VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI -590 014

Mini Project Report On Computer Graphics And Visulization Lab[17CSL67]

"MYSORE PALACE"

submitted in partial fulfillment of the requirement for the award of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by:
ABHISHEK T 4SM17CS002
ARPITHA M 4SM17CS011

Under the Guidance of :
Mrs.M K SHRUTHI M.Tech
Assoc.Prof.,Dept of CS&E,
SJMIT, Chitradurga.

Under the Guidance of :
Mr.PRAVEEN S M.Tech,MISTE,(Ph.D.)
Asst.Prof.,Dept of CS&E,
SJMIT, Chitradurga.

Head of the Department
Dr.Nagabhushana
Prof.,Dept.of CS&E,
SJMJT, Chitradurga.

2019-2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SJM Vidyapeetha®
Sri Jagadguru Mallikarjuna Murugharajendra Institute of Technology
(Affiliated to Visvesvaraya Technological University, Belagavi, Recognized by AICTE, New Delhi and Approved by Government of Karnataka)
P.B. No. 73, NH4 By-pass, Chitradurga -577502, Karnataka State, INDIA.

SJM VIDYAPEETHA®

S.J.M INSTITUTE OF TECHNOLOGY

(Affiliated to the Visvesvaraya Technological University Belagavi,
P.B. No.73, NH-4 By-pass, Chitradurga -577 502, Karnataka State, INDIA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

2019-2020

CERTIFICATE

This to certify that the mini project in Computer Graphics and Visulization[17CSL67] entitled "**MYSORE PALACE**" is a bonfide work carried out by **ABHISHEK T (4SM17CS002)** and **ARPITHA M (4SM17CS011)** in partial fulfillment for the **VI sem** of **Bachelor of Engineering in Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year 2019-2020. It is certified that all corrections/suggestions indicated for the Internal Assessment have been approved as it is satisfies the academic requirements in respect of mini project work prescribed for the **Bachelor of Engineering Degree.**

.....

Signature of the guide
Mrs.M K Shruthi M.Tech
Assoc.Prof. Dept. of CS & E
SJMIT, Chitradurga

.....

Signature of the guide
Mr.Praveen s M.Tech,MISTE,(Ph.D.)
Asst.Prof., Dept. of CS & E
SJMIT, Chitradurga

.....

Signature of the HOD
Dr.Nagabhushana M.Tech., Ph.D.
Prof. Dept. of CS & E
SJMIT, Chitradurga

.....

Signature of the Principal
Dr.B.C.Shanthappa M.Tech., Ph.D.
Principal
SJMIT, Chitradurga

External Viva

Name of the Examiner	**Signature with date**
1) _____	1)_____
2) _____	2)_____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

We consider it as a privilege to express our gratitude and respect to all those who guided us in completion of Project. We would like to convey my pranams and sincere gratitude to his Holiness, ***Dr. Shivamurthy Murugha Sharanaru, President, S.J.M Vidyapeetha®, Sri Bruhanmatha, Chitradurga*** for providing us all the required facilities.

We will remain indebted forever to all the ***Management Authorities of S.J.M Vidyapeetha®, Sri Bruhanmatha, Chitradurga*** and support for carrying out this Project work successfully.

We extend our sincere and heart full thanks to **Dr. B C Shanthappa** **M.Tech., Ph.D.,** **Principal,** for providing us the right ambience, constant inspiration and support for carrying out this Project work successfully.

We extend our sincere and heart full thanks to **Dr. Nagabhushana** **M.Tech., Ph.D.,** **Professor and HOD, Department of Computer science and Engineering,** for providing us the right ambience, constant inspiration and support for carrying out this Project work successfully.

We profoundly indebted to our mini Project Guides **Mrs.M K Shruthi** **M.Tech** and **Mr.Praveen s** **M.Tech,MISTE,(Ph.D.)** **Department of Computer Science & Engineering**, for innumerable acts of timely advice, encouragement and we sincerely express our gratitude to them.

We express our enormous pleasure and thankfulness to all teaching and non-teaching staff of the **Computer Science and Engineering Department.**

Project Associates

ABHISHEK T (4SM17CS002)

ARPITHA M (4SM17CS011)

ABSTRACT

MYSORE PALACE is 2D representation of a palace which shows the person standing in front of the palace and shows the sun rising and sunset where the links from one screen to another screen is given through the mouse it is easy to use and has a graphical user interface and In this Computer graphics it is concerned with all the aspects of producing pictures or images using computer.We can model all the real world objects, render them, give various material properties for them and create animations using computer graphics

