AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Assignment Title:	**A Village Scenario**		
Report No:	**01**	Date of Submission:	**14 May 2023**
Course Title:	**Computer Graphics**		
Course Code:	CSE4118	Section:	**A**
Semester:	Spring **2022-23**	Course Teacher:	**Mahfujur Rahman**
Group Name/No.:	**01**		

NO	NAME	ID	PROGRAM	SIGNATUIRE
1	MD. SHAHRIAR PARVEZ SHAMIM	**21-44998-2**	BSc CSE	
2	MD. AL FAIAZ RAHMAN FAHIM	**21-45080-2**	BSc CSE	
3	AHAMAD SAFAT	**21-45017-2**	BSc CSE	
4	ASHIK AHAMED	**21-45368-2**	BSc CSE	
6	AMINA TAZIN	**21-45141-2**	BSc CSE	

Faculty use only

FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

A Village Scenario

Introduction:

In this project, we aimed to create a visually engaging and interactive environment that simulates a typical village setting. The village scenario we developed includes various elements such as clouds, boats, mountains, trees, grass, houses, fans, birds, sun, and mountains. Creating such a scenario involved overcoming several challenges, including designing the graphics, implementing the user interaction, and ensuring smooth performance. We employed various techniques to address these challenges, including optimizing the code and using appropriate data structures. This report will provide a detailed overview of the project, discussing the various components and the techniques used to create them. We will also highlight the significance of such a project and the potential educational or entertainment value it may have. Overall, we hope this report will provide insights into the process of creating a village scenario using GLUT on C++.

Literature review:

OpenGL

The Village Scenarios in OpenGL would likely focus on the use of the library for creating realistic and immersive environments that simulate life in a rural community. This could include the use of 3D models to represent buildings, landscapes, and other features of the village, as well as the use of lighting, textures, and other effects to create a convincing atmosphere.

One potential area of research in this field is the use of procedural generation techniques to create realistic village environments. This approach involves using algorithms to generate terrain, buildings, and other features of the environment based on a set of parameters and rules. This can be used to create large-scale environments that are highly detailed and varied, without requiring manual input from the developer.

Another area of research could be the use of virtual reality technologies to create immersive village scenarios. This could involve using head-mounted displays and other hardware to create a fully immersive experience for the user, allowing them to explore the village environment in detail and interact with the various objects and characters in the scene.

Overall, there is a wide range of research opportunities in the area of village scenarios in OpenGL, and the use of the library continues to be an important tool for creating rich and engaging visualizations in a variety of fields.

Windows.h library

Windows.h is a header file in the Windows API (Application Programming Interface) for Microsoft Windows operating systems. The Windows API provides a set of functions, structures, and constants that are used to create Windows applications and interact with the Windows operating system.

The Windows.h header file contains declarations for various functions and data types that are used to interact with the Windows operating system, including functions for creating and managing windows, handling messages, and accessing system information. It also includes data types for representing Windows-specific data, such as handles, messages, and pointers.

GL/glut.h:

GL/glut.h is a header file in the OpenGL utility toolkit (GLUT) library. The GLUT library is a set of tools and functions that make it easier to create graphical user interfaces and handle user input in OpenGL applications. It provides a simplified interface to the lower-level OpenGL API, making it easier for developers to create cross-platform OpenGL applications.

The GL/glut.h header file contains declarations for various functions, data types, and constants that are used in OpenGL applications built using the GLUT library. These include functions for creating and managing windows, handling input events, rendering text, and more.

Math.h:

math.h is a header file in the C Standard Library that provides various mathematical functions and constants. It is used in C and C++ programming to perform mathematical calculations and manipulate numbers.

The math.h header file includes many mathematical functions, such as trigonometric functions (e.g. sin, cos, tan), exponential functions (e.g. exp, log, pow), and rounding functions (e.g. ceil, floor, round). These functions can be used to perform complex mathematical operations in C and C++ programs.

The math.h header file also provides many mathematical constants, such as pi (π), e (the base of the natural logarithm), and infinity. These constants can be used in mathematical calculations and expressions in C and C++ programs.

GL/gl.h:

GL/gl.h is a header file in the OpenGL (Open Graphics Library) API that provides declarations for functions, data types, and constants used in OpenGL applications. OpenGL is a cross-platform graphics API used to render 2D and 3D graphics in computer applications.

The GL/gl.h header file contains the core OpenGL API declarations, including functions for initializing and configuring the OpenGL context, drawing geometric shapes and images, and manipulating the projection and model view matrices.

glPushMatrix():

In computer graphics, "push matrix" is a command used to save the current state of the transformation matrix onto a stack. The transformation matrix is used to apply various transformations, such as scaling, rotation, and translation, to the vertices of a 3D object to position and orient it in the scene.

When a "push matrix" command is issued, the current state of the transformation matrix is saved onto a stack. This allows subsequent transformations to be applied without affecting the original state of the matrix. This is useful, for example, when you want to apply a transformation to only a portion of an object while preserving the original position and orientation of the rest of the object.

glPopMatrix():

In computer graphics, "pop matrix" is a command used to retrieve the previous state of the transformation matrix from a stack. The transformation matrix is used to apply various transformations, such as scaling, rotation, and translation, to the vertices of a 3D object in order to position and orient it in the scene.

