**VISVESVARAYA TECHNOLOGICALUNIVERSITY**

**"JNANASANGAMA", BELAGAVI-590018**

**A MINI PROJECT REPORT**

**ON**

# SMART VILLAGE

**Submitted in partial fulfilment of the requirements for**

**COMPUTER GRAPHICSLABORATORY WITH MINI PROJECT [18CSL67]**

**For the award of degree of**

**Bachelor of Engineering**

**In**

**Computer Science and Engineering**

**By**

**KEERTHAN GOWDA S**

**1KS19CS043**

**Under the guidance of**

**Mrs. Beena K**
Assistant Professor

**Mrs. Geetha R**
Assistant Professor

**Department of Computer Science & Engineering**

**K.S. INSTITUTE OF TECHNOLOGY**

**#14, Raghuvanahalli, Kanakpura Main Road, Bengaluru-560109**

**2021-2022**

# K.S. INSTITUTE OF TECHNOLOGY

**#14, Raghuvanahalli, Kanakpura Main Road, Bengaluru-560109**

**Department of Computer Science & Engineering**

## CERTIFICATE

This is to certify that mini project work entitled **"SMART VILLAGE"** carried out by **Mr. Keerthan Gowda S** bearing USN **1KS19CS043** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering** in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belagavi, during the year 2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree for the 6th semester.

| | |
|---|---|
| **Mrs. Beena K** | **Mrs. Geetha R** |
| Assistant Professor | Assistant Professor |
| **Dr. Rekha. B. Venkatapur** | **Dr. Dilip Kumar. K** |
| Professor& Head | Principal/Director |

| **Name of the Examiners with date** | **Signature** |
|---|---|
| **1.** | |
| **2.** | |

# INDEX

# LIST OF FIGURES

# ACKNOWLEDGEMENT

The successful completion of mini project would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I take this opportunity to express my sincere gratitude to our **Management** K S Institute of Technology, Bengaluru for providing all the resources required for the mini project.

I would express my gratitude to **Dr. K.V.A. Balaji,** C.E.O. K.S. Institute of Technology, Bengaluru, for facilitating me to complete mini project.

I would like to extend my gratitude to **Dr. Dilip Kumar. K,** Principal/Director, K.S. Institute of Technology, Bengaluru, for his encouragement and providing all facilities for the accomplishment of this project.

I thank **Dr. Rekha B. Venkatapur**, Professor and Head, Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for her encouragement.

I whole heartedly thank project guides, **Mrs. Beena K and Mrs. Geetha R**, Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for their support and guidance.

Finally, I would like to thank all the teaching and non-teaching staff of the college for their co-operation. Moreover I thank all my **family** and **friends** for their invaluable support and cooperation.

**KEERTHAN GOWDA S**

**1KS19CS043**

# ABSTRACT

OpenGL is the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide varietyof computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions.

The main idea behind this project is to display an aquarium with computer graphics. This graphics package is based on the OpenGL library functions. The programming language used here is C using OpenGL libraries. In this project, we are demonstrating the screensaver in which the birds will have a motion along with the aeroplane and there is movement of lights turning on and off.

I have created some artifacts in this mini project, like a home, sun, clouds, trees, and birds. In the project a house is present in a small village and in the sky couple of clouds will be passing in the sky and there will be few trees from where the set of birds will be flying from and in the house we can turn off the light and turn on the light whenever it is needed and it has got two modes in it day mode and the night mode the above-explained things are attained in the day mode and in the night mode an airplane and the comet will be passing in the night sky.

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction to Computer Graphics

Computer Graphics started with the display of data on hardcopy plotters and cathode ray tube screens soon after the introduction of computer themselves. Now it has grown to that extent that it has included the creation, storage and manipulation of models and images of objects.

