**Final Project Report**

**Project title:** A landscape view from the remoteness of Bangladesh

**Course:** Computer Graphics

**Sec:** A **Group:** 05

**Faculty:** Md. Kishor Morol

**Project Members:**

**Name**  **ID**

1. Fahim Fayaz 16-32748-3

2. Shihab Uddin 17-34216-1

3. Nazmul Alam 17-34473-2

4. Kazi Shakawat Absar 18-36913-1

**1.INTRODUCTION**

In the project "Ferry Ghat" , the project is programmed using C++.It is a scenario-based project which includes the scenic beauty of the ferry ghat along with some common substances that is usually scene. The most notable part of the objects is moving train on the bridge, a ferry boat which is used to transport the cars from one bank to another. The projects also include a sample cityscape, sky with clouds, the sun and the moon. Some other visible objects are ferry wheel, mountain and rotating windmill.

Keyboard interaction is given for the user to change the speed of the moving objects like cloud, train etc. and also for changing the context like switch to night mode from day mode.

**1.1 Computer Graphics:**

Graphics provides one of the most natural means of communicating within a computer, since our truly highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and effectively.

Due to relationships between the input devices and the display, the handling of such devices is included in the study of computer graphics. The advantages of the interactive graphics are many in number. Graphics provides one of the most natural means of communicating with a computer, since our higly developed 2D or 3D pattern-recognition abilities allow us to perceive and process data rapidly and efficiently. In many design, implementation, and construction processes today, the information pictures can give is virtually indispensable. Scientific visualization became an important field in the 1980s when the scientists and engineers realized that they could not interpret the prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

**1.2 OpenGL Interface:**

OpenGL is an application program interface (API) offering various fuctions to implement primitives, models and images. This offer functions to create and manipulate render lighting, coloring, 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 threee 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 unctions 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:**

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

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

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

⦁ It is a state machine. At any moment the execution of a program there is a current model transformation.

⦁ It is a rendering pipeline. The rendering pipeline consists of the following steps:

i. Defines objects mathematically.

ii. Arranges objects in space relative to a viewpoint.

iii. Calculates the colors of the objects.

iv. Rasterizes the object.

**2.System Specification**

**Functional Requirements:**

**OpenGL APIs:**

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

**GL/glut.h:**

We use available library called the OpenGL Utility Toolkit (GLUT), which provides the minimum functionality that should be expected in any modern windowing system.

The application program uses only GLUT functions and can be recompiled with the GLUT library for other window system. OpenGL makes a heavy use of macros to increase code readability.

**3. ABOUT THE PROJECT**

**3.1 Overview:**

Our project is a simple scenery of representing the landscape view from the remoteness of Bangladesh. A train will be moved at a certain speed. User can control the speed of the train. Clouds can be moved with the keyboard pressing and also can be paused. View of the scenario can be switched to night mode from day mode and vice versa. In the night mode the sun of the day mode will be replaced by a crescent moon.

**3.2 User Interface:**

The interface is mainly concentrated on use of keyboard.

For the day mode the user has to press 'b', for night mode 'n'. To move the cloud 'd' should be pressed and to pause it 'a' should be used. We used 'q' button to stop the train and 'w' to resume its movement. Also, the train can be accelerated and decelerated by pressing 'e' and 'r' respectively.

**3.3 Purpose:**

The aim of the project is to develop a graphical scenario of the real-life objects using OpenGL. This is also includes a user-friendly interface. The objectives of developing this model was to design and apply the skills we learnt in class.

**3.4 Scope:**

It provides most of the graphical features that we have learnt in the class. It is developed in C++ language. It has been implemented on Windows platform.

**4.Implementation**

**4.1 Functions in Open GL:**

⦁ void glClear(glEnum mode):

Clears the buffers namely color buffer and depth buffer mode refers to GL\_COLOR\_BUFFER\_BIT or DEPTH\_BIT.

⦁ void glTranslate(TYPE x, Type y,Type z)

Alters the current matrix by displacement of (x,y,z) TYPE is either GLfloat or GLdouble.

⦁ void glutSwapBuffers()

Swaps the back and front buffers.

⦁ void glLoadIdnetity()

Sets the current transformation matrix to identity matrix.

⦁ void glPushMatrix()

⦁ void glPopMatrix()

Pushes to and pops from the matrix stack corresponding to the current matrix mode.

⦁ void glutInit(int \*argc, char \*\*argv)

Initializes GLUT; the arguments from main are passed in and can be used by the application.

