VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

A PROJECT REPORT
ON

PERPLEXITY
A SIMPLE MAZE GAME

BY

PN CHANDRAMOHAN (4SF17CS102)

PUSHPARAJ BR (4SF17CS117)

In the partial fulfillment of the requirement for VI Sem. B. E. (CSE)

COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT

Under the guidance of

Dr.Manjula V
Asst.Prof. Dept. of CSE

Department of Computer Science & Engineering
SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
Adyar, Mangaluru-575007

2019-20

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(Affiliated to Visvesvaraya Technological University, BELAGAVI)
Adyar, Mangaluru – 07

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**PERPLEXITY A SIMPLE MAZE GAME"** is submitted in partial fulfillment for the requirement of VI sem. B. E. (Computer Science & Engineering), "COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT" during the year 2019 – 20 is a result of bonafide work carried out by

P N CHANDRAMOHAN	**4SF17CS102**
PUSHPARAJ B R	**4SF17CS117**

……
Dr.Manjula V
Asst. Prof. Dept. of CSE
SCEM, Mangaluru

……
Dr. Pushpalatha K
HOD, Dept. of CSE
SCEM, Mangaluru

Signature of the Examiners

1. ……..

2. …….

ABSTRACT

“PERPLEXITY” is a 2D Maze game. The game is created using OpenGL where the player tries to solve the maze. The objective of the game is to navigate through the maze and complete the game within a minute then he wins the game either he loses the game.

ACKNOWLEDGEMENT

1.

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible. Whose consistent guidance and encouragement crowned our effort with success. We consider it our privilege to express our gratitude and respect to all those who guided us in the completion of this project.

We express our sincere thanks to our guide **Dr.Manjula V** associate professor department of computer science and engineering for their best guidance and support throughout the tenure of the project.

We are grateful to the **Dr. Pushpalatha K** Head of the Deapartment, Computer science and Engineering for providing us with the right academic atmosphere in the department encouragement and support, which made our entire task appreciable.

We thank our principal **Dr. Rajesha S** for providing us with congenial environment to carry out Project and Academic.

We would also wish to convey our profound to our lab assistants, non-teaching staff, who directly and indirectly helped us in making project successful.

Last but not the least we would like to extend our gratitude to our parents and all those who helped for this project to be cries out.

PN CHANDRAMOHAN

PUSHPARAJ BR

PAGE INDEX

CHAPTER 1
INTRODUCTION

1.1 Computer Graphics

"A picture is worth a thousand words" is a well-known saying and highlights the advantages and benefits of the visual presentation of our data. Computer graphics is concerned with all aspects of producing images using a computer. It concerns with the pictorial synthesis of real or imaginary objects from their computer-based models. Years of research and development were made to achieve the goals in the field of computer graphics. In 1950 the first computer driven display was used to generate only simple pictures. This display made use of a cathode ray tube similar to the one used in television sets. Although the term often refers to the study of three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing. Computer graphics studies the manipulation of visual and geometric information using computational techniques. It focuses on the mathematical and computational foundations of image generation and processing rather than purely aesthetic issues. Computer graphics is often differentiated from the field of visualization, although the two fields have many similarities.[1]

Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with help from specialized graphical hardware and software. It is often abbreviated as CG, though sometimes erroneously referred to as computer-generated imagery (CGI).Some topics in computer graphics include user interface design, sprite graphics, vector graphics, 3D modelling, shaders, GPU design, implicit surface visualization with ray tracing, and computer vision, among others. The overall methodology depends heavily on the underlying sciences of geometry, optics, and physics.

1.2 Open GL

Open GL emerged from Silicon Graphics Inc, in 1992, and has become a widely adopted graphics programming interface (API). It provides the actual drawing tools through a collection of functions that are called within an application. It is easy to install and learn, and its longevity as a standard API is being nurtured and overseen by the OpenGL Architecture Review Board (ARB), an industry consortium responsible for guiding its evolution.[3]

One aspect of OpenGL that suits it so well for use in computer graphics course is its device independence or portability. A student can develop and run a program on any available computer.

OpenGL offers rich and highly usable API for 2D graphics and image manipulation, but its real power emerges with 3D graphics.

Our project is based on the movement of the racing cars in the pit lane. They sort themselves based on the number on the cars. These are drawn and moved across the screen by using multiple GL functions like GL_POLYGON, GL_LINES, GL_POINTS, etc. glutPostRedisplay is used to redraw the elements on the screen when they are translated or rotated.

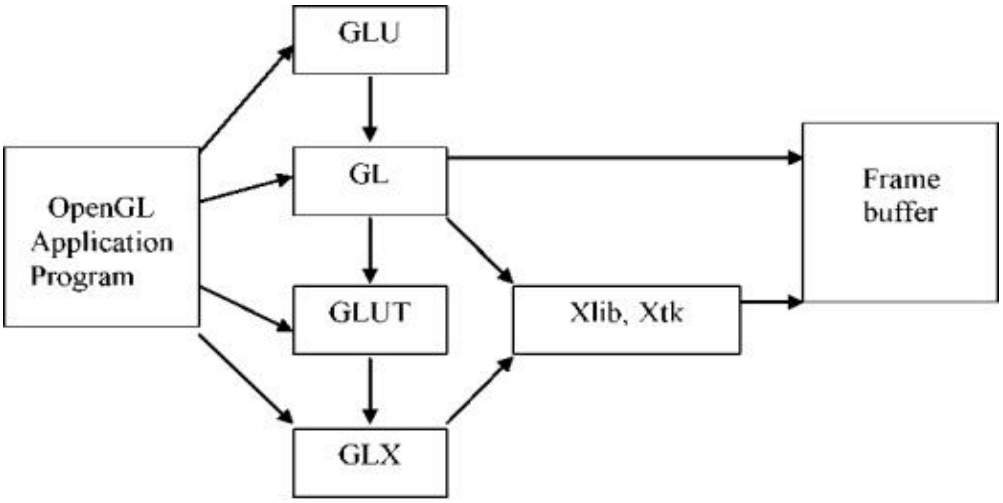


Figure 1.2.1 OpenGL Library Organization

1.3 THE MAZE GAME

- The objectives of this study are summarized below:
- To develop a Open GL software called “PERPLEXITIY”.
- To build the environment for the player to improve his quick thinking/accuracy.
- To build the basic platform of problem solving for the player.
- To progress the thinking ability of the player to solve the game.
- To navigate through the maze and complete the game within a minute then he wins the game either he loses the game.

1.4 Applications of Maze game

- It is basically a visual treat and we have tried our level best Muzzily the people

CHAPTER 2

REQUIREMENT ANALYSIS

Requirement specification is focused specially on functioning of the system, functions to be carried out and performance levels to be obtained and corresponding interfaces to be established.
This report gives the description of the roles of users, the functional overviews of the project, input and output characteristics and also the hardware and software for the project.

2.1 Functional Requirements

For the execution of this project,the graphics has been written in C language and many simple user defined functions are used. The project requires access to OpenGL utility toolkit through the use of the header file "GL /glut.h". This header file, in addition to the usual header files is needed for the working of the project. For running the program, any basic C running compatible version of Code Blocks or Linux (Ubuntu) based platform is sufficient.

2.2 Non Functional Requirements

The software should not accept wrong inputs and produce outputs. This should be meticulously taken care of. It should use memory as less as possible. For this, dynamic memory allocation is preferable to accomplish this task. The project needs to be memory efficient and be able to produce quality output at the same time.

2.3 Hardware Requirements

The minimum/recommended hardware configuration required for developing the proposed software is given below:

- INTEL dual core and above compatible systems.
- 2GB RAM.
- Approximately 200MB free space in the hard disk.
- Hard disk access time must be less than 20milliseconds.

2.4 Software Requirements

- OpenGL libraries.
- C language compiler.
- Windows
- NetBeans

CHAPTER 3

SYSTEM DESIGN

3.1 Flowchart

When we run the program, home window appears. On clicking ‘Enter’ button Main window is opened. In main window list of options like New Game, Instructions & Quit appears. By selecting any of these options we can perform the specified operation in the game.

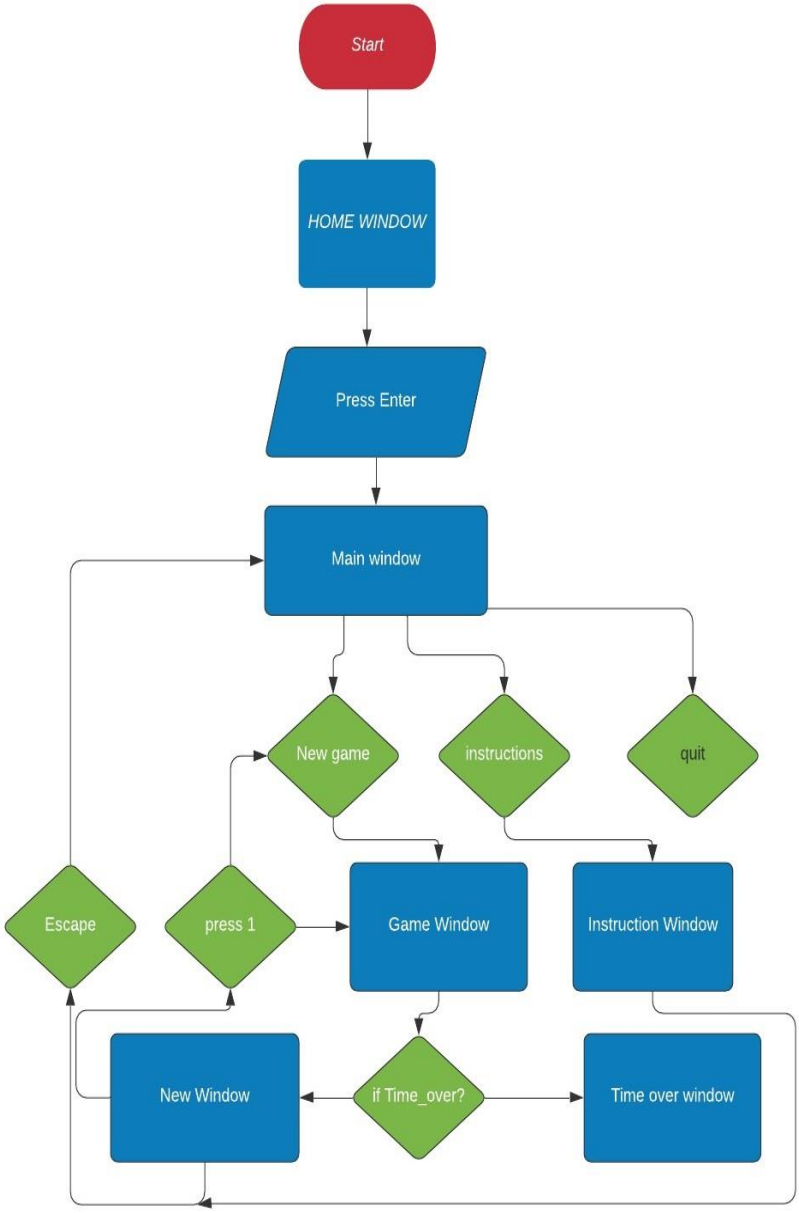


Figure 5.1:Flow Chart

CHAPTER 4
IMPLEMENTATION

4.1 OVERVIEW

This project is a demonstration of "Maze Game". We have taken the help of built in functions present in the header file. To provide functionality to our project we have written sub functions. These functions provide us the efficient way to design the project. In this chapter we are describing the functionality of our project using these functions.

Keyboard interactions are provided where, when a Enter button is pressed, menu displays and we can select options from menu displayed.

4.2 USER INTERFACE

The Project which we have done uses OpenGL functions and is implemented using C. Our Project is to demonstrate MAZE GAME. User can perform operations using keyboard.

Keyboard interaction

 Firstly, after compiling we get a Home Page.

 Then we click the Enter button to display the Main window here we get three options in which user has to specify his choices: New Game: To start the new game

 Instructions: It Guides the user how to play the game.

 Exit: Quits the Game.

 As the player clicks 1 i.e. To open the new game.

 Now in game the player uses the arrow key to complete the game.

 Regardless of a win or a lose the player is redirected to pop-up page, where again he has to specify his choice.

4.3 STRUCTURE

- void point();
- void point1();
- void point2();
- void output(int x,int y,char *string);
- void draw_string(int x,int y,char *string);
- void frontscreen(void);
- void winscreen();
- void startscreen();
- void instructions();
- void timeover();
- void idle();
- void wall();
- void specialkey(int key,int x,int y);
- void display();
- void keyboard(unsigned char key,int x,int y);
- void myinit();
- void myreshape(int w,int h);
- int main(int argc,char** argv);.

4.4 ANALYSIS

FUNCTIONS

A function is a block of code that has a name and it has a property that it is reusable that is it can be executed from as many different points in a c program as required.
The partial code of various function that have been used in the program are:

4.4.1 myinit

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

This function is used to initialize the graphics window.glMatrixMode(GL_PROJECTION), glLoadIdentity() are used to project the output on to the graphics window.

4.4.2 Display

```
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();
//line();
glColor3f(1.0,1.0,1.0);
wall(-4,-4,0,-4,0,162,-4,162);

……

……
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();
}
```

If df==10, i.e., it will call the frontscreen(), else if df==0 then startscreen() is called, Now the game has been started with the timer of 60sec displaying the MAZE to be solved by the player

4.4.3 Wall

```
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();
}
```

This function is used to display the Wall forming the Maze.

.

4.4.4 Point

```
void point() {
 glColor3f(0.0,0.0,1.0);
glBegin(GL_POINTS);
glVertex2f(px,py);
glEnd();
}
```

This function is used to create a color point in the game to identify the start & end point. In the game starting point is green & end point is red, and the player's color point is blue which he uses to play the game

4.4.5 Frontscreen

```
void frontscreen(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(72,30,"(B.E.)",GLUT_BITMAP_HELVETICA_12);
output(70,20,"Lecturer,Dept. of CSE");
glFlush();
}
```

This is the function which helps in opening the Home page of the game. This page is linked to all other pages described before. After clicking enter in this page the Main page is opened.

4.4.6 Idle

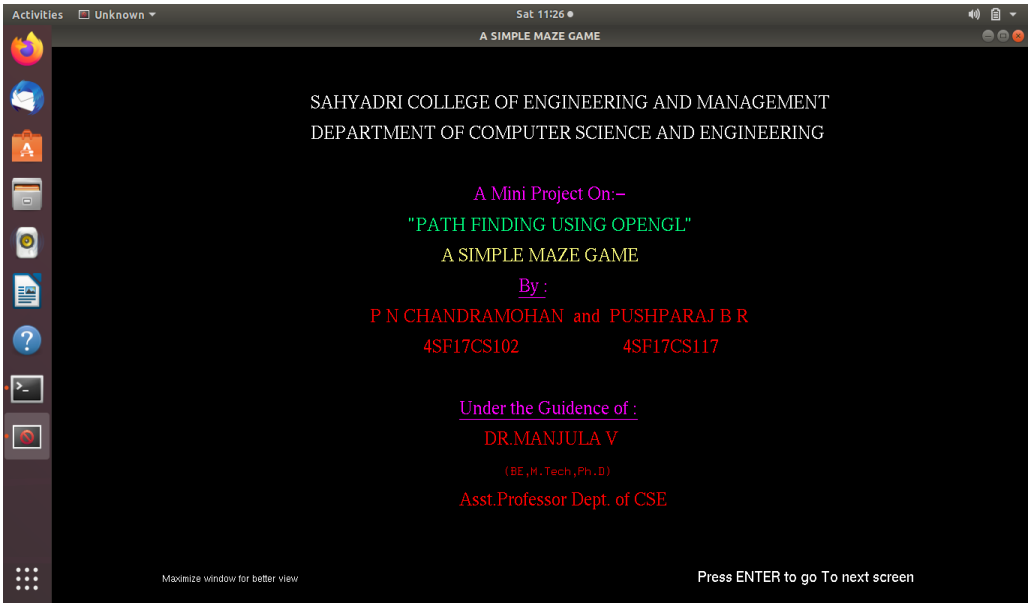
```
void idle()
{
 if(df==1)
{
end=clock();
count=(end-start)/CLOCKS_PER_SEC;
if(count==60)
df=4;
else if((count<60) && ((px>=0 && px<=4) && (py>=162 && py<=168)))
df=5;
}
 glutPostRedisplay();
}
```

This function is the major criteria of this game as it sets a timer for the player which limits the player to finish his game within 60sec else he loses the game.

CHAPTER 5

RESULTS

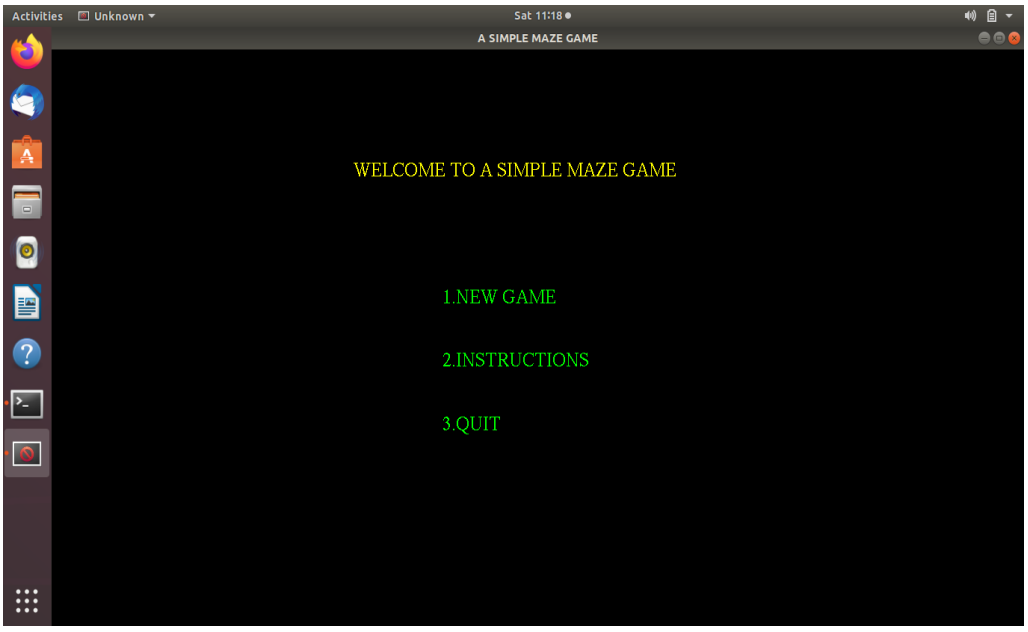
1. After running the program



Home page

The above snapshot shows the screen displayed when the program gets Execute

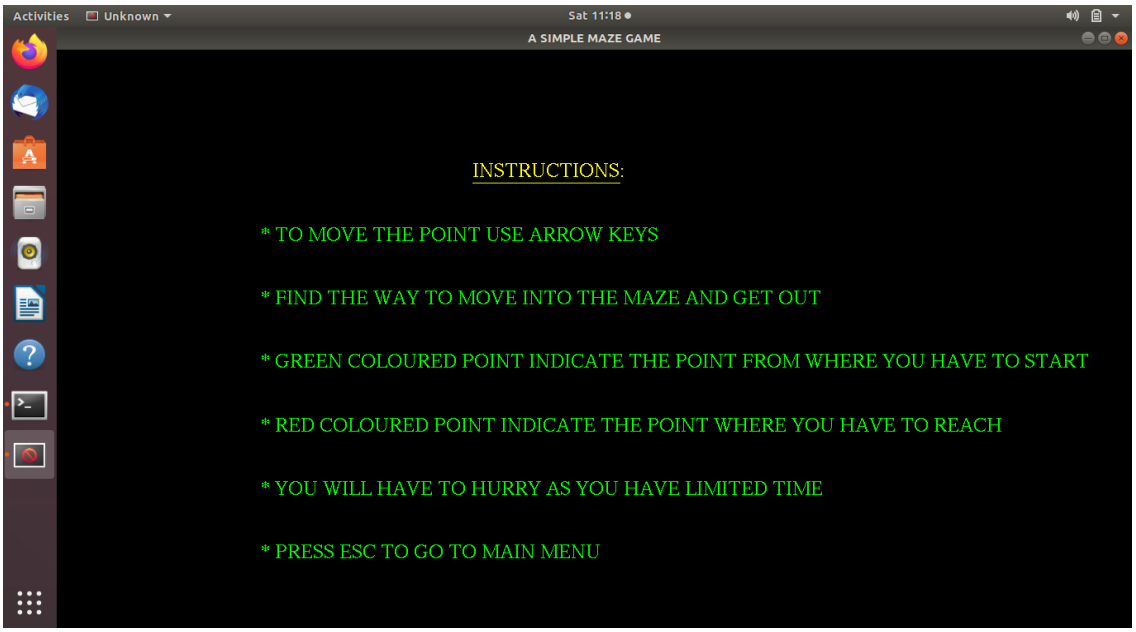
2. Game Menu



Main Window

The above snapshot shows the Game Menu 1st one to start the game , 2nd option guides the player how to use the game & 3rd option Exits the game.

2. Instruction



Instruction page

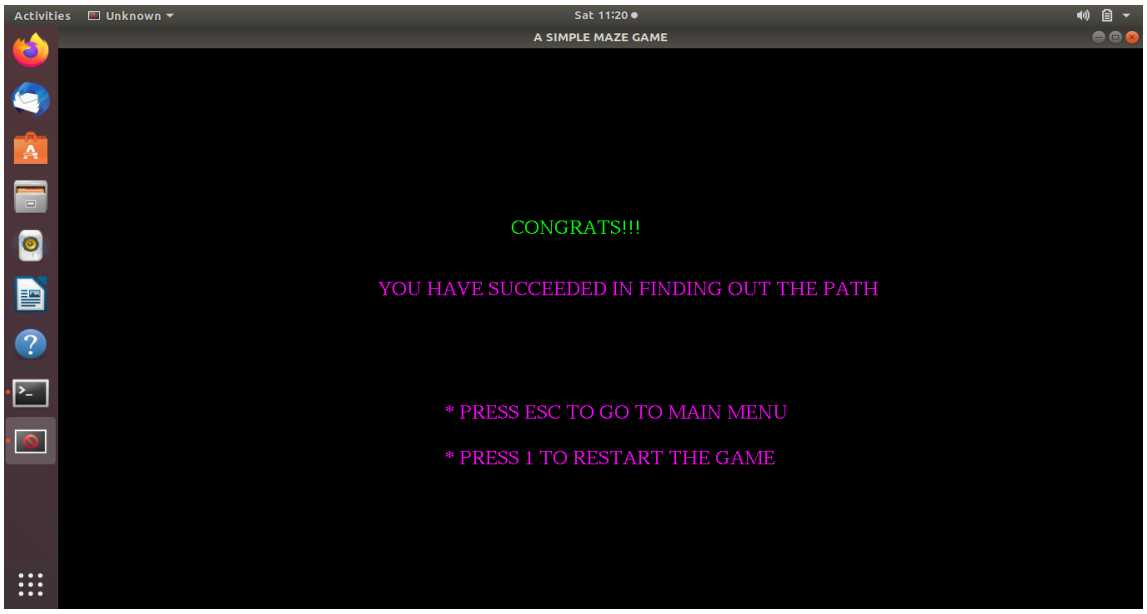
The above snapshots shows the instructions must be follow to play the game effectively

3. The Game

Game Page

The above snapshot our Maze Game with green mark as the starting point & red mark is the end point & the player uses the blue block.

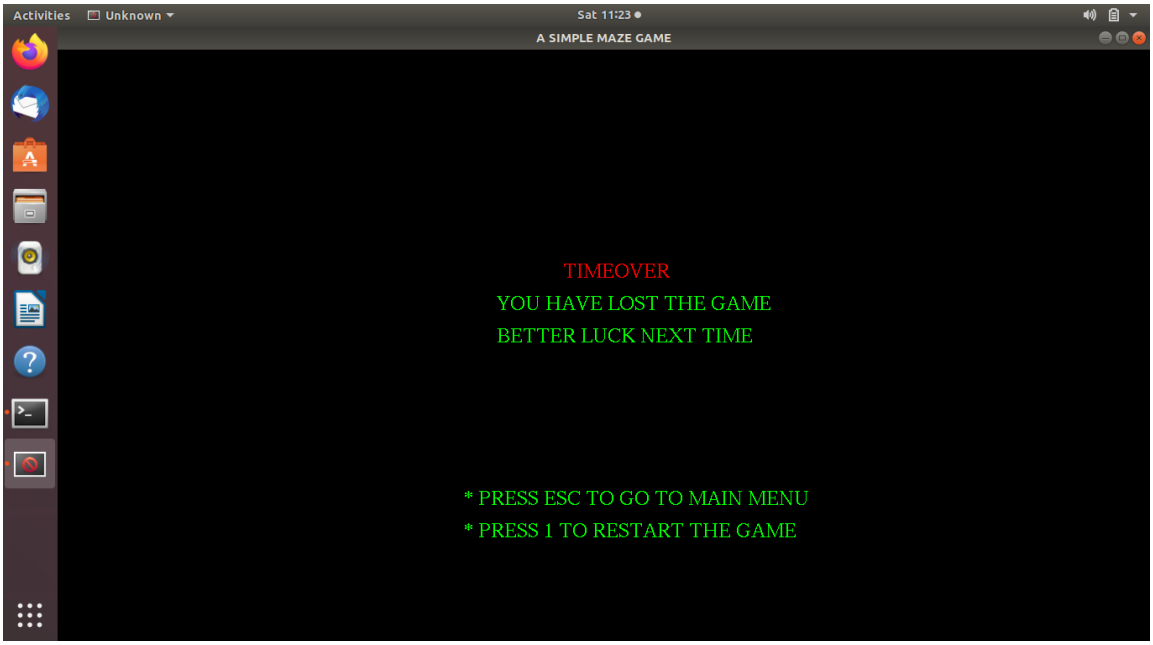
4. Win Screen



Win Page.

The above snapshot shows the win screen as the player has finished the game within 60 Sec

5. Game Over (Lost!!)



Game Over Page

The above snapshot shows the lost screen as the player has not finished the game within 60 Sec

CHAPTER 6

CONCLUSION

PERPLEXITY is designed and implemented using a graphics software system called OpenGL which has became a widely accepted standard for developing graphic application. Using OpenGL functions user can create geometrical objects and can use translation, rotation, scaling with respect to the co-ordinate system. The development of this project has enabled us to improve accuracy, problem solving skills while providing a fun and interactive experience to the player.

.

REFERENCES

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011

2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008

3. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education

4. Xiang, Plastock : Computer Graphics , sham's outline series, 2nd edition, TMG.

List of websites:

1. http://www.opengl.org/ 2. http://www.academictutorials.com/graphics/graphics-flood-fill.asp 3. http://www.glprogramming.com/ 4. https://www.opengl.org/discussion_boards/showthread.php/167379-Making-a-maze- using-arrays