Table of content

Chapter 1: Introduction 1-5

1.1 Introduction to Computer Graphic

1.2 OpenGL Interface

1.3 OpenGL Overview

1.4 About Project

Chapter 2: System Requirement Specification 5

2.1 Hardware Requirements

2.2 Software Requirements

Chapter 3: System Design 6-7

3.1 Design

3.2 Data Flow Diagram

3.3 Flow Chart

Chapter 4: Implementation 8-11

4.1 Call back Functions

4.2 GL Functions

4.3 GLUT Functions

4.4 User Defined Functions

4.5 Pseudo code

Chapter 5: Snapshots 12-14

Chapter 6: Conclusion and FutureEnhancement 15

6.1 conclusion

6.2 FutureEnhancement

Chapter 7: Bibliography 16

CHAPTER 1:- INTRODUCTION

1.1 Introduction to Computer Graphic

Computer Graphics is a complex and diversified technology. To understand the technology it is necessary to subdivide it into manageable parts. This can be accomplished by considering that the end product of computer graphics is a picture. The picture may, of course, be used for a large variety of purpose; e.g., it may be an engineering drawing, an architectural rendering for a proposed construction or design project, or a single frame from an animated movie. The picture is the fundamental cohesive concept in computer graphics.

Consider how pictures are represented in computer graphics.

- Pictures are prepared for presentation.
- Previously prepared pictures are presented.

Interaction with the picture is accomplished.
Here “picture” is used in its broadest sense to mean any collection of lines, points, text, etc. displayed on a graphics device.

1.1 Computer Graphics:

Graphics provides one of the most natural means of communicating within a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and effectively. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with the computer, we can make pictures not only of concrete real world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

Computer Graphics started with the display of data on hardcopy plotters and cathode ray tube screens soon after the introduction of computer themselves. It has grown to include the creation, storage, 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 the objects and of their displayed images by using input devices, such as keyboard, mouse, or touch screen. 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. In many design, implementation, and construction processes today, the information pictures can give is visually indispensable.

Scientific visualisation 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 OpenGL Interface

OpenGL is an application program interface (API) offering various functions to implement primitives, models and images. This offers functions to create and manipulate render lighting, colouring, viewing the models. OpenGL offers different coordinate system and frames. OpenGL offers translation, rotation and scaling of objects.

Most of our applications will be designed to access OpenGL directly through functions in three libraries.

They are:

1. Main GL: Library has names that begin with the letter gl and are stored in a library usually referred to as GL.
2. OpenGL Utility Library (GLU): This library uses only GL functions but contains code for creating common objects and simplifying viewing.
3. OpenGL Utility Toolkit (GLUT): This provides the minimum functionality that should be accepted in any modern windowing system.

1.3 OpenGL Overview

1. OpenGL (Open Graphics Library) is the interface between a graphic program and graphics hardware. It is streamlined. In other words, it provides low-level functionality. For example, all objects are built from points, lines, and convex polygons. Higher level objects like cubes are implemented as six four-sided polygons.

2. OpenGL supports features like 3-dimensions, lighting, anti-aliasing, shadows, textures, depth effects etc.

3. It is system-independent. It does not assume anything about hardware or operating system and is only concerned with efficiently rendering mathematically described scenes. As a result, it does not provide any windowing capabilities.

4. It is a state machine. At any moment during the execution of a program there is a current model transformations.

5. It is a rendering pipeline. The rendering pipeline consists of the following steps:
    - Defines objects mathematically.
    - Arranges objects in space relative to a viewpoint.
    - Calculates the color of the objects.
    - Rasterizes the objects.

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. 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. OpenGL is a portable to many platforms (Win, Mac, Unix, Linux). It is also used in video games. OpenGL serves two main purpose

- To hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API.
- To hide the differing capabilities of hardware platforms, by requiring that all implementations support the full OpenGL, feature set.

OpenGL has historically been influential on the development of 3D accelerator, providing a basic level of functionality that is now common in consumer level hardware:

- Rasterized points, lines and polygons are basic primitives.
- A transform and lighting pipeline.
- Z buffering.
- Texture Mapping.
- Alpha.
- Blending.

1.4 About Project

This Project is on **"MYSORE PALACE"** Computer Graphics using OpenGL Functions. It is a User Interactive program where the user can view the required display by making use of input devices like Keyboard. This project mainly consists of two phases. In this we just visualize the Scenario of Mysore Palace.

- The aim of the project is to graphically represent an visiting a Mysore Palace.
- As Windows/Lunix doesn't provide graphics editor, it should be designed in such a way that it provides a very useful graphical implementation interface.
- It should be easy to understand, user interactive interface.
- Providing human interaction through Keyboard.

In this project, we represents the motion of sun and vistors visiting a palace, the Mysore Palace contains in unique of colors and which makes visitors to looks attractivly.

CHAPTER 2:- SYSTEM REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

- Processor : Intel Pentium 4 CPU and higher versions
- RAM : 256 MB and higher
- Hard Disk : 80GB
- Mouse : 3 Button Mouse or Track pad
- Keyboard : QWERTY Keyboard
- Monitor : Standard VGA

2.2 Software Requirements:

- Programming Language : C/C++ using OpenGL
- Operating System : Windows /Linux Operating System
- Compiler : Code blocks/ C Compiler
- Graphics Library : GL/glut.OpenGL 3.0

CHAPTER 3:- DESIGN

3.1 Design

Software design is essential to develop a model of system before writing any software that is used to control the system or to interact with it. During design process we try to develop system models at different levels of abstraction.

Design process involves design of algorithms, modules, components and subsystems. The main aim of the project is to show the phases of hunting that each object passes through in a pictorial form by using OpenGL functions. This project also has some additional features such as we can see the transformation from day to night mode, animation of a scenery. The objects are drawn by using GLUT functions.

3.2 Data Flow Diagram

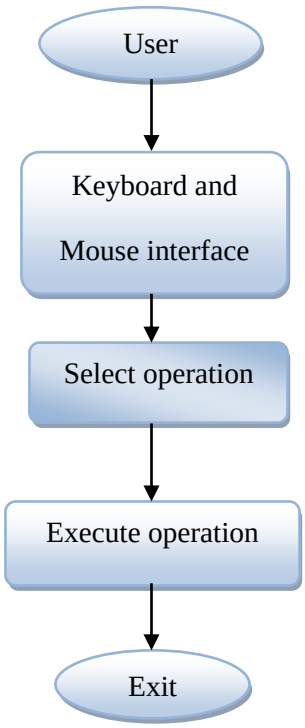


Fig 3.2 Data Flow Diagram

A data flow diagram (DFD)is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create anoverview of the system. A DFD shows what kind of information will be input to and output from the system, how data will advance through the system, and where the data will be stored.

3.3 Flow Chart

Flow chart is the diagrammatic representation of algorithm showing the steps as boxes and their order by connecting with arrows .It shows the flow of execution when different options are specified.

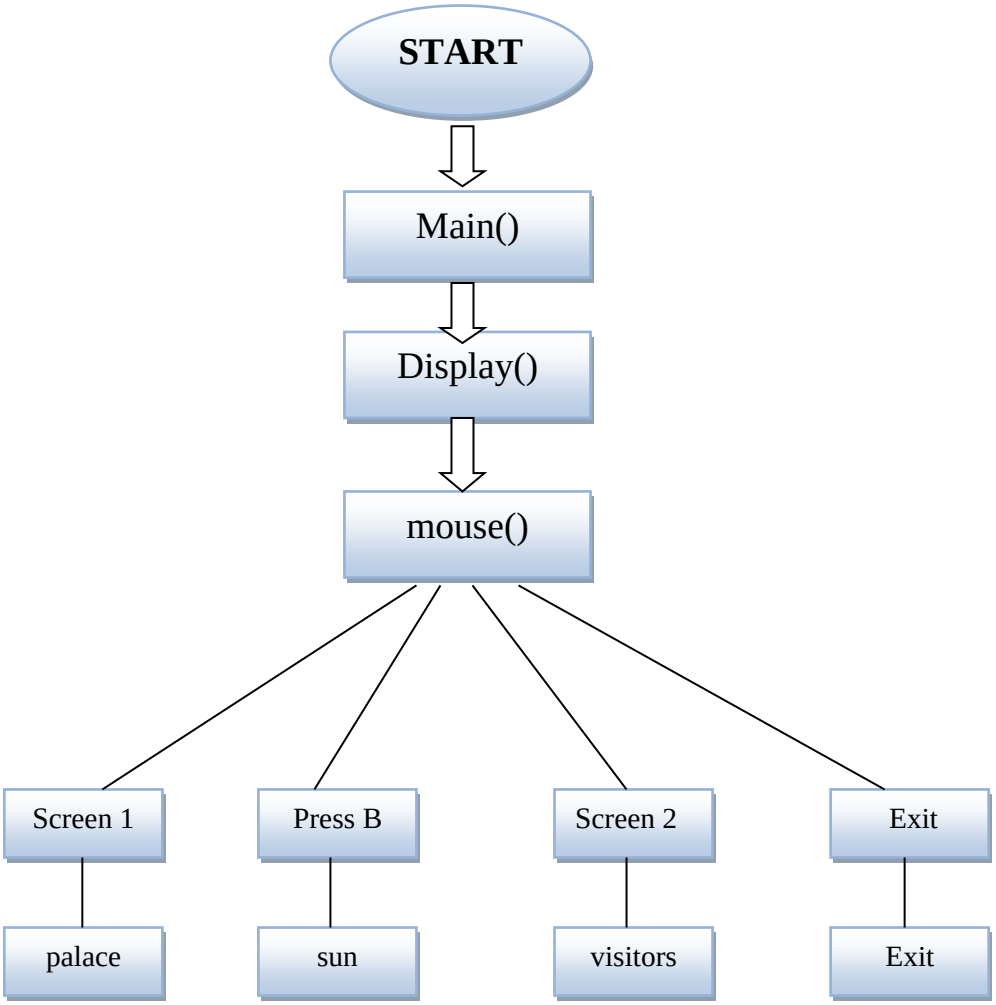


Fig 3.3 Flow chart

CHAPTER 4 :-IMPLEMENTATION

This mini project designed extensively uses the OpenGL graphics library functions. So it would be very much suitable to brief about it at this context. The most commonly used functions of the graphics library are:

- ✓ Call back functions
- ✓ GL functions
- ✓ GLUT functions

4.1 Call back Functions

- **Void display( ):** This function is passed to glutDisplaFunc( ) in order to display OpenGL contents on window.

4.2 GL Functions

- **glMatrixMode( ):** It specifies which matrix will be affected by subsequenttransformations. Mode can be GL_MODEL_VIEW, GL_PROJECTION OR GL_TEXTURE.
- **glPushMatrix( ):** glPushMatrix pushes the current matrix stack down by one, duplicating the current matrix. That is, after a glPushMatrix call, the matrix on top of the stack is identical to the one below it.
- **glLoadIdentity( ):**This function replaces the current matrix with the identity matrix.
- **glPopMatrix( ):** glPopMatrix pops the current matrix stack, replacing the current matrix with the one below it on the stack.
- **glBegin( ),End( ):** This function delimits the vertices or a primitive or a group of liking primitive**s.**
- **glClear ( ):** Clears the buffers to present values.
- **glRotatef( ):** This produces a rotation of angles, degrees around the vector(x, y, z). The current matrix is multiplied by a rotation matrix with the product replacing the current matrix.
- **glTranslatef( ):** This displaces a point to new positions defined by a displacement vector.
- **glPointSize( ):** This specify the diameter of rasterized points.

4.3 GLUT Functions

- **glutDisplayFunc( ):** Graphics are sent to screen through a display callback function and registered with window system. Here the function name will be called whenever the windowing system determines that the OpenGL window needs to be redisplayed.

- **glutInit(&argc,argv):** This is used to initiate a communication between the windowing system and OpenGL. The two arguments enable the programmer to pass command line arguments.
- **glutInitDisplayMode(unsigned int mode):** Requests a display with the properties in mode. The value of mode is determined by the logical OR of options including the color model and buffering..
- **glutCreateWindow( ):** This creates a top-level window. The name will be provided to the window system as the window's name. The intent is that the window system will label the window with the name.
- **glutPostRedisplay( ):** registers that the display callback be executed after the current callback returns .
- **glutMainLoop( ):** Cause the program to enter an event processing loop. It should be the last statement in main function.
- **glutInitWindowSize(int width, int height):** Specifies the initial position of the top left corner of the window in pixels.
- **glutKeyboardFunc(void(*func)(void)):** This function is called every time when you press 'n' or 'N' it goes to next phase, 'b' or 'B' it goes to previous phase, 'e' or 'E' to exit from the application.
- **glutSwapBuffers( ):** This performs a buffer swap on the layer in use for the current window. This promotes the contents of the back buffer of the layer in use of the current window to become the contents of the front buffer.

4.4 User Defined Functions

- **void circle( )**

  {

  //This function is called to draws a sun and palace objects.

  }
- **void palace()**

  {

  //This function draws a palace pillars, wall, arch etc.

  }
- **void doll( )**

  {

  //This function draws a person in front of palace.

  }

- **void flowers( )**

  {

  //This function is called to draw the flowers.

  }

- **void drawCircleFilled( )**

  {

  //This function is to draw a clouds, palace sticks etc.

  }

4.5 Pseudo code

```
#include<stdio.h>
#include<GL/glut.h>
#include<math.h>
#include<string.h>
# define PI 3.142
int i=0;
int disp=0,mainmenu;
void *currentfont;
 float degInRad=0.0;
void drawString(float x,float y,float z,char *string);

void display(void)
{

if(disp==0)
{
    glClearColor(0,0,0,0);
    //glClear(GL_COLOR_BUFFER_BIT);
    text();
//glFlush();
}

if(disp==1)
{
 glClearColor(0,0,0,0);
//glClear(GL_DEPTH_BUFFER_BIT|GL_COLOR_BUFFER_BIT);
screen1();
}
if(disp==2)
{
 glClearColor(0,0,0,0);
//glClear(GL_DEPTH_BUFFER_BIT|GL_COLOR_BUFFER_BIT);
screen2();
}
glutSwapBuffers();
}
```

```
void menu1(int value)
{
if(value==1)
{
disp=1;
glutPostRedisplay();
}
if(value==2)
{
disp=2;
glutPostRedisplay();
}
if(value==0)
    exit(0);
}
void createMenu()
{
mainmenu=glutcreateMenu(menu1);
}
void keys(unsigned char key,int x,int y)        // KEY BOARD //
{
            if(key=='b'||key=='B')
            {
            screen1();
            movesun();
            }
}
 void init()
 {
  glClearColor(0.0,0.0,0.0,0.0);
  glColor3f(0.0,0.0,0.0);
 }
 int main(int argc,char **argv)
 {
  glutInit(&argc,argv);
  glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
  glutInitWindowSize(1300,1300);
  glutInitWindowPosition(0,0);
  glutCreateWindow("palace");
  glutDisplayFunc(display);
  glutKeyboardFunc(keys);
  init();
CreateMenu();
glutAddMenuEntry("SCREEN 1",1);
glutAddMenuEntry("SCREEN 2",2);
glutAddMenuEntry("Exit",0);
glutAttachMenu(GLUT_RIGHT_BUTTON);
  glutMainLoop();
return 0;
}
```

CHAPTER 5 :-SNAPSHOTS

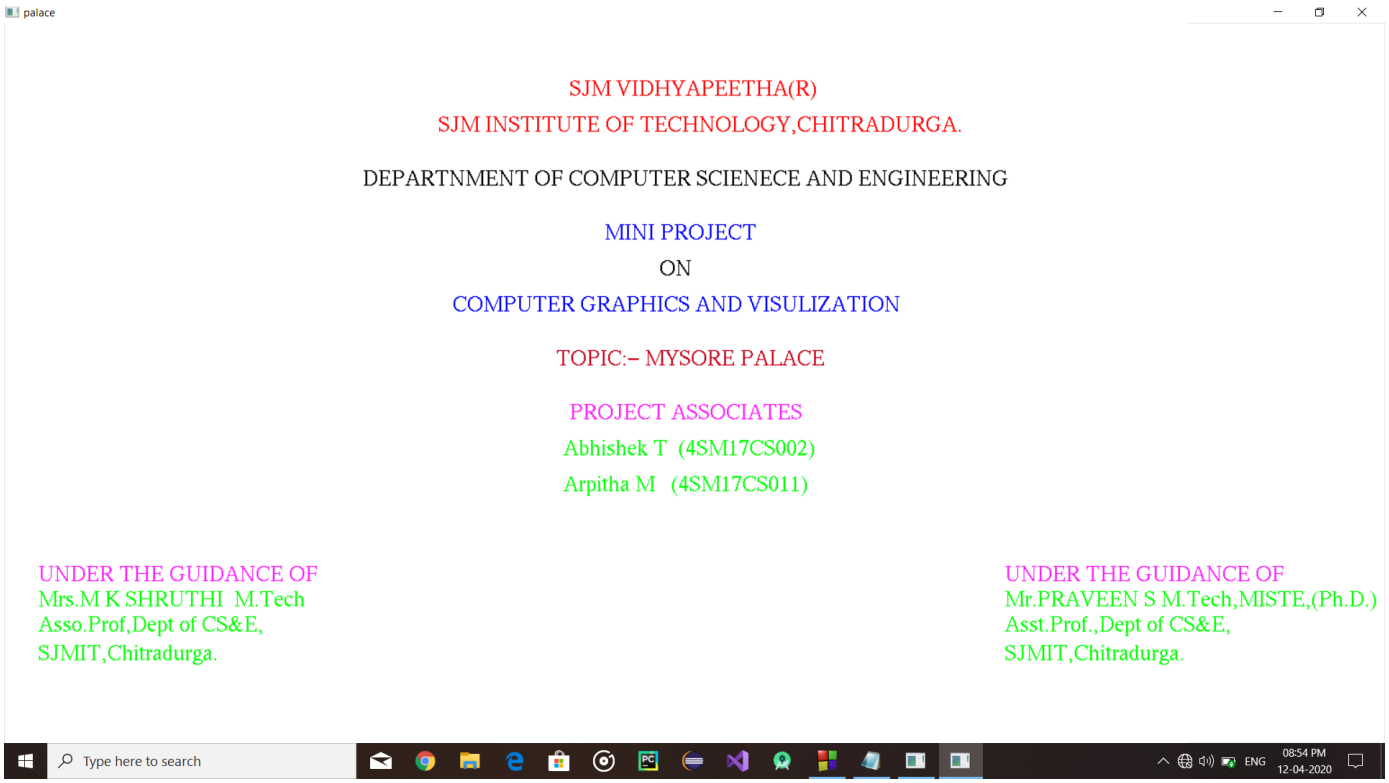


Fig 5.1 Front Screen

Fig 5.1 The above screenshot represents the front screen details such as Project guide, Project coordinator and project assistants

Fig 5.2 Front Menu Screen

Fig 5.2 The above screenshot represents the Menu of the project

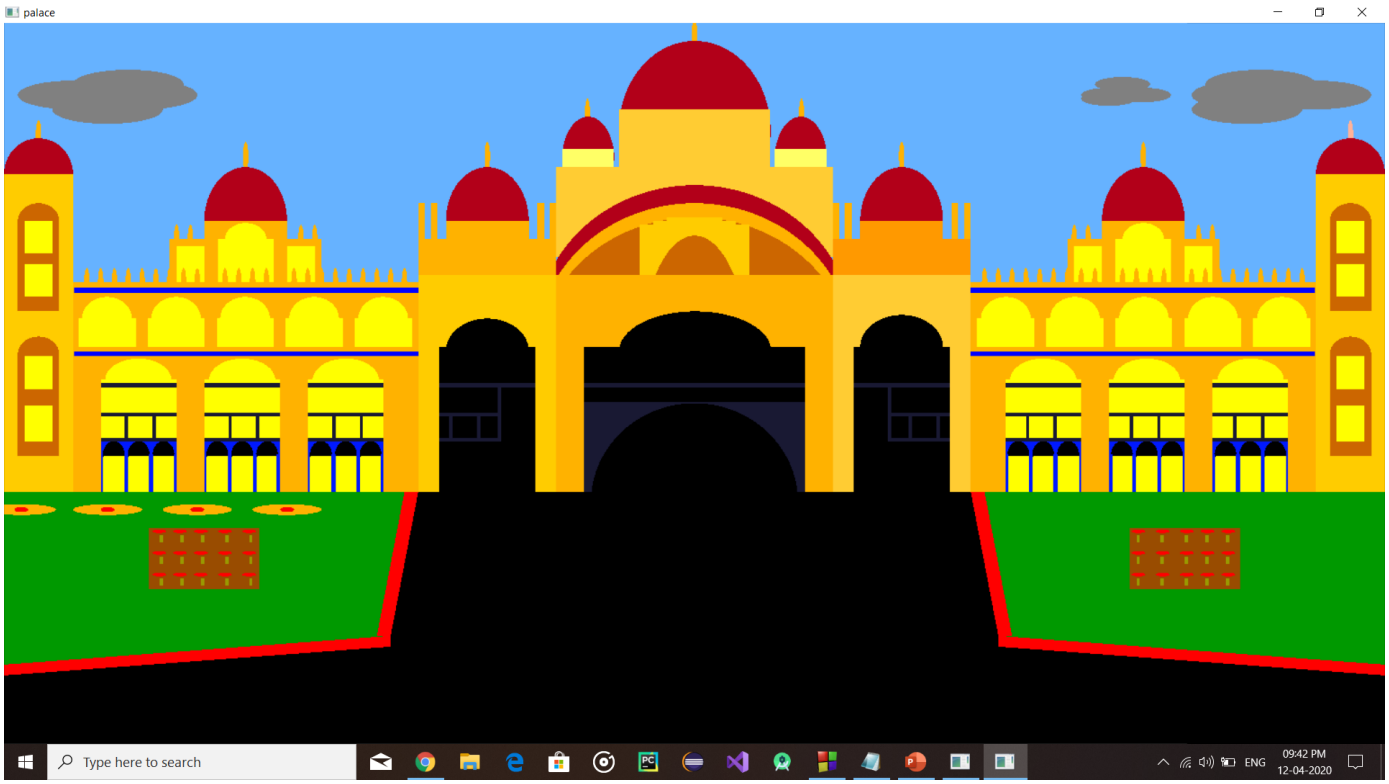


Fig 5.2 Mysore Palace

Fig 5.3 The above screenshot represents the Historical Mysore Palace, One of the oldest empire in the world.

