CSCI 4110U – Assignment 1

Rhys Agombar – 100515147

To construct the terrain, I passed the program a text file containing: A number representing the dimensions of the height field (ie. 30 = 30x30), the scale factor representing the number of iterations for the subdivision algorithm to do, and the actual height field itself. The heights technically have no upper/lower bounds, but numbers between 0 and 12 work best.

The colour of the terrain is determined by its height. Values close to 0 and below are shaded blue (for water), whereas the heights > 9.0 are shaded white (for snow), between 9.0 and 7.0 is grey (for stone), between 7.0 and 4.0 is dark green (for evergreen forest), between 1.0 and 4.0 is light green (for normal forest), and between 0 and 1.0 are khaki (for sand). Gradients have also been implemented, so that the closer the height gets to the upper bound of a colour, the more it bleeds into the next.

The terrain itself is determined by the height map. We start with an array of vertices with the same dimensions listed in the file, assign them evenly spread x and z values, and then map the y values from the file onto them. From there, we start to do subsurface division to create more arrays of vertices with progressively higher resolution, using the fractal algorithms discussed in class. The height of each sub-vertex was varied using a random number generator to give a more organic and bumpy look. The distribution of the random numbers also changes based on height, so that terrain above 'sea level' has more variance than the terrain below. The number of iterations of this algorithm is specified in the height map file, as the scale factor.

The camera controls have also been modified so that the wasd keys move the camera forwards/backwards/side to side, and the mouse can be used to change the viewing angle by clicking and dragging. This is accomplished by changing the camera position to move when the keys are pressed, and changing the yaw/pitch of the viewing angle by comparing the start and end positions of the mouse drag.

Pictures of example execution: