VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI

Mini Project Report on

"STEAM ENGINE SIMULATION"

Submitted in the partial fulfillment for the requirements of Computer Graphics & Visualization Laboratory of 6th semester CSE requirement in the form of the Mini Project work

Submitted By

VEDASHRUTHA D S **USN: 1BY19CS178**

SRINIVAS R **USN: 1BY19CS157**

Under the guidance of

Prof. Muneshwara M S
Assistant Professor
Dept. of CSE

Prof. Chethana C
Assistant Professor
Dept. of CSE

Prof. Shankar R
Assistant Professor
Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064.

2021-2022

BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT
YELAHANKA, BENGALURU – 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"STEAM ENGINE SIMULATION"** is a bonafide work carried out by **Vedashrutha D S (1BY19CS178) and Srinivas R (1BY19CS157)** in partial fulfillment for *Mini Project* during the year 2021-2022. It is hereby certified that this project covers the concepts of *Computer Graphics & Visualization*. It is also certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report.

Signature of the Guide
Prof. Muneshwara M S/
Prof. Chethana C/
Prof. Shankar R
Assistant Professor
CSE, BMSIT&M

Signature of the HOD
Dr. Thippeswamy G
Professor & HOD
CSE, BMSIT&M

Name and Signature of the Examiners

Internal Examiner

External Examiner

INSTITUTE VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

INSTITUTE MISSION

Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

DEPARTMENT VISION

To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

DEPARTMENT MISSION

Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

PROGRAM EDUCATIONAL OBJECTIVES

1. Lead a successful career by designing, analyzing and solving various problems in the field of Computer Science & Engineering.
2. Pursue higher studies for enduring edification.
3. Exhibit professional and team building attitude along with effective communication.
4. Identify and provide solutions for sustainable environmental development.

PROGRAM SPECIFIC OUTCOMES

1. Analyze the problem and identify computing requirements appropriate to its solution.
2. Apply design and development principles in the construction of software systems of varying complexity.

ACKNOWLEDGEMENT

We are happy to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals.

We would like to express our deep sense of gratitude and indebtedness to each and every one who has helped us make this project a success.

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

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

We gracefully thank our Project Guide, **Prof. Muneshwara M S, Prof. Chethana C** and **Prof. Shankar R** Assistant Professors, Department of Computer Science and Engineering for their intangible support and for being constant backbone for our project.

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

Lastly, we thank our parents and friends for the support and encouragement given throughout in completing this precious work successfully.

VEDASHRUTHA D S (1BY19CS178)

SRINIVAS R (1BY19CS157)

ABSTRACT

A **“STEAM ENGINE”** is a heat engine that performs mechanical work using steam as its working fluid. Heat is obtained from fuel burnt in a closed firebox. The heat is transferred to the water in a pressurized boiler, ultimately boiling the water and transforming it into saturated steam. Steam in its saturated state is always produced at the temperature of the boiling water, which in turn depends on the steam pressure on the water surface within the boiler. The steam is transferred to the motor unit which uses it to push on pistons to power machinery. The used, cooler, lower pressure steam is exhausted to atmosphere.

The project simulates the working of a steam engine. It illustrates how the linear motion of the piston is converted into rotary motion. The engine is initially at rest. On right clicking, the user is provided with a menu which provides five options-shaded, animate, increase speed, decrease speed, transparent.

The animate option starts the steam engine from rest or stops the engine if it is running. The speed of the engine can then be increased by the increase speed option or decreased by the decrease speed option. The texture of the engine can be changed by the shaded option. There are two textures. One being the normal solid fill and the other being wireframe. The transparent option makes the front portion of the cylinder transparent and shows the up and down motion of the piston. This project mimics a steam engine's operation. It demonstrates how the piston's linear motion is changed into rotary motion. At first, the engine is not moving. When the user does a right click, a menu appears with five options: transparent, animated, increased and decreased speed, and shaded.

The steam engine can be animated to start from a standstill or to shut down if it is already functioning. The increase speed option or reduce speed option can then be used to change the engine's speed. The shaded option allows you to modify the engine's texture. Two textures are present. The first is the typical solid fill, and the second is wireframe. The transparent option reveals the piston's up and down action and makes the front portion of the cylinder transparent.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Computer Graphics

Computer graphics are graphics created using computers and, more generally, the representation and manipulation of image data by a computer hardware and software. The development of computer graphics, or simply referred to as CG, 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 of media and have revolutionized the animation and video game industry.

2D computer graphics are digital images—mostly from two-dimensional models, such as 2D geometric models, text (vector array), and 2D data. 3D computer graphics in contrast to 2D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering images.

The user controls the contents, structure, and appearance of the objects and of their displayed images by using input devices, such as keyboard, mouse, or touchscreen. Due to close relationships between the input devices and the display, the handling of such devices is included in the study of computer graphics. The advantages of the interactive graphics are many in number. Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D patter-recognition abilities allow us to perceive and process data rapidly and efficiently.

