Water Vertex Shader

Description:
You have had some practice working with the Fragment shader in WebGL and GLSL. Now we are going to work with the vertex Shader to see how we can move vertices around to displace geometry. In order to do this, we are going to build a water effect and take advantage of the vertex shader.

Your Task:
- Modify the vertex shader to create a water effect.
  - The goal here is to create some sort of 'noise' function that translates our mesh along the 'z-axis' to give a scrolling effect.
    - The noise function should make use of sine and cosine functions in order to update the height of the 'z' position. This gives the illusion of movement, emulating a wave.
      - newX = sin(vertex.x * ...)
      - newY = cos(...)
- Modify the fragment shader to create a scrolling texture
  - The goal here is to pass a variable associated with time that modifies the 's' and/or 't' components of a texture to give a nice scrolling effect.
    - This can be done by modifying one line of code in the fragment shader once you have passed in a variable that is associated with time 😊.

Files Given:
Index.html – Edit the vertex shader in this file and the fragment shader

A Reminder on the GLSL Pipeline

GLSL Refresher
1. sin(float) – Computes sine in glsl
2. cos(float) – Computes cosine in glsl

Going Further:
Did you enjoy this in class assignment?

- Try creating a ripple effect with the water.
- Pass normal data in order to create nice specular highlights on the water.
- Try creating more advanced water here: