

Chapter 1

INTRODUCTION

The project Maze Game is basically a mini game of arcade. The game has the scores displayed to have knowledge about the record of your performance. This project has the square object moving up, down, left and right the area to get the proper space for moving along the walls. It has limited time to reach the destination along the walls. At the end the result is displayed whether the user has won or lost in the given limited time.

1.1 Purpose of Project

- To help visualize the working of the code and the functions used.
- To explain the working of 2D space.
- To describe the structural aspect of bringing forth the idea of the project.

1.2 Advantages of Project

- Very User-Friendly.
- Easy to understand.
- Mode of entertainment as well.
- Helps in giving a visual aspect to the viewer.
- Provides a wide range of user defined options.

Chapter 2

REQUIREMENTS SPECIFICATION

Some of the basic requirements for the development of this project are as follows:

2.1 Hardware Requirements

- Processor : Intel core i3
- Hard Disk : 20 GB, 80 GB, 160 GB or above
- Monitor : 15 VGA colour, 1024*768 resolution
- RAM : 2GB or above
- Input Device : Keyboard and Mouse

2.2 Software Requirements

- Operating System : Windows 7
- Tools used : Codeblocks and OpenGL interfaces
- Coding Language : C/C++
- Library used : GLUT

Chapter 3

DESIGN AND IMPLEMENTATION

3.1 Data Flow Diagram

The complete overview of the project is shown in the block diagram, it shows how it perform the actions using the user interactions.

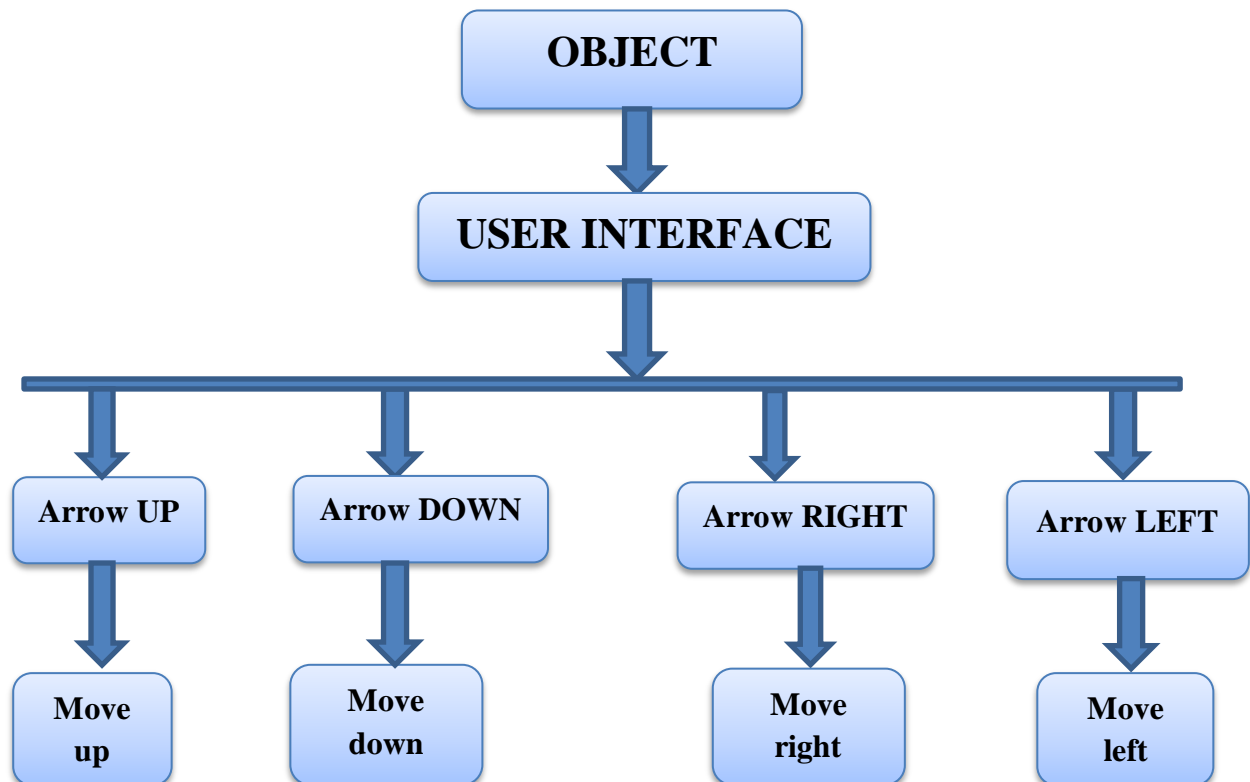


Fig. 3.1: Data Flow Diagram of Maze game

Fig. 3.1 shows the inputs given to the program for the purpose of interaction and the functionality of those inputs as to what actions is performed.

3.2 Functions Used

In this section, the algorithm regarding the “Maze Game” is explained, as well as the various OpenGL and user-defined functions. An algorithmic approach is a good practice with which the flow within a program goes about.

The most commonly used functions of the graphics library in our game are:-

- Call back functions
- GL functions
- GLUT functions

Call Back Functions:

A callback function is a function which the library(GLUT) calls when it needs to know how to process something.

- **Void display():** This function is passed to glutDisplayFunc() in Order to display OpenGL contents on window.

GL Functions:

GL(open graphics library) Functions that are used for creating 2D/3D objects and screens.

- **glPushMatrix() :** glPushMatrix pushes the current matrix stack down by one, duplicating the current matrix. That is, after a glPushMatrix call, the matrix on top of the Stack is identical to the one below it.
- **glLoadIdentity() :** This function replaces the current matrix with the Identity matrix.
- **glPopMatrix() :** glPopMatrix pops the current matrix stack, replacing the current matrix with the one below it on the stack.
- **glRotatef() :** This produces a rotation of angles, degrees around the vector(x, y, z). The current matrix is multiplied by a rotation matrix with the product replacing the current matrix.
- **glTranslatef() :** This displaces a point to new positions defined by a Displacement vector.

GLU Functions:

GLU (OpenGL Utility Library) is a graphics library consisting of utility functions which can be used with OpenGL which primarily focuses on primitive rendering and mapping between screen and world coordinates.

- **gluLookAt ():** This defines a viewing transformation.
- **gluOrtho2D ():** This function defines a 2-D orthographic Projection matrix.

GLUT Functions:

GLUT (OpenGL Utility Toolkit) is a library of utilities for OpenGL, which primarily focuses on window definition, window control and monitoring of keyboard and mouse input.

- **glutCreateWindow ():** This creates a top-level window. The name will be provided to the window system as the window's name.
- **glutDisplayFunc():** Graphics are sent to the screen through a function called the display callback and registered with the window system. Here the function name will be called whenever the windowing system determines that OpenGL window needs to be redisplayed.
- **glutCreateMenu():** This creates a new pop-up menu and returns a unique small integer identifier.
- **glutInitWindowSize(int width, int size):** specifies the size, in pixels, of your window.
- **glutCreateWindow(char *string):** creates a window with an OpenGL context. It returns a unique identifier for the new window.
- **glutDisplayFunc(mydisplay):** is used to invoke the call back display.
- **glutKeyboardFunc():** is the call back for events generated by pressing a key.
- **glutMainLoop():** causes the program to begin an event processing loop.

3.3 User Defined Functions Used

- **int main()** :A program shall contain a global function main which is designated start of the program. The main function is called at program startup after initialization of the nonlocal objects with the static storage duration.

Argc - non negative value representing the number of arguments passed to the program from the environment in which the program is run.

Argv - pointer to the first elements of an array pointers to null-terminated multi byte strings that represent the arguments passed to the program from the execution environment.

- **glutInit (int argc, char **argv)** : initializes GLUT and processes any command line arguments.
- **glutInitDisplayMode(unsigned int mode)** :specifies whether to use an RGBA or color-index color model. We can also specify whether you want a single or double buffered window.

Chapter 4

TESTING

4.1 GENERAL INTRODUCTION:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a project. It provides a way to check the functionality of components and functions used in the source code. It is the process of exercising software with the intent of ensuring that the software system meets its user requirements and user expectation and does not fail in any manner.

4.2 TEST CASES:

Table 4.1: Test Cases for maze game

S.No	Test Case	Actual Output	Expected Result	Remarks
1	Functionality of arrow key UP.	Move object towards UP.	Move object towards UP.	Result Achieved
2	Functionality of arrow key DOWN.	Move object towards DOWN.	Move object towards DOWN.	Result Achieved
3	Functionality of arrow key RIGHT.	Move object towards RIGHT.	Move object towards RIGHT.	Result Achieved
4	Functionality of arrow key LEFT.	Move object towards LEFT.	Move object towards LEFT.	Result Achieved

Table 4.1 shows the different tests done on different input cases with the outputs obtained.

Chapter 5

SNAPSHOTS

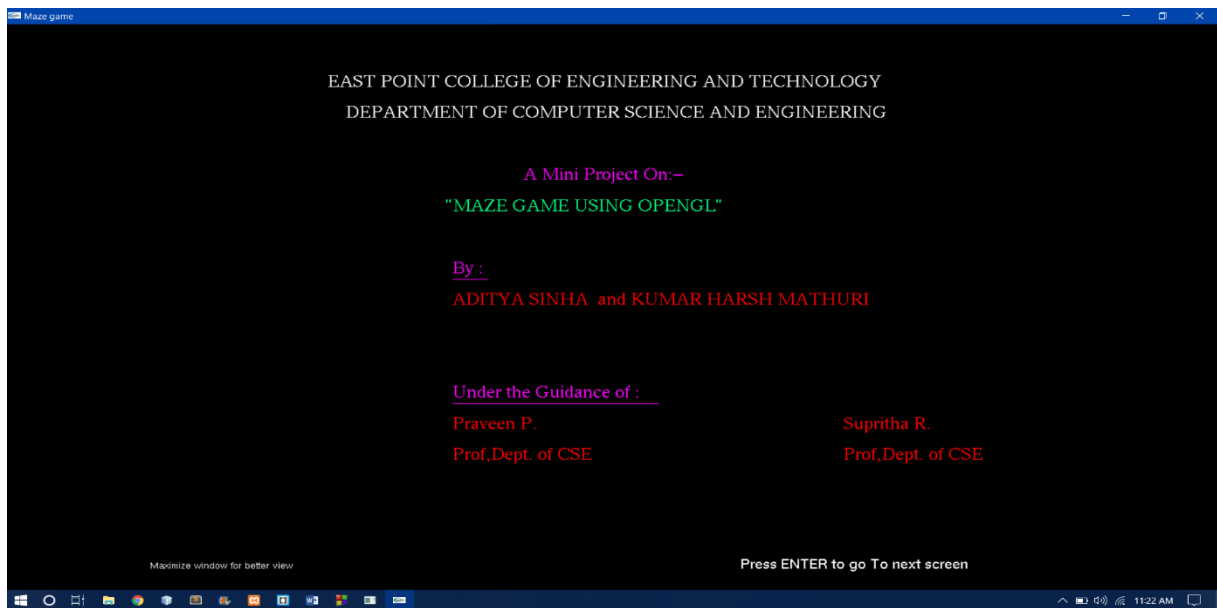


Fig. 5.1: Front Page View

Fig. 5.1 shows the front page of the project which describes the institution of education, project name, student information and guidance information.

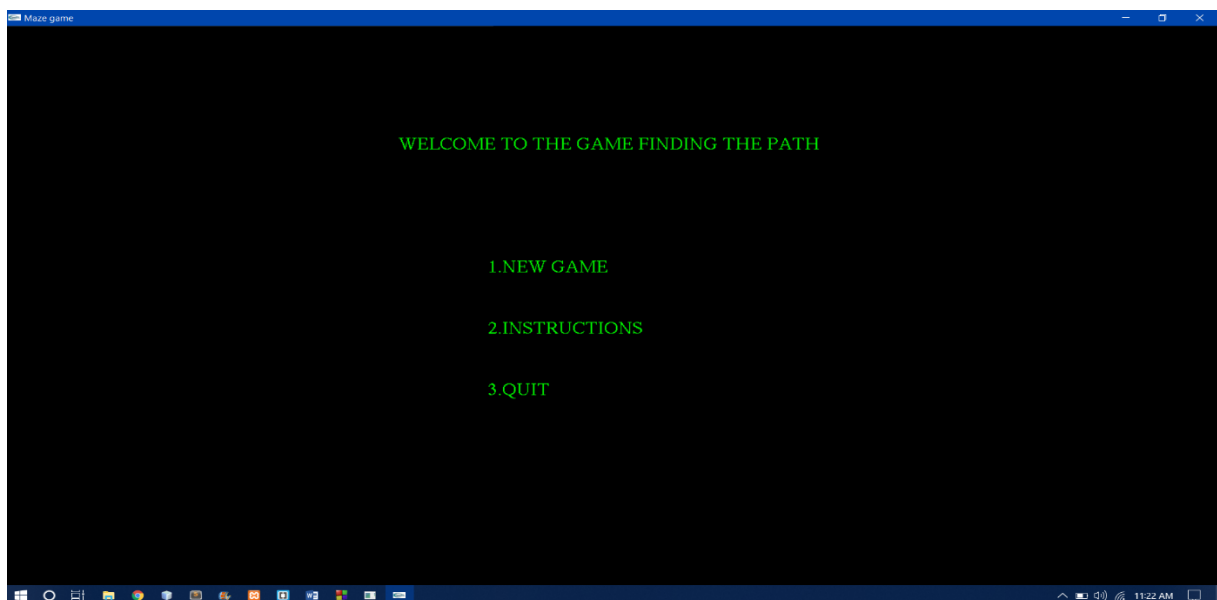


Fig. 5.2: Welcome page of The Game

Fig. 5.2 shows the welcome page of the game.

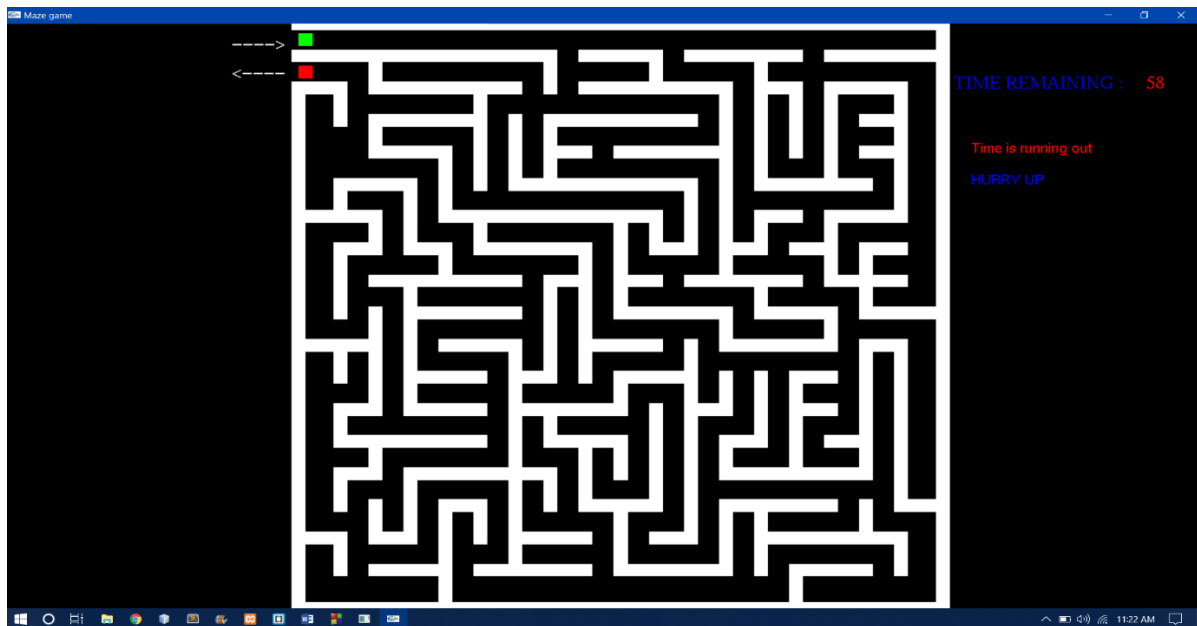


Fig. 5.3: Screen After Starting the game

Fig. 5.3 shows the display of the game screen when the player starts the game.

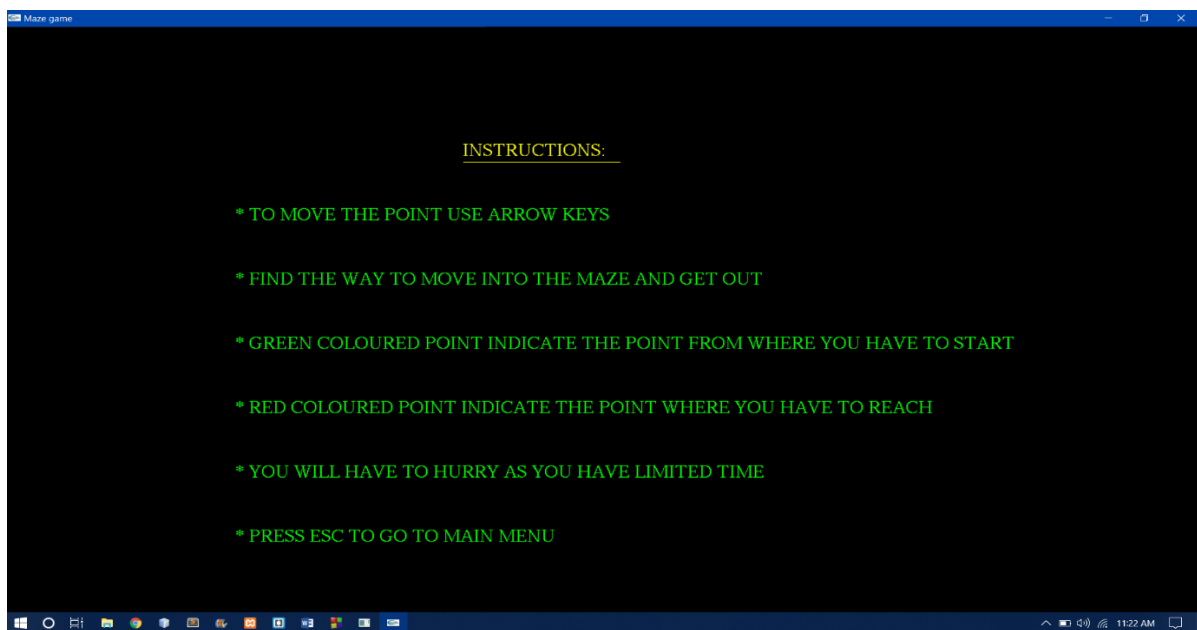


Fig. 5.4: Screen showing the instructions

Fig. 5.4 show the game window when the player view the instructions.

