Shri B.V.V.Sangha's

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS)

BAGALKOT-587 103.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

For the academic year

2020-2021

Seminar Report
On

"SURVIVE ON KNOWLEDGE"

Under the Guidance of	Seminar Coordinator	Head of the Department
Prof. Smitha.K	Prof.Savita.S.Hanji	Dr.V.B. Pagi

Submitted by:

Name	USN
Vivek S Halakatti	2BA18CS081

Shri B.V.V.Sangha's

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS)

BAGALKOT-587 103.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

For the academic year

2020-2021

CERTIFICATE

This is to certify that Mr.VIVEK HALAKATTI(2BA18CS081) has satisfactorily completed the seminar report on ***"SURVIVE ON KNOWLEDGE"***, for the fulfillment of their courses in Computer Science and Engineering as prescribed by Basaveshwar Engineering College(Autonomous), Bagalkot during the academic year 2020-2021.

PLACE: Bagalkot

DATE:30/6/2021

Under the Guidance of	Seminar Coordinator	Head of the Department
Prof. Smitha.K	Prof.Savita.S.Hanji	Dr.V.B.Pagi

ACKNOWLEDGEMENT

Our special thanks to principal “**Dr S. S. Injaganeri**“ for being supported towards our department for providing all the facilities needed.

We express our gratitude to “**Prof V.B.Pagi**“ for providing us the infrastructure to carry out the project and all staff members who were directly and indirectly instrument an enabling us to stay committed for the project.

Our heartly thank to “**Prof Smita.K**” for being a source of inspiration and for their constant support in the design, implementation and evaluation of the project on “**SURVIVE ON KNOWLEDGE**”.

It would be our most pleasure to express our warm thanks to their cooperation without which we might not be able to accomplish this project. Lastly I would like to thank my project partners for the dedication and hard work through the process and their valuable ideas and enthusiasm.

ABSTRACT

We use a particular graphics software system. OpenGL which has become a widely accepted standard for developing graphics applications.
Game is just simulation of real world situation. Since hardware is getting cheaper day by day gaming is taken one step closer to reality. Using edge technology game makers are putting a lot of effort to make the gamers enjoy.
This game is simple in construction and interactive. It uses OpenGL utility library to provide graphics package. In this project we used objects such as helicopter, obstacle. We will illustrate the process of this game.

The last experiment used questionnaires to assess the extent of immersion and presence of the users in the virtual environment. The results indicate that such a setting provides an immersive effect for flying in virtual reality.

INDEX

CHAPTER 1

INTRODUCTION

Graphics provides one of the natural means of communicating with a computer. Graphics has also become a key technology for communicating ideas, data and trends in most areas of commerce, science, engineering and education. So, graphics refers to pictures, sketch of building, flowcharts, control flow diagrams, bar charts, and pie charts. Computer graphics is the creation, manipulation and storage of models and images of picture objects by the aid of computers. The term computer graphics has been used in a broad sense to describe almost everything on computers that is not text sound.

The first version of OpenGL, version 1.0, was released on June 30, 1992 by Mark Segal and Kurt Akeley. Since then, OpenGL has occasionally been extended by releasing a new version of the specification. Such releases define a baseline set of features which all conforming graphics cards must support, and against which new extensions can more easily be written.

Each new version of OpenGL tends to incorporate several extensions which have widespread support among graphics-card vendors, although the details of those extensions may be changed.

CHAPTER 2

Overview of OpenGL

PIONEERS IN GRAPHIC DESIGN

Charles Csuri

Charles Csuri is a pioneer in computer animation and digital fine art and created the first computer art in 1964.Csuri was recognized by Smithsonian as the father of digital art and computer animation, and as a pioneer of computer animation by the Museum of Modern Art (MoMA) and (ACM-SIGGRAPH).

Donald P.Greenberg

Donald P.Greenberg is a leading innovator in computer graphics. Greenberg has authored hundreds of articles and served as a teacher and mentor to many prominent computer graphics artists, animators, and researchers such as Robert L. Cook, Marc Levoy, and Wayne Lytle. Many of his former students have won academy awards for technical achievements and several have won the SIGGRAPH achievement award. Greenberg was the founding director of the NSF center for computer graphics and scientific visualization.

Michael Noll

Noll was one of the first researchers to use a digital computer to create artistic patterns and to formalize the use of random processes in the creation of visual arts. He began creating digital computer art in 1962, making him one of the earliest digital computer artists. In 1965, Noll along with FriederNake and Georg Nees were the first to publicly exhibit the computer art. During April 1965, the Howard Wise Gallery exhibited Noll’s computer art along with random-dot patterns by BelaJulesz.

2.1 OpenGL

OpenGL has its origins in the earlier GL (“Graphics Library”) system was invented by Silicon Graphics Inc. As the means for programming their high-performance specialized graphics workstations.
As time went on, people became interested in porting GL to other kinds of machine, and in 1992 a variation of GL -called OpenGL _was announced. Unlike GL, OpenGL was specifically designed to be platform-independent, so it would work across a whole range of computer hardware not just Silicon Graphics machine.

The combination of OpenGL’s power and portability led to its rapid acceptance as a standard for computer graphics programming. OpenGL itself isn’t a programming language, or a software library. It’s the specification of an Application Programming Interface (API) for computer graphics programming. In other words, OpenGL defines a set of functions for doing computer graphics.

2.2 Portability

We had heard that OpenGL is portable and that our application will be able to run on Linux, Mac and Windows without issues. That it could even be ported to a mobile phone if you restrict your usage to OpenGL ES.
Unfortunately, the opposite is true -OpenGL is far from

portable. Direct X is at least portable across Microsoft products. But lets not turn this into one of those rants, because we really like OpenGL, and We feel so sorry for the poor sods roped into writing graphics drivers. We acknowledge that they have a hard job, which leaves us, unfortunately, with a hard job too.

