VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belagavi-590018, Karnataka

A Mini Project Report on

"NATURE SCENERY WITH COLLEGE BUILDINGS"

Submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Engineering

In

Computer Science and Engineering

Submitted by

NAVYASHREE S (4NN20CS030)

PRATHEEK RAJ URS C P (4NN20CS036)

Under the Guidance of

Ms. SHEEBAN E TAMANNA

Assistant Professor

Dept. of CSE

ESTD-2008

Department of Computer Science and Engineering

NIE Institute of Technology

Mysuru -570018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NIE Institute of Technology, Mysuru

ESTD-2008

CERTIFICATE

This is to certify that the mini project work entitled ***"NATURE SCENERY WITH COLLEGE BUILDINGS"*** is carried out by ***NAVYASHREE S*** bearing ***USN:*****4NN20CS030** and ***PRATHEEK RAJ URS C P*** bearing ***USN: 4NN20CS036*** in the partial fulfilment for the sixth semester of **Bachelor of Engineering degree** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2022-23**. The project report has been approved as it satisfies the academic requirements with respect to project work prescribed for the Bachelor of Engineering.

Signature of the Guide

Ms. SHEEBAN E TAMANNA
Asst. Professor,
Dept. of CS & Engineering
NIEIT, Mysuru

Signature of the HOD

Dr. USHA M.S
Associate Professor and Head,
Dept. of CS & Engineering
NIEIT, Mysuru

External Viva

Name of the examiners

1……………………....

2……………………….

Signature with Date

1……………………….

2……………………….

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all people who helped and guided us in completing this project work.

We are thankful to **Dr. Rohini Nagapadma**, Principal, NIEIT, Mysuru, for having supported us in our academic endeavors.

We are thankful to **Dr. Usha M S.** Associate Professor and Head**,** Department of Computer Science and Engineering, NIEIT for providing us timely suggestion, encouragement and support to complete this mini-project.

We would like to sincerely thank our project guide**, Ms. Sheeban E Tamanna.** Asst. Professor in Dept. of Computer Science and Engineering for providing relevant information, valuable guidance and encouragement to complete this mini-project.

We would also like to thank all our **teaching and non-teaching staff members** of **Dept. of Computer Science and Engineering**. We are grateful to the college for keeping labs open whenever required and providing us Systems and Required software.

We are always thankful to our Parents for their valuable support and guidance in every step. Also thank all our friends for their support and guidance throughout the project.

We express our deepest gratitude and indebted thanks to NIEIT which has provided us an opportunity in fulfilling our most cherished desire of reaching our goal.

Yours Sincerely,

NAVYASHREE S (4NN20CS030)
PRATHEEK RAJ URS C P (4NN20CS036)

ABSTRACT

The project "Nature Scenery with College Buildings" is a computer graphics endeavor that aims to create a visually captivating depiction of college buildings amidst a breathtaking natural landscape.

The scene showcases a seamless integration of architectural structures with elements of nature, including a vast sky adorned with majestic clouds, vibrant birds gracefully soaring overhead, and lush grass extending beneath the buildings. The composition captures the serene ambiance of both sunrise and sunset, casting a mesmerizing palette of warm hues that illuminates the entire scene.

Through the seamless fusion of nature and man-made structures, this project seeks to evoke a sense of harmony and tranquility, emphasizing the symbiotic relationship between human creations and the awe-inspiring beauty of the natural world.

TABLE OF CONTENTS

LIST OF FIGURES PAGE NO

CHAPTER 1

INTRODUCTION

1.1 Introduction to Computer Graphics:

Graphics are defined as any sketch or a drawing or a special network that pictorially represents some meaningful information. Computer Graphics is used where a set of images needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer.

Computer Graphics can be used in digital photography, film, entertainment, electronic gadgets, and all other core technologies which are required. It is a vast subject and area in the field of computer science. Computer Graphics can be used in UI design, rendering, geometric objects, animation, and many more. In most areas, computer graphics is an abbreviation of CG. There are several tools used for the implementation of Computer Graphics. The basic is the <graphics.h> header file in
Turbo-C; Unity for advanced and even OpenGL can be used for its
Implementation.

1.2 Overview of Computer Graphics:

The term computer graphics has been used in a broad sense to describe “almost everything on computers that is not text or sound. It is one of the most powerful and interesting facts of computer. There is a lot that you can do apart from drawing figures of various shapes.

Today, computers and computer-generated images touch many aspects of daily life. Computer image is found on television, in newspapers, for example in weather reports, in all kinds of medical investigation and surgical procedures. A well constructed graph can present complex statistics in a form that is easier to understand and interpret. In the media such graphs are used to illustrate papers, reports, and other presentation material.

Many powerful tools have been developed to visualize data. Computer generated imagery can be categorized into several different types: 2D, 3D, 5D, and animated graphics. As technology has improved, 3D computer graphics have become more common, but 2D computer graphics are still widely used. Computer graphics has emerged as a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Over the past decade, other specialized fields have been developed like information visualization, and scientific visualization more concerned with "the visualization of three dimensional phenomena" (architectural, meteorological, medical, biological, etc.), where emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time component).

1.3 History of Computer Graphics:

Computer graphics was first created as a visualization tool for scientists and engineers in government and corporate research centres such as Bell Labs and Boeing in the 1950s. Later the tools would be developed at Universities in the 6os and 70s at places such as Ohio State University, MIT, University of Utah Corn shell, North Carolina and the New York Institute of technology. The early breakthroughs that took place in academic centres continued at research centres such as the famous
Xerox PARC in the 1970's. These efforts broke first into broadcast video graphics and then major motion pictures in the late 70's and early 1980's. Computer graphic research continues today around the production companies. Companies such as George Luca's Industrial light and magic are constantly redefining the cutting edge of computer graphic technology in order to present the world with a new synthetic digital reality.

1.4 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.

Graphs and Charts:

Graphs & charts are commonly used to summarize functional, statistical, mathematical, engineering and economic data for research reports, managerial summaries and other types of publications. Typically examples of data plots are line graphs, bar charts, pie charts, surface graphs, contour plots and other displays showing relationships between multiple parameters in two dimensions, three dimensions, or higher-dimensional spaces.

Computer-Aided Design:

A major use of computer graphics is in design processes-particularly for engineering and architectural systems. CAD, computer-aided design or CADD, computer-aided drafting and design methods are now routinely used in the automobiles, aircraft, spacecraft, computers, home appliances. Circuits and networks for communications, water supply or other utilities are constructed with repeated placement of a few geographical shapes.

Education and Training:

Computer generated models of physical, financial, political, social, and economic & other systems are often used as educational aids. For some training applications, special hardware systems are designed. Examples of such specialized systems are the simulators for practice sessions, aircraft pilots, and air traffic-control personnel.

Entertainment:

Television production, motion pictures, and music videos routinely computer graphics methods. Some television programs also use animation techniques to combine computer generated figures of people, animals, or cartoon characters with the actor in a scene or to transform an actor’s face into another shape.

Computer Art:

The picture is usually painted electronically on a graphics tablet using a stylus, which can simulate different brush strokes, brush widths and colours. Fine artists use a variety of other computer technologies to produce images. To create pictures the artist uses a combination of 3D modelling packages, texture mapping, drawing programs and CAD software etc.

User Interface:

A major component of graphical interface is a window manager that allows a user to display multiple, rectangular screen areas called display windows. It is a common for software packages to provide a graphical interface. A major component of a graphical interface is a window manager that allows a user to display multiple window area. Interface also displays menus, icons for fast selection and processing.

1.5 INTRODUCTION TO OPENGL:

As a software interface for graphics hardware, OpenGL's main purpose is to render two- and three-dimensional objects into a frame buffer. These objects are described as sequences of vertices (which define geometric objects) or pixels (which define images). OpenGL performs several processing steps on this data to convert it to pixels to form the final desired image in the frame buffer.

OpenGL basic (core) library: A basic library of functions is provided in OpenGL for specifying graphics primitives, attributes, geometric transformations, viewing transformations, and many other operations.

Basic OpenGL Syntax:

Function names in the OpenGL basic library (also called the OpenGL core library) are prefixed with gl. The component word first letter is capitalized. For eg:- glBegin, glClear, glCopyPixels, glPolygonMode.

Symbolic constants that are used with certain functions as parameters are all in capital letters, preceded by "GL", and component are separated by underscore. For eg:- GL_2D, GL_RGB etc. The OpenGL functions also expect specific data types. For example, an OpenGL function parameter might expect a value that is specified as a 32-bit integer. To indicate a specific data type, OpenGL uses special built-in, datatype names, such as GLbyte, GLshort, GLint, GLfloat, GLdouble, GLboolean.

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. It is a state machine. At any moment during the execution of a program there is a current model transformation. It is a rendering pipeline. The rendering pipeline consists of the following steps:

- Define objects mathematically.
- Arranges objects in space relative to a viewpoint.
- Calculates the color of the objects.

OpenGL serves two main purposes:

- 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.

1.5.1 Features of OpenGL:

Stable:

OpenGL implementations have been available for more than seven years on a wide variety of platforms. Additions to the specification are well controlled, and proposed updates are announced in time for developers to adopt changes. Backward compatibility requirements ensure that existing applications do not become obsolete.

Reliable and portable

All OpenGL applications produce consistent visual display results on any OpenGL API- compliant hardware, regardless of operating system or windowing system.

Scalable:

OpenGL API-based applications can run on systems ranging from consumer electronics to PCs, workstations, and supercomputers. As a result, applications can scale to any class of machine that the developer chooses to target.

Easy to use:

OpenGL is well structured with an intuitive design and logical commands. Efficient OpenGL routines typically result in applications with fewer lines of code than those that make up programs generated using other graphics libraries or packages. In addition, OpenGL drivers encapsulate information about the underlying hardware, freeing the application developer from having to design for specific hardware features.

Well-documented:

Numerous books have been published about OpenGL, and a great deal of sample code is readily available, making information about OpenGL inexpensive and easy to obtain.

1.5.2 THE OPENGL INTERFACE:

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

GL – Graphics Library:

Functions in the main GL (or OpenGL in Windows) library have names that begin with the letters gl and are stored in a library usually referred to as GL (or OpenGL in Windows).

GLU – Graphics Utility Library:

This library uses only GL functions but contain code for creating common objects and simplifying viewing. All functions in GLU can be created from the core GL library but application programmers prefer not to write the code repeatedly. The GLU library is available in all OpenGL implementations; functions in the GLU library begins with the letters glu.

GLUT – OpenGL Utility Toolkit:

To interface with the window system and to get input from external devices into our programs we need at least one more library. For the X window System, this library is called GLX, for Windows, it is wgl, and for the Macintosh, it is agl. Rather than using a different library for each system, we use a readily available library called the OpenGL Utility Toolkit (GLUT) , which provides minimum functionality that should be expected in any modern windowing system.

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 wx 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. Now, presently with version-10.05.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Software Requirements:

- Programming language – C++ using OpenGL
- Compiler/IDE – Code::Blocks IDE
- Operation system – Windows 8 or above
- Graphics library – GL/glut.h
- OpenGL 2.0

2.2 Hardware Requirements:

- Intel®i5
- 6GB RAM
- Hard Disk: 40 GB
- Hard disk speed in RPM: 5400 RPM
- Mouse: 2 or 3 Button mouse
- Monitor: 1024 x 768 display resolution
- Keyboard: QWERTY Keyboard

CHAPTER 3

SYSTEM DESIGN

3.1 INITIALIZATION:

- Initialize to interact with the Windows.
- Initialize the display mode that is double buffer and RGB colour system.
- Initialize window position and window size.
- Initialize and create the window to display the output.

3.2 DISPLAY:

- Sunrise Page.
- Keys will be provided.
- Suitable operations will be performed According to your need.
- Sunset Page

CHAPTER 4

ABOUT THE PROJECT

Main aim of this project is to illustrate the concepts of Nature Scenery with College Buildings in OpenGL. The objects are drawn using the GLUT functions. This project has been developed using Code::Blocks IDE on Windows 10 operating system with OpenGL package

4.1 Overview:

Our computer graphics project titled "Nature Scenery with College Buildings" aims to create a captivating and immersive digital representation of a college campus harmoniously integrated with the beauty of the natural environment. Through meticulous attention to detail and innovative design techniques, we seek to showcase the seamless blend of architectural brilliance with elements such as the vast sky, gracefully soaring birds, ethereal clouds, mesmerizing sunrises or sunsets, and vibrant green grass. By combining the serene tranquility of nature with the educational realm of a college campus, our project endeavors to inspire viewers to appreciate the delicate balance between human creativity and the awe-inspiring beauty of the natural world, fostering a sense of environmental consciousness and preservation.

In an era marked by rapid urbanization and growing environmental concerns, it becomes imperative to explore and celebrate the harmonious coexistence of human-made structures with the natural environment. Our computer graphics project aims to demonstrate this delicate balance through the creation of a visually stunning and immersive representation of a college campus nestled within a captivating natural backdrop. By meticulously combining elements of nature, such as the sky, birds, clouds, sunrise or sunset, and grass, with the architectural beauty of college buildings, we strive to illustrate the profound connection between human creativity and the magnificent wonders of the natural world.

The Majesty of the Sky : The sky plays a vital role in our nature scenery, serving as a canvas that showcases the ever-changing beauty of nature. Through dynamic sky rendering techniques, we simulate the progression of time, transitioning seamlessly from the soft hues of dawn to the vibrant colors of sunset. The patterns of clouds add a sense of dynamism to the scene, and contributing to the overall visual richness. By accurately representing the sky's colors, and atmospheric effects, we aim to evoke a sense of awe and appreciation for the vastness and grandeur of the natural world.

Graceful Birds in Flight : Birds soaring through the sky bring an element of life to the nature scenery with college buildings. These animated elements add a dynamic and lively dimension to the project, evoking a sense of freedom, harmony, and the interconnectedness of all living beings within the natural world.

Capturing the Beauty of Sunrise and Sunset : The serene beauty of sunrises and sunsets casts a captivating aura over the nature scenery with college buildings. Through intricate color schemes, we aim to recreate the breathtaking hues that fill the sky during these magical moments. The blue and warm golden tones of a sunrise or the vivid oranges and red of a sunset create a visually stunning backdrop, enhancing the overall aesthetic appeal of the scene and conveying a sense of tranquility, inspiration, and the promise of new beginnings.

Lush Grass and Nature's Abundance : The presence of lush green grass is an integral part of our nature scenery, symbolizing vitality, growth, and the harmony between human structures and the natural environment. By utilizing modeling and texturing

4.2 Objective:

The primary objective of our computer graphics project, "Nature Scenery with College Buildings," is to create a visually captivating and immersive digital representation that seamlessly integrates the architectural beauty of college buildings with the enchanting elements of nature. Through this project, we aim to achieve the following goals:

1. Realism and Immersion: Our objective is to create a nature scenery with college buildings that is visually realistic and immersive, enabling viewers to feel a deep sense of connection with the environment. By employing advanced rendering techniques, lifelike textures, and accurate lighting simulations, we strive to transport viewers into a virtual world where the boundaries between the man-made structures and the natural elements blur.

2. Harmonious Integration: We seek to demonstrate the harmonious integration of human creativity and the natural world by showcasing how college buildings seamlessly coexist with the surrounding environment. Our objective is to design and render the buildings in a way that complements and enhances the beauty of nature, rather than overpowering or detracting from it. Through careful attention to architectural details and the integration of sustainable features, we aim to promote an ethos of environmental consciousness and sustainability.

3. Dynamic Elements: To create a sense of dynamism and realism, our objective is to incorporate dynamic elements such as a transitioning sky, soaring birds, and clouds. By simulating the passage of time, the movement of Sun, and the patterns of clouds, we aim to evoke a sense of life, energy, and the ever-changing nature of the environment.

4. Emotional Impact: We aim to evoke emotional responses from viewers through our project. By capturing the breathtaking beauty of sunrises and sunsets, the serenity of the sky, and the vibrancy of nature, our objective is to elicit feelings of awe, tranquility, and a deep appreciation for the natural world. We hope to inspire viewers to develop a stronger connection with nature, fostering a sense of responsibility and a desire to protect and preserve our environment.

5. Educational Insight: Through our project, we aspire to provide an educational insight into the coexistence of human architecture and nature. By showcasing the integration of college buildings within a natural setting, we aim to highlight the importance of sustainable design, eco-friendly practices, and the balance between human progress and environmental preservation. Our objective is to inspire a broader understanding and appreciation of the symbiotic relationship between human-made structures and the natural world.

4.3 USER INTERFACE:

The Project which we have done uses OpenGL functions and is implemented using C++.

Our Project is to demonstrate **Nature Scenery with College Buildings**. User can perform operations using Mouse.

Mouse Interaction

- Firstly, after compiling we get directly Main window Page.
- Then we click the maximize button to display the Main window here the sunrise view is presented.
- If we right click on the Screen we get options in which user has to specify his choices:
- We need to double click on the options to see the proper changes.
- Sunrise option for sunrise view
- Sunset option for sunset view
- EXIT option to exit the screen.

4.4 OpenGL functions:

- glutInit(): Interaction between the windowing system and OpenGL is initiated.
- glutInitDisplayMode(): Used when double buffering is required and depth information is required.
- glutCreateWindow(): This opens the OpenGL window and displays the title at top of the window.
- glutInitWindowSize(): Specifies the size of the window.
- glutInitWindowPosition(): Specifies the position of the window in screen coordinates.
- glutKeyboardFunc(): This function handles normal ascii symbols.
- glutSpecialFunc(): This function handles special keyboard keys.
- glutReshapeFunc(): Setup the call back function for reshaping the window.
- glutIdleFunc(): This handles the processing of the background.
- glutMainLoop(): This starts the main loop, and it never returns.
- glViewport(): Used to setup the viewport.

- glVertex3fv(): Used to setup the points or vertices in three dimensions.
- glColor3fv(): Used to render colour to faces.
- glFlush(): Used to flush the pipeline.
- glutPostRedisplay(): Used to trigger an automatic redraw of the project.
- glMatrixMode(): Used to setup the required mode of the matrix.
- glLoadIdentity(): Used to load or initialize to the identity matrix.
- glTranslatef(): Used to translate or move the rotation centre from one point to another in three dimensions.
- glRotatef(): Used to rotate an object through a specified rotation angle.
- glBegin() , glEnd() : delimit the vertices of a primitive or a group of like primitives.

CHAPTER 5

INTRODUCTION TO TESTING

TESTING:

Verification and validation are a generic name given to checking processes, which ensures that the software conforms to its specifications and meets the demands of users.

- Validation: Are we building the right product? Validation involves checking that the program has implanted meets the requirement of the users.
- Verification: Verification involves checking that the program confirms to its specification.

Results:

Several errors were detected and rectified and the whole project is working as it should have to work with proper output and high efficiency.

Test case ID	Test case	Steps to execute the test case	Expected result	Actual result	Pass/ Fail
1	Window Screen	Double Click on ‘Sunrise’ option	Changes to Rising Sun	Changes to Rising Sun	Pass
2	Window Screen	Double Click on ‘Sunset’ option	Changes to Sundown	Changes to Sundown	Pass
3	Window Screen	Double Click on ‘EXIT’ option	Exit from the Screen	Exit from the Screen	Pass

Table 5.1 Testing for the “Nature Scenery with College Buildings” Project

CHAPTER 6

SNAPSHOTS

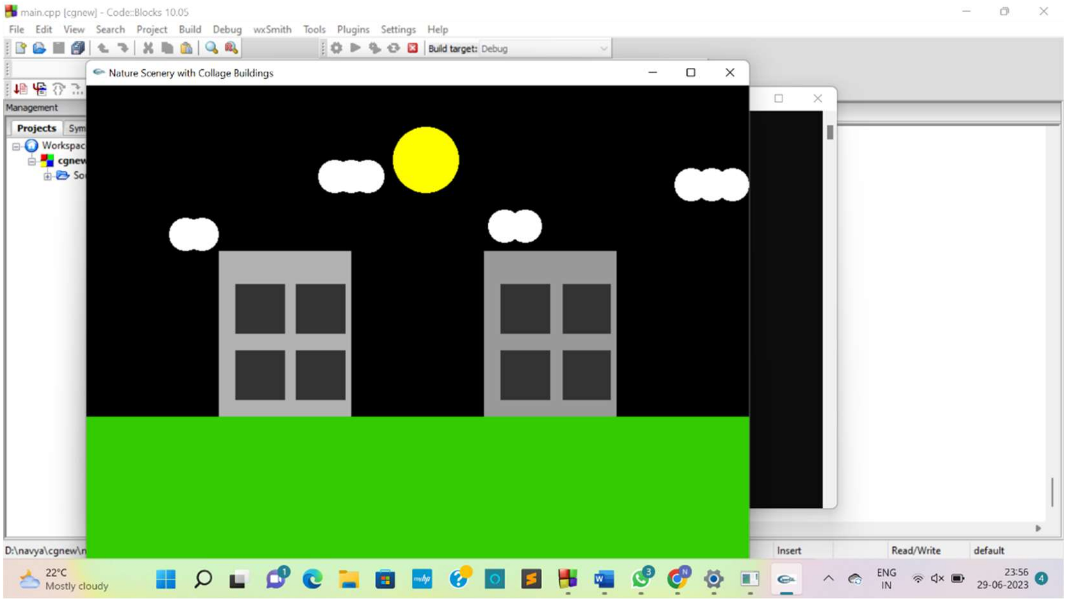
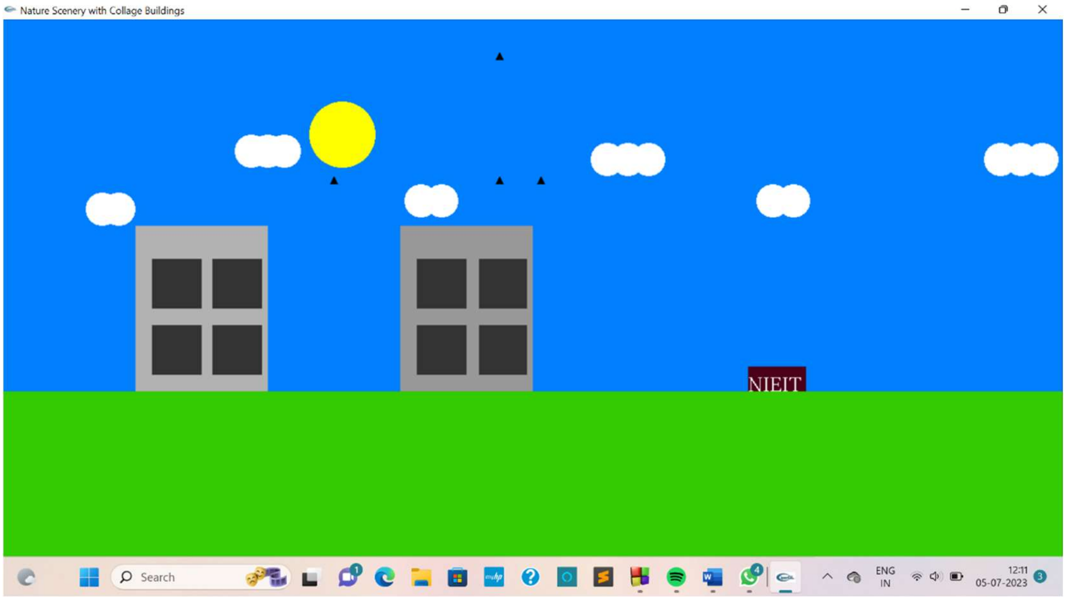