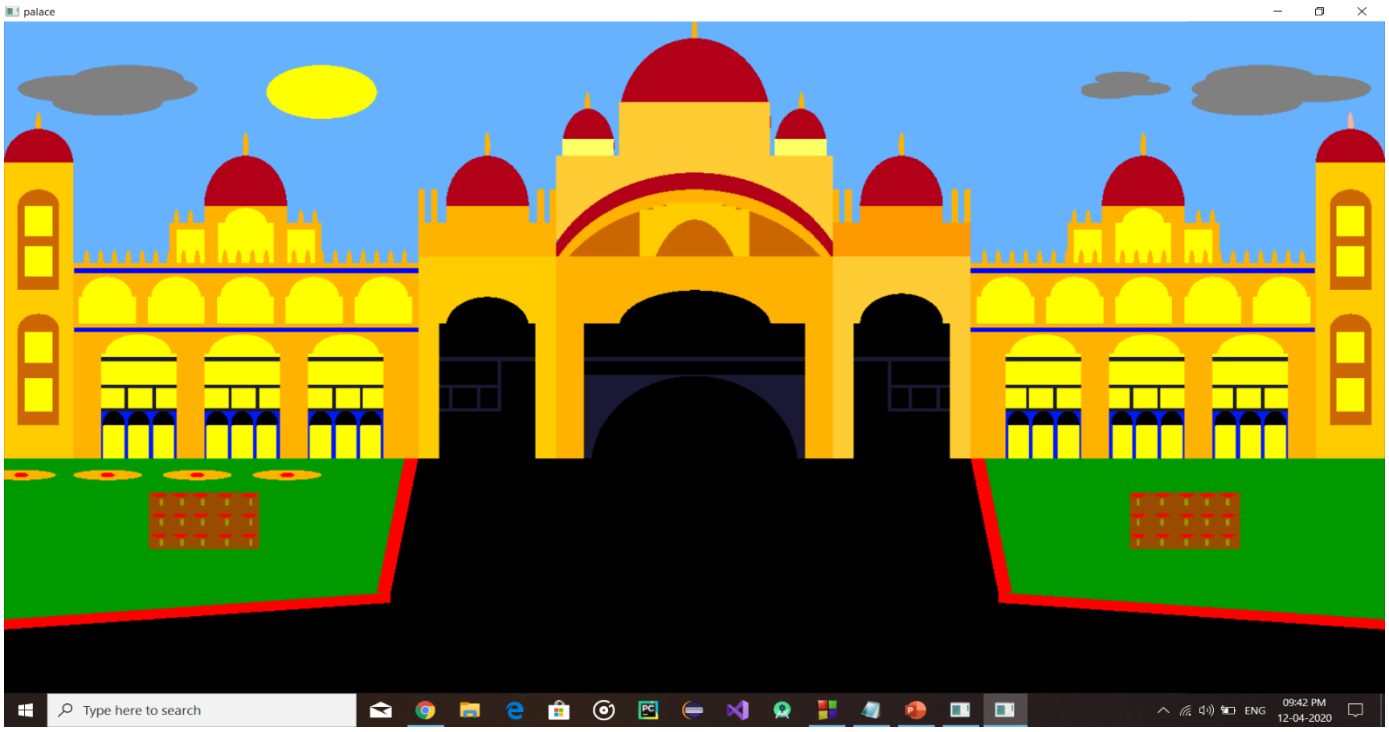
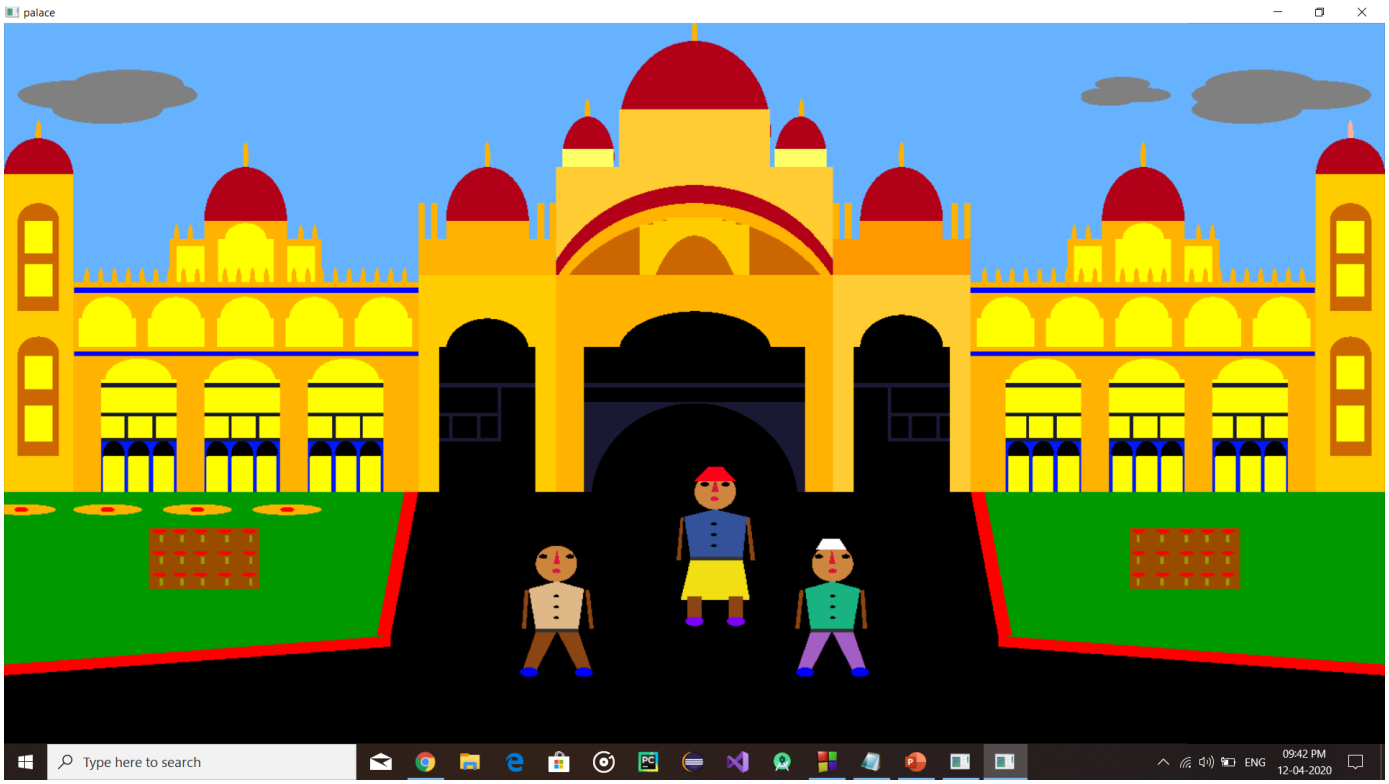


Fig 5.3 Mysore Palace Sun rise

Fig 5.4 The above screenshot represents the Historical Mysore Palace, during the sun rise.



