Cameron Taylor
CAP 4720

Graphics Project Proposal

Algorithm: SelectionSort(list);
The algorithm that I would like to visualize for my proposal is selection sort. This algorithm is fairly simple to implement, but I think that there is some interesting potential for visualization to help illustrate what is happening behind the scenes. The project will show include an option to generate a random list of numbers to sort and also a manual entry for a list of numbers (up to 10).

Implementation:
Once the list of numbers has been set and the cubes representing those numbers have been generated, the sort button will then be pressed. This will activate the selection sort algorithm which will highlight the current number being compared against in purple, the already sorted values in yellow and the current lowest value found in red. Once a pass has been completed the lowest value (marked in red) and the value being compared against (marked in purple) will be swapped. Then the newly swapped lowest value will be added to the yellow values, marking its sorted status.

JavaScript

1. Create an empty scene.
2. Create DatGUI menu with random button, a sort button and a numeric entry field.
3. Create a cube object that will represent an individual number to be sorted.
4. Create logic to place cubes into the scene along the x-axis with even spacing.
5. Create a selection sort algorithm which sorts the cube objects based on their height field and also updates uniforms to the shader material used for the cube to adjust color based on status within the algorithm.
6. Create an animation algorithm to move cubes based on the status of the selection sort algorithm.

GLSL

1. Set color based on material color and status uniforms coming from selection sort algorithm.

Deliverable:
The final deliverable will be an HTML file which contains the javascript and shader code that will execute the program described above. Based on my current ideas, there should not be any external files such as image or obj.