main.cpp [cgnew] - Code::Blocks 10.05
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
Build target: Debug
Nature Scenery with Collage Buildings
Management
Projects
Workspace
cgnew
D:\navya\cgnew\
Insert
Read/Write
default
22°C
Mostly cloudy
ENG
IN
23:56
29-06-2023

Figure 6.1: Nature Scenery with College Buildings front page

Starting display window when we build the program displaying the overview scene



. Figure 6.2: Nature Scenery with College Buildings Sunrise Window

Sunrise Window that is showing rising sun, birds, clouds, buildings with blue sky.

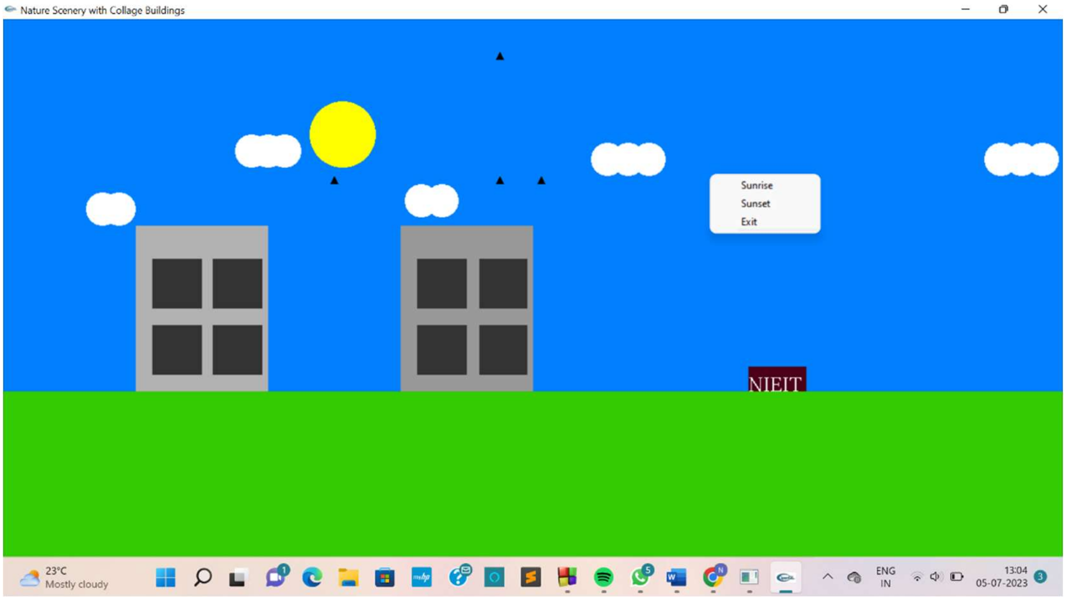


Figure 6.3. Nature Scenery with College Buildings Menu option Window

Window that is showing the Menu options like sunrise, sunset and exit provided for the specific project