2.3 OpenGL Basics

Open Graphics Library (OpenGL) is a cross-language (language independent), cross-platform (platform-independent) API for rendering 2D and 3D Vector Graphics(use of polygons to represent image). OpenGL API is designed mostly in hardware.

Design : This API is defined as a set of functions which may be called by the client program. Although functions are similar to those of **C** language but it is language independent.

Development : It is an evolving API and **Khronos Group** regularly releases its new version having some extended feature compare to previous one. GPU vendors may also provide some additional functionality in the form of extension.

Associated Libraries : The earliest version is released with a companion library called OpenGL utility library. But since OpenGL is quite a complex process. So in order to make it easier other library such as OpenGL Utility Toolkit is added which is later superseded by free glut. Later included library were GLEE, GLEW, and gliding.

Implementation : Mesa 3D is an open source implementation of OpenGL. It can do pure software rendering and it may also use hardware acceleration on BSD, Linux, and other platforms by taking advantage of Direct Rendering Infrastructure.

2.4 List of OpenGL Libraries

- OpenGL core library OpenGL32

  GL on most Unix/Linux systems (libGL.a)
- OpenGL Utility Library (GLU)

  Provides functionality in OpenGL core but avoids having to rewrite code

  Will only work with legacy code
- Links with window system

  GLX for X window systems

  WGL for windows
- OpenGL a window
- Get input from mouse and keyboard

2.5 GLUT

Free glut is a free-software/open-source alternative to the OpenGL Utility Toolkit (GLUT) library. GLUT was originally written by Mark Kilgard to support the sample programs in the second edition OpenGL 'RedBook'. Since then, GLUT has been used in a wide variety of practical applications because it is simple, widely available and highly portable.

GLUT (and hence free glut) takes care of all the system-specific chores required for creating windows, initializing OpenGL contexts, and handling input events, to allow for truly portable OpenGL programs. Free glut is released under the X-Consortium license.

The original GLUT library seems to have been abandoned with the most recent version (3.7) dating back to August 1998. Its license does not allow anyone to distribute modified library code. This is really unfortunate, since GLUT is getting old and really needs improvement. Also, GLUT's license is incompatible with some software distributions (e.g., XFree86).

Free glut was originally written by Pawel W. Olszta with contributions from Andreas Umbach and Steve Baker.John F. Fay, John Tsiombikas, and Diederick C. Niehorster are the current maintainers of the free glut project.

2.6 The support libraries: GLU and GLUT

A key feature of the design of OpenGL is the separation of interaction (input and windowing functions) from rendering. OpenGL itself is concerned only with graphics rendering. You can always identify an OpenGL function: all OpenGL Utility Toolkit", or GLUT: GLU provides functions for drawing more complex primitives.

OpenGL, such as curves and surfaces, and also functions to help specify 3D views of seems. All GLU function name starts with “glu”. GLUT provides the facilities for interaction that OpenGL lacks. It provides functions for managing windows on the display screen, and handling input events from the mouse and keyboard.

It provides some rudimentary tools for creating Graphical User Interface (GUIs). It also includes functions for conveniently drawing 3D objects like the platonic solids, and a teapot. All GLUT function names start with “glut”. However, somewhat confusingly, when most people say “OpenGL”, what they really mean is “OpenGL plus GLU plus GLUT”, It’s a slightly lazy terminology, but we’ll use it too. This is a set of functions to create texture mipmaps from a base image, map coordinates between screen and object space, and draw quadric surfaces and NURBS.

CHAPTER 3

LITERATURE SURVEY

Computer graphics started with the display of data on hardcopy plotters and cathode ray tube (CRT) screens soon after the introduction of computers.

Computer graphics today largely interactive, the user controls the contents, structure, and appearance of objects and of displayed images by using input devices, such as keyboard, mouse, or touch-sensitive panel on the screen. Graphics based user interfaces allow millions of new users to control simple, low-cost application programs, such as spreadsheets, word processors, and drawing programs.

OpenGL (Open Graphics Library) 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. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization, and flight simulation. It is also used in video games, where it competes with Direct3D on Microsoft Windows platforms (see Direct3D vs. OpenGL).

OpenGL is managed by the non-profit technology consortium, the Chromos Group. In the 1980s, developing software that could function with a wide range of graphics hardware was a real challenge. By the early 1990s, Silicon Graphics (SGI) was a leader in 3D graphics for workstations.

SGI's competitors (including Sun Microsystems, Hewlett-Packard and IBM) were also able. In addition, SGI had a large number of software customers; by changing to the OpenGL API they planned to keep their customers locked onto SGI (and IBM) hardware for a few years while market support for OpenGL matured to bring to market 3D hardware, supported by extensions made to the PHIGS standard.

In 1992, SGI led the creation of the OpenGL architectural review board (OpenGL ARB), the group of companies that would maintain and expand the OpenGL specification took for years to come.These early computer graphics were Vector graphics, composed of thin lines whereas modern day graphics are Raster based using pixels. The difference between vector graphics and raster graphics can be illustrated with a shipwrecked sailor.

The improvements of our project over the previous one are:-

- ✓ The previous game only required some basic gaming skills but our game requires gaming skills and also knowledge.
- ✓ Previous projects works on Windows OS but we implemented through Ubuntu OS and Codeblocks as our Editor and C as compiling language.
- ✓ It also increases your basic general knowledge.
- ✓ It is very efficient than the previous game because of addition of quiz.

CHAPTER 4

PROBLEM DEFINITON AND DESCRIPTION

❖ PROBLEM DEFINITON

We would like to accomplish a video game, that is Helicopter Game. The aim of our game is to score high, the player's goal is trying to avoid bumping to other obstacles and get high score. In this game, we can control the altitude of helicopter.

❖ PROBLEM DESCRIPTION:-

In the Helicopter Game in C/C++, certain keys have been fixed to play the game. The Left Mouse button, lifts the helicopter up or helps to gain the height where as the Right Mouse button is for landing the helicopter. If no key is pressed, the helicopter loses its height.

In the game, you have to keep the helicopter flying without any collision with the vertical walls or the obstacles in the game. If the helicopter collides with any obstacles, the game is stopped and you will get back to main menu. The more you are able to keep the helicopter flying without collision, the more score you will **gain.**

4.1 SOFTWARE REQUIREMENTS

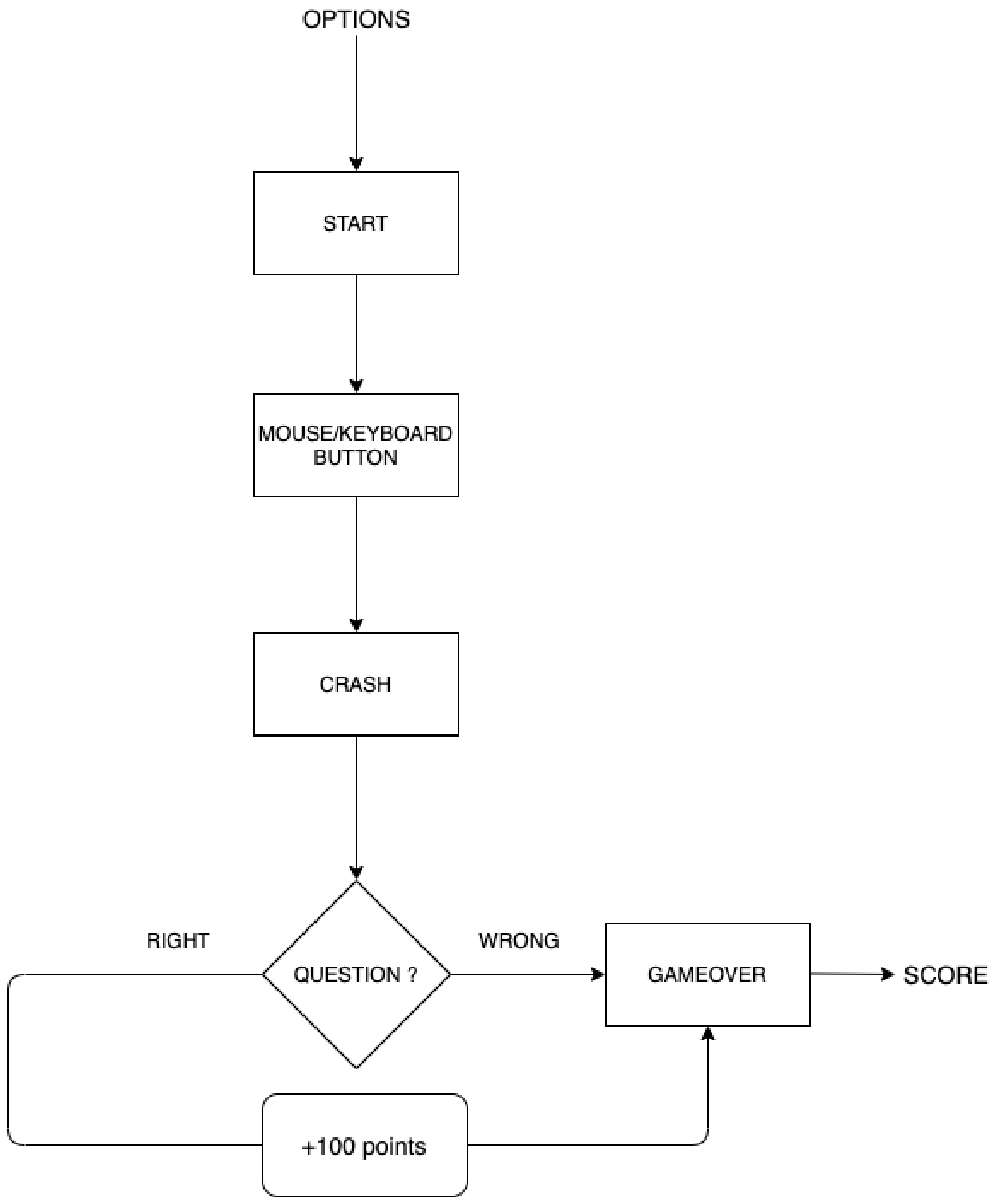
Operating System	**:**	**Ubuntu 20.04 .**
Language	**:**	**C/C++**
Compiler	**:**	**Codeblocks 20.03.**
OpenGL	**:**	**Basic library Files And glut files**

4.2 HARDWARE REQUIREMENTS (MINIMUM)

Processor	**:**	**Pentium PC**
RAM	**:**	**128 MB**
Hard Disk	**:**	**20 GB(approx.)**
Monitor	**:**	**VGA Color Monitor**

CHAPTER 5

IMPLEMENTATION



Working:-

The objective of the game is to fly a helicopter in space with restricted upward and downward motion using either mouse or keyboard, meanwhile walls will move towards player’s copter and player have to avoid a collision between them.
The game will enter into next level as soon as player crosses next certain fixed unit distance and speed of wall will increase by a certain fixed amount each time player enter next level.
The basic feature of the 2D game were analysed to be: -

1. A welcome screen which contains following buttons: -

i) PROFILE: - player can make his/her own profile to save his level and score.

ii) ABOUT: - Display about the game.

iii) CONFIGURATION: Player can either choose the key for up and down or can also use mouse button for the action of copter and can change the color of copter.

iv) START: - Clicking on which game will start with the customized configuration.

2. Calculates distance travelled and level by copter and displays and updates it continuously as score of player while game is on.

3. After a crash, there will be a question popping up, depending on your answer being right or wrong, your score will either be incremented by 100 or remains the same respectively.

4. We can create a timer that counts down to the end of the flight and ends the game. Your timer should not be so short that the game is unplayable or unwinnable, but it should not be so long that there is no challenge to successfully playing the game

5. Create code that puts the helicopter in a specific location at the beginning of the game. Make sure it's starting on the sky, but not from the ground!!

6. **Bonus:** Modify your landing code to make the helicopter start in a random location on the sky.

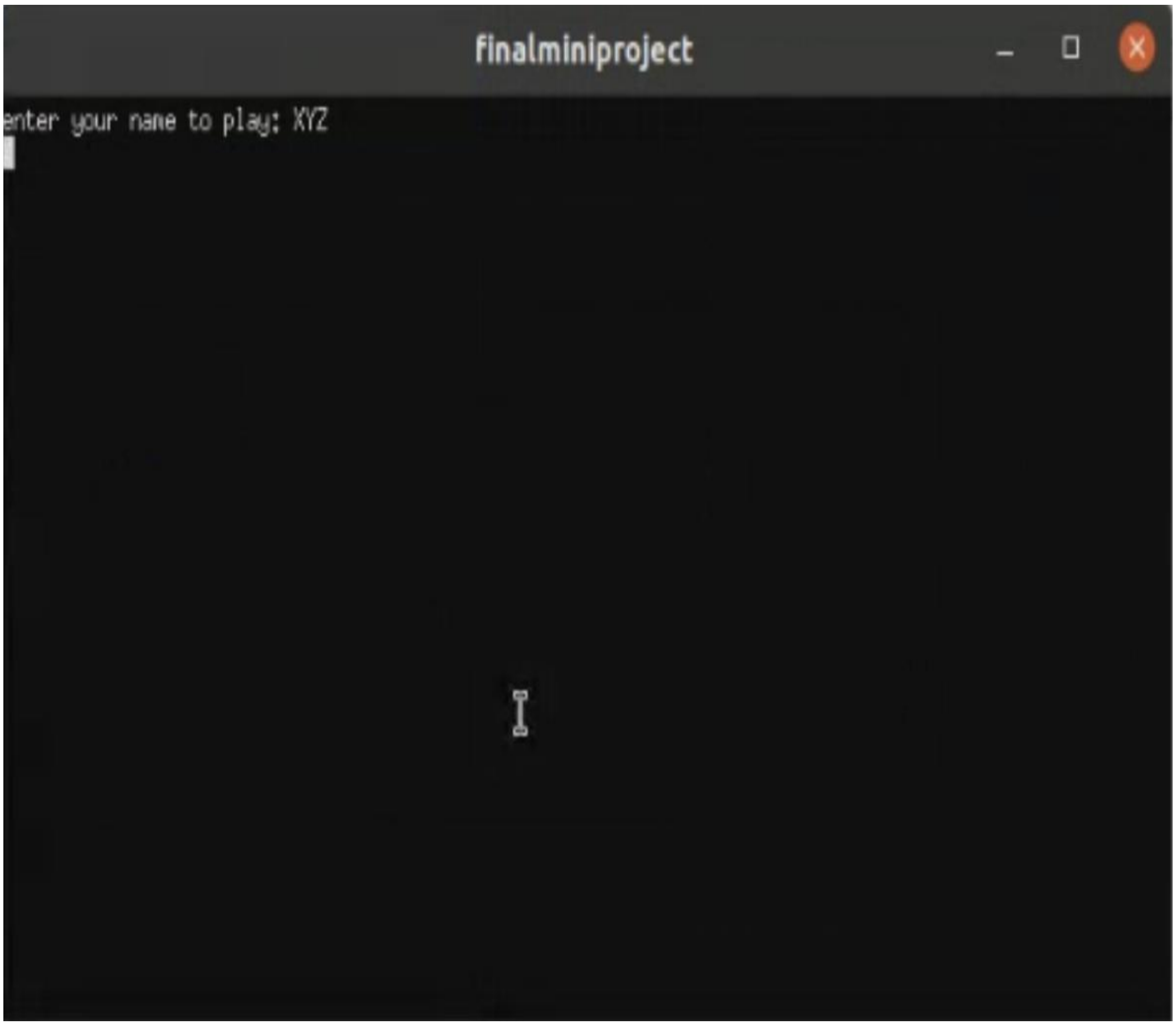
7. Make the helicopter fly up, down, left and right on the screen using four different keys on the keyboard.
   Hint: See if a block in the **Control** section will help.

8. Specify how far the helicopter will move each time one of the keys defined above is pressed. Select a number that makes the helicopter movement appear smooth when the key is pressed and held.
   Hint: See if a block in the **Motion** section will help.

CHAPTER 6

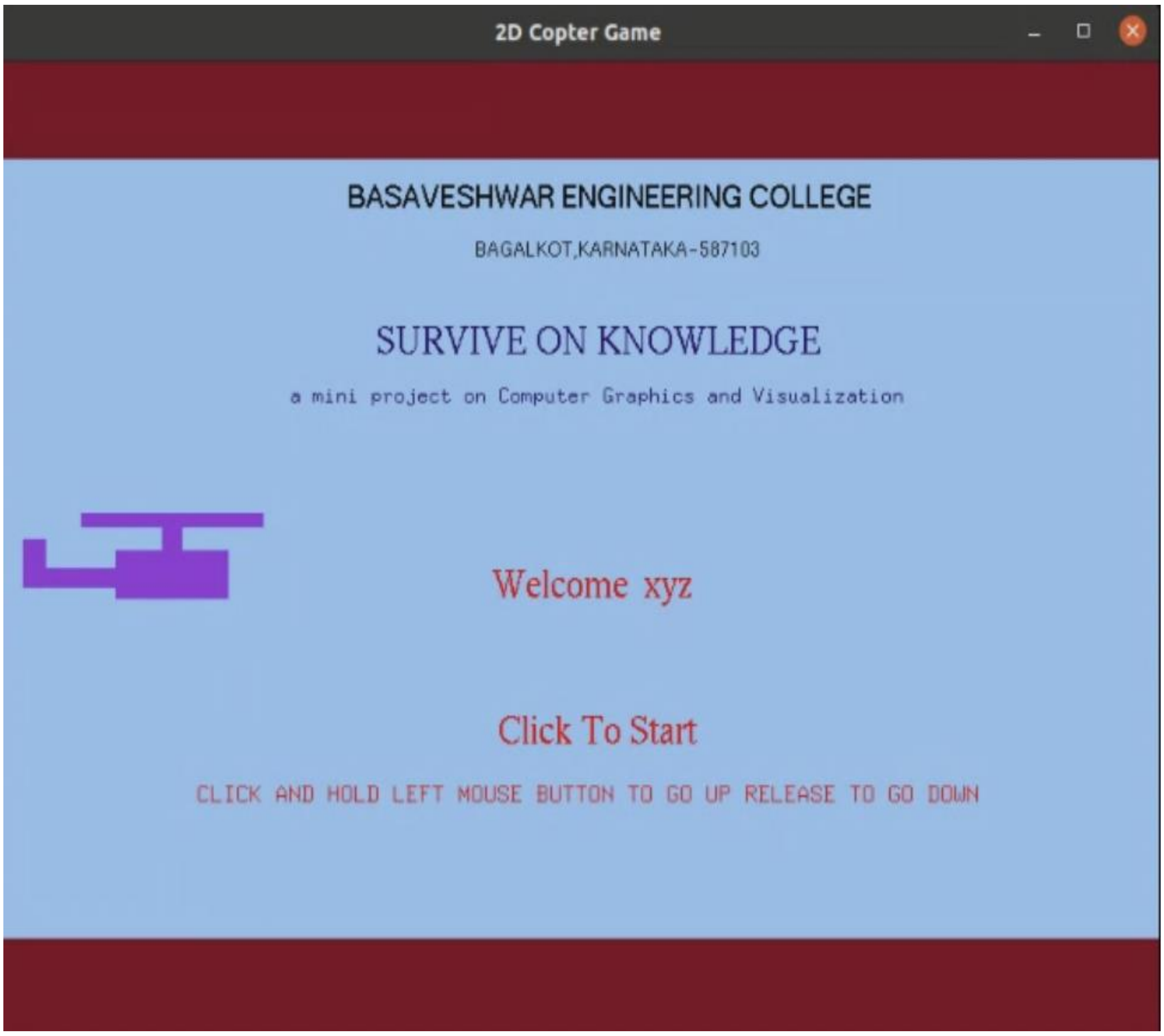
SNAPSHOTS: -

1.WELCOME PAGE



```
enter your name to play: XYZ
```

2.START PAGE



BASAVESHWAR ENGINEERING COLLEGE

BAGALKOT,KARNATAKA-587103

SURVIVE ON KNOWLEDGE

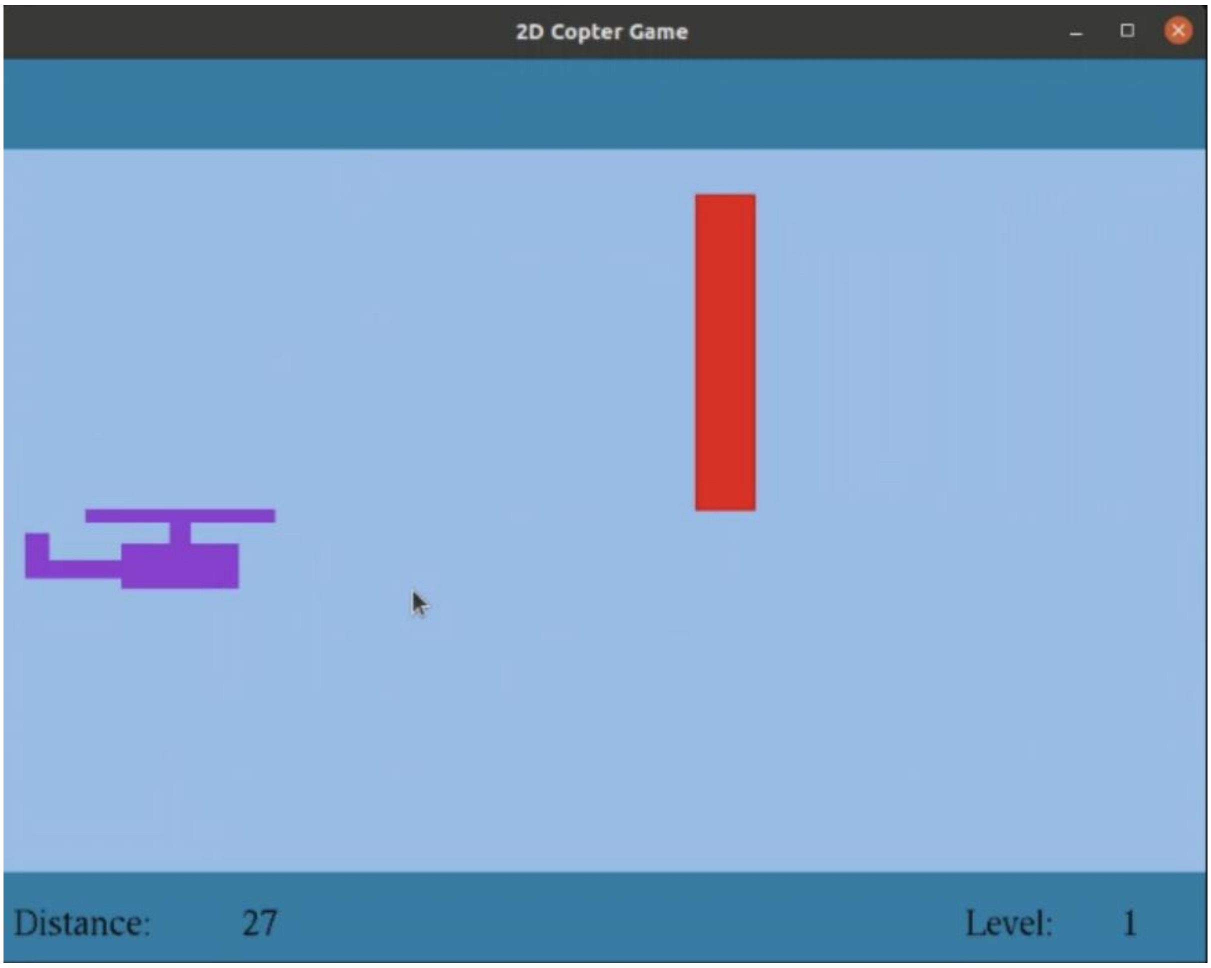
a mini project on Computer Graphics and Visualization