. **Fig 5.4 Visitors visiting Mysore palace**

Fig 5.6 The above screenshot represents the visitors visiting the Historical Mysore Palace

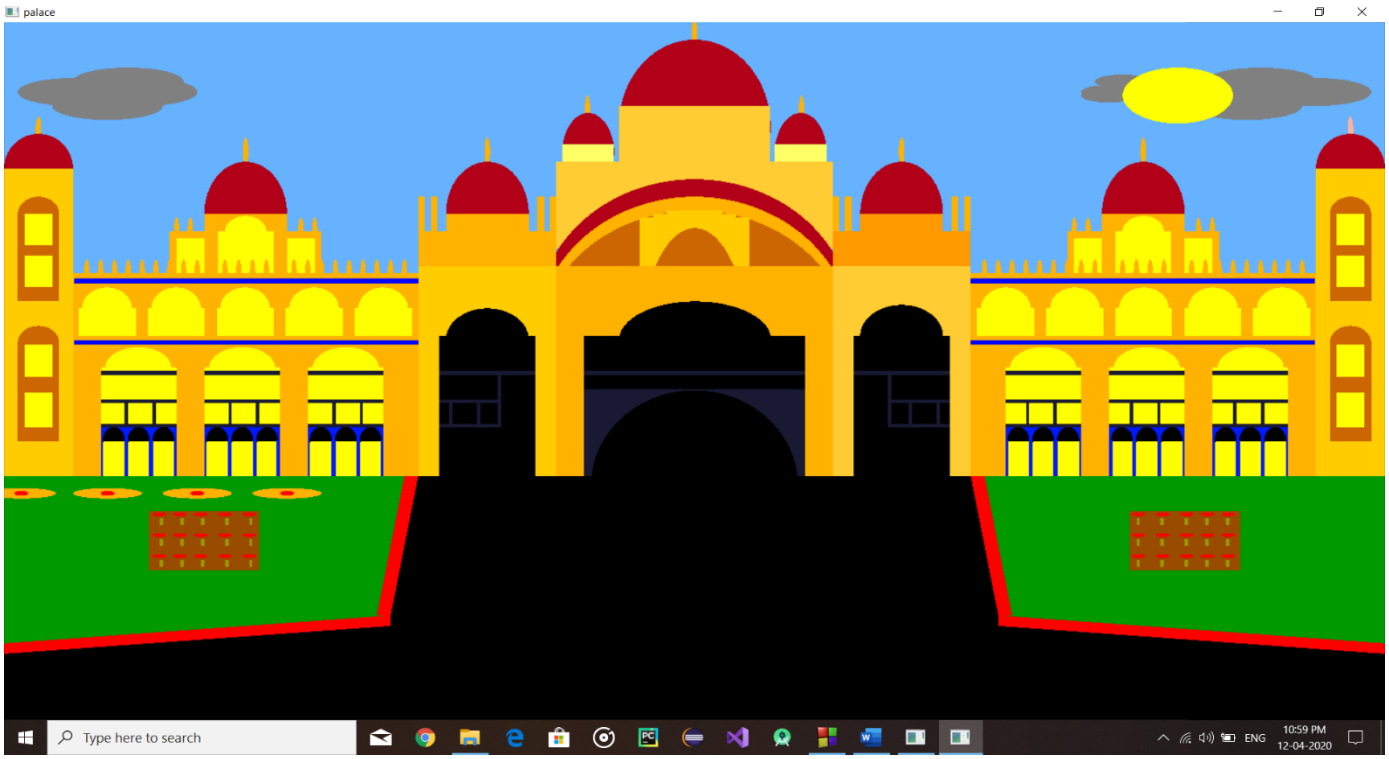


Fig.5.5 Sun Movement

Fig 5.7The above screenshot represents the movement of sun rising from one place to another.

CHAPTER 6:-CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The mini project on “MYSORE PALACE” has been efficiently developed with OpenGL. The illustration of graphical principles and OpenGL features are included and application program is efficiently developed. This project is very user friendly since it provides the basic information about the various OpenGL functions and its component utilities. This is indeed an interactive project which has efficient interactions given through the keyboard as well as the mouse. Blending of colors and many details have been used to make it appear realistic.

The Present project is an interactive project since it is based upon the user’s input. It allows the user to perform some actions. The user can interact with project through the keyboard and Mouse. By using simple functions the interactions has been made easier.

6.2 FUTURE ENHANCEMENT

The project has graphical usage where in different functions which can be applicable in graphical packages are used. The application developed so far is in its basic working stage. There can be more enhancements like designing of detailed 2D buildings with the capabilities of lighting, visitors, and also more details can be added by using other functions and it can also be made more interactive, also we can develop application program for the user on this basis.

CHAPTER 7:-BIBLOGOGRAPHY

[1] Edward Angel, Interactive Computer Graphics: A Top down approach using OpenGL, Pearson Education Asia, Fifth Edition.

[2] Computer Graphics using OpenGL-F.S Hill, Jr. Second Edition.

[3] Computer Graphics-James D Foley, Andres Van Dam, Steven K Feiner, John F Hughes, Addison-Wesley.

[4] Computer Graphics-OpenGL version Hearn and Pauline Baker, Second Edition, Pearson Education.