When a "pop matrix" command is issued, the previous state of the transformation matrix is retrieved from the stack and becomes the current state. This allows you to restore the original state of the matrix after applying one or more transformations using "push matrix" commands.

Objective Of this project:

The objective of the project was to create a visually engaging and interactive village scenario using GLUT on C++, with the aim of providing a realistic and immersive environment that users can explore and interact with.

Methodology:

The scenario is made with the help of some of the methods and functions. Each section is coded by calling its own method. For example, moving the boat in river with day and night environment, moving the clouds and birds, shifting day and night environment. Here are some different functions for the sound and text section.

renderBitmapString (): This method is used for writing text. It also sets a default text font and font size.

playSound (): This method can be used to play sound effects, background music, or any other audio component in a multimedia application.

renderBitmapString (): This method is used for writing text. It also sets a default text font and font size.

keyboard (): The purpose of the function is to handle keyboard input events and execute different actions based on the key that was pressed. The switch statement inside the function checks the value of the "key" parameter and executes a specific block of code based on its value.

update (): The purpose of the function is to update the graphics display at a fixed time interval. The function is called repeatedly by the event loop, and each time it is called, it requests a redisplay of the graphics window using the "glutPostRedisplay()" function. This causes the display to be redrawn with updated graphics based on any changes that have occurred since the last time the display was drawn.
The function also sets up a timer for the next update by calling the "glutTimerFunc()" function. This causes the "update()" function to be called again after the specified time interval has elapsed (in this case, 35 milliseconds).
Overall, the purpose of the "update()" function is to provide a mechanism for continuous animation or updating of the graphics display in an interactive OpenGL application. By calling "glutPostRedisplay()" and "glutTimerFunc()" at regular intervals, the function ensures that the display is updated and redrawn with new graphics as the application runs.

cloud (): This function is used for moving the cloud.

Bird (): This function is used for moving the Bird.

moveBoat (): The purpose of the function is to move an object (presumably a boat) in the graphics display by a specified amount.

renderBoat (): The function "renderBoat()" is a custom-defined function in a computer graphics program. Its purpose is to render a boat object in the graphics display.

The function likely achieves this by calling a series of OpenGL functions to specify the geometry and texture of the boat, as well as any transformations required to position and orient the boat in the scene. These functions might include:

- glBegin() and glEnd() to specify the beginning and end of a set of vertices that make up the boat's geometry.
- glVertex3f() to specify the x, y, and z coordinates of each vertex in the boat's geometry.
- glTexCoord2f() to specify the texture coordinates of each vertex (if a texture is used).
- glNormal3f() to specify the normal vectors of each vertex (if lighting is enabled).
- glTranslatef(), glRotatef(), and glScalef() to specify transformations that position and orient the boat in the scene.

The specific implementation of the "renderBoat()" function will depend on the details of the boat object being rendered and the overall structure of the computer graphics program. However, in general, the function is responsible for creating the visual representation of the boat in the graphics display.

Features:

1. Moving Clouds with Birds
2. Day & Night Shifting
3. Boat moving depends on Day & Night mode.
4. Sounds for different scenarios.

Significant Of the Project:

A console-based scenario project has several significant benefits includes:

1. Ease of development.
2. Lightweight.
3. Portability.
4. User-friendly.
5. Less memory.
6. No development cost.
7. Customizability.
8. Less system requirements.

Conclusion:

In conclusion, the development of a village scenario using GLUT on C++ was a challenging but rewarding project. We were able to successfully create a visually engaging and interactive environment that simulates a typical village setting. The scenario includes several elements such as clouds, boats, mountains, trees, grass, houses, fans, birds, sun, and mountains. Overall, the project provided a valuable opportunity to explore the potential of using GLUT on C++ to create visually appealing and interactive environments. It demonstrates the importance of careful planning, attention to detail, and effective implementation in the development of computer graphics projects.

Reference:

1. https://www.opengl.org
2. https://www.glprogramming.com/red/chapter01.html
3. https://youtu.be/WbTwwXncb1s

Links of the project:

1. https://github.com/spshamim/GLUT_Project

2. https://youtu.be/oX18H6LDtto

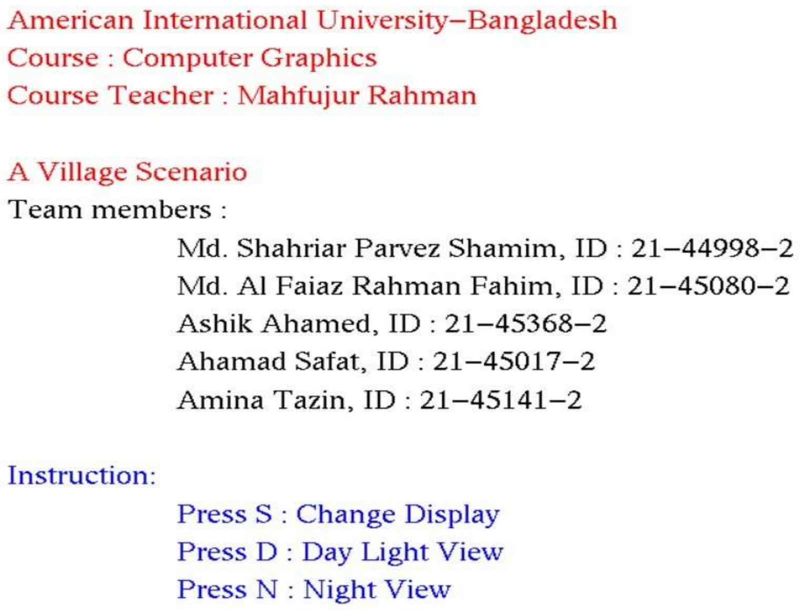
Future Plans:

As it was our console scenario development using OpenGL, we faced some difficulties in adding some features that we would love to have. We are hoped to integrate the features that are given below:

- Change the river view.
- Adding winter Season.
- Adding rain season.
- Adding the bullock cart.
- Adding human on the Boat.
- Adding humans on the Road.
- Change the mountain view.
- Adding vehicles on the road.

Screenshot of each section:

Cover Page Menu and Instruction Menu



American International University–Bangladesh
Course : Computer Graphics
Course Teacher : Mahfujur Rahman

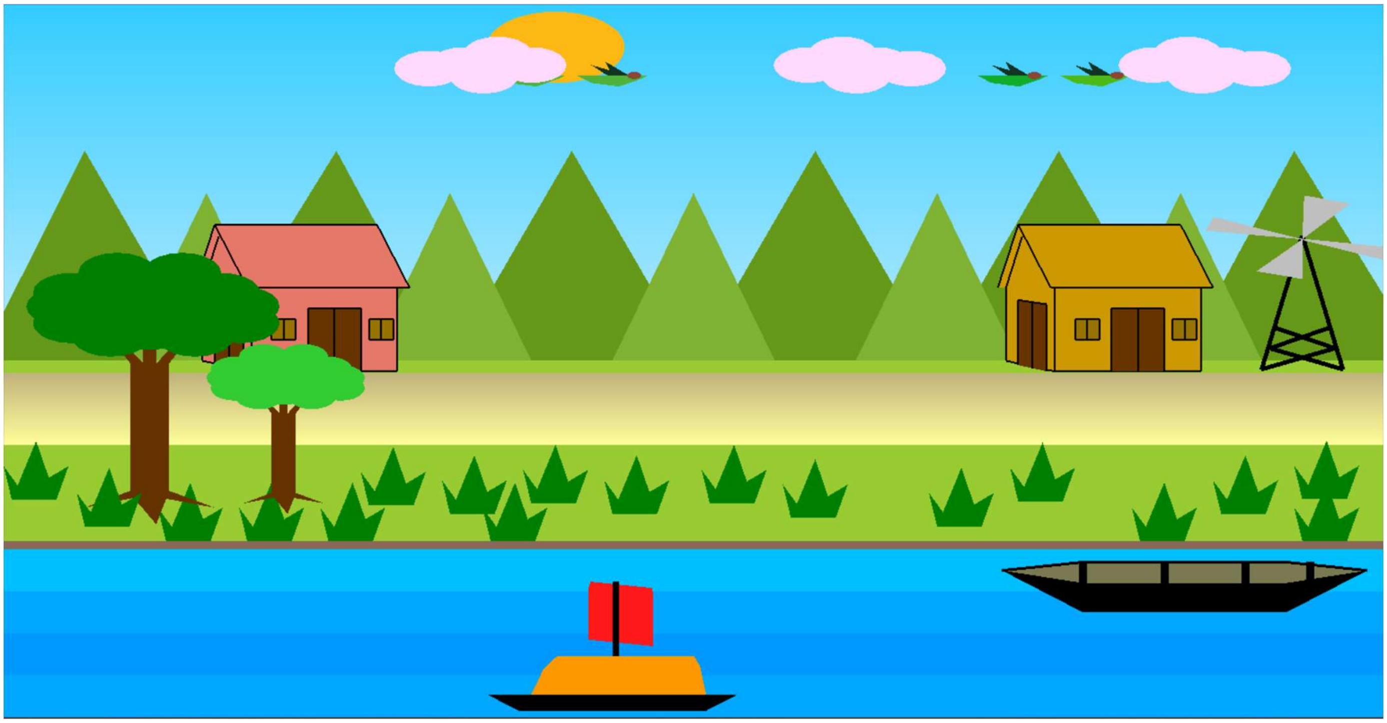
A Village Scenario
Team members :

Md. Shahriar Parvez Shamim, ID : 21–44998–2
Md. Al Faiaz Rahman Fahim, ID : 21–45080–2
Ashik Ahamed, ID : 21–45368–2
Ahamad Safat, ID : 21–45017–2
Amina Tazin, ID : 21–45141–2

Instruction:

Press S : Change Display
Press D : Day Light View
Press N : Night View

Day view



Night view