Daniel Son

Progress Report 3

Program Summary

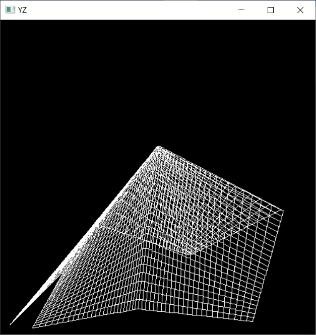
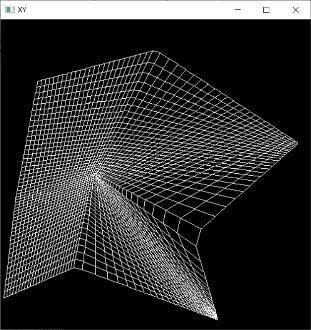
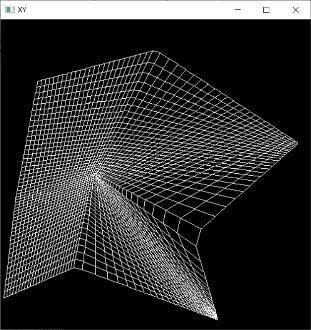
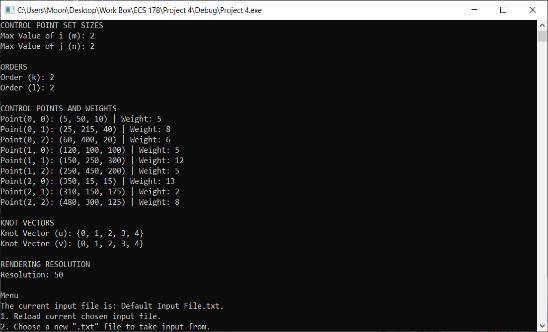
**Compiling and Running**

The program is created using FreeGLUT in Visual Studio. Otherwise, it can also be compiled through CSIF using the compile command “g++ main.cpp -lglut -lGL”, and the corresponding display window can be displayed using “Xming”.

**Using the Program**

1. Menu and Display Windows

The program displays an interactive menu window and three separate display windows for orthographic projections. The display windows are labeled by the title of “XY”, “XZ”, and “YZ” for their corresponding projections.

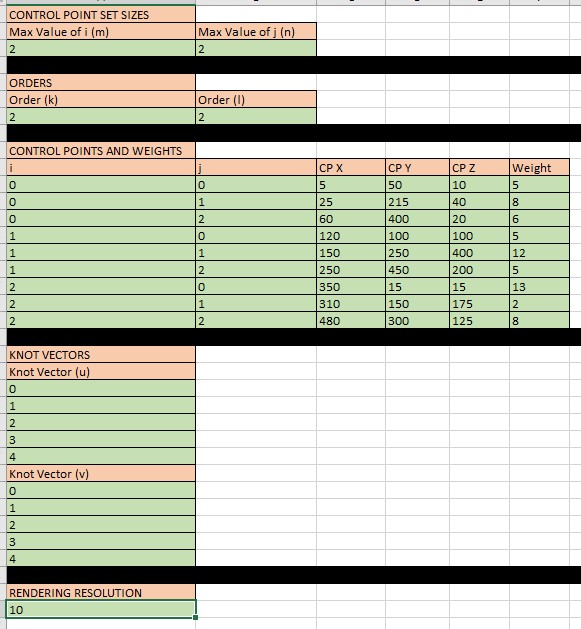


Display Window XY XZ YZ

1. Input Files and Data Management

The program works by taking in input files in the “.txt” format. The default input file is titled, “DefaultInputFile.txt”. It is included in the submission. Any adjustments can be made to an input file by opening the file, making necessary changes, and saving the file. This can be done while the program is running. However, the format must stay consistent. (Explained later). It is recommended that the user edits the file using programs like Excel, because it is easier to visualize.