Computer Graphics today is largely interactive; it is the user who controls the contents, structure and appearance of objects and of their displayed images by using input devices, such as keyboard, mouse or touch-sensitive panel on the screen. Although early applications in engineering and science had to rely on expensive and cumbersome equipment, advance in computer technology have made interactive graphics as a practical tool.

Computer Graphics is an integral part of all computer user interfaces, and is indispensable for visualizing two dimensional, three-dimensional, and higher-dimensional objects: areas as diverse such as education, science, Computer Graphics started with the display of data on hardcopy plotters and cathode ray tube screens soon after the introduction of computer themselves. Now it has grown to that extent that it has included the creation, storage and manipulation of models and images of objects.

Computer Graphics today is largely interactive; it is the user who controls the contents, structure and appearance of objects and of their displayed images by using input devices, such as keyboard, mouse or touch-sensitive panel on the screen. Although early applications in engineering and science had to rely on expensive and cumbersome equipment, advance in computer technology have made interactive graphics as a practical tool. Computer Graphics is an integral part of engineering, medicine, commerce, the military; advertising and entertainment all rely on computer graphics.

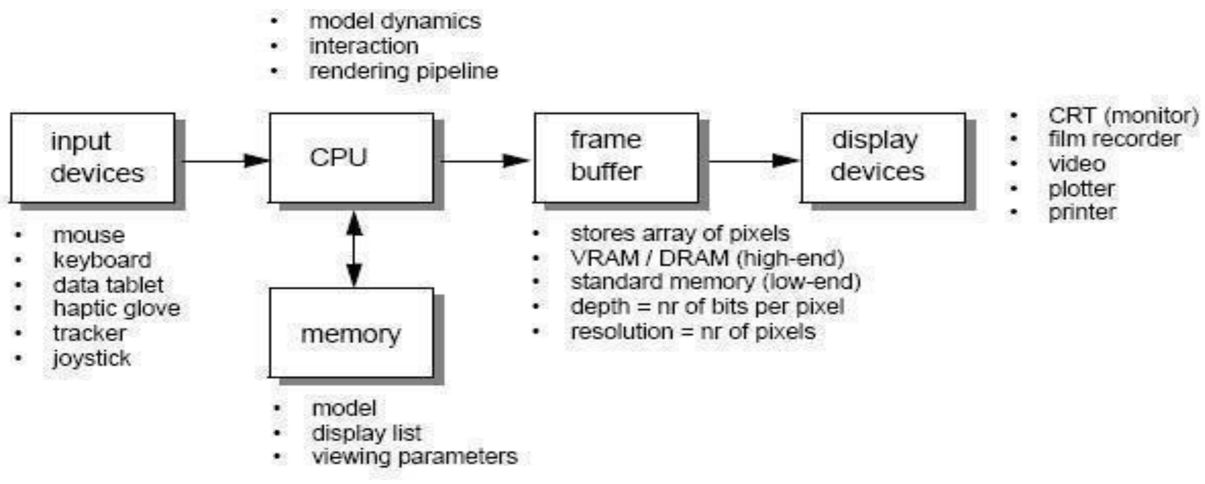

Fig. 1.1: Graphics System

Until the early 1980's, Computer Graphics was a small, specialized field, largely because the hardware was expensive and graphics-based application programs that were easy to use and cost effective were few. Then, personal computers with built in raster graphics displays popularized the use of bitmap graphics for array of points or also called as pixels on the screen. Once bitmap graphics became affordable, an explosion of easy- to- use and inexpensive graphics-based applications soon followed. Graphics-based interfaces allowed millions of news users to control simple, low cost application programs, such as spreadsheets, word processors, and drawing programs. Graphics has its own hardcopy technologies, input technologies and also display technologies. Some of the hardcopy technologies are printers, pen plotters etc. Some of the display technologies such direct view storage tube, liquid crystal displays, plasma panels etc. Input technologies like keyboard, mouse, touch panel, tablets etc.