⦁ void glutInitDisplayMode(unsigned int mode)

Requests a display with the properties in the mode, the value of mode is determined by the logical OR of options including the colors model (GLUT\_RGB, GLUT\_INDEX) and buffering (GLUT\_SINGLE, GLUT\_DOUBLE).

⦁ void glutCreateWindow(char \*title)

Creates window on display, the string title can be used to label the window. The return value provides a reference to the window that can be used when the there are multiple windows.

⦁ void glutMainLoop()

Causes the program to enter an event-processing loop.

⦁ void glutDisplayFunc(void(\*func)(void))

Register the display function func that is executed when the window needs to be redrawn.

⦁ void glutKeyboardFunc(void \*f(char key,int width,int height))

Register the keyboard callback function f.The callback function returns the ASCII code of the key pressed and the position of the mouse.

⦁ void glColor3ub(TYPE r,TYPE g,TYPE b)

Sets the present RGB colors.

⦁ void glutInitWindowSize(int width, int height)

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

**4.2 User Defined Functions:**

⦁ int main() function:

The main function is used for creating the window for display of the model of the atom. The callback function registered in main are:

glutDisplayFunc(Display);

glutKeyboardFunc(keyboard);

Some of the notable functions for creating the object:

⦁ void windmill1()

⦁ void windmill2()

⦁ void Train()

⦁ void Traintimer()

⦁ railline1()

⦁ railline2()

⦁ raillineshadow()

⦁ trainshadow()

⦁ nighttrainshadow()

⦁ void tunnel1()

⦁ void tunnel2()

⦁ void ferry()

⦁ void ferryCarshadow()

⦁ void nightferryCarshadow()

⦁ void car1()

⦁ void car2()

⦁ void sun()

⦁ void suntimer()

⦁ void moon()

⦁ void eidmoon()

⦁ void river()

⦁ void skyday()

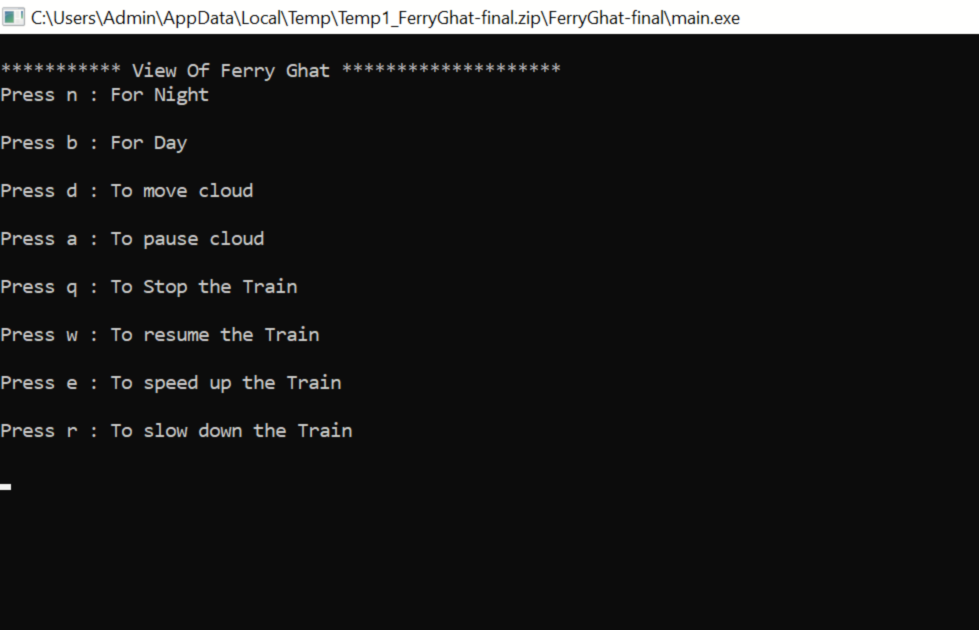
⦁ void cityscape()

⦁ void buildingWindows()

⦁ void ferrytimer()

⦁ void nightWindow()

**5.Snapshots**





**6.Conclusion**

The project was started with no prior experience in any programming projects as this, but ended up in learning many things. We have gained the experience of practicing new programming skills. Due to the participation to this project we have acknowledged with the concept of computer graphics. We have come up with some of the algorithm which we have taught in this course and also, we implement it through our project like circle drawing algorithm. We have tried to fix the bugs that were occurring frequently in this project. Through this we have gained the ability to solve problems. After all, doing this project was actually gaining some lessons for us. We have learnt about teamwork, leading skills and problem-solving skills etc while contributing together.