Welcome xyz

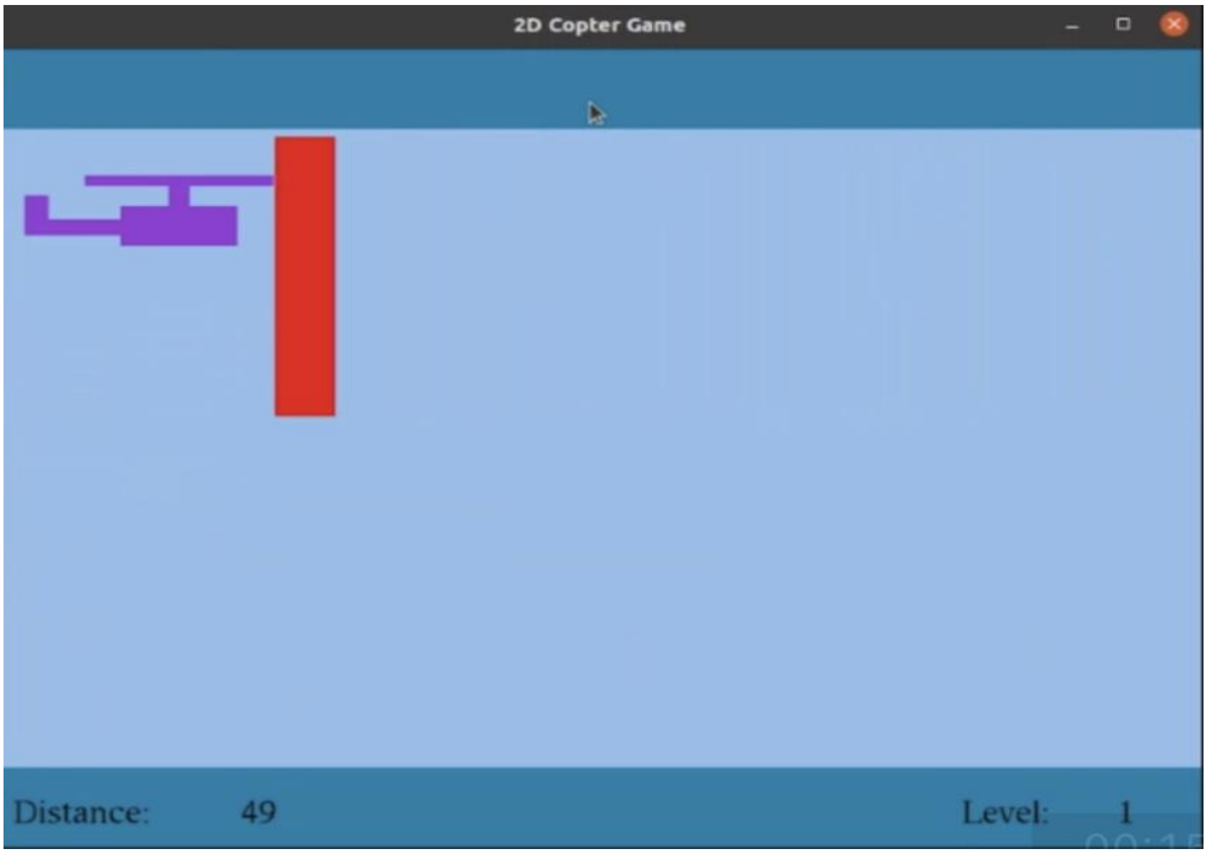
Click To Start

CLICK AND HOLD LEFT MOUSE BUTTON TO GO UP RELEASE TO GO DOWN

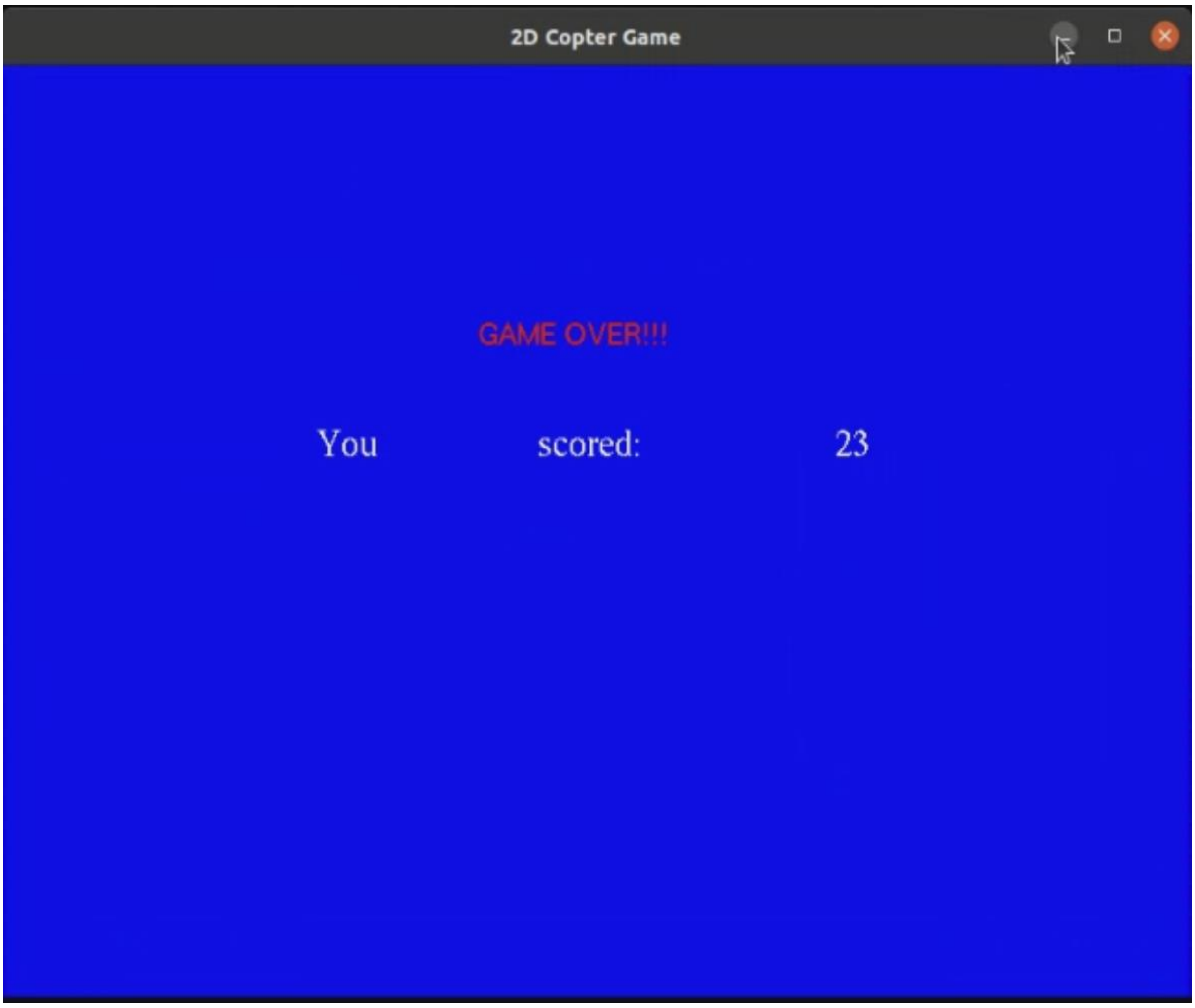
3.DURING PLAY



4.DURING CRASH



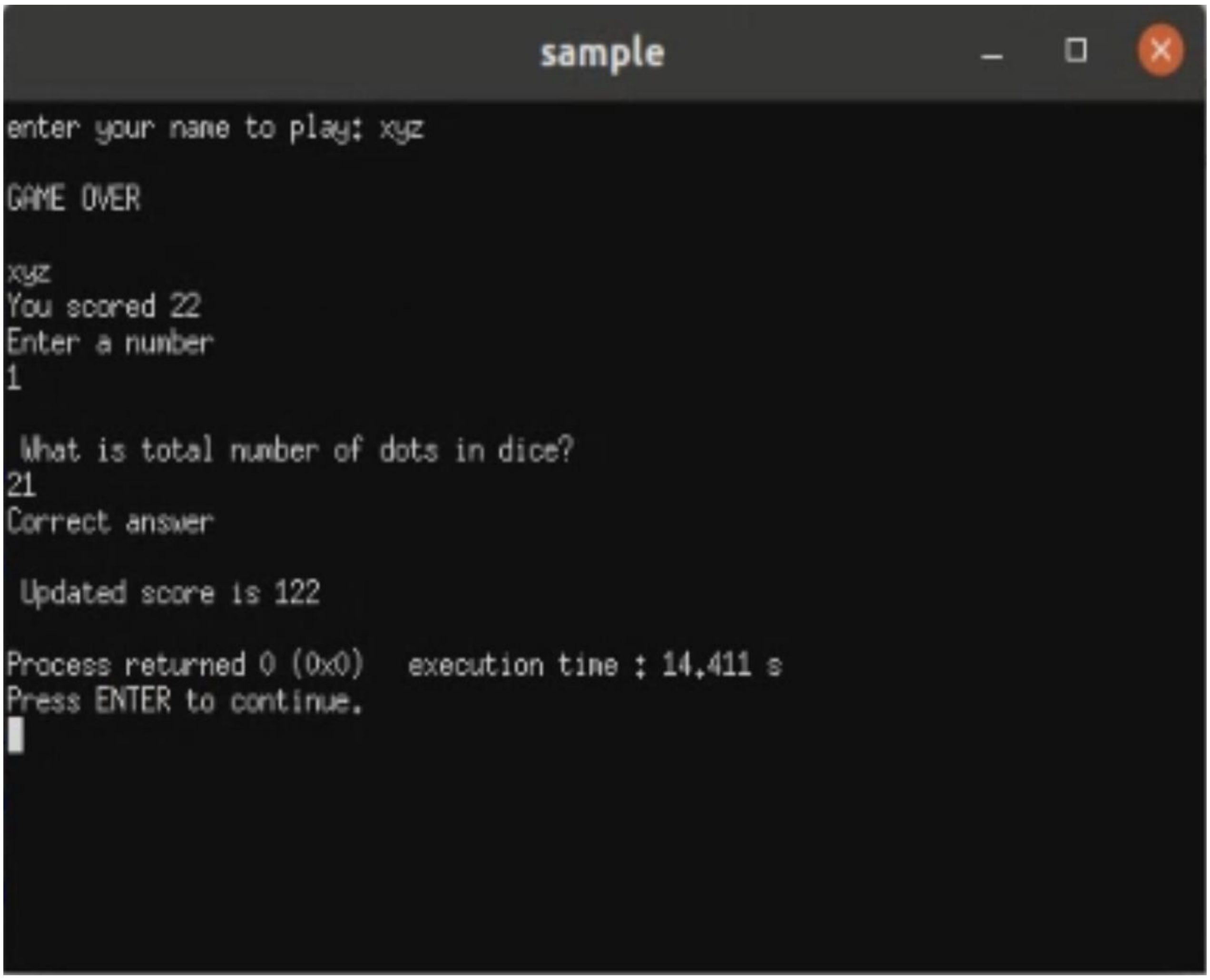
5.GAME OVER



GAME OVER!!!