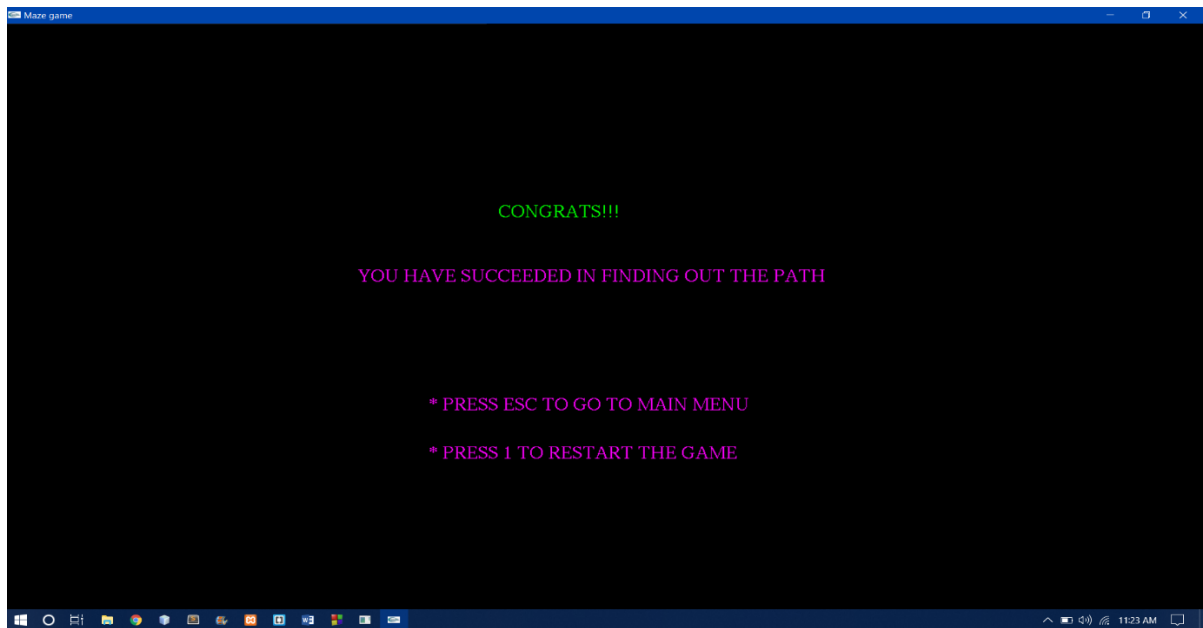


Fig. 5.5: Screen Displayed if User wins

Fig. 5.5 displays the game window when the player wins the game.

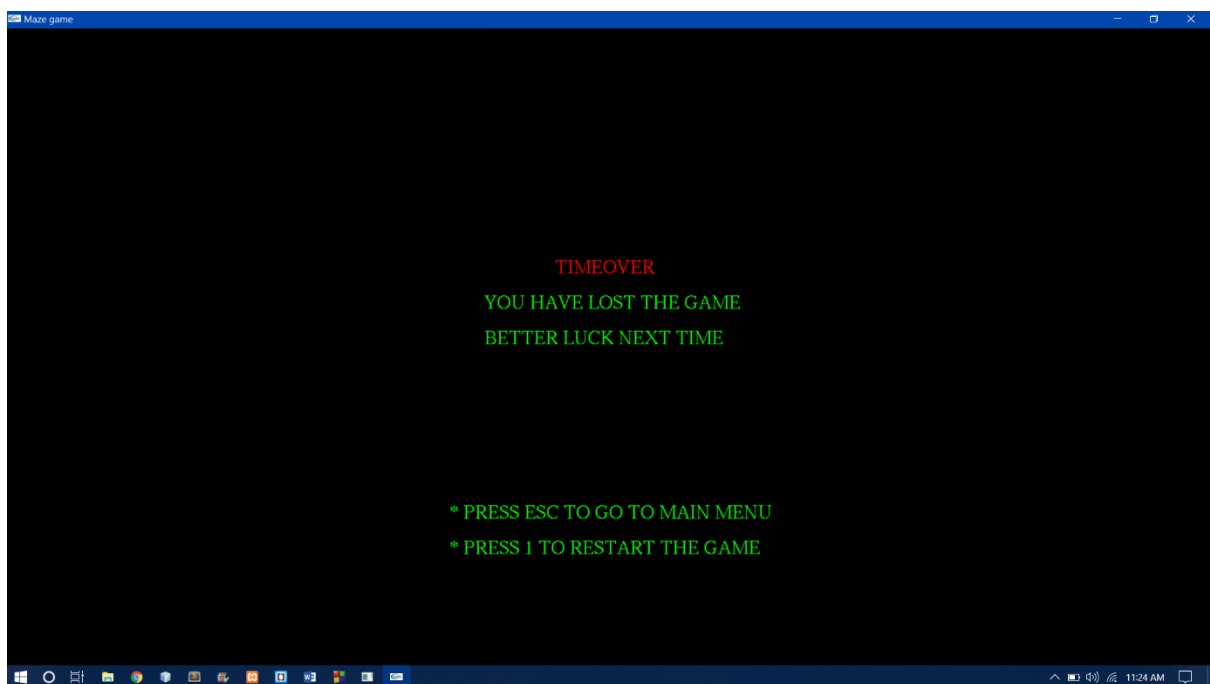


Fig. 5.6: Screen Display if User loses

Fig. 5.6 shows the window when the player loses the game.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The implementation of Maze Game project has given us the knowledge of how the graphic functions works. This project is a very basic and simple way to implement the working of a graphics program. The project was constructed with the use of OpenGL functions and primitives. The project was to reach the destination in limited time. The object, on reaching destination decides which the player wins and on failing which the player loses.

6.2 Future Enhancement

The project, for the future enhancement, can be added with some modification code to make the game more attractive. The following things can be done:

- Make the game more realistic.
- Add more levels to the game.
- Add different types of object with different speeds.
- Add different difficulty levels.

BIBLIOGRAPHY

- [1]. Edward Angel: “*Interactive Computer Graphics A Top-Down Approach with OpenGL*”, 5th Edition, Addison-Wesley, 2008.
- [2]. Donald Hearn and Pauline Baker: “*Computer Graphics using OpenGL*”, 3rd Edition, Pearson Education, 2011.

APPENDIX

SOURCE CODE

```
#include<windows.h>
#include<stdio.h>
#include<stdlib.h>
#include<GL/glut.h>
#include<math.h>
#include<string.h>
#include<time.h>
int x,y;
int i,count;
char t[2];
float px=0.0,py=175.0;
int flag, df=10;
clock_t start,end;
static GLfloat rotated=0.0;
static GLint rotateq=0;
void point()
{
    glColor3f(0.0,0.0,1.0);
    glBegin(GL_POINTS);
    glVertex2f(px,py);
    glEnd();
}
void point1()
{
    glColor3f(0.0,1.0,0.0);
    glBegin(GL_POINTS);
    glVertex2f(0.0,175.0);
    glEnd();
}
void point2()
{
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POINTS);
    glVertex2f(0.0,165.0);
    glEnd();
}

void output(int x, int y, char *string)
{
    int len, i;
    glRasterPos2f(x,y);
    len=(int) strlen(string);
    for (i = 0; i < len; i++)
    {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24,string[i]);
    }
}
```

```
    }  
}
```

```
void drawstring(int x, int y, char *string,void *font)
```

```
{  
    int len, i;  
    glRasterPos2f(x,y);  
    len=(int) strlen(string);  
    for (i = 0; i < len; i++)  
    {  
        glutBitmapCharacter(font,string[i]);  
    }  
}
```

```
void frontend(void)
```

```
{  
    glClear(GL_COLOR_BUFFER_BIT);  
    glLoadIdentity();  
    glColor3f(1,1,1);  
    drawstring(120,5," Press ENTER to go To next screen",GLUT_BITMAP_HELVETICA_18);  
    drawstring(-45,5,"Maximize window for better view",GLUT_BITMAP_HELVETICA_12);  
    glColor3f(1,1,1);  
    output(5,160,"EAST POINT COLLEGE OF ENGINEERING AND TECHNOLOGY");  
    glColor3f(1,1,1);  
    output(10,150,"DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING");  
    glColor3f(1,0,1);  
    output(60,130,"A Mini Project On:-");  
    glColor3f(0,1,0.5);  
    output(38,120,"\"MAZE GAME USING OPENGL\"");  
    glColor3f(1,0,1);  
    output(40,100,"By :");  
    glBegin(GL_LINES);  
    glVertex2f(40,98);  
    glVertex2f(50,98);  
    glEnd();  
    glColor3f(1,0,0);  
    output(40,90,"ADITYA SINHA and KUMAR HARSH MATHURI ");  
    output(40,80,"");  
    glColor3f(1,0,1);  
    output(40,60,"Under the Guidance of :");  
    glBegin(GL_LINES);  
    glVertex2f(40,58);  
    glVertex2f(98,58);  
    glEnd();  
    glColor3f(1,0,0);  
    output(40,50,"Praveen P.");  
    //glColor3f(1,0,0);  
    //drawstring(72,50,"(B.E.)",GLUT_BITMAP_HELVETICA_12);  
    glColor3f(1,0,0);  
    output(40,40,"Prof,Dept. of CSE");  
}
```

```
    glColor3f(1,0,0);
    output(150,50,"Supritha R.");
    //glColor3f(1,0,0);
    //drawstring(72,30,"(B.E.)",GLUT_BITMAP_HELVETICA_12);
    output(150,40,"Prof,Dept. of CSE");
    glFlush();

}

void winscreen()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glColor3f(0.0,1.0,0.0);
    output(55,120,"CONGRATS!!!");
    glColor3f(1.0,0.0,1.0);
    output(15,100,"YOU HAVE SUCCEEDED IN FINDING OUT THE PATH");
    output(35,60,"* PRESS ESC TO GO TO MAIN MENU");
    output(35,45,"* PRESS 1 TO RESTART THE GAME");
    glFlush();
}

void startscreen()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0,1.0,0.0);
    output(25,140,"WELCOME TO THE GAME FINDING THE PATH");
    output(50,100,"1.NEW GAME");
    output(50,80,"2.INSTRUCTIONS");
    output(50,60,"3.QUIT");
    glFlush();
}

void instructions()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0,1.0,0.0);
    output(45,140,"INSTRUCTIONS:");
    glBegin(GL_LINES);
    glVertex2f(45,138);
    glVertex2f(90,138);
    glEnd();
    glColor3f(0,1,0);
    output(-20,120,"* TO MOVE THE POINT USE ARROW KEYS");
    output(-20,100,"* FIND THE WAY TO MOVE INTO THE MAZE AND GET OUT");
    output(-20,80,"* GREEN COLOURED POINT INDICATE THE POINT FROM WHERE YOU
HAVE TO START");
    output(-20,60,"* RED COLOURED POINT INDICATE THE POINT WHERE YOU HAVE TO
REACH");
    output(-20,40,"* YOU WILL HAVE TO HURRY AS YOU HAVE LIMITED TIME");
    output(-20,20,"* PRESS ESC TO GO TO MAIN MENU");
```

```
    glFlush();
}

void timeover()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0,0,0);
    output(70,110,"TIMEOVER");
    glColor3f(0,1,0);
    output(50,100,"YOU HAVE LOST THE GAME");
    output(50,90,"BETTER LUCK NEXT TIME");
    output(40,40,"* PRESS ESC TO GO TO MAIN MENU");
    output(40,30,"* PRESS 1 TO RESTART THE GAME");
    glFlush();
}

void idle()
{
    if(df==1)
    {
        end=clock();
        count=(end-start)/CLOCKS_PER_SEC;
        if(count==60)
        {
            df=4;
        }
        else
            if((count<60) && ((px>=0 && px<=4) && (py>=162 && py<=168)))
            {
                df=5;
            }
        }
    glutPostRedisplay();
}

void wall(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2,GLfloat x3,GLfloat y3,GLfloat x4,GLfloat
y4)
{
    glBegin(GL_POLYGON);
    glVertex3f(x1,y1,0);
    glVertex3f(x2,y2,0);
    glVertex3f(x3,y3,0);
    glVertex3f(x4,y4,0);
    glEnd();
}

void SpecialKey(int key, int x, int y)
```



```
{
switch (key)
{
case GLUT_KEY_UP:
    flag=0;
    if(py<175)
        if(!((px>=8 && px<=12) && (py>=145 && py<=162)))
            if(!((px>=168 && px<=172) && (py>=5 && py<=22)))
                if(!((px>=132 && px<=172) && (py>=15 && py<=22)))
                    if(!((px>=128 && px<=132) && (py>=5 && py<=32)))
                        if(!((px>=142 && px<=162) && (py>=5 && py<=12)))
                            if(!((px>=118 && px<=152) && (py>=25 && py<=32)))
                                if(!((px>=88 && px<=122) && (py>=5 && py<=12)))
                                    if(!((px>=48 && px<=82) && (py>=5 && py<=12)))
                                        if(!((px>=62 && px<=82) && (py>=15 && py<=22)))
                                            if(!((px>=8 && px<=12) && (py>=5 && py<=18)))
                                                if(!((px>=0 && px<=12) && (py>=15 && py<=22)))
                                                    if(!((px>=42 && px<=52) && (py>=25 && py<=32)))
                                                        if(!((px>=18 && px<=42) && (py>=5 && py<=12)))

                                                            if(!((px>=168 && px<=180) && (py>=25 && py<=32)))
                                                                if(!((px>=98 && px<=108) && (py>=15 && py<=22)))
                                                                    if(!((px>=108 && px<=112) && (py>=15 && py<=82)))
                                                                        if(!((px>=88 && px<=92) && (py>=35 && py<=48)))
                                                                            if(!((px>=68 && px<=92) && (py>=45 && py<=52)))
                                                                                if(!((px>=98 && px<=102) && (py>=29 && py<=62)))
                                                                                    if(!((px>=78 && px<=82) && (py>=25 && py<=48)))
                                                                                        if(!((px>=68 && px<=72) && (py>=25 && py<=38)))
                                                                                            if(!((px>=32 && px<=72) && (py>=35 && py<=42)))
                                                                                                if(!((px>=22 && px<=32) && (py>=19 && py<=22)))
                                                                                                    if(!((px>=22 && px<=32) && (py>=15 && py<=22)))
                                                                                                        if(!((px>=18 && px<=22) && (py>=15 && py<=32)))
                                                                                                            if(!((px>=78 && px<=102) && (py>=25 && py<=32)))
                                                                                                                if(!((px>=88 && px<=92) && (py>=9 && py<=28)))
                                                                                                                    py=py+5;
                                                                                                                    glutPostRedisplay();
                                                                                                                    break;
case GLUT_KEY_DOWN:
    flag=0;
    if(py>5)
        if(!((px>=0 && px<=8) && (py>=158 && py<=165)))
            if(!((px>=8 && px<=12) && (py>=148 && py<=165)))
                if(!((px>=0 && px<=72) && (py>=168 && py<=175)))
                    if(!((px>=18 && px<=62) && (py>=158 && py<=165)))
                        if(!((px>=18 && px<=48) && (py>=148 && py<=155)))
                            if(!((px>=18 && px<=38) && (py>=138 && py<=145)))
                                if(!((px>=72 && px<=82) && (py>=138 && py<=145)))
                                    if(!((px>=68 && px<=112) && (py>=148 && py<=155)))
                                        if(!((px>=88 && px<=118) && (py>=138 && py<=145)))
```

```
if(!((px>=132 && px<=138) && (py>=158 && py<=165)))
if(!((px>=138 && px<=152) && (py>=68 && py<=75)))
if(!((px>=108 && px<=142) && (py>=168 && py<=175)))
if(!((px>=128 && px<=162) && (py>=128 && py<=135)))
if(!((px>=68 && px<=72) && (py>=158 && py<=175)))
if(!((px>=78 && px<=122) && (py>=158 && py<=165)))
if(!((px>=78 && px<=102) && (py>=168 && py<=175)))
if(!((px>=148 && px<=180) && (py>=168 && py<=175)))
if(!((px>=148 && px<=172) && (py>=158 && py<=165)))
if(!((px>=148 && px<=172) && (py>=168 && py<=175)))
if(!((px>=148 && px<=172) && (py>=98 && py<=105)))
if(!((px>=158 && px<=168) && (py>=148 && py<=155)))
if(!((px>=158 && px<=168) && (py>=138 && py<=145)))
if(!((px>=148 && px<=172) && (py>=118 && py<=125)))
if(!((px>=138 && px<=148) && (py>=108 && py<=115)))
if(!((px>=158 && px<=172) && (py>=108 && py<=115)))
if(!((px>=162 && px<=172) && (py>=108 && py<=115)))
if(!((px>=158 && px<=180) && (py>=88 && py<=95)))
if(!((px>=132 && px<=142) && (py>=118 && py<=125)))
if(!((px>=122 && px<=132) && (py>=108 && py<=115)))
if(!((px>=108 && px<=142) && (py>=98 && py<=105)))
if(!((px>=128 && px<=152) && (py>=88 && py<=95)))
if(!((px>=118 && px<=152) && (py>=78 && py<=85)))
if(!((px>=92 && px<=122) && (py>=88 && py<=95)))
if(!((px>=92 && px<=102) && (py>=98 && py<=105)))
if(!((px>=118 && px<=152) && (py>=28 && py<=35)))
if(!((px>=88 && px<=92) && (py>=88 && py<=121)))
if(!((px>=58 && px<=62) && (py>=128 && py<=155)))
if(!((px>=108 && px<=112) && (py>=112 && py<=131)))
if(!((px>=58 && px<=112) && (py>=128 && py<=135)))
if(!((px>=98 && px<=102) && (py>=112 && py<=121)))
if(!((px>=38 && px<=42) && (py>=102 && py<=115)))
if(!((px>=32 && px<=42) && (py>=108 && py<=115)))
if(!((px>=12 && px<=28) && (py>=128 && py<=135)))
if(!((px>=28 && px<=32) && (py>=108 && py<=135)))
if(!((px>=8 && px<=12) && (py>=122 && py<=135)))
if(!((px>=0 && px<=22) && (py>=118 && py<=125)))
if(!((px>=18 && px<=22) && (py>=112 && py<=125)))
if(!((px>=12 && px<=22) && (py>=108 && py<=115)))
if(!((px>=8 && px<=12) && (py>=88 && py<=115)))
if(!((px>=98 && px<=112) && (py>=108 && py<=115)))
if(!((px>=38 && px<=102) && (py>=118 && py<=125)))
if(!((px>=48 && px<=52) && (py>=112 && py<=118)))
if(!((px>=48 && px<=78) && (py>=108 && py<=115)))
if(!((px>=82 && px<=102) && (py>=78 && py<=85)))
if(!((px>=78 && px<=82) && (py>=68 && py<=115)))
if(!((px>=72 && px<=78) && (py>=98 && py<=105)))
if(!((px>=68 && px<=72) && (py>=72 && py<=105)))
if(!((px>=62 && px<=72) && (py>=68 && py<=75)))
if(!((px>=28 && px<=32) && (py>=122 && py<=155)))
```

```
if(!((px>=78 && px<=82) && (py>=28 && py<=53)))
if(!((px>=68 && px<=72) && (py>=28 && py<=43)))
if(!((px>=32 && px<=72) && (py>=38 && py<=45)))
if(!((px>=28 && px<=32) && (py>=18 && py<=45)))
if(!((px>=22 && px<=32) && (py>=18 && py<=25)))
if(!((px>=18 && px<=22) && (py>=18 && py<=35)))
if(!((px>=78 && px<=102) && (py>=28 && py<=35)))
if(!((px>=88 && px<=92) && (py>=12 && py<=31)))
if(!((px>=118 && px<=122) && (py>=8 && py<=35)))
if(!((px>=158 && px<=162) && (py>=22 && py<=35)))
if(!((px>=168 && px<=172) && (py>=8 && py<=25)))
if(!((px>=132 && px<=172) && (py>=18 && py<=25)))
if(!((px>=128 && px<=132) && (py>=8 && py<=35)))
if(!((px>=142 && px<=162) && (py>=8 && py<=15)))
if(!((px>=138 && px<=142) && (py>=0 && py<=15)))
if(!((px>=88 && px<=122) && (py>=8 && py<=15)))
if(!((px>=48 && px<=82) && (py>=8 && py<=15)))
if(!((px>=62 && px<=82) && (py>=18 && py<=25)))
if(!((px>=8 && px<=12) && (py>=8 && py<=21)))
if(!((px>=0 && px<=12) && (py>=18 && py<=25)))
if(!((px>=58 && px<=62) && (py>=8 && py<=85)))
if(!((px>=48 && px<=52) && (py>=18 && py<=31)))
if(!((px>=42 && px<=52) && (py>=28 && py<=35)))
if(!((px>=38 && px<=42) && (py>=0 && py<=35)))
if(!((px>=18 && px<=42) && (py>=8 && py<=15)))
py=py-5;
glutPostRedisplay();
break;
```

case GLUT_KEY_LEFT:

```
    flag=0;
    if(px>0)
    if(!((px>=8 && px<=15) && (py>=148 && py<=162)))
    if(!((px>=68 && px<=75) && (py>=158 && py<=168)))
        if(!((px>=138 && px<=155) && (py>=68 && py<=72)))
        if(!((px>=18 && px<=25) && (py>=162 && py<=168)))
        if(!((px>=18 && px<=65) && (py>=58 && py<=62)))
        if(!((px>=48 && px<=55) && (py>=128 && py<=158)))
        if(!((px>=18 && px<=25) && (py>=142 && py<=148)))
        if(!((px>=38 && px<=45) && (py>=122 && py<=142)))
        if(!((px>=72 && px<=85) && (py>=138 && py<=142)))
        if(!((px>=68 && px<=75) && (py>=132 && py<=148)))
        if(!((px>=68 && px<=115) && (py>=148 && py<=152)))
        if(!((px>=138 && px<=145) && (py>=138 && py<=162)))
        if(!((px>=18 && px<=145) && (py>=168 && py<=172)))
        if(!((px>=128 && px<=135) && (py>=132 && py<=172)))
        if(!((px>=128 && px<=165) && (py>=128 && py<=132)))
        if(!((px>=68 && px<=75) && (py>=158 && py<=172)))
        if(!((px>=78 && px<=125) && (py>=158 && py<=162)))
    if(!((px>=98 && px<=105) && (py>=162 && py<=172)))
```

```
if(!((px>=78 && px<=105) && (py>=168 && py<=172)))
if(!((px>=108 && px<=145) && (py>=168 && py<=172)))
if(!((px>=148 && px<=180) && (py>=168 && py<=172)))
if(!((px>=148 && px<=155) && (py>=132 && py<=162)))
if(!((px>=148 && px<=175) && (py>=158 && py<=162)))
if(!((px>=148 && px<=175) && (py>=168 && py<=172)))
if(!((px>=148 && px<=175) && (py>=98 && py<=102)))
if(!((px>=168 && px<=175) && (py>=122 && py<=162)))
if(!((px>=152 && px<=175) && (py>=118 && py<=122)))
if(!((px>=148 && px<=155) && (py>=98 && py<=122)))
if(!((px>=158 && px<=175) && (py>=108 && py<=112)))
if(!((px>=162 && px<=175) && (py>=108 && py<=112)))
if(!((px>=158 && px<=165) && (py>=92 && py<=112)))
if(!((px>=158 && px<=180) && (py>=88 && py<=92)))
if(!((px>=132 && px<=145) && (py>=118 && py<=122)))
if(!((px>=128 && px<=135) && (py>=112 && py<=122)))
if(!((px>=122 && px<=135) && (py>=108 && py<=112)))
if(!((px>=118 && px<=125) && (py>=102 && py<=162)))
if(!((px>=108 && px<=145) && (py>=98 && py<=102)))
if(!((px>=128 && px<=135) && (py>=92 && py<=98)))
if(!((px>=128 && px<=155) && (py>=88 && py<=92)))
if(!((px>=148 && px<=155) && (py>=82 && py<=88)))
if(!((px>=118 && px<=155) && (py>=78 && py<=82)))
if(!((px>=92 && px<=125) && (py>=88 && py<=92)))
if(!((px>=92 && px<=105) && (py>=98 && py<=102)))
if(!((px>=118 && px<=125) && (py>=78 && py<=88)))
if(!((px>=88 && px<=95) && (py>=88 && py<=118)))
if(!((px>=58 && px<=65) && (py>=128 && py<=152)))
if(!((px>=108 && px<=115) && (py>=112 && py<=128)))
if(!((px>=58 && px<=115) && (py>=128 && py<=132)))
if(!((px>=138 && px<=145) && (py>=0 && py<=12)))
if(!((px>=88 && px<=125) && (py>=8 && py<=12)))
if(!((px>=48 && px<=85) && (py>=8 && py<=12)))
if(!((px>=62 && px<=85) && (py>=18 && py<=22)))
if(!((px>=8 && px<=15) && (py>=8 && py<=18)))
if(!((px>=0 && px<=15) && (py>=18 && py<=22)))
if(!((px>=58 && px<=65) && (py>=8 && py<=82)))
if(!((px>=48 && px<=55) && (py>=18 && py<=28)))
if(!((px>=42 && px<=55) && (py>=28 && py<=32)))
if(!((px>=38 && px<=45) && (py>=0 && py<=32)))
px=px-5;
glutPostRedisplay();
break;
case GLUT_KEY_RIGHT:
    flag=0;
    if(px<175)
    if(!((px>=115 && px<=122) && (py>=98 && py<=162)))
    if(!((px>=5 && px<=12) && (py>=148 && py<=162)))
    if(!((px>=65 && px<=72) && (py>=158 && py<=168)))
    if(!((px>=15 && px<=22) && (py>=162 && py<=168)))
```

```
if(!((px>=45 && px<=52) && (py>=128 && py<=158)))
if(!((px>=15 && px<=22) && (py>=138 && py<=152)))
if(!((px>=35 && px<=42) && (py>=122 && py<=142)))
if(!((px>=65 && px<=72) && (py>=132 && py<=148)))
if(!((px>=65 && px<=112) && (py>=148 && py<=152)))
if(!((px>=85 && px<=118) && (py>=138 && py<=142)))
if(!((px>=135 && px<=142) && (py>=138 && py<=162)))
if(!((px>=105 && px<=142) && (py>=168 && py<=172)))
if(!((px>=125 && px<=132) && (py>=132 && py<=172)))
if(!((px>=125 && px<=162) && (py>=128 && py<=132)))
if(!((px>=65 && px<=72) && (py>=158 && py<=172)))
if(!((px>=75 && px<=122) && (py>=158 && py<=162)))
if(!((px>=95 && px<=102) && (py>=162 && py<=172)))
if(!((px>=75 && px<=102) && (py>=168 && py<=172)))
if(!((px>=145 && px<=180) && (py>=168 && py<=172)))
if(!((px>=145 && px<=152) && (py>=132 && py<=162)))
if(!((px>=145 && px<=172) && (py>=158 && py<=162)))
if(!((px>=145 && px<=172) && (py>=168 && py<=172)))
if(!((px>=145 && px<=172) && (py>=98 && py<=102)))
if(!((px>=155 && px<=168) && (py>=148 && py<=152)))
if(!((px>=155 && px<=168) && (py>=138 && py<=142)))
if(!((px>=165 && px<=172) && (py>=122 && py<=162)))
if(!((px>=149 && px<=172) && (py>=118 && py<=122)))
if(!((px>=135 && px<=148) && (py>=108 && py<=112)))
if(!((px>=145 && px<=152) && (py>=98 && py<=122)))
if(!((px>=95 && px<=102) && (py>=112 && py<=118)))
if(!((px>=35 && px<=42) && (py>=102 && py<=108)))
if(!((px>=29 && px<=42) && (py>=108 && py<=112)))
if(!((px>=9 && px<=28) && (py>=128 && py<=132)))
if(!((px>=25 && px<=32) && (py>=108 && py<=132)))
