Fractal Landscape Generator
frac-ls-gen
Calvin Hu – ch2880

This project involves the generation of landscape looking graphics using the fractals and outputting to a computer monitor.

Because mountains and landscapes are relatively self similar objects, they can be imitated using fractals. This data can be stored as a 2D array of height/color values and using those values and 2D array indices, be drawn as vertices for a wireframe of a landscape. This would be written to a video buffer which would then be output to the display via VGA. A rotation transform would be applied to the point before display to more easily

The project would receive just input from the buttons.

VGA output would be utilized to display the video.

The landscapes would be generated with the diamond-square algorithm implemented iteratively. First you create a square array and seed the corners with height values. The first step is the "diamond step" where you take four corner points of each square, average their height values and place it in the square midpoint (where the two diagonals of the square meet) and then displace that height value with a random value. This step creates a new series of diamonds in the array. The next step, the "square step", takes each diamond (the using the newly calculated midpoints from the last step), and place the average of the corner height values at the center of this diamond and then displace this value with a random value as well. The square step creates a new set of squares. Repeat the diamond and square steps to increase the number of squares 2 fold. Performing a pass of the algorithm makes the model more detailed, but takes twice as much memory and time each time. Connecting the lines creates a wireframe of the landscape. Alternatively, the height values could act as colors and a topography can be drawn. If a wireframe is drawn, it would have to be rotated to be view properly. The points would have to have a rotation matrix applied before writing to the video buffer.

The hardware would handle the button input and VGA output. The hardware would would read from the video buffer and output through the VGA. The software would handle the diamond-square algorithm and rotation transform (if used instead of topography colors) and write to the video buffer.