## 1.2 Applications of Computer Graphics

Computer graphics is used today in mainly different areas of industry, business, government, education, entertainment, and most recently, the home. The list of applications is enormous and is growing rapidly as computers with graphics capabilities become commodity products. Now let us see some of the applications.

• **Computer Aided Design**

Computer graphics is used in design process, particularly for engineering and architectural systems, but almost all products are now computer designed. Generally referred to as CAD, computer aided design methods are now routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft, computers, textiles, and many other products.

• **Image Processing**

In computer graphics, a computer is used to create a picture. Image processing on the other hand, applies techniques to modify or interpret existing pictures, such as photographs and TV scans. Two principal applications of image processing are improving picture quality and machine perception of visual information, as used in robotics.

• **Education and Training**

Computer generated models of physical, financial and economic systems are often used as educational aids. Models of physical systems, physiological systems, population trends, or equipment, can help trainees to understand the operation of the system.

• Entertainment

Computer graphics methods are now commonly used in making motion pictures, music videos, and televisionshows.

## 1.3 History of OpenGL

In the 1980s, developing software that could function with a wide range of graphics hardware was a real challenge. Software developers wrote custom interfaces and drivers for each piece of hardware. This was expensive and resulted inmuch duplication of effort. By the early 1990s, Silicon Graphics (SGI) was a leader in 3D graphics for workstations. Their IRIS GL API was considered the state of the art and became the de facto industry standard, overshadowing the open standards-based PHIGS. This was because IRIS GL was considered easier to use, and because it supported immediate mode rendering. By contrast, PHIGS was considered difficult to use and outdated in terms of functionality.

SGI's competitors (including Sun Microsystems, Hewlett-Packard and IBM) were also able to bring to market 3D hardware, supported by extensions made to the PHIGS standard. This in turn caused SGI market share to weaken as more 3D graphics hardware suppliers entered the market. In an effort to influence the market, SGI decided to turn the IrisGL API into an open standard.

SGI considered that the IrisGL API itself wasn't suitable for opening due to licensing and patent issues. Also, the IrisGL had API functions that were not relevant to 3Dgraphics. For example, it included a windowing, keyboard and mouse API, in part because it was developed before the X Window System and Sun's News systems were developed.

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. Meanwhile, SGI would continue to try to maintain their customers tied to SGI hardware by developing the advanced and proprietary Iris Inventor and Iris Performer programming APIs. As a result, SGI releasedthe OpenGL standard.

The OpenGL standardized access to hardware, and pushed the development responsibility of hardware interface programs, sometimes called device drivers, to hardware manufacturers and delegated windowing functions to the underlying operating system. With so many different kinds of graphic hardware, getting them all speak the same language in this way had a remarkable impact by giving software developers a higher level platform for 3D-software development.

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 for years to come. OpenGL evolved from (and is very similar in style to) SGI's earlier 3D interface, *IrisGL*. One of the restrictions of IrisGL was that it only provided access to features supported by the underlying hardware. If the graphics hardware did not support a feature, then the application could not use it. OpenGL overcame this problem by providing support in software for features unsupported by hardware, allowing applications to use advanced graphics on relativelylow powered systems.

### 1.3.1 Important Functions of OpenGL

The OpenGL Utility Toolkit is a libraryof utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. The glut library is organized as shown.

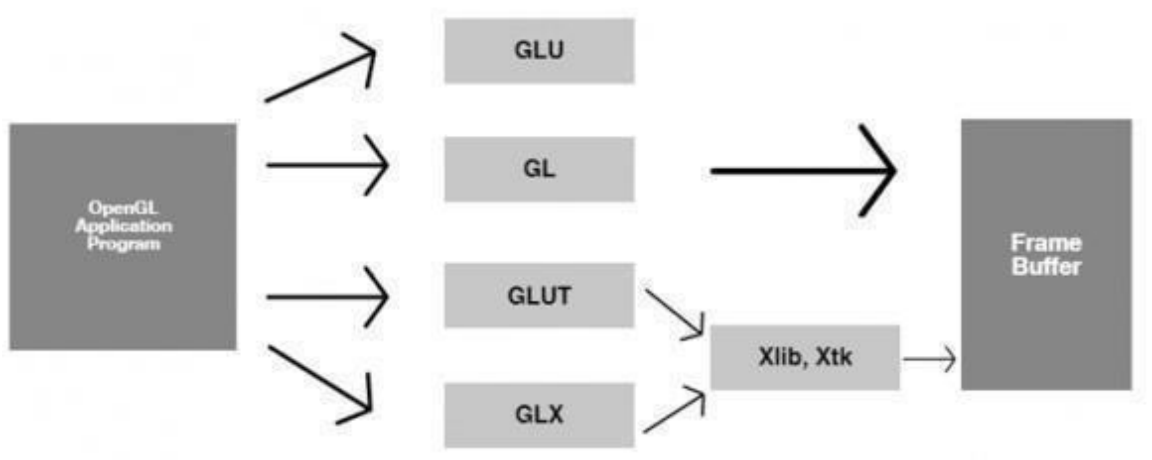

Fig. 1.2: OpenGL libraries

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.

GLUT is designed for constructing small to medium sized OpenGL programs. While GLUT is well-suited to learning OpenGL and developing simple OpenGL applications, GLUT is not a full-featured toolkit so large applications requiring sophisticated user interfaces are better off using native window system toolkits. GLUT is simple, easy, and small.

The GLUT library has both C, C++ (same as C), FORTRAN, and Ada programming bindings. The GLUT source code distribution is portable to nearly all OpenGL implementations and platforms.

Examples:
**gl functions:** glBegin(), glEnd(), glClear(), glFlush(), glOrtho()
**glu functions:** gluLookAt(), gluOrtho2D(), gluSphere(), gluPerspective()

**glut functions:** glutInit(), glutFullScreen(), glutMainLoop(), glutCreateMenu()

**Some of the important openGL built in functions with their functionalities are listed in following table**

| | |
|---|---|
| glBegin, glEnd | The glBegin and glEnd functions delimit the vertices of a primitive or a group of like primitives. |
| glClear | The glClear function clearsbuffers to preset values. |
| glColor | These functions set the current color |
| glFlush | The glFlush function forces execution of OpenGL functions in finite time. |
| glLoadIdentity | The glLoadIdentity function replacesthe current matrix with the identity matrix. |
| glMatrixMode | The glMatrixMode function specifies which matrix is the current matrix. |
| glVertex | These functions specify a vertex. |
| gluOrtho2D | The gluOrtho2D function defines a 2-D orthographic projection matrix. |

Fig.1.3: Important OpenGL built in functions with their functionalities.

### 1.3.2 Important Features of OpenGL

OpenGL provides a set of commands to render a three-dimensional scene. That means you provide the data in an OpenGL-useable form and OpenGL will show this data on the screen (render it). It is developed by many companies and it is free to use.

OpenGL is an API and system-independent interface. An OpenGL-application will work on every platform, as long as there is an installed implementation.

OpenGL is a collection of several hundred functions that provide access to all of the features that your graphics hardware has to offer. Internally it acts like a state machine-a collection of states that tell OpenGL what to do. Using the API you can set various aspects of this state machine, including current color, blending, lighting effect, etc.

Because it is system independent, there are no functions to create windows etc., but there are helper functions for each platform. A very useful thing is GLUT.

## 1.4 Advantagesof OpenGL

- It is user-friendly and speed up the user's work.
- It is more attractive for non-technical people.
- In general, it looks more professional (but this does not mean it is always the best solution).
- OpenGL is a cross-platform graphics API, which means that the same code can be used on multiple operating system types with minimal changes.
- OpenGL runs on every computer with graphics output capability and requires no extra downloads.

## 1.5 Drawbacks of OpenGL

- When it is not properly built, it can be very difficult to work with.
- It generally requires more resources than a non-graphical one.
- It might require the installationof additional software e.g.,
  the "runtime environment" in the case of java.

# CHAPTER 2

# LITERATURE SURVEY

In late 1960's the development of Free-form curves and surfaces for computer graphics begins. Free-form curves and surfaces where developed to describe curved 3-D objects without using polyhedral representations which are bulky and intractable. To get a precise curve with polygons might require thousands of faces, whereas curved surfaces requires much less calculations. The UNISURF CAD system was created for designing cars which utilized the curve theories.

A research was made but was never published so designers get most of the credit. The men were pioneers in Computer Aided Geometric Design(CAGD) for the auto industry, which replaced the use of hand drawn French-curve templates in design of auto bodies. The curves were based on Bernstein Polynomials which had been developed by the mathematician Bernstein much earlier. Another kind of basic curve predated that was the Hermite Curve developed by the mathematician Hermite. Also, in the same era as, Schoenberg, a mathematician at the university of Wisconsin was working on Mathematical Splines, which would influence the work of S.Coons at MIT in Splines, Bicubic Surface Patches, Rational Polynomials around 1968. A surface patch is freeform curved surface defined by two or more curves.

In 1973, designers based their research into parametric B-Splines on Coon's work. The main difference between B-Splines and Bezier curves is the former allows for local control system. B-Splines and Bezier curves is the former allows for local control of key control points and the later has more of a global control system. B-Splines are also faster to calculate for a computer than cubic polynomial based curves like the Hermite and Bezier.Rosenfield's pioneering development of B-Splines later influenced the E.catmull's researchat Utah.

# CHAPTER3

# REQUIREMENT ANALYSIS

## 3.1 Domain Understanding

The main objective is to develop a suitable OpenGL graphics package to implement basic computer graphics skills. The aim of the project is storing based animation of "SMART VILLAGE".

## 3.2 Classification of Requirements

### 3.2.1 User Requirements

Program to demonstrate the Rendering of a Beautiful story-based animation, in which the user can interact with the past which will have an impact on the future.

### Requirement Collection

The character and objects used in the project are textured rectangles which are referred from websites and other libraries. This project makes use of objects and functions defined by the glut libraries. It makes use of C programming language. The functions defined are:

- void declare(char *string)
- void draw_pixel(GLint cx, GLint cy)
- void plotpixels(GLint h, GLint k, GLint x, GLint y)
- void draw_circle(GLint h, GLint k, GLint r)
- void draw_object()
- void idle()
- void mouse(int btn, int state, int x, int y)
- void keyboardFunc(unsigned char key, int x, int y)
- void main_menu(int index)
- void computeNcR(int n, int *hold_ncr_values)
- void computeBezierPoints(float t, point *actual_bezier_point, int number_of_control_points, point *control_points_array, int *hold_ncr_values)
- void Bezier(point *control_points_array, int number_of_control_points, intnumber_of_bezier_points)

### 3.2.2 System Requirements

Here we are using header files namely <GL/glut.h>, functions in the OpenGL in windows are stored in a library usually referred as GL. GLUT uses only GL functions and also contains code for creating objects and simplifying, viewing, rather than using different library for each system here we can use readily available library called the OPENGL UTILITY TOOL KIT(GLUT).

**Software Requirements**

- Operating system: Windows
- Programming language- C
- Compiler (g++)
- OpenGL library

**Hardware Requirements**

- Processor: Pentium processor
- Memory: 32MB RAM.
- Hard Disk:40GB Hard disk
- Interface Device: Mouse, Keyboard, Monitor of Resolution 1280x720

# CHAPTER 4
# SYSTEM ANALYSIS AND DESIGN

## 4.1 System Analysis

- In this project we are mainly concentrating on using mouse and keyboard.
- Press any key and hit Enter.
- P ress ‘ d’ or ‘ D’ to make it day.
- P ress ‘ n’ or ‘ N’ to make it night.
- P ress ‘ b’ or ‘ B’ to fly birds.
- P ress ‘ l’ or ‘ L’ to turn On the lights.
- P ress ‘ f ’ or ‘ F’ to turn Off the lights.
- P ress RIGHT MOUSE BUTTON to display menu.
- P ress LEFT MOUSE BUTTON to quit the program.