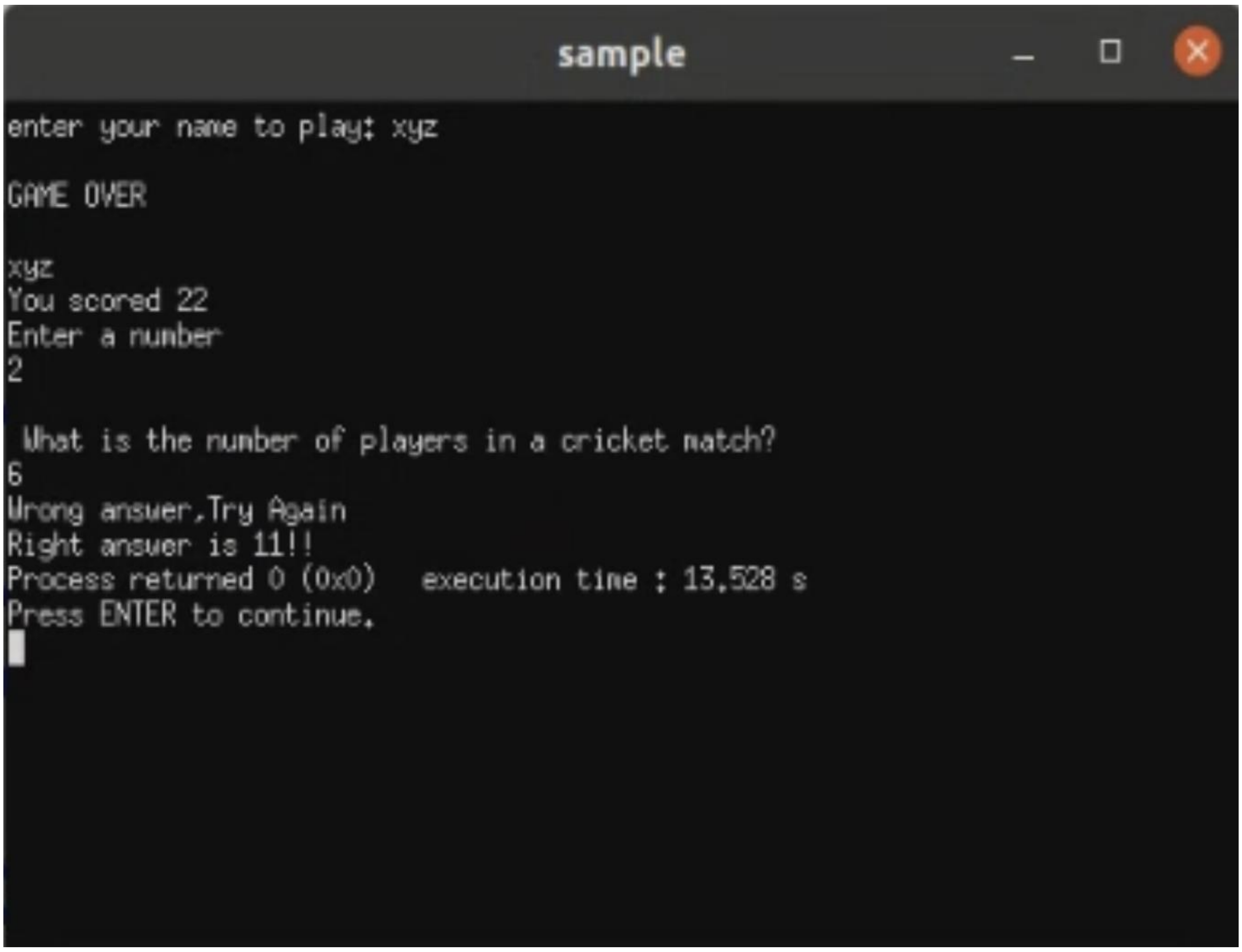
You scored: 23

6. ANSWER WHEN RIGHT



```
enter your name to play: xyz

GAME OVER

xyz
You scored 22
Enter a number
1

 What is total number of dots in dice?
21
Correct answer

 Updated score is 122

Process returned 0 (0x0)   execution time : 14.411 s
Press ENTER to continue.
```

7. ANSWER WHEN WRONG



```
enter your name to play: xyz

GAME OVER

xyz
You scored 22
Enter a number
2

 What is the number of players in a cricket match?
6
Wrong answer,Try Again
Right answer is 11!!
Process returned 0 (0x0)   execution time : 13.528 s
Press ENTER to continue.
```

CHAPTER 7

CONCLUSION

We have attempted to design and implement “2D helicopter”.
OpenGL supports enormous flexibility in the design and the use of OpenGL graphics programs. The presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler.

The project was started with the designing phase in which we figured the requirements needed, the layout design, then comes the detail designing of each function after which, was the testing and debugging stage. We have tried to implement the project making it as user-friendly and error free as possible.

We regret any errors that may have inadvertently crept in.

Future Scope

OpenGL is always been an interesting and most popular cross platform graphics APIs. It will remain among popular graphics APIs even though Krono's developers have introduce next generation low level Vulkan APIs, OpenGL always best APIs for small application where you have reasonable performance (FPS).

Only in case of performance challenges you can move to Vulkan APIs which are much low level and gives better control on GPU. Otherwise, OpenGL is always in demand. It is commonly used to make UI animations more responsive or to handle embedded video or to draw vector graphics – really any visual element you put on the screen is fair game for OpenGL

CHAPTER 8

Bibliography

1. Computer Graphics – Principals And Practice (Foley, Van Dam, Fenier and Hughes) helped me to understand graphics generation algorithms, user interface and dialogue design

2. OpenGL Programming Guide (Addison-Wesley Publishing Company) helped me to get through all OpenGL functions and Commands and understandings of all aspects of them.

3. www.cplusplus.com: - provided references regarding all c++ functions and their uses.

4. www.stackoverflow.com: - help to get rid of all types of error occurred regarding uses of OpenGL functions.

5. www.lighthouse3d.com: - OpenGL tutorial for implementing the OpenGL functions in Source code.

Thank You!