In many design, implementation, and construction processes today, the information pictures can give is virtually indispensable. Scientific visualization became an important field in the 1980s when the scientists and engineers realized that they could not interpret the prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

1.2 Open GL

OpenGL is the most extensively documented 3D graphics API (Application Program Interface) to date. Information regarding OpenGL is all over the Web and in print. It is impossible to exhaustively list all sources of OpenGL information. OpenGL programs are typically written in C and C++. One can also program OpenGL from Delphi (a Pascal-like language), Basic, Fortran, Ada, and other languages. To compile and link OpenGL programs, one will need OpenGL header files. To run OpenGL programs, one may need shared or dynamically loaded OpenGL libraries, or a vendor-specific OpenGL Installable Client Driver (ICD).

OpenGL is a low-level graphics library specification. It makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered (drawn).

1.3 GLUT

The OpenGL Utility Toolkit (GLUT) is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres, and cylinders. GLUT even has some limited support for creating pop-up menus. The two aims of GLUT are to allow the creation of rather portable code between operating systems (GLUT is crossplatform) and to make learning OpenGL easier. All GLUT functions start with the glut prefix (for example, glutPostRedisplay marks the current window as needing to be redrawn).

CHAPTER 2

LITERATURE SURVEY

A steam turbine is a mechanical device that extracts thermal energy from pressurized steam, and converts it into rotary motion. Its modern manifestation was invented by Sir Charles parson in 1884.It has almost completely replaced the reciprocating piston steam engine primarily because of its greater thermal efficiency and higher power-to-weight ratio. Because the turbine generates rotary motion, it is particularly suited to be used to drive an electrical generator – about80% of all electricity generation in the world is by use of steam turbines.

The steam turbine is a form of heat engine that derives much of its improvement in thermodynamic efficiency through the use of multiple stages in the expansion of the steam, which results in a closer approach to the ideal reversible process. Turbines are rotating machines with unique construction and very efficient operation. The operating principle is simple; when a forcing agent such as wind, water, steam or gas is directed at the curved blades of the rotor mounted on a shaft, the forcing agent changes speed and direction and the kinetic energy produced, spins the rotor.

The shaft upon which the rotor is mounted, in turn produces mechanical power that can be used to operate motors, generators, hydraulic systems, compressors, gear drives, or any number of machines used for many applications. While the principle of operation is simple, turbines themselves can be very complex in design and construction and each type offers its own challenges to the operator with regard to lubrication, maintenance, troubleshooting and overall reliability. The utilisation of many stages in the expansion of the steam, which results in a closer approach to the ideal reversible process, is a major contributor to the steam turbine's gain in thermodynamic efficiency. Turbines are rotating devices with distinctive design and highly effective operation. The working principle is straightforward: the rotor spins when a forcing agent, such as wind, water, steam, or gas, is directed at the curved blades fixed on a shaft. The forcing agency changes speed and direction as it does this.

The rotor's mounting shaft generates mechanical power that can be utilised to drive a variety of devices, including motors, generators, hydraulic systems, compressors, gear drives, and many more.

Sir Charles Parson created it in its current form in 1884. Because of its improved thermal efficiency and higher power-to-weight ratio, it has almost entirely supplanted the reciprocating piston steam engine.

About 80% of the world's power is produced by steam turbines, which are uniquely adapted to operate an electrical generator since they produce circular motion. The activity on other steam lines that need to be electrified for other reasons, and comparing the intrinsic characteristics of steam and electric locomotives seems to be the most effective way to gain a complete understanding of the issue. It is best to determine the general characteristics of the two locomotives before focusing on the electric locomotive, which is considerably simpler.

Limitations of the steam locomotive from the perspective of the electrical engineer, in order to clearly define the contrasts and fully comprehend the magnitude of the issue. strongly drawn.This initial investigation of the steam locomotive is made necessary by the fact that modern railroad practise is largely based on the steam locomotive and is constrained by the practises and operational procedures required by the usage of the steam locomotive as a form of motive power. When considering the largest advantages to be gained, the introduction of the electric locomotive is not the result of little savings in coal consumption and locomotive repair costs.

Therefore, it is necessary to look for benefits that have a wider impact. Without taking into account the factors that led to the introduction of electric locomotives at terminals and in tunnels, we find that there is a stark contrast between the characteristics of the steam and electric locomotives that not only demonstrates the superiority of the electric locomotive for general railway conditions but also suggests fundamental changes to current operating procedures.