## 4.2 System Design

The main function calls the following functions:

```
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_SINGLE| GLUT_RGB);
glutInitWindowSize(1100,700);
glutInitWindowPosition(0,0);
glutCreateWindow(“SMART VILLAGE”);
glutDisplayFunc(display);
glutKeyboardFunc(keyboardFunc);
glutAddMenuEntry(“Aeroplane", 1);
glutAddMenuEntry(“Comet”, 2);
glutAttachMenu(GLUT_RIGHT_BUTTON);
glutMainLoop();
```

# CHAPTER 5
# IMPLEMENTATION

## 5.1 Description of ImplementationModules

In this project I have created a smart village using “OpenGL” functional APIby the help of built in the functions present in the header file. To provide functionality to the project I 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.

## 5.2 List of Implementation Functions

**User Defined Functions:**

void keyboardFunc(unsigned char key ,int x ,int y)

This function is used to add the keyboard listener in an OpenGL.

void mouse(int btn, int state, int x, int y)

This function is used to add the keyboard listener in an OpenGL.

void main_menu(int index)

This function creates a new pop-up menu.

void display()

This function is used to displayon screen

int main(int argc, char ** argv)

The program begins execution from this function.

## 5.3 Description of inbuilt functions

**main():**

The execution of the program starts from the main().

**glutInit():**

Initializes GLUT. The argument from main are passed in and can be used by the application.

**glutInitWindowsSize():**

Specifies the initial height and width of the window in pixels**.**

**glutCreateWindow():**

Creates the window on the display. The string can be used to label the window **.**

**glutDisplayFunc():**

Registers the display function that is executed when the window needs to redraw.

**glutMainLoop():**

Causes the program to enter an event processing loop.

**init():**

This function is defined to initialize the window parameters**.**

# CHAPTER 6

## SNAPSHOTS

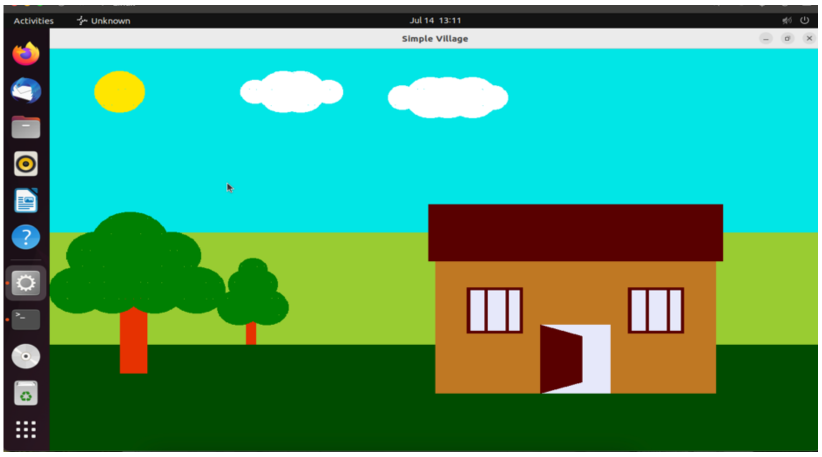

**Fig 6.1: Day Mode in the Smart Village**

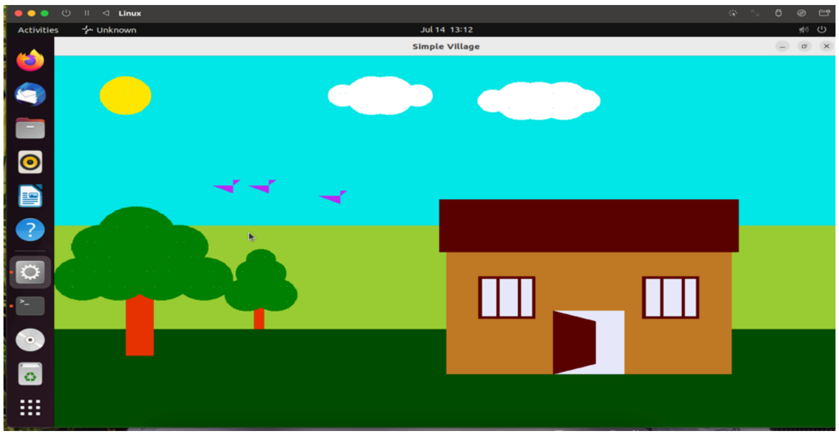

**Fig 6.2: Day Mode with the movement of Birds**

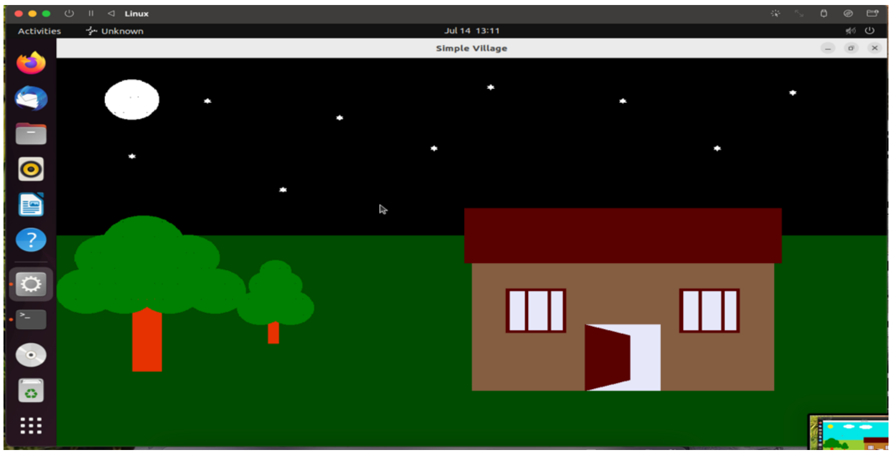

**Fig 6.3: Night Mode with menu Comet options**

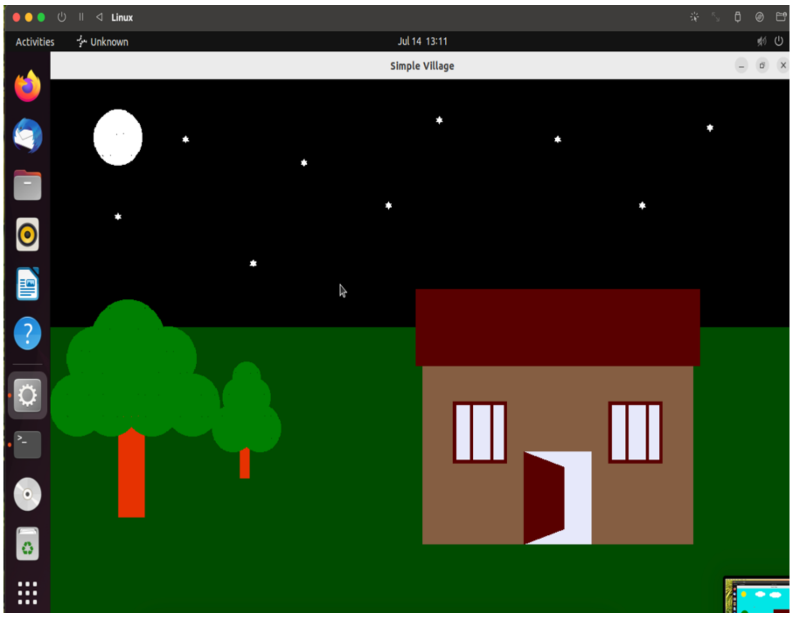

**Fig 6.4: Night Mode with Comet**

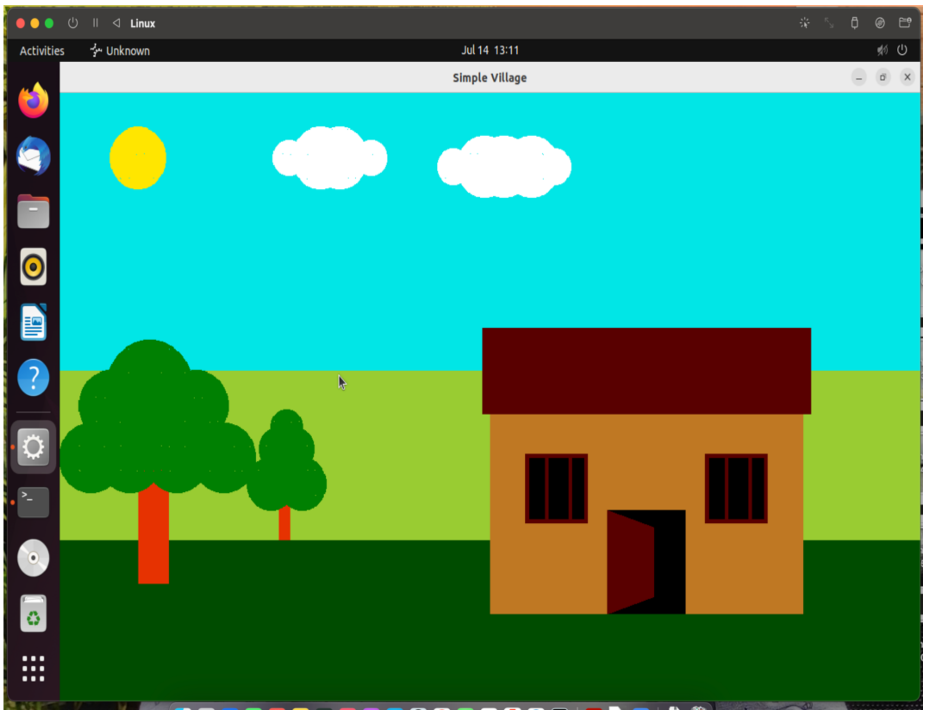

**Fig 6.5: Day Mode with house lights OFF**

# CONCLUSION

This project helped me learn a lot about the proper utilization of various graphics library functions that are defined in GLUT, GLU and GL. The package can be used for educational purposes to demonstrate the various curves, area filling algorithm, non-convex polygon and other OpenGL Primitives.

This project also helped me learn how to design the clouds, trees, house, aeroplane, comet, stars, sun, moon, grass, birds, sky and the flag using the Bezier curve. Also able to move the birds along with the aeroplane, as well as there is movement of turning lights on and off. The flag animation has also been added.

# FUTURE ENHANCEMENTS

The currently developed project can improve in many areas. Time constraint and deficient knowledge has been a major factor in limiting the features offered by this displaymodel.

The following features were thought and will be implemented to enhance the project.

- More interactive feature addition.
- Hardware interface inclusion.
- Using an openGLframework

Creating a more simulated environment to carry forward few calculations made in regard with the architecture of the different rides in the amusement park is a future enhancement.

Simulating the rides in a manner to employ them in AR-VR stimulation for 4D experience.

# REFERENCES

[1] Interactive Computer Graphics A Top-Down Approach with OpenGL-Edward Angel, 6th Edition, Addison-Wesley, 2008.

[2] http://www.opengl-tutorial.org/ - Tutorials.

[3] For textbooks - A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood, N.J., Prentice Hall, 3 Edition, 1975.