Landscape Generator

CSEE 4840
Calvin Hu
Overview

Goal: Create heightmap using diamond square algorithm and display on monitor.

- Generate heightmap
- Rotate vertices
- Backface cull vertices
- Z-sort/Grid sort
- Draw wireframe lines
Diamond-Square Algorithm

- Use to generate terrain like heightmaps.
- Setup square grid of pixels with $2^n + 1$ rows and columns.
- Find average at center of "square" and "diamond" and add random value.
- Iterate and repeat until grid is filled with values.
Backface Culling

- Find orientation of face to determine whether to draw.
- Cross product of two edges in the face.
- Find dot product with camera vector (in this case, \( <0,0,1> \)).
- If dot product is negative, face is facing away from camera, and can be culled.
Rotation

- To rotate vertices, represent coordinates in column matrix and multiply with the corresponding rotation matrix.
- Take advantage of 9 bit multipliers in DE2.
- Read cosine/sine values from lookup ROM.

\[
\begin{bmatrix}
  d_x \\
  d_y \\
  d_z
\end{bmatrix} =
\begin{bmatrix}
  a_x \\
  a_y \\
  a_z
\end{bmatrix}
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \theta & -\sin \theta \\
  0 & \sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]
Grid/Z-sorting

- Because we know the model is organized as a grid, drawing the faces in different orders depending on the orientation relative to the screen.

<table>
<thead>
<tr>
<th>Orientation</th>
<th>0 - 90 degrees</th>
<th>90 - 180 degrees</th>
<th>180 - 270 degrees</th>
<th>270 - 360 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example table entries</td>
<td>0 2</td>
<td>1 4</td>
<td>2 4</td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td>1 4</td>
<td>2 0</td>
<td>4 1</td>
<td>1 0</td>
</tr>
<tr>
<td></td>
<td>4 1</td>
<td>2 0</td>
<td>1 0</td>
<td>4 2</td>
</tr>
</tbody>
</table>

Fig. 2
Memory Units

- 2 Framebuffers: 2-bit word, 76800 words
  - 4 colors
- Heightmap RAM: 36-bit word, 1024 words
  - 1 36-bit coordinate per vertex
- Sine, Cosine lookup roms: 20-bit word, 181 words
  - Each address represents the value of sine/cosine of address in degrees
Fixed Numbers

36 bit signed (2's complement) fixed point to represent vertex coordinates

<table>
<thead>
<tr>
<th>Integer</th>
<th>Fractional</th>
</tr>
</thead>
<tbody>
<tr>
<td>35 to 18</td>
<td>17 to 0</td>
</tr>
</tbody>
</table>

Used to handle trigonometric values/rotation.
Draw Lines and VGA output

- Uses provided Bresenham Line Algorithm component.

- Output to VGA at 640x480 60Hz
Lessons Learned

- Start early and plan well before executing.
- VHDL is much harder to debug than C.
- Don't work alone.