CHAPTER 3

PROJECT IN DETAIL

3.1 Problem Statement

Computer graphics is no longer a rarity. It is an integral part of all computer user interfaces, and is indispensable for visualizing 2D; 3D and higher dimensional objects. Creating 3D objects, rotations and any other manipulations are laborious process with graphics implementation using text editor. OpenGL provides more features for developing 3D objects with few lines by built in functions.

In this project the main problem statement is to simulate the working of a “steam engine” using the concepts of Open GL. The simulation should cover all the aspects and principles involved in the conversion of steam energy into mechanical energy to rotate a piston. It is a crucial component of all computer user interfaces and essential for viewing objects in higher dimensions. The implementation of graphics using a text editor is a lengthy procedure when creating 3D objects, rotating them, or performing any other operations. With the help of built-in functions, OpenGL offers more options for creating 3D objects with little code.

3.2 Brief Introduction

“Steam Engine” operates on the force of water vapor (steam) heated to high pressure. It is a technology that converts thermal energy (heat) to mechanical energy (work).The boiler heats the water turning it to steam. The pressure generated is then used to drive a piston within a cylinder. The piston is attached to a connecting rod for transforming the translational movement into a rotational movement.Steam turbines generate almost 80% of the world's energy, and because they generate circular motion, they are specially designed to run an electrical generator. The activity on other steam lines that require electrification for different reasons, as well as a comparison of the inherent qualities of steam and electric locomotives, appears to be the most efficient ways to fully comprehend the problem. Prior to concentrating on the electric locomotive, which is much simpler, it is best to ascertain the main characteristics of the two locomotives.

The steam engine simulation shows the steam engine of the Scottish inventor, James Watt. It has many improvements over its predecessors’ machines. He invented, in 1782, the principle of the double effect machine (or double action) in which a sliding valve distributing the pressure on the piston is driven in both directions.

The centrifugal regulator (rotating balls) is another innovation introduced by James Watt. It maintains an almost constant rate despite fluctuations in available pressure.Main objectives of this project are :

- To understand the operating principle of a steam engine.
- To simulate converting thermal energy (heat) to mechanical energy (work).
- To identify the function of each part of the Watt engine.

3.3 Scope and Motivation

It has become evident that by simulating an object makes it easier to understand the basic principles, mechanics and technology behind the working of a system. Steam Engine Simulation provides the cross sectional view of the engine, rotor and the piston motion to provide an overall viewing experience to the users. Automobiles are involved with our day to day life. We all use automobiles without knowing their working. By steam engine simulation we analyse the basics principles and working of an IC engine.

Modelling an object makes it simpler to comprehend the fundamental ideas, physics, and technology that underlie a system's operation. To give customers a complete viewing experience, Steam Engine Simulation offers a cross-sectional view of the engine, rotor, and piston action. Our daily lives involve automobiles in some way. All of us use cars without fully understanding how they function. We examine the fundamental principles and operation of an IC engine using steam engine simulation.

In this project we are trying to develop most enriched graphical model that renders high quality graphics for the simulation of steam engine. Utilizing a steam engine simulation, we look at the core concepts and operation of an IC engine.We're attempting to create the most comprehensive graphical model that produces stunning visuals for the simulation of a steam engine.

3.4 Proposed System

The whole idea behind the simulation is to provide a better view and working of the system. In order to make the simulation much more efficient enriched graphical objects, efficient algorithms that renders high quality encoded graphics and complex mathematics for 3D rendering is used. Because visual is always more interesting and easier to remember. With the use of different functionality like of making polygons, using various functions of mouse and creation of menu, the whole story is plotted in this project.

The simulation's main goal is to provide users a clearer understanding of how the system operates. Complex mathematics for 3D rendering and effective algorithms that output high-quality encoded graphics are employed to make the simulation much more effective. because visual information is always more fascinating and simpler to recall. The entire story is plotted in this project by utilising various features, such as creating polygons, employing various mouse operations, and creating menus.

This project gives a user-friendly interface for the end user to interact with the frames. This project contains several different frames that are used in a particular sequence to help the user the engine working from different view points.

3.5 User Interface

The legend for user interaction is as follows

- ‘a’ - > To rotate crank anti-clockwise.
- ‘z’ - > To rotate crank clock wise.
- ‘+’ and ‘-‘ - > To speed up and speed down
- ‘o’ - > Transparency.
- ‘0’ and ‘1’ - > Right light and Left light respectively
- ‘s’ and ‘t’ - > Shading and Texture respectively

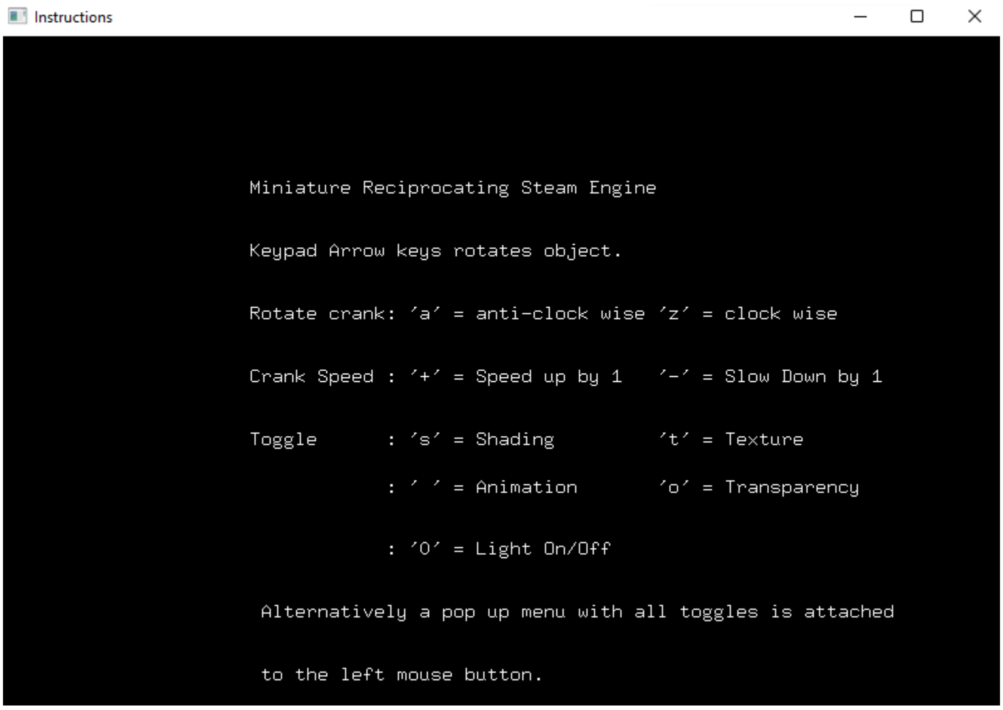

Figure 3.5: **Control and Instructions menu**

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

Function scope simulation: - design to animate the engine model to show the basic working and operations of a steam engine. Animation and movement of pistons to show the working of an IC steam engine. Operating environment to develop this system we used software and hardware operating environment. The software requirements are description of features and functionalities of the target system. Create an animated 3D model to demonstrate a steam engine's fundamental operations. An IC steam engine's operation is demonstrated through animation and piston movement. We employed both a software and a hardware operating environment to construct this system.

The target system's features and functionalities are described in the software requirements. Function scope simulation: - Create an animated engine model to demonstrate a steam engine's fundamental operations. An IC steam engine's operation is demonstrated through animation and piston movement. We employed both a software and a hardware operating environment to construct this system. The target system's features and functionalities are described in the software requirements. To illustrate the basic workings of a steam engine, make an animated 3D model. Animation and piston movement are used to show how an IC steam engine works. To build this system, we used both a software and a hardware operating environment. The software requirements outline the features and functionalities of the target system.

4.1 Hardware Requirements

Processor: Intel Pentium or Higher

Hard Disk: At least 10GB

RAM: Minimum 2GB

Keyboard: US English Keyboard

Mouse: 2 or 3 Button mouse

Monitor: 1024 x 768 60Hz display

4.2 Software Requirements

Programming language – C/C++ using OpenGL

Operating system – Windows7 or higher/2000/XP/Vista/UBUNTU

Compiler/IDE - Code::Blocks IDE

Graphics library – GL/glut.h

OpenGL 2.0

4.3 Functional Requirements

OpenGL APIs: If we want to have a control on the flow of program and if we want to interact with the window system then we use OpenGL API'S. Vertices are represented in the same manner internally, whether they are specified as two-dimensional or three-dimensional entities, everything that we do are here will be equally valid in three dimensions. Although OpenGL is easy to learn, compared with other APIs, it is nevertheless powerful. It supports the simple three-dimensional programs and also supports the advanced rendering techniques.

Everything we do here will be equally valid in three dimensions since vertices are internally represented in the same way whether they are described as two-dimensional or three-dimensional entities. Compared to other APIs, OpenGL is simpler to understand, yet it still has a lot of capability. Both basic three-dimensional programmes and cutting-edge rendering methods are supported by it.

GL/glut.h: We use a readily available library called the OpenGL Utility Toolkit (GLUT), which provides the minimum functionality that should be expected in any modern windowing system. The application program uses only GLUT functions and can be recompiled with the GLUT library for other window system. OpenGL makes a heavy use of macros to increase code readability and avoid the use of magic numbers. In most implementation, one of the include lines.

The application programme can be recompiled with the GLUT library for usage with various window systems and solely uses GLUT functions. To improve code readability and eliminate the usage of magic numbers, OpenGL heavily utilises macros. One of the include lines in the majority of implementations.

Code::Blocks Integrated Development Environment (IDE): 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 Widgets 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. Code::Blocks is being developed for Windows, Linux, and MacOS and has been ported to FreeBSD, OpenBSD and Solaris. Code::Blocks is based on a plugin framework. Plugins can be used to extend blocks. Installing or creating a plugin allows for the addition of any type of functionality. For instance, plugins offer the ability to compile events and debug them!

CHAPTER 5

SYSTEM DESIGN AND IMPLEMENTATION

5.1 System Design

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats,output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

Purpose and Scope: This section provides a brief description of the Systems Design Document's purpose and scope.
The main purpose of documenting the simulation is to provide a better understanding and real time working knowledge of the steam engine. This knowledge can be used to further develop high end jet engines ad well as automobile engine.

Project Executive Summary: This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

System Overview:
This section describes the system in narrative form using non-technical terms.It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

Design Constraints:
This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design

Future Contingencies:

This section describes any contingencies that might arise in the design of the system that may change the development direction.

The description of the functions in the design process is discussed as follows:

void glutMouseFunc(myMouse)

Refers to the mouse callback function. The function to callback is defined as void myMouse(int button, int state, int x)

```
{
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
        if (glutGetModifiers() & GLUT_ACTIVE_SHIFT)
            decrease a level of recursion
        else
            increase a level of recursion
}
```

Here mouse interface is given to increase a level of recursion by clicking mouse button and also to decrease a level of recursion by doing the same holding the shift on the keyboard.

void glutKeyboardFunc(myKey)

Refers to the keyboard callback function.The function to callback is defined as void myKey(unsigned char key, int x, int y)

```
{
    if (c == 't')
    exit
    if (c == 's')
        //STATEMENTS and repeat when finished
}
```

Here keyboard interface is given to quit, the user can quit by pressing ‘q’ and to see next example of the implementation, the user should press ‘n’.

5.2 System Implementation

The “Steam Engine Simulation” is implemented using OpenGL inbuilt functions along with some of the user defined functions. This section comprises of the complete explanation of the design of the project using various opengl functions and other user defined methods along with its explanation. This chapter is divided into two sections

- OpenGL Functions – This section throws light on the various openGL functions used and its uses.
- User Defined Functions – This section demonstrates the various user defined functions and its purpose.

OpenGL Functions

The different OpenGL functions that are used in the project is described below:

- **glColor3f(float, float, float)** :- this function specifies the red, green, blue, and alpha values used by glClear() to clear the color buffers. Values specified by glClearColor() are clamped to the range 0, 1
- **glClear()** :-Takes a single argument that is the bitwise OR of several values indicating
- which buffer is to be cleared.
- **glClearColor()** :-Specifies the red, green, blue, and alpha values used by glClear to
- clear the color buffers.
- **glMatrixMode(GL_MODELVIEW)** :- specify which matrix is the current matrix. Specifies which matrix stack is the target for subsequent matrix operations. Three values are accepted: GL_MODELVIEW, GL_PROJECTION, and GL_TEXTURE. The initial value is GL_MODELVIEW.
- **glFlush()** :- Force execution of GL commands in finite time. glFlush empties all of these buffers, causing all issued commands to be executed as quickly as they are accepted by the actual rendering engine. Though this execution may not be completed in any particular time period, it does complete in finite time.
- **glutDisplayFunc(display1)** :- registers the callback function (or event handler) for handling window-paint event. The OpenGL graphic system calls back this handler when it receives a window re-paint request. In the example, we register the function display() as the handler.

- **glutMainLoop()** :- glutMainLoop enters the GLUT event processing loop. This routine should be called at most once in a GLUT program. Once called, this routine will never return. It will call as necessary any callbacks that have been registered.

User Defined Functions:

- **void draw_piston(void)** :- This function is used to create a piston that oscillates inside the cylinder as the turbine rotates.
- **void myBox(GLdouble x, GLdouble y, GLdouble z)** :-This function creates a rectangular shaped object for engines body.
- **void myCylinder(GLUquadricObj * object, GLdouble outerRadius,GLdouble innerRadius, GLdouble lenghmydisplay(void)** :- this function on the value of the parameters creates a cylinder that houses the piston.
- **void draw_engine_pole(void)** :- Draws the engine pole and the pivot pole for the cylinder head.
- **void draw_cylinder_head(void)** :-Draws the cylinder head at the appropreate angle, doing the necesary
- **void draw_crankbell(void)** :-Draws the crank bell, and the pivot pin for the piston. Also calls the appropriate display list of a piston doing the necessary rotations beforehand.
- **void draw_crank(void)** :-Draws the complete crank. Piston also gets drawn through the crank bell function.
- **void animation(void)** :-Called when the window is idle. When called increments the crank angle by ANGLE_STEP, updates the head angle and notifies the system that the screen needs to be updated.
- **void display(void)** :-Main display routine. Clears the drawing buffer and if transparency is set, displays the model twice, 1st time accepting those fragments with a ALPHA value of 1 only, then with DEPTH_BUFFER writing disabled for those with other values.
- **void myReshape(int w, int h)** :-Called when the model's window has been reshaped by transformations.
- **void myinit(void)** :-Initialises texturing, lighting, display lists, and everything else associated with the model.

5.3 Snapshots

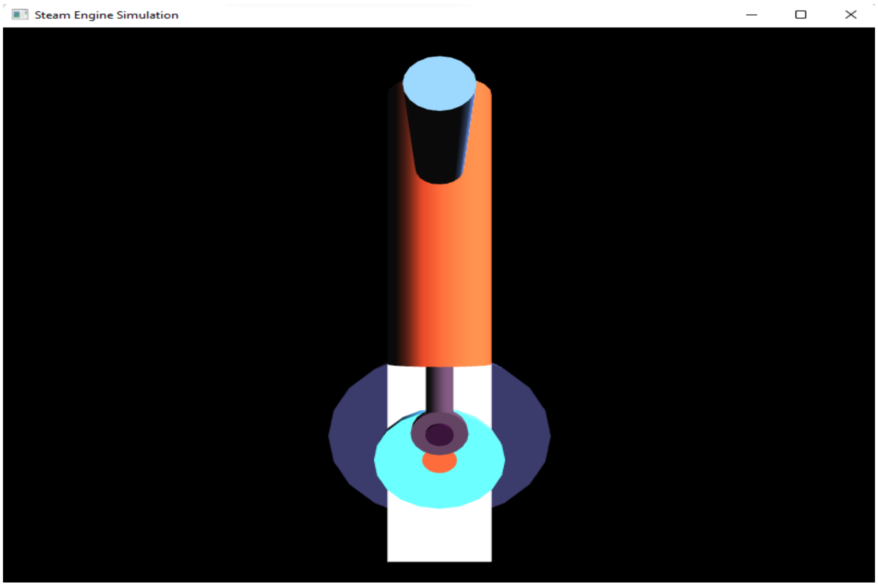

Fig 5.3.1 : Shaded view of Engine

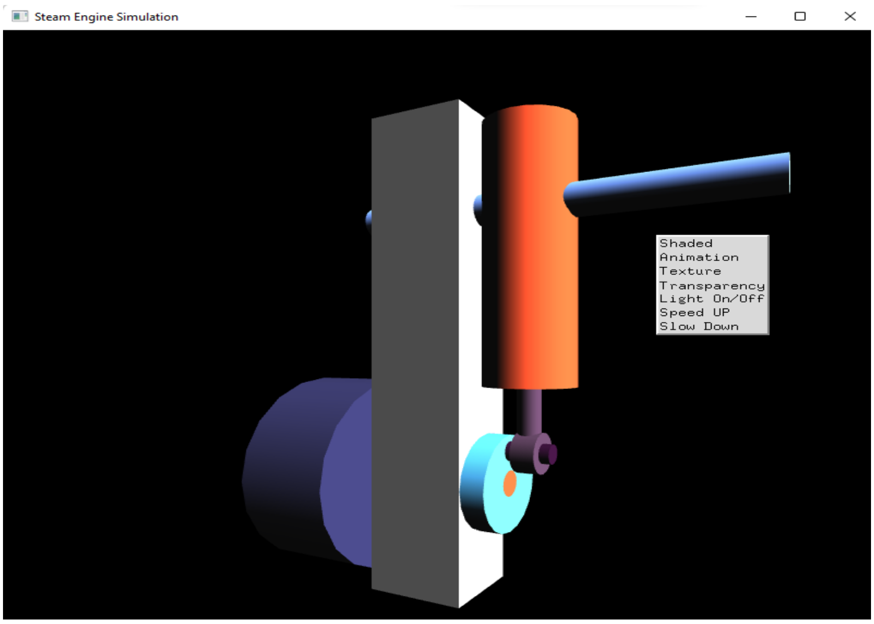

Fig 5.3.2:Rotation about the Origin with menu list

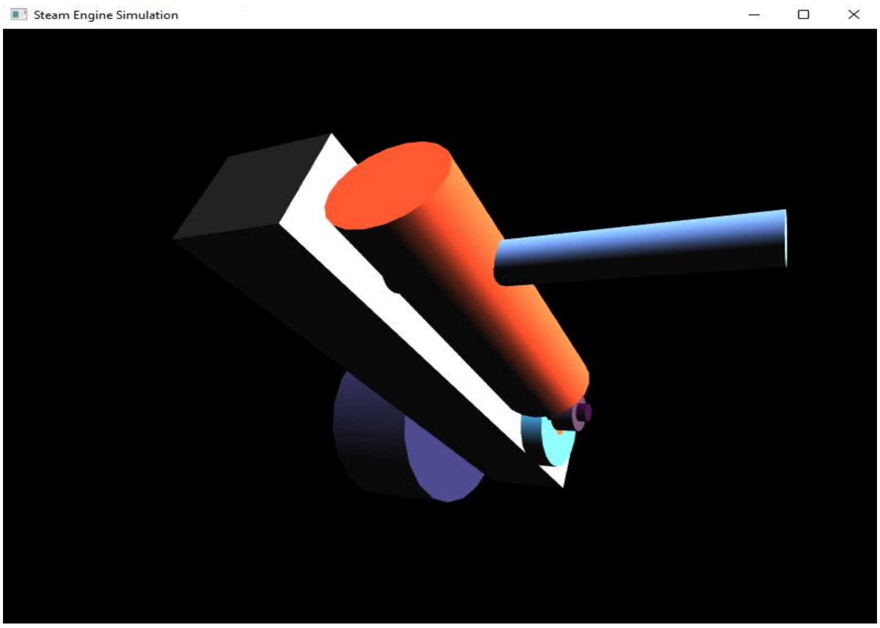

Fig 5.3.3 Rotation Front

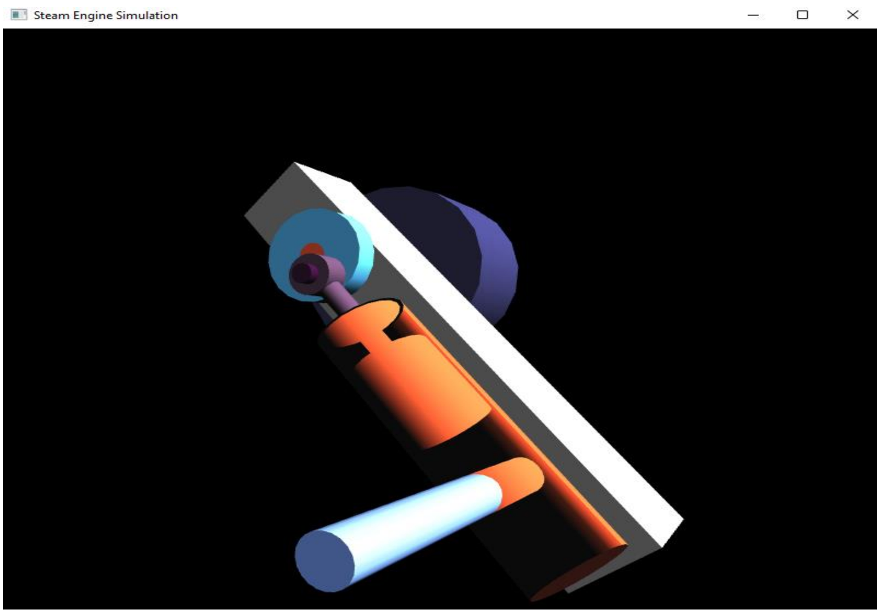

Fig 5.3.4 Transparent Cylinder showing piston

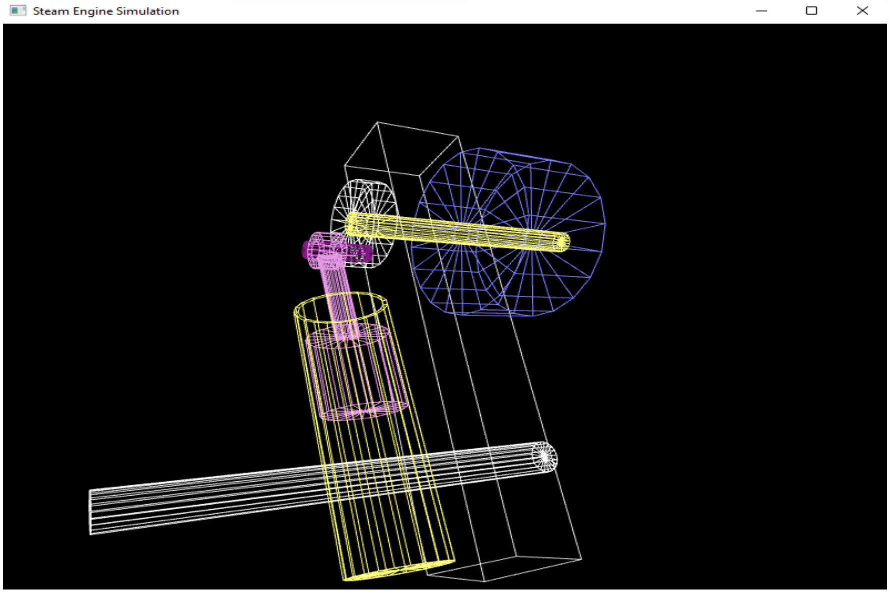

Fig 5.3.5:Textured view of Engine

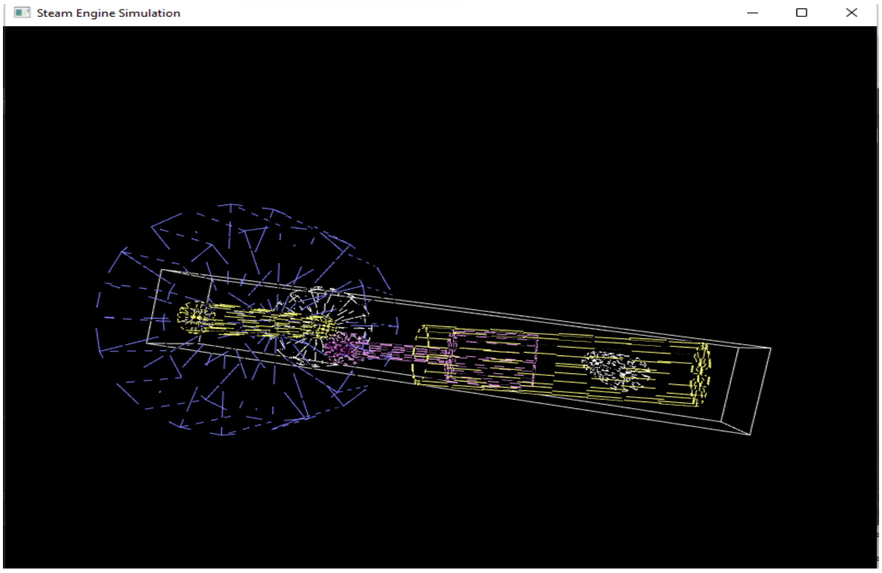

Fig 5.3.6. Dotted Cross sectional view of Engine

CHAPTER 6

CONCLUSION

The very purpose of developing this project is to exploit the strength of OpenGL graphics capabilities for a interesting cause. This project allows the user to rotate the piston in a Steam Engine similar to a Miniature Steam Engine Simulation.

Steam Engine Simulation has been tested under Windows platform, and has been found to provide ease of use and manipulation to the user. With the availability of graphics card the simulation provides best overall experience for the users. The Simulation created for the Windows operating system can be used to draw lines, boxes, circles, ellipses, and polygons. It has a very simple and effective user interface. The simulation outputs high quality graphics which are rendered through a graphic engine in support to the pre-installed graphic drivers.

