

Introduction

This assignment will build on the previous assignment. We will add grid lines and a 1×1 square that will be drawn sequentially in each of the cells created (visually) by the grid lines. The square will be given a color so that we can verify that it was drawn.

User Interface

The default rectangle will be 1×1 . If the user changes the window size, the window boundaries will be recomputed as necessary.

If the user presses the **S** key, then, in the text window, the program will prompt the user to enter the width and height of the rectangle. The program will then draw that rectangle with the grid lines and the colored square in each cell and with the window margins adjusted as necessary. In all cases, the rectangle will remain centered in the window.

If the user presses the escape key, the program will quit.

Program Description

Save a copy of `Margins.cpp` as `Grid.cpp`. Also, save copies of the shader programs as `Grid.vert` and `Grid.frag`, even though they will be unchanged from the previous assignment.

To draw the grid, I recommend that you create a single horizontal line of the length 1.0 and a single vertical line of length 1.0. Each of them will be represented as an array of four `floats`, representing the endpoints. (two x values and two y values). It is possible to create a single line and then rotate it, but that gets unnecessarily complicated. To create the lines, you should follow the pattern set by the creation of the original rectangle. That is, you will need to add an `enum` for each line in the lists of VAOs and VBOs, and so on.

To draw the horizontal grid lines, begin with the lowest one, at the bottom edge of the rectangle. Scale the line to the width of the rectangle. Then use a `for` loop with translations to work your way from bottom to top, drawing the line at each level. In a similar way, draw the vertical lines from left to right.

To draw a colored square in each cell, create a 1×1 rectangle. You probably can use the same 1×1 rectangle that you created to draw the one large rectangle. Then, beginning in the lower-left corner, draw the square. Using nested `for` loops, translate across the row inside the inner `for` loop, drawing the square after each translation. In the outer `for` loop, translate back to the left edge and up one row, and repeat the inner `for` loop.

One question to consider: Should you draw the rectangles first and the grid lines later, or vice versa, or does it not matter?

Due Date

Drop the program `Grid.cpp` and the associated shader programs in the dropbox by midnight Friday, September 20.