VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI

#### vtulogo

*A Mini Project Report on*

MAP OF BMSIT

Submitted in the partial fulfillment for the requirements for the conferment of degree of

BACHELOR OF ENGINEERING

In

#### **COMPUTER SCIENCE AND ENGINEERING**

#### By

**Name:**

USN:

Under the guidance of

**Mrs. SHANKAR R**

Asst. / Assoc. Professor,

Department of CSE, BMSIT&M.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# B.M.S. INSTITUTE OF TECHNOLOGY & MANAGEMENT

**Yelahanka, BENGALURU-560064**

**2017-2018**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI**

**B.M.S INSTITUTE OF TECHNOLOGY & MANAGEMENT**

**Yelahanka, BENGALURU-560064**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the Mini Project work entitled **“TOWER OF HANOI”** is a bonafide work carried out by **Mr. Aayush Nandkeolyar (1BY15CS002)** in partial fulfillment for the award of **Bachelor of Engineering Degree in Computer-Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2017-18. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The Mini project report has been approved as it satisfies the academic requirements in respect of project work for the B.EDegree.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of the Guide Signature of the HOD**

Mr. SHANKAR R. Dr. THIPPESWAMY G.

**Name of the Examiners Signature with Date**

**1.**

**2**.

**ABSTRACT**

The Tower of Hanoi simulation is an interactive way for the user to understand how the tower of Hanoi algorithm works. It consists of three rods and three disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest on the top thus making a conical shape. The objective of the simulation is to simulate the movement of the entire stack from one rod to another obeying all the rules.

**ACKNOWLEDGEMENT**

I am happy to present this Mini project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals. I would like to express my deep sense of gratitude and indebtedness to each and every one who has helped me to make this project a success.

I heartily thank our **Principal, Dr. MOHAN BABU G N,** **BMS Institute of Technology & Management** for his constant encouragement and inspiration in taking up this Mini project.

I heartily thank our **Head of Department, Dr. Thippeswamy G, Dept. of Computer Science** **and** **Engineering, BMS Institute of Technology & Management** for his constant encouragement and inspiration in taking up this Mini project.

I gracefully thank Project guide**, Mr. SHANKAR R, Asst. Professor, Dept. of** **Computer Science and Engineering,** for his/her encouragement and advice throughout the course of the Project work.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation.

Lastly I thank my parents and friends for their encouragement and support given to me in order to finish this precious work.

By,

**Aayush Nandkeolyar & Ankit Agrawal**

**1BY15CS002 & 1BY15CS010**

**TABLE OF CONTENTS**

**ACKNOWLEDGEMENT**

**ABSTRACT**

1. **Introduction 06**
   1. Motivation 07
   2. Scope 08
   3. Objective 09

**2. Literature Survey**

**3. Software and Hardware Requirements**

3.1 Software Requirements

3.2 Hardware Requirements

**4. Requirements Analysis**

**5. System Design**  24

**6. Implementation 27**

**7. Conclusion**

**1. INTRODUCTION**

**Computer Graphics**

Computer graphics is one of the most exciting and rapidly growing computer fields. It is also an extremely effective medium for communication between man and computer; a human being can understand the information content of a displayed diagram or perspective view much faster than he can understand a table of numbers or text containing the same information. Thus computer graphics is being used more extensively.

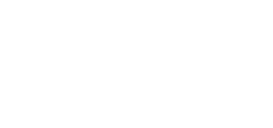
There is a lot of development in hardware and software required to generate images, and nowadays the Cost of hardware and software is dropping rapidly. Due to this, interactive computer graphics is becoming available to more and more people.

Computer graphics started with the display of data on hardcopy plotters and cathode ray tube (CRT) screens soon after the introduction of computers themselves. It has grown to include the creation, storage and manipulation of models and manipulation of models and images of objects. These models come from a diverse and expanding set of fields, and include physical, mathematical, engineering, architectural, and even conceptual structures, natural phenomena, and so on.

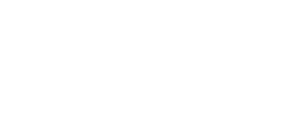
Computer graphics today is largely interactive. The user controls the contents, structure and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch sensitive panel on the screen. The handling of such devices is included in the study of computer graphics, because of the close relationship between the input devices and the display.

Computer Graphics Architecture

Early graphics system used general purpose computers with the standard von Neumann architecture, such computers are characterized by a single processing unit that processes a single instruction at a time. The display in these systems was based on a calligraphic CRT display that included the necessary circuitry to generate the line segment connecting two points. The job of first computer was to run the application program and to compute the endpoints of the line segment in the image. This information had to be sent to the display at a rate high enough to avoid flickers on the display.



HOST



Digital to

Analogy

#### Figure 1.1 Early Graphics System

To reduce the burden of basic systems we use the following support applications.

1.Display processor

2.Pipeline architecture

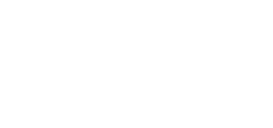
3.Graphics pipeline

4.Vertex processing

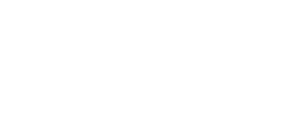
5.Clipping and primitive assembly

6.Rasterization

7.Fragment processing

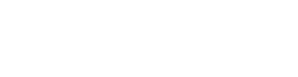


HOST



Display

Processor



Display List

Figure1.2 Display Processor Architecture

**MOTIVATION**

The development of computer graphics has been driven both by the needs of user community and by advances in hardware and software.

The four major areas are:

1. **Display of information**

Classical graphics techniques are used as a medium to convey information among people. In ancient times Greeks were able to convey their architectural ideas graphically even though the relevant mathematics was not developed.

Today the same type of information is generated by architectures mechanical designers and drafts people using computer based drafting system.

2. **Design**

Professionals such as engineering and architecture are concerned with design. Starting with a set of specifications, engineers and architects seek cost effective and esthetic solution that satisfies the specifications. Design is an iterative process.

3. **Simulation and Animation**

Once graphics system evolved to be capable of generating sophisticated images in real time, engineers and researchers began to use them as simulators. Graphical flight simulator as proved to increase safety and reduce training expenses, Use of graphics for animation in television, motion pictures and advertising industry.

**SCOPE**

This mini project under Computer Graphics & Visualization Laboratory is an implementation of a of the very famous tower of Hanoi algorithm using the OpenGL Graphics Library and GLUT Toolkit.

The user can rotate the view in all the three axes i.e., X-axis, Y-axis and Z-axis using the mouse buttons. This gives the user a complete picture of the three stacks moving from all the directions, and provides ease of navigation.

**OBJECTIVE**

The objective of the program is to render a easy to understand simulation of the Tower of Hanoi algorithm and understand its method of flow. The user can view from all the directions rotating the view using the assigned buttons.

The view contains three rods and a total of three disks that have to be rotated

**2. LITERATURE SURVEY**

Computer graphics is graphics created using computers and, more generally, the representation and manipulation of pictorial data by a computer. The development of computer graphics has made computers easier to interact with and better for understanding and interpreting many types of data. Developments in computer graphics have had a profound impact on many types and have revolutionized animation and video game industry.

The term computer graphics includes almost everything on computers that is not text or sound. Today nearly all computers use some graphics and users expect to control their computer through icons and pictures rather than just by typing. Today computers and computer-generated images touch many aspects of our daily life. Computer imagery is found on television, in newspapers, in weather reports, and during surgical procedures.

The project has been developed using Code::Blocks which is a commercial integrated development environment (IDE) with OpenGL (Open Graphics Library). OpenGL is a standard specification to produce 2D and 3D computer graphics. We use, the OpenGL Utility Toolkit called GLUT which is a library of utilities for OpenGL programs.

**2.1 Code::Blocks**

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Features of Code::Blocks are as follows:

* Compilers

Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel

C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including Fortran and D. A plug-in system is included to support other programming languages.

* Code editor

The IDE features syntax highlighting and code folding (through its Scintilla editor component), C++ code completion, class browser, a hex editor and many other utilities. Opened files are organized into tabs. The code editor supports font and font size selection and personalized syntax highlighting colours.

* Debugger

The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

* GUI designer

As of version 13.12 Code::Blocks comes with a GUI designer called wxSmith. It is a derivative port of wxWidgets version 2.9.4.[6] To make a complete wxWidgets application, the appropriate wxWidgets SDK must be installed.

* User migration

Some of Code::Blocks features are targeted at users migrating from other IDE's - these include Dev-C++, Microsoft Visual C++ project import (MSVC 7 & 10), and Dev-C++ Devpak support.

* Project files and build system

Code::Blocks uses a custom build system, which stores its information in XML-based project files. It can optionally use external makefiles, which simplifies interfacing with projects using the GNU or qmake build systems.

**2.2 OpenGL and Glut**

OpenGL (Open Graphics Library) is a standard specification defining a cross-language, [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [API](http://en.wikipedia.org/wiki/Application_programming_interface) for writing applications that produce [2D](http://en.wikipedia.org/wiki/2D_computer_graphics) and [3D computer graphics](http://en.wikipedia.org/wiki/3D_computer_graphics), describing a set of functions and the precise behaviors that they must perform. From this specification, hardware vendors create implementations - libraries of functions created to match the functions stated in the OpenGL specification, making use of hardware acceleration.

GLUT is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL programming. GLUT provides a portable API so you can write a single OpenGL program that works across all PC and workstation OS platforms.

**2.3 Applications of computer graphics**

Nowadays Computer Graphics used in almost all the areas ranges from science, engineering, medicine, business, industry, government, art, entertainment, education and training.

* CG in the field of CAD

Computer Aided Design methods are routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft computers, textiles and many other applications.

* CG in presentation Graphics

Another major application area presentation graphics used to produce illustrations for reports or generate slides. Presentation graphics is commonly used to summarize financial, statistical, mathematical, scientific data for research reports and other types of reports.2D and 3D bar chart to illustrate some mathematical or statistical report.

* CG in computer Art

CG methods are widely used in both fine art and commercial art applications. Artists use a variety of computer methods including special purpose hardware, artist’s paintbrush program (lumena), other pain packages, desktop packages, maths packages, animation packages that provide facility for designing object motion. Ex: cartoons decision is an example of computer art which uses CG.

* Entertainment

Computer graphics methods are now commonly used in making motion pictures, music, videos, games and sounds. Sometimes graphics objects are combined with the actors and live scenes.

* Education and Training

Computer generated models of physical financial, economic system is often used as education aids. For some training application special systems are designed. Ex: specialized system is simulator for practice sessions or training of ship captain, aircraft pilots and traffic control.

* Image Processing

Although the methods used in CG image processing overlap, the 2 areas are concerned with fundamentally different operations. In CG a computer is used to create picture. Image processing on the other hand applies techniques to modify existing pictures such as photo scans, TV scans.

The various methods used in this project are as follows:

* display()

The entire working of the program is graphically displayed on the screen by the contents defined in the function.

* main()

The execution first start with main() function.

* mouse()

The button click on the mouse will display various scenes of celebration.

* keyboard()

Pressing the enter key displays scene1 after displaying the title page.

**3. SOFTWARE AND HARDWARE REQUIREMENTS**

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

**3.1 Software requirements:**

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

OPERATING SYSTEM : Windows 7/ XP/8/10

LANGUAGE : C++

TOOL : Codeblocks

LIBRARY : OpenGL

**3.2 Hardware requirements:**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

PROCESSOR : Intel dual Core i3, i5, i7

RAM : 4 GB or more

HARD DISK : 80 GB or more

**4. REQUIREMENT ANALYSIS**

The big problem with using a higher resolution height map is the memory requirements. To this end there are multiple different solutions that "optimize" the height map data that is stored in memory. Nearby landscape does not need as much details as the landscape far away.

A really good technique for rendering VAST amounts of highly detailed landscape geometry is geometry clipmaps. It's quite complicated though. Another scheme I have used to great success in the past is Geo-Mipmapping. It's very simple and very fast. It produces excellent results, too, if you apply trilinear filtering techniques to your Geo-mipmaps. It is also fairly trivial to write an asynchronous chunk loader for such a landscape to provide good streaming performance of huge landscapes.

The basic feature of the 3D map were analyzed to be: -

1.A welcome screen which contains following buttons: -

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

ABOUT: - display about the game, its version and owners of game. iii) CONFIGURATION: - player can choose the key for up and down action of copter and can change the color of copter. iv) START: - clicking on which game will start with the customized configuration.

2.Calculating distance travelled by copter and displaying and updating it continuously as score of player while game is on.

3.Calculating level and continuously displaying and updating it as game is on.

4.A game over window which will show score and level of player and also a button

“PLAY AGAIN” clicking on which game will start again.

5.If player score manages among top ten scorer, after game over it should prompt for entering his name so it can store and display player’s name among top ten scorer with player’s name in the list.

**5. SYSTEM DESIGN**

Algorithm of display\_function()

// start

{

// Checking collision condition

if ( ( top & bottom check ) ORed ( propeller front check ) ORed ( lower body

check ) ORed ( lower body check ) )

{

display game over window

exit }

else if( welcome\_window\_flag is set to 1 )

{

reset welcome\_window\_flag to zero

display welcome window screen

}

else {

// increase in level by 1 and block\_speed by certain amount if( (score multiple of 50) ANDed level\_flag is set to 1)

{

reset level\_flag to zero increase level by 1 increase block\_speed by a fixed small amount

}

// during playing a level set level\_flag to 1 else if( (score not multiple of 50) ANDed ( level\_flag is not zero) ) {

Set level\_flag to 1

}

// show level and score show level during play increase score by 1

show score

// controlling helicopter movement

//applying translation function about y-axis as there is a change occur on

every

mouse click or key pres

translatef(0.0, dy, 0.0);

draw\_helicopter\_function();

// controlling block movement

// if block move till the last of left window translate back to first position to the right of window

if (block x value become negative)

{

block\_x\_value=50;

// generate y value using rand() function

block\_y\_value= rand() % fix\_amount ;

} else

increase block\_x\_value by an certain fix amount

// drawining bloack

glTranslatef( block\_x\_value, -dy, 0.0); glRectf(x1,y1,x2,y2);

glPopMatrix(); glutSwapBuffers(); glFluash(); }

// end of external if…else statement

}

// end of display\_function

**6. IMPLEMENTATION**

Implementation is the stage where the theoretical design is turned into a working system. It is the most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

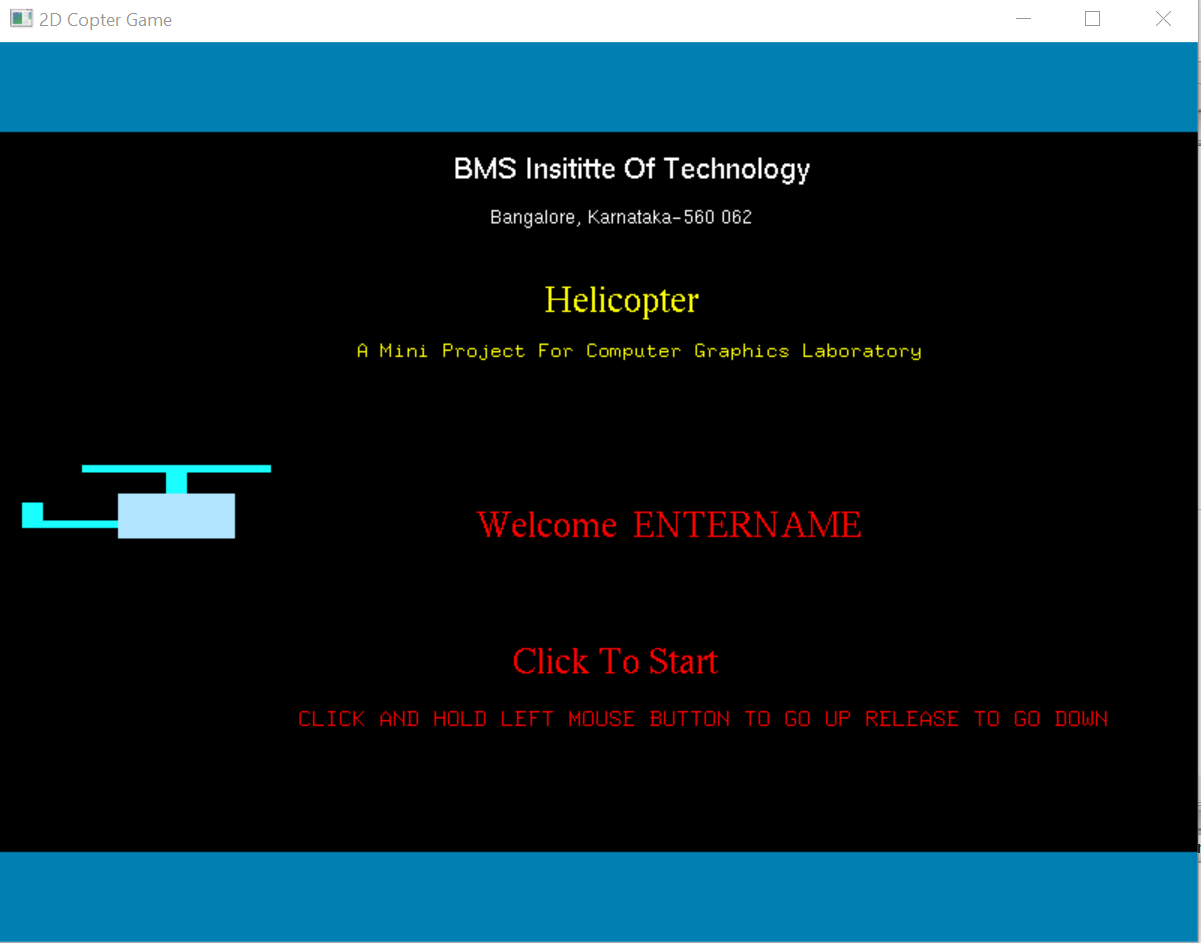
      The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