if(!((px>=5 && px<=12) && (py>=122 && py<=132)))
if(!((px>=-3 && px<=22) && (py>=118 && py<=122)))
if(!((px>=15 && px<=22) && (py>=112 && py<=122)))
if(!((px>=9 && px<=22) && (py>=108 && py<=112)))
if(!((px>=5 && px<=12) && (py>=88 && py<=112)))
if(!((px>=95 && px<=112) && (py>=108 && py<=112)))
if(!((px>=35 && px<=102) && (py>=118 && py<=122)))
if(!((px>=45 && px<=52) && (py>=112 && py<=118)))
if(!((px>=45 && px<=78) && (py>=108 && py<=112)))
if(!((px>=75 && px<=82) && (py>=68 && py<=112)))
if(!((px>=65 && px<=72) && (py>=72 && py<=102)))
if(!((px>=25 && px<=32) && (py>=108 && py<=132)))
if(!((px>=85 && px<=92) && (py>=62 && py<=72)))
if(!((px>=35 && px<=58) && (py>=78 && py<=82)))
if(!((px>=65 && px<=72) && (py>=48 && py<=62)))
if(!((px>=5 && px<=12) && (py>=68 && py<=82)))
if(!((px>=25 && px<=52) && (py>=68 && py<=72)))
if(!((px>=25 && px<=32) && (py>=58 && py<=98)))
if(!((px>=15 && px<=62) && (py>=98 && py<=102)))
if(!((px>=0 && px<=18) && (py>=78 && py<=82)))
```

```
        if(!((px>=15 && px<=22) && (py>=62 && py<=92)))
        if(!((px>=5 && px<=12) && (py>=52 && py<=62)))
        if(!((px>=5 && px<=52) && (py>=48 && py<=52)))
        if(!((px>=15 && px<=22) && (py>=42 && py<=48)))
        if(!((px>=5 && px<=12) && (py>=28 && py<=42)))
        if(!((px>=115 && px<=122) && (py>=62 && py<=72)))
        if(!((px>=109 && px<=122) && (py>=58 && py<=62)))
        if(!((px>=145 && px<=158) && (py>=48 && py<=52)))
        if(!((px>=135 && px<=142) && (py>=42 && py<=72)))
        if(!((px>=125 && px<=132) && (py>=52 && py<=72)))
        if(!((px>=115 && px<=122) && (py>=38 && py<=52)))
        if(!((px>=115 && px<=158) && (py>=38 && py<=42)))
        if(!((px>=155 && px<=162) && (py>=38 && py<=78)))
        if(!((px>=155 && px<=168) && (py>=78 && py<=82)))
        if(!((px>=165 && px<=172) && (py>=8 && py<=22)))
        if(!((px>=125 && px<=132) && (py>=8 && py<=32)))
        if(!((px>=139 && px<=162) && (py>=8 && py<=12)))
        if(!((px>=135 && px<=142) && (py>=0 && py<=12)))
        if(!((px>=85 && px<=122) && (py>=8 && py<=12)))
        if(!((px>=45 && px<=82) && (py>=8 && py<=12)))
        if(!((px>=5 && px<=12) && (py>=8 && py<=18)))
        if(!((px>=0 && px<=12) && (py>=18 && py<=22)))
        if(!((px>=55 && px<=62) && (py>=8 && py<=82)))
        if(!((px>=45 && px<=52) && (py>=18 && py<=28)))
        if(!((px>=35 && px<=42) && (py>=0 && py<=32)))
        if(!((px>=15 && px<=42) && (py>=8 && py<=12)))
        px=px+5;;
        glutPostRedisplay();
        break;
    }
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    if(df==10)
        frontscreen();
    else if(df==0)
        startscreen();
    else if(df==1)
    {
        output(-21,172,"---->");
        output(-21,163,"<----");
        glColor3f(0.0,0.0,1.0);
        output(185,160,"TIME REMAINING : ");
        drawstring(190,130,"HURRY UP",GLUT_BITMAP_HELVETICA_18);
        glColor3f(1,0,0);
        drawstring(190,140,"Time is running out",GLUT_BITMAP_HELVETICA_18);
        sprintf(t,"%d",60-count);
```

```
    output(240,160,t);
    glutPostRedisplay();
        point();
        point1();
        point2();
    glColor3f(1.0,1.0,1.0);
        wall(-4,-4,0,-4,0,162,-4,162);
    wall(-4,178,-4,184,184,184,184,178);
        wall(180,178,184,178,184,-4,180,-4);
        wall(72,158,72,168,68,168,68,158);
        wall(12,162,12,148,8,148,8,162);
        wall(8,162,8,158,0,158,0,162);
        glutPostRedisplay();

}

else if(df==2)
    instructions();
else if(df==3)
{
    exit(1);
}
else if(df==4)
{
    timeover();
}
else if(df==5)
{
    winscreen();
}

    glFlush();
}

void keyboard(unsigned char key,int x,int y)
{
    if(df==10 && key==13)
        df=0;

    else if((df==0 || df==4 || df==5)&& key=='1')
    {
        df=1;
        start=clock();
        glutPostRedisplay();
    }
    else if(df==0 && key=='2')
        df=2;
    else if(df==0 && key=='3')
        df=3;
    else if(key==27)
```



```
{
    df=0;
}
if((key=='0' || key=='1') && (df==4 || df==1))
{
    px=0.0;
    py=175.0;
}
glutPostRedisplay ();
}

void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glPointSize(18.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.0,0.0,0.0,0.0);
}

void myreshape(int w, int h)
{
    glViewport(0,0,w,h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if(w<=h)
        gluOrtho2D(45.0,135.0,-2.0*(GLfloat)h/(GLfloat)w,180.0*(GLfloat)h/(GLfloat)w);
    else
        gluOrtho2D(-45.0*(GLfloat)w/(GLfloat)h,135.0*(GLfloat)w/(GLfloat)h,-2.0,180.0);
    glMatrixMode(GL_MODELVIEW);
    glutPostRedisplay();
}

int main(int argc,char **argv)
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(600,600);
    glutInitWindowPosition(0,0);
    glutCreateWindow("Maze game");
    glutReshapeFunc(myreshape);
    glutDisplayFunc(display);
    glutIdleFunc(idle);
    glutSpecialFunc(SpecialKey);
    glutKeyboardFunc(keyboard);
    myinit();
    glutMainLoop();
    return 0;
}
```