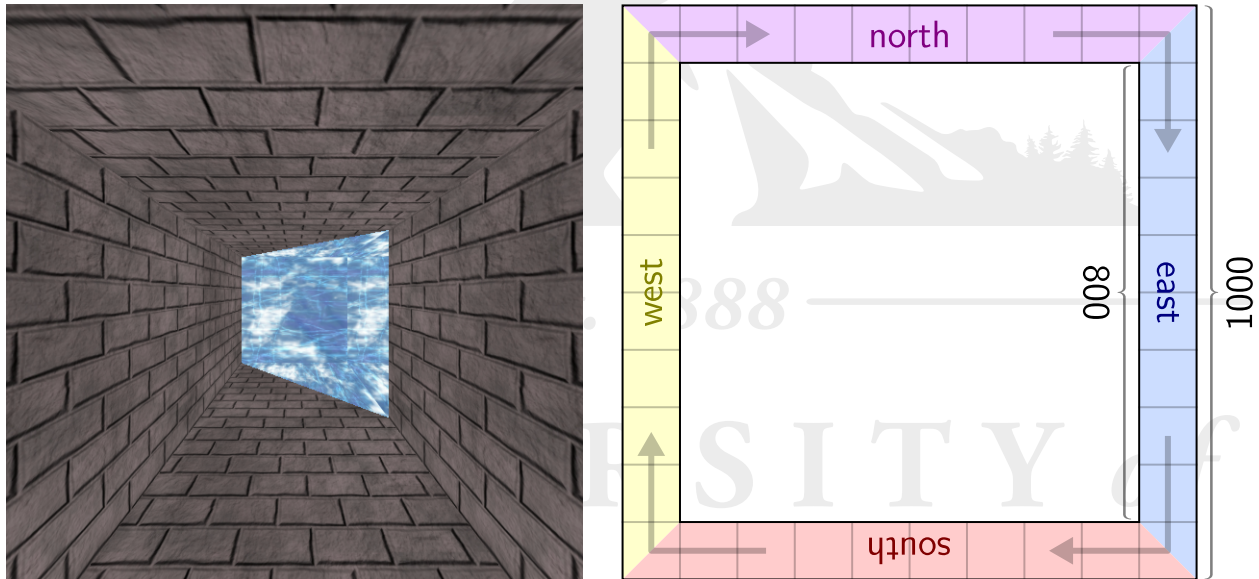


HW 4: Textured Corridor

Your job is to make 4 connecting corridors in a square formation, that will each have a different texture. The camera should then appear to “walk” circuits through the structure in a clockwise direction. The main class you create should be called `Corridor`, and will of course be accompanied by `corridor-vertex.glsl` and `corridor-fragment.glsl`. (If you wish, you may also create helper classes, CSV or TSV text files to hold data, and PNGs and JPEGs to hold textures.)



The corridors will be in a square formation, as shown in the above right figure. The full square map is 1000 units on a side, and the inside width and height of each corridor is 100 units. Each of the 4 regions will have a different texture, as determined by the user at runtime. The user will enter between 5 and 8 command-line arguments, in this order:

1. **Walking period:** a positive number indicating how many seconds it takes to walk from one corner to another.
2. **Turning period:** a positive number indicating how many seconds it takes to turn 90° , when in a corner.
3. **Step count:** a positive integer indicating how many steps are taken from one corner to the next, to be used when bouncing the camera.
4. **Step height:** a positive number indicating how far each step should raise the camera above base level, at its greatest height.
5. **1-4 texture files:** Strings indicating the names of texture files to use, which will be PNGs or JPEGs.

Textures should be placed on each 100×100 region of a surface. Textures on the wall should have the texture’s up direction facing up. Textures on the wall or ceiling should have the texture’s up direction facing away from the camera as it makes its clockwise circuit. Corner

areas will likely have noticeable seams on the floor and ceiling, since they will be painted with two different textures.

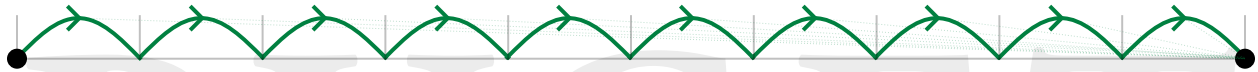
Textures should be seen in the order they were entered by the user. If only one texture is given, it will be used for the whole map. If two are given, the first will be used for regions 1 and 3 (where 1 is the first texture the user sees), and the second for the others. If three are given, the first will again be used for regions 1 and 3, with the second being used for region 2 and the third for region 4. (It may be helpful to label the regions “north”, “south”, “east”, and “west” as in the diagram, but this is not required.)

The camera will always be centered left-to-right in the current hallway. Its base height will be $\frac{2}{3}$ of the height off the floor ($\frac{1}{3}$ down from the ceiling), to simulate the vantage of a person. While walking, it will always be focused on the imaginary point on the far wall, centered left-to-right and at the base height.

To simulate footsteps, the camera should be given a “bounce” according to this formula:

$$b = h |(\sin(kd))|$$

The value b will be added to the camera’s base height. d is the distance from the last corner. h is the maximum step height, as given by the user. k is a constant that you will have to determine, to make sure that the right number of steps happen between each corner. The camera will still stay focused on the same point, no matter where it is in its bouncing cycle. (Thus, the camera will usually be facing *very slightly* downward.)



Note that the bobbing of the camera should be subtle—the bounce height that the user gives should be quite low. There will be no bouncing when the camera rotates. It will simply turn 90° to the right at constant speed, according to the parameter given by the user.

You must optimize by making use of back-face culling.

You do not need to worry about lighting or shadows.

A couple of hints:

- The path of the camera is symmetrical on each of the four sides of the complex. It may help to assume that it is always on a given side, and then to rotate it 90° , 180° , or 270° around the center of the complex after all calculations have been made.
- A good way to test for back-face culling is to turn the bounce height up far too high, so that the camera actually bounces through the ceiling. You should be able to look through the ceiling from above it.
- If you wish to start on this early, do the assignment using solid surfaces rather than textures. Those can be added in later.

Good luck!