         It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

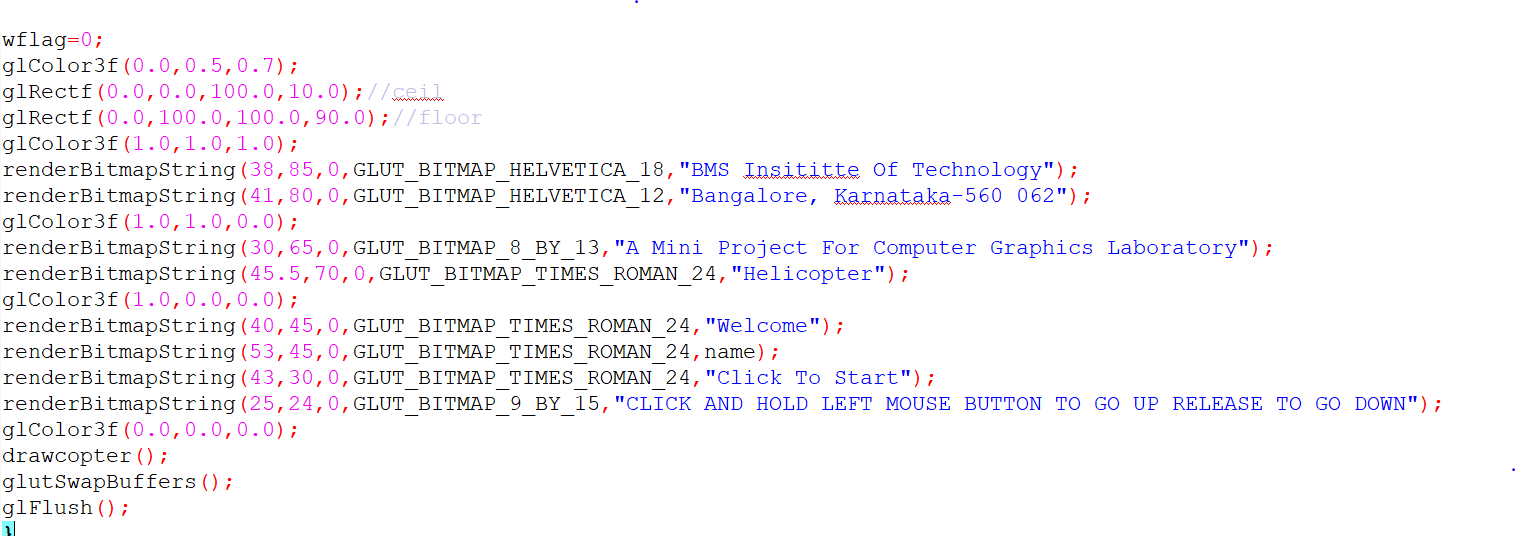
          The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

       The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

* **THE FRONT SCREEN**



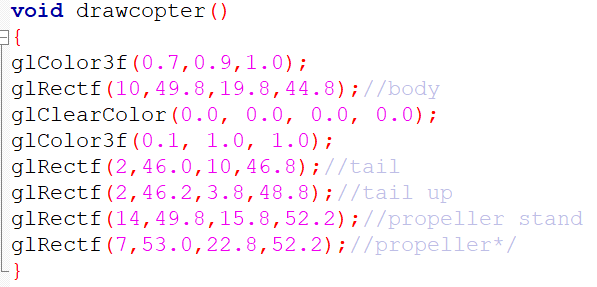
The code used for the above screen is as below:



* **PLAYING WINDOW**

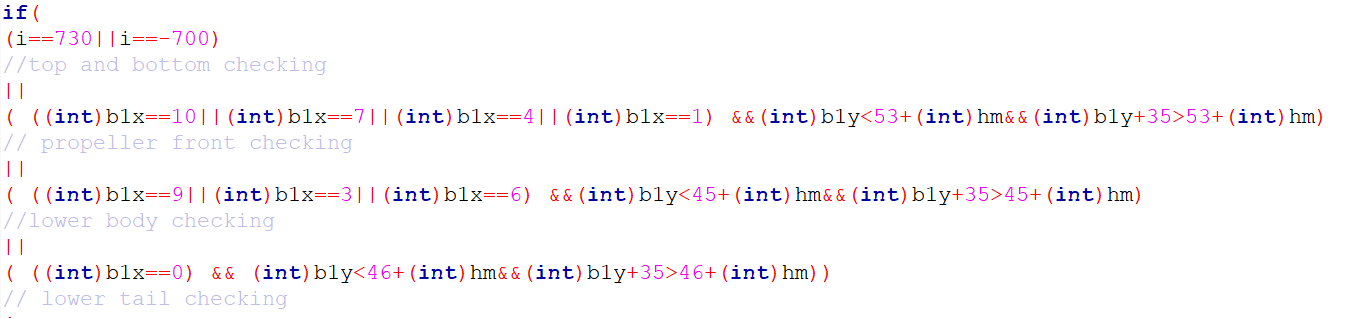


**To draw helicopter :**

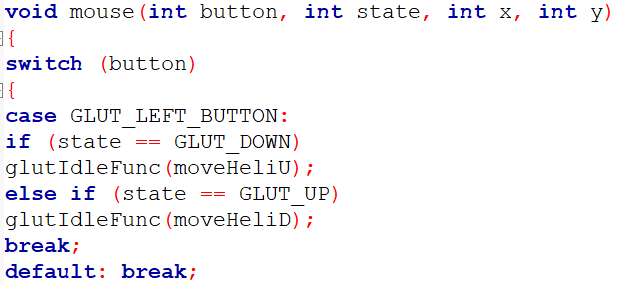


The different parts of the helicopter like tail, propeller, body etc. are made up of individual rectangles of different sizes.

**To check if the copter has touched the bottom or top of the screen/ any of the obstacles:**



**Mouse button click functions:**



moveHeliU – moves the copter up when the mouse button is clicked if the copter is moving down.

moveHeliD – moves the copter down when the mouse button is clicked if the copter is moving up

* **END SCREEN**



**CONCLUSION**

We have attempted to design and implement “3D map of BMSIT&M”. 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 ENHANCEMEMTS**

1. Add the functionality to Zoom-in and Zoom-out using keyboard and mouse which will provide even more details to the users.

2. Increase the level of details on all the objects in the map, making them more lifelike and appealing.

3. Allow the users to click on any structure on the map view the interior of that structure in 3D and view accurate and lifelike details.

**BIBILIOGRAPHY**

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

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

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

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

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