

## 10 - Terrain

### Overview

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

2

### 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 won't work -- moves with player
- two scenarios: "walkover" vs. "flyover"
  - both have issues to deal with*
- viewer proximity to detail

3

### Terrain generation

How do we create the desired terrain ?

### Terrain rendering

There can be millions of polygons

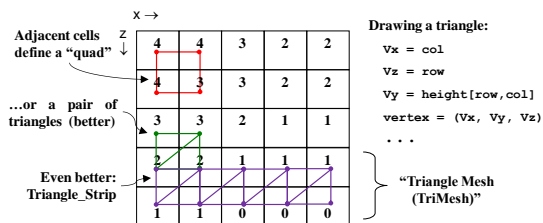
### Terrain appearance

How do we assign color/lighting to the terrain ?

4

### Height Maps

A 2D grid of numbers representing heights



5

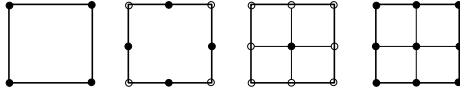
### Height Map Generation

- Algorithmic Generation
  - Functional description, e.g.  $h(x, z) = 10 * \sin\left(\frac{x}{24}\right) + 7 \cos\left(\frac{z-50}{18}\right)$
  - Midpoint displacement
  - Diamond-Square
  - Fault line
  - Hill raising
  - Genetic Algorithms
- Image-based Descriptions
- Procedural ("Fractal") Models

6

## Height Map Generation Algorithms

### "Midpoint Displacement"



Assign random  
height values to  
four corners

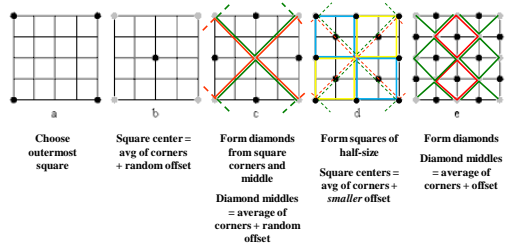
Assign height  
values to edge  
midpoints by  
averaging  
corners and  
adding small  
displacement

Assign height  
value to center  
point by  
averaging edge  
midpoints and  
adding small  
displacement

Repeat  
recursively for  
each smaller  
square

7

## Diamond-Square



Choose  
outermost  
square

Square center =  
avg of corners  
+ random offset

Form diamonds  
from square  
corners and  
middle  
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

Image credit: Generating Random Fractal Terrain, Paul  
Martz, <http://www.gameprogrammer.com/fractal.html>

8

## Fault Line

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

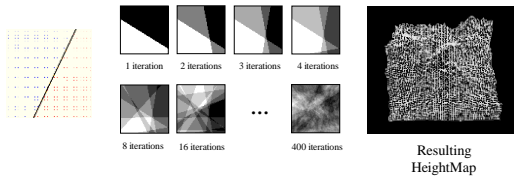
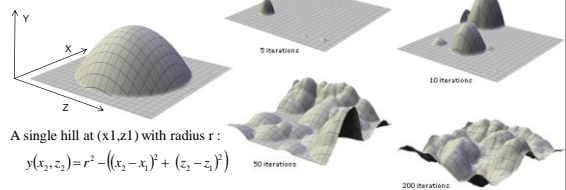


Image credit:  
[www.lighthouse3d.com](http://www.lighthouse3d.com)

9

## Hill-Raising

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



A single hill at  $(x_1, z_1)$  with radius  $r$ :

$$y(x_2, z_2) = r^2 - ((x_2 - x_1)^2 + (z_2 - z_1)^2)$$

Image credit <http://www.stuffwithstuff.com/robot-frog/3d/hills/hill.html>

10

## 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

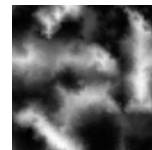
11

## Terrain Map Examples

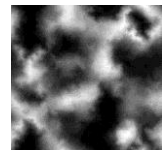


128 x 128 jpg

Low variation in height



512 x 512 bmp



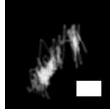
257 x 257 jpg

Wide variation in height

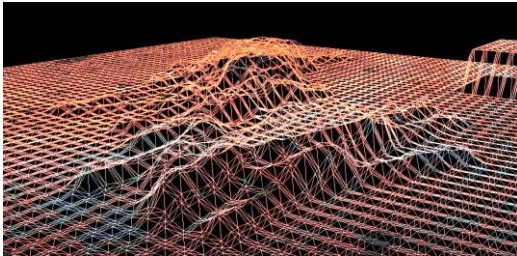
12

## Hardware Support

height map:



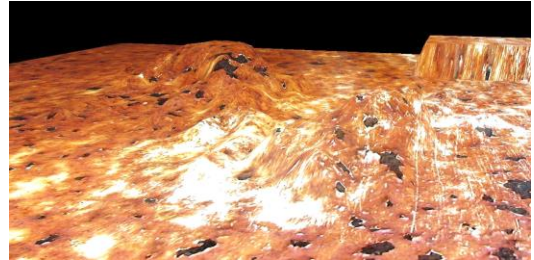
Tessellation shader example:



13

## Hardware Support

Tessellation shader example:



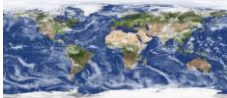
14

Incorporating lighting:

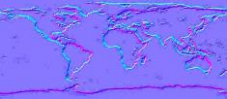
height map



standard texture

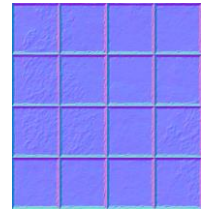
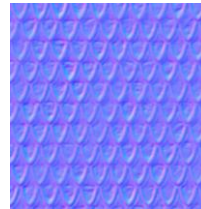


normal map



15 of 24

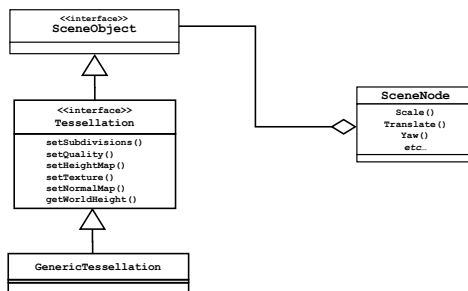
Normal Maps:



Values in normal maps are usually stored as offsets from vertical relative to the plane tangent to the surface, with the Z (or B) coordinate set to 1.0. This is why normal maps appear "bluish".

16 of 24

## RAGE Terrain Classes



17

## using the RAGE terrain classes

- Create height map
  - Instantiate Tessellation class (similar to Entity)
  - Load a height map, texture, normal map as desired
- Instantiate a SceneNode for the height map
  - attach the tessellated height map to the scene node
  - Translate, Scale, Rotate as desired
  - Scale on Y axis adjusts actual heights

18

## Following Terrain Height

Adjust avatar height after moving:

```
moveForward();  
updateVerticalPosition();
```

- Get avatar world X and Z coordinates
- Retrieve height from height map object
- Adjust avatar y-translation appropriately
- *optional*:  
adjust avatar tilt based on neighboring heights

19

## other Terrain features

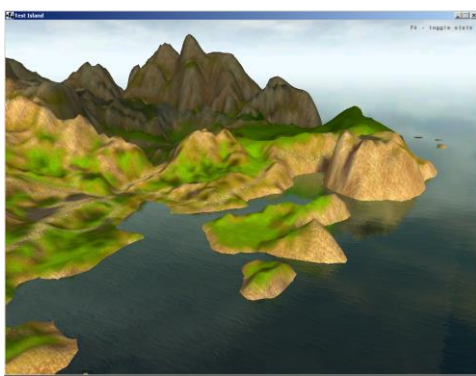
### Lakes and Islands

- 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\*)

20



21

## other Terrain issues

Terrain map may not cover the world

remedies:

- repeated terrain
- scaling

Terrain is huge, renderer can't keep up

remedies:

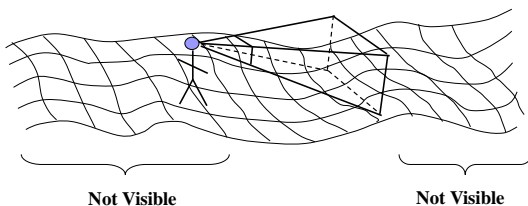
- Frustum Culling
- Terrain Blocks and Paging
- Level of Detail (LOD) - supported in RAGE
- Hardware support (e.g., tessellation)

22

## Frustum Culling for Terrain

Most triangles are outside frustum

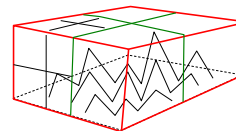
Can we avoid sending them down the pipeline?



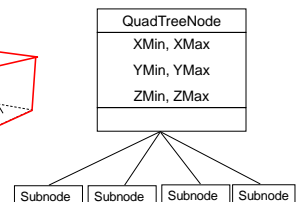
23

## QuadTrees For Visibility Culling

Bounding Box



HeightField



24