We found designing and developing this simulation as a very interesting and learning experience. It helped us to learn about computer graphics, design of Graphical User Interfaces, interface to the user, user interaction handling and screen management. The graphical engine has a vital role that renders high quality pictures and detailed textures to the users. This simulation's conception and development was a really interesting and educational experience for us. It aided in our learning of computer graphics, GUI design, user interface, handling of user interaction, and screen management.

The graphical engine plays a crucial role in providing consumers with high definition images and intricate textures. The simulation gives users the best overall experience when a graphics card is available. Draw lines, boxes, circles, ellipses, and polygons with the Simulation designed for the Windows operating system. Its user interface is incredibly basic and efficient.

CHAPTER 7

FUTURE ENHANCEMENTS

The future enhancement that can be made include:

- **Enhanced Simulator:** Not only the movement of piston, we can make the whole parts in the steam engine working so that it will be a simulator of steam engine. By modifying this project, we can construct a fully-fledged simulator. Students who are studying about the steam engine can work through this and it will be very helpful for them. Almost a complete picturization of a steam engine can be done through this.
- **Design of Steam Engines:** Engineers who build Automobile Engines can design their model by looking this project. They get a good picturization by seeing this and it will be helpful for them in building much more complex jet and automobile engines.
- **Optimized algorithm for high quality rendering:** The algorithm used for this simulation is a straight forward one. By developing a new algorithm that involves much more complex mathematics for high picture quality and sharpness of the image layer, the project will be able to render high quality graphics.

The simulation's algorithm is an easy-to-understand one. The project will be able to create high quality graphics by creating a new algorithm that uses considerably more complicated mathematics for high picture quality and sharpness of the image layer.

This project is developed by considering only one piston. Further it can be improved to check the fuel efficiency and basic horsepower output for inline engines or v-8 engines that comes with a turbo and transmission modes.

BIBLIOGRAPHY

- Edward Angel's Interactive Computer Graphics Pearson Education 5th Edition.
- Computer Graphics Using OpenGL – F.S. Hill, Jr. 2nd Edition, Pearson Education, 2001.
- Steam Engine: https://ieeexplore.ieee.org/document/6152649
- OpenGl: https://www.khronos.org/opengl/wiki/Getting_Started
- https://learnopengl.com/