Overview

- Basic Issues
- Height Maps
  - Concepts
  - Generation Algorithms
  - Image-based Methods
- Terrain Size
  - Culling, Paging, & Level of Detail (LOD)
- Sage Terrain Support

Outdoor Terrains

Terrain: a “world object” defining the ground
- not a texture
- not a skybox

issues:
- outdoors - ground is very seldom “flat”
- sky box ground moves with player
- two scenarios: “walkover” vs. “flyover”
  - both have issues to deal with
- viewer proximity to detail

Terrain generation
How do we create the desired terrain?

Terrain rendering
There can be millions of polygons

Terrain appearance
How do we assign color to the terrain?

Height Maps

A 2D grid of numbers representing heights

**Drawing a triangle:**
- $Vx = col$
- $Vy = row$
- $Vz = height[row, col]$
- $\text{vertices} = (Vx, Vy, Vz)$

Even better: TriangleStrip

Height Map Generation

- Algorithmic Generation
  - Functional description, e.g. $A(x, z) = 10 \times \sin \left( \frac{\sqrt{3}}{2} \cdot \left( -\frac{50}{10} \cdot \frac{z}{z} \right) \right) + 7 \cos \left( \frac{z}{10} \right)$
  - Midpoint displacement
  - Diamond-Square
  - Fault line
  - Hill raising
  - Genetic Algorithms
- Image-based Descriptions
- Procedural (“Fractal”) Models
Height Map Generation Algorithms

"Midpoint Displacement"

- Assign random height values to four corners.
- Assign height values to edge midpoints by averaging corners and adding small displacement.
- Assign height value to center point by averaging edge midpoints and adding small displacement.
- Repeat recursively for each smaller square.

Diamond-Square

- Choose outermost square.
- Square center = avg of corners + random offset.
- Form diamonds from square corners and midpoints. Diamond middles = average of corners + random offset.
- Form squares of half-size. Square centers = avg of corners + smaller offset.
- Form diamonds. Diamond middles = average of corners + offset.

Fault Line

- Generate random fault line.
- Increase heights on one side, decrease on the other.
- Repeat.

Hill-Raising

- Choose a random point and radius.
- Raise a hill with the chosen radius at the point.
- Repeat.

Image-based Height Fields

Basic Idea: use image pixels as “height”

Most common form: “gray-scale”

- 0 = black = low height
- 1 = 255 = white = high height

Easy to create in any paint program

Terrain Map Examples

- Low variation in height
- Wide variation in height
Sage Terrain Classes

**AbstractHeightMap**
- float[] heightData
- int edgeSize

**ImageBasedHeightMap**
- Image heightImage
- IBHM(String imageFile)
- IBHM(Image image)

**MidPointHeightMap**
- MPHM(size, roughness)
- load(size, roughness)

**HillHeightMap**
- HHM(size, itr, min, max…)
- load(size, itr, minR, maxR, flatness, seed)

**FaultFractalHeightMap**
- FFHM(size, itr, min, max)
- load(size, itr, minD, maxD)

**TerrainBlock**
- int size
- HeightMap map
- int quadrant
- float sx, sz

**TriMesh**

using the SAGE terrain classes

- Instantiate height map
  - ImageBasedHeightMap, or
  - HillHeightMap
- Instantiate TerrainBlock; includes:
  - height map
  - texture state
  - terrain
- Scale terrain size and height
- Adjust vertical position based on height at origin

Following Terrain Height

Adjust avatar height after moving:
- moveForward();
- updateVerticalPosition();

- Verify that avatar is inside terrain bounds
- Determine heightmap height at avatar position
- Adjust for heightmap height relative to origin
- Scale adjustment based on heightmap scale
- Adjust avatar y-translation

other Terrain features

**Lakes and Islands**
- Force all height-map values within some boundary to be equal
- Use “hill raising” with center forced to lie within lake boundary
- Or, add a ground plane

**Rivers**
- one approach: manipulate terrain between “start” and “end”
- Use “path-finding” algorithms (e.g. A*)

other Terrain Issues

Terrain map may not cover the world
remedies:
- repeated terrain
- scaling

Terrain is huge, renderer can’t keep up
remedies:
- Frustum Culling
- Terrain Blocks and Paging
- Level of Detail (LOD) Management
- Hardware support (e.g., tessellation)
Hardware Support

Tessellation shader example:

Sage Terrain Pages

Frustum Culling for Terrain

QuadTrees For Visibility Culling