_x=245;d=rand()%5; break;
            case 8:ball_x=380;d=rand()%5; break;
        }
    }
    else
    {
        printf("\n\n\t\t\t\tGAME OVER\n\n");
        print_score();
        init_score();
    }

}
//colours for the balls
void color()
{
    switch(d)
    {
        case 0:glColor3f(0,0,1);p=15;break;     //blue
        case 1:glColor3f(1,1,0);p=100;break;    //yellow
        case 2:glColor3f(0,1,0);p=10;break;     //green
        case 3:glColor3f(1,0,0);p=5;break;      //red
        case 5:glColor3f(1,1,1);p=1;break;      //white
    }
     glFlush();
}

// score function increments the score counters
void score()
{
    if(ball_y<=50 && (ball_x>=basket_x && ball_x<=basket_x+60))
    {
        printf("\a");

        balls_caught++;

        switch(d)
        {
            case 0: blue_ball++;break;
            case 1: yellow_ball++;break;
            case 2: green_ball++;break;
            case 3: red_ball++;break;
            case 5: white_ball++;break;
        }
        points+=p;
        ball_y=-10;

    }
    missed_balls=dropped_balls-balls_caught;

}


void display(void)
```

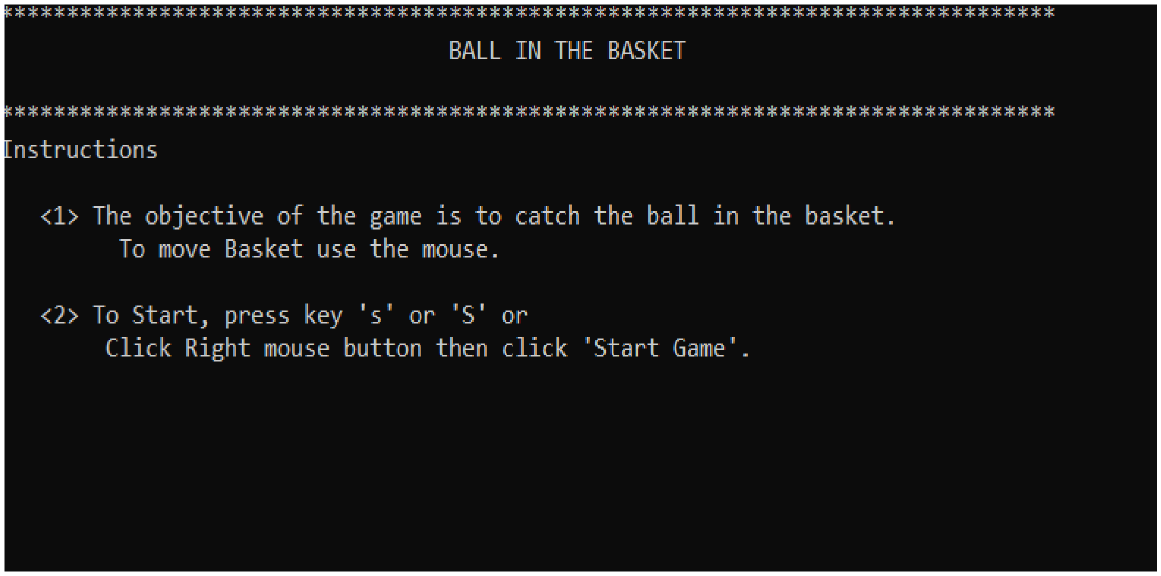
```
{

    char level1[12]="LEVEL 1";
    char level2[12]="LEVEL 2";
    char level3[12]="LEVEL 3";
    char level4[12]="LEVEL 4";

    glClear(GL_COLOR_BUFFER_BIT);


    if(s>=1)
    {
        //Diplaying the level
        if(level_count==1)
        {
            glColor3f(1,1,1);
            glRasterPos2i(400,300);
            int i;
            for(i=0;i<12;i++)
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13,level1[i]);
        }
        else if(level_count==2)
        {
            glColor3f(1,1,1);
            glRasterPos2i(400,300);
            int i;
            for(i=0;i<12;i++)
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13,level2[i]);
        }
        else if(level_count==3)
        {
            glColor3f(1,1,1);
            glRasterPos2i(400,300);
            int i;
            for(i=0;i<12;i++)
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13,level3[i]);
        }
        if(level_count==4)
        {
            glColor3f(1,1,1);
            glRasterPos2i(400,300);
            int i;
            for(i=0;i<12;i++)
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13,level4[i]);
        }
        if(ball_y<=10)
        {
            ball_start(); //ball_start() is the balls dropping function
        }
        color();
        ball();
        basket(basket_x,basket_y);

        //changing the level based on the balls caught

        if(balls_caught>=5)
        {

            ball_y-=speed_2;
            level_count=2;
        }
        if(balls_caught>=10)
        {

            ball_y-=speed_3;
            level_count=3;
        }
        if(balls_caught>=15)
        {

            ball_y-=speed_4;
            level_count=4;
        }

        else
            ball_y-=speed_1;
        score();
    }

    else
        //No game start event. Display initial screen
        start_screen(40,300);
    glFlush();
    glutSwapBuffers();

}


void basket_set(int a,int b)
{
    basket_x=a;
```

```
   basket_y=b;
   glutPostRedisplay();
   glFlush();
}

void myReshape(int w,int h)
{
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0,(GLdouble)w,0.0,(GLdouble)h);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glViewport(0,0,w,h);
   a=w;
   b=h;
}


void keys(unsigned char key,int x,int y)
{

   if(key=='q'||key=='Q')
   {
      printf("\n\n\n\t\tQUIT BY PLAYER\n\n");
      print_score();
      exit(0);
   }
   if(key=='s'||key=='S')
      s+=1;
   if(key=='a'||key=='A') //moving the ball to the left
   {
      ball_x-=10;
      if(ball_x<=0) ball_x=10;
   }
   if(key=='d'||key=='D') //moving the ball to the right
   {
      ball_x+=10;
      if(ball_x>=500) ball_x=490;
   }

}


void menu(int id)
{
   switch(id)
   {
      case 1: s+=1;
          break;

      case 2: printf("\n\n\n\t\tPLAYER HAS QUIT\n");
          print_score();
          exit(0);
   }
   glutPostRedisplay();

}


int main(int argc,char **argv)
{

printf("*****************************************************************
**************");
   printf("\n\t\t\t\t  BALL IN THE BASKET \n\n");

printf("*****************************************************************
**************");
   printf("\nInstructions\n\n   <1> The objective of the game is to catch the ball in the
basket.\n\t To move Basket use the mouse.\n");
   printf("\n   <2> To Start, press key 's' or 'S' or \n\tClick Right mouse button then click
'Start Game'.\n");

   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
   glutInitWindowSize(a,b);
   glutCreateWindow("BALL IN THE BASKET");
   myinit();

   glutCreateMenu(menu);
   glutAddMenuEntry("Start game",1);
   glutAddMenuEntry("Quit",2);
   glutAttachMenu(GLUT_RIGHT_BUTTON);

   glutDisplayFunc(display);
   glutKeyboardFunc(keys);
   glutPassiveMotionFunc(basket_set);
   glutIdleFunc(display);
   glutReshapeFunc(myReshape);
   glutMainLoop();

  }
```

CHAPTER 6

SCREENSHOTS

6.1 Instruction Window:



```
********************************************************************************
                                  BALL IN THE BASKET

********************************************************************************
Instructions

   <1> The objective of the game is to catch the ball in the basket.
         To move Basket use the mouse.

   <2> To Start, press key 's' or 'S' or
        Click Right mouse button then click 'Start Game'.
```

Fig 6.1: This figure shows the instructions of the game.

6.2 Main Window:

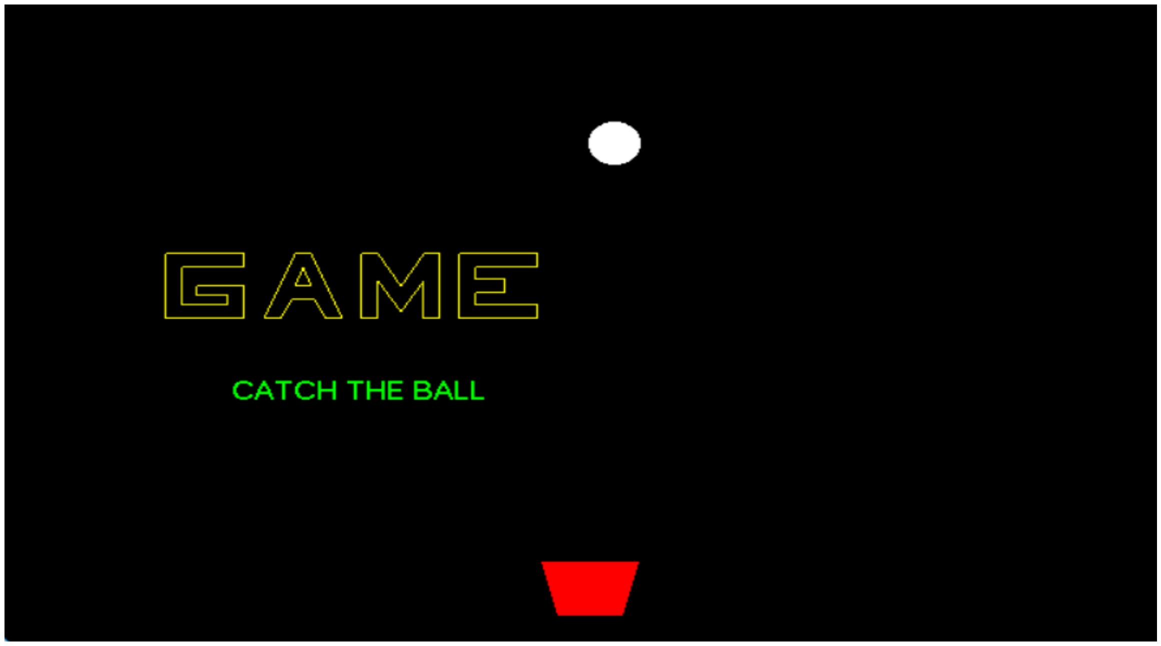


Fig 6.2: This figure shows the main window of the game.

6.3 Level-1:

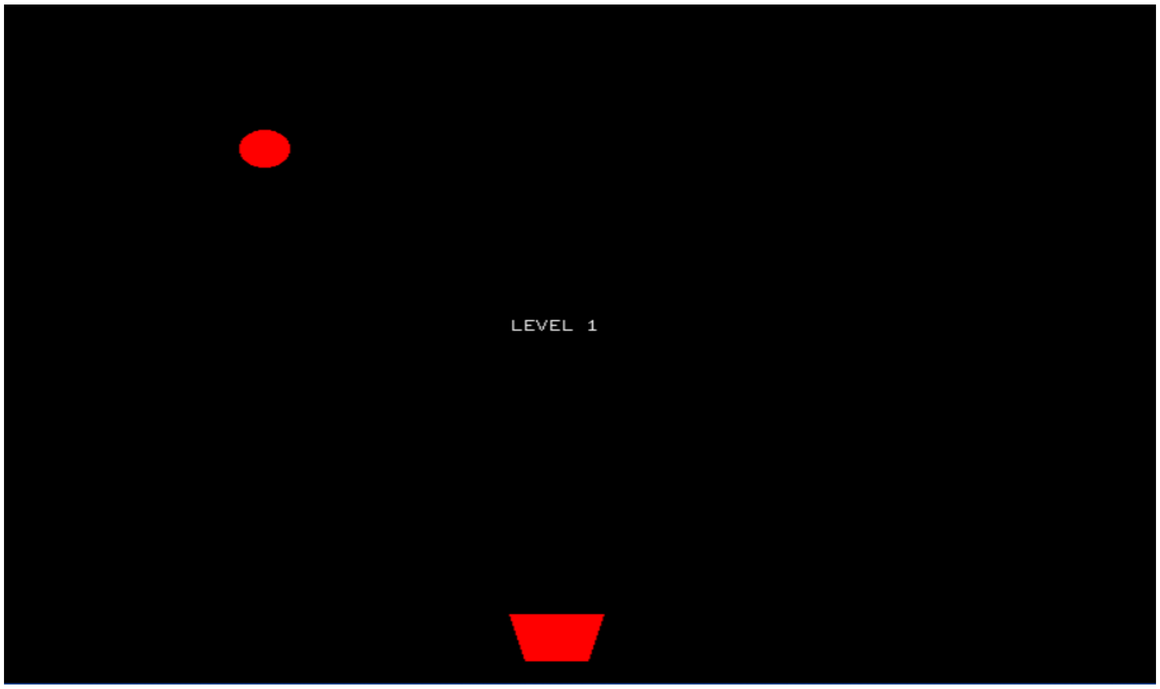


Fig 6.3: This figure shows the first level of the game.

6.4 Level-2 :

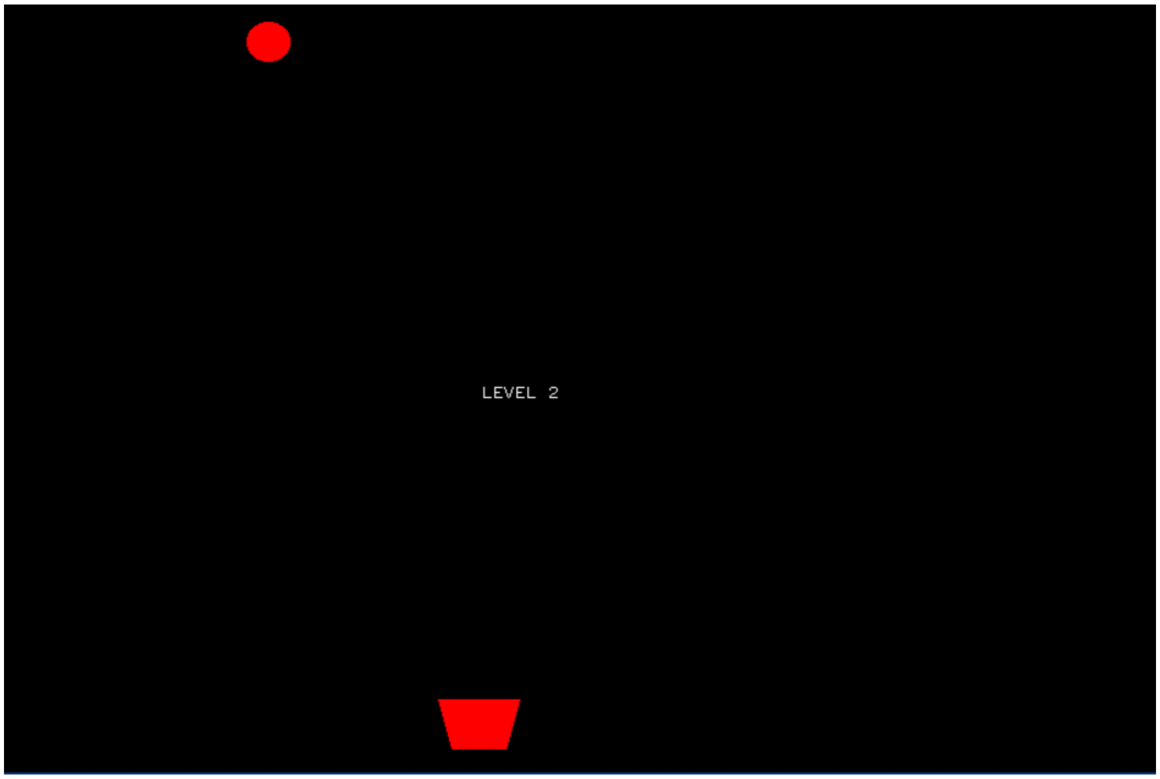
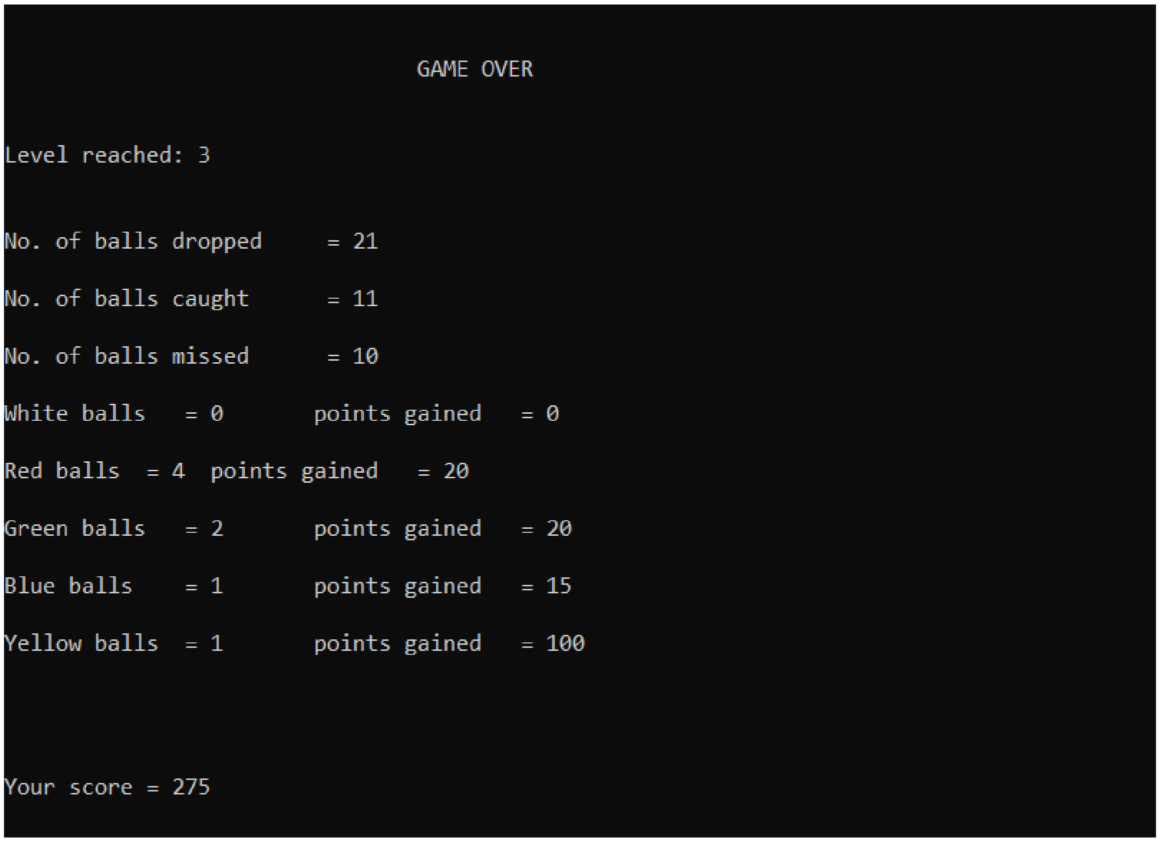


Fig 6.4: This figure shows the second level of the game.

6.5 Score Board:



```
                                GAME OVER


Level reached: 3


No. of balls dropped     = 21

No. of balls caught      = 11

No. of balls missed      = 10

White balls   = 0       points gained   = 0

Red balls   = 4  points gained   = 20

Green balls   = 2       points gained   = 20

Blue balls    = 1       points gained   = 15

Yellow balls  = 1       points gained   = 100



Your score = 275
```

Fig 6.5: This figure shows the score of the player.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The Catching Ball Graphics package has been developed Using OpenGL. The main idea of the program is to provide a game through the keyboard interaction, which includes catching balls in the basket at different levels. Further, we increment the speed of the game (dropping balls) at each level and then finally display the score achieved by the player in the game.

7.2 Future Enhancement

The project has graphical usage where in different functions which can be applicable in graphical packages are used. The applications developed so far is in its basic working stage. There can be more enhancements like

- Designing of a different balls and baskets with complex interior structures.
- Adding the features of displaying score at each level.
- By providing keyboard-based interaction to every level.
- We can also change the colour and provide more effects.

BIBLIOGRAPHY

BOOKS

- Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson Education, 2008.
- Donald Hearn and Pauline Baker: Computer Graphics- OpenGL Version, 3rd Edition, Pearson Education, 2008.
- F.S Hill Jr.: Computer Graphics Using OpenGL, 3rd Edition, PHI, 2009.

WEBSITES

- http://www.opengl.org/resources/libraries/glut
- http://www.opengl.ord
- http://en.wikipedia.org/wiki/opengl

Contact Details:

NAME: ABHISHEK BARNWAL

USN: 1DT19CS004

SEMESTER AND SECTION: 6TH SEM, A SEC

COLLEGE: DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

PHONE: 7301059777

EMAIL: AABARNWAL01@GMAIL.COM

NAME: ABHISHEK MISHRA

USN: 1DT19CS005

SEMESTER AND SECTION: 6TH SEM, A SEC

COLLEGE: DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

PHONE: 7273870237

EMAIL: ABHI727387@GMAIL.COM