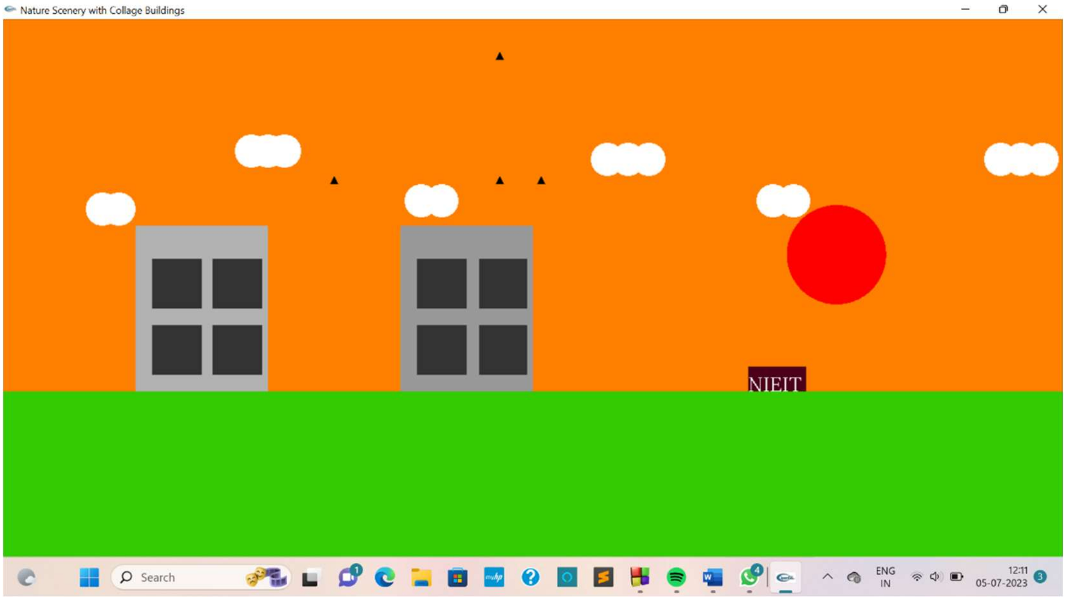


Figure 6.4. Nature Scenery with College Buildings Sunset Window

Sunset Window that is showing falling sun, birds, clouds, buildings with orange sky

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the provided code demonstrates a simple graphical application using OpenGL and GLUT libraries. The application displays a scene with buildings, grass, and a sun that moves across the sky. The user can switch between a sunrise and sunset view using a menu.

The code showcases fundamental concepts of OpenGL, including setting up the rendering environment, drawing basic shapes, applying colors and lighting, handling user input, and implementing basic animation. It provides a starting point for further exploration and development of more complex graphical applications.

FUTURE ENHANCEMENT:

The future enhancements that can be made include:

- Mouse interaction can be improvised.
- User Interaction: Implement more interactive features such as allowing the user to control the movement of objects or change the scene dynamically using keyboard or mouse input.
- Additional Objects: Add more objects to the scene, such as trees, clouds, birds, or other elements to create a more immersive environment.
- Textures: Apply textures to the objects to make them look more realistic and visually appealing. For example, add texture to the buildings, grass, or sky.
- Shaders and Effects: Utilize advanced shading techniques, such as per-pixel lighting, shadow mapping, or post-processing effects like bloom or depth of field, to enhance the visual quality of the scene.
- Multiple Views: Implement multiple camera views to provide different perspectives of the scene. This could include a first-person view or a top-down view, allowing the user to switch between views.

BIBLOGRAPHY

BOOKS:

- Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 3rd / 4th Edition, Pearson Education, 2011.
- Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008.
- James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
- Xiang, Plastock : Computer Graphics , sham‟s outline series, 2nd edition, TMG.
- Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning
- Angel, Edward, and Dave Shreiner. "Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL." Addison-Wesley Professional, 2011.
- Hawkins, David. "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V." Addison-Wesley Professional, 2016.
- Woo, Mason et al. "OpenGL Insights." CRC Press, 2012.

FURTHER LINKS:

- www.google.com
- https://opengl.org/