1. Format for Input Files

This is how “DefaultInputFile.txt” looks when opened in Excel with the exception of color-coded parts. Use it as reference as the format for input files is explained.

The parts in pink are guide text to help label the sections for the user’s convenience. The black bar signifies an extra new line required in between each section. Neither parts should be changed by the user.

The parts in green are blocks in which the user must fill in data.

The first section is “CONTROL POINT SET SIZES”. The following line labels two blocks indicating “Max Value of i (m)” and “Max

Value of j (n)”. **IN THE LINE AFTER, THE USER MUST**

**ENTER A POSITIVE INTEGER VALUE FOR EACH OF THE**

**TWO CORRESPONDING BLOCKS FOR THE VALUE OF**

**“m” AND “n”.** In the case of the default file, m = 2 and n = 2.

The second section is “ORDERS”. The following line labels two blocks indicating “Order (k)” and “Order (l)”. **IN THE LINE**

**AFTER, THE USER MUST ENTER A POSITIVE INTEGER VALUE FOR EACH OF THE TWO CORRESPONDING**

**BLOCKS FOR THE VALUE OF “k” AND “l” for k <= m and l <= n.** In the case of the default file, k = 2 and l = 2.

The third section is “CONTROL POINTS AND WEIGHTS”. The following line labels six blocks indicating “i”, “j”, “CP X”,

“CP Y”, “CP Z”, and “Weight”. **THE NEXT (m+1) \* (n+1) LINES SHOULD BE LINES RESERVED FOR USER INPUT**

**FOR CORRESPONDING CONTROL POINTS. THE BLOCKS UNDER “i” and “j” WILL BE THE i AND j LABEL VALUES OF EACH OF THE CONTROL POINTS. 0 <= i <= m AND 0 <= j <= n. THE NEXT THREE BLOCKS**

**UNDER “CP X”, “CP Y”, AND “CP Z” WILL BE THE X, Y, AND Z VALUES OF THE CORRESPONDING CONTROL POINT LABELED BY i AND j. 0 <= X, Y, Z <= 499. THE LAST BLOCK LABELED “Weight” IS FOR DETERMINING THE WEIGHT VALUE OF THE CORRESPONDING CONTROL POINT.**

The fourth section is “KNOT VECTORS”. The following line labels a block indicating “Knot Vector (u)”. **THE NEXT**

**m + k + 1 LINES SHOULD BE RESERVED FOR THE VALUES OF THE KNOTS IN THE KNOT VECTOR “u”.**

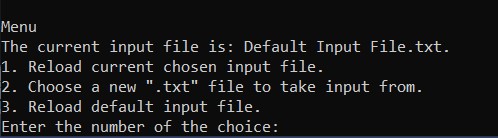
**EACH KNOT VALUE SHOULD BE THE SAME OR GREATER THAN THE PREVIOUS KNOT VALUE.**

The following line labels a block indicating “Knot Vector (v)”. **THE NEXT n + l + 1 LINES SHOULD BE RESERVED FOR THE VALUES OF THE KNOTS IN THE KNOT VECTOR “v”. EACH KNOT VALUE SHOULD BE THE SAME OR GREATER THAN THE PREVIOUS KNOT VALUE.**

The fifth and last section is “RENDERING RESOLUTION”. **IN THE NEXT LINE, THE USER INPUTS THE RENDERING RESOLUTION VALUE IN THE FORM OF A POSITIVE INTEGER.**

4. Menu

In the display window, all the relevant data information should be displayed to the user whenever a new graphic is displayed. Furthermore, the user will be presented with a small menu that allows the user to interact with the input files. The user can enter the number of the menu option to select that option.



Here’s what each option does:

1. Reload current chosen input file.

Reloads the current input text file. Therefore, the user should make changes to the input file, save it, and choose this option to update the graphic.

1. Choose a new “.txt” file to take input from.

Allows the user to enter the name of a new “.txt” file that follows the input file format.

1. Reload default input file.

Changes the selected input file back to “DefaultInputFile.txt”.