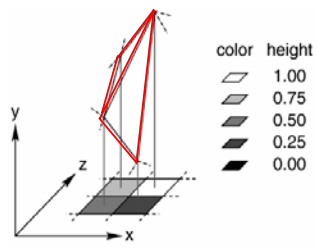
Odds and Ends

Height Fields
Slope
Walking

Height Field

- Simulate natural terrain
- An image controls terrain creation
- Simple base geometry
- Brighter → higher

Height Field



POV-Ray height field

- One unit wide by one unit long square with a mountainous surface on top.
- The height of the mountain at each point is taken from the color number or the brightness in a graphic image file.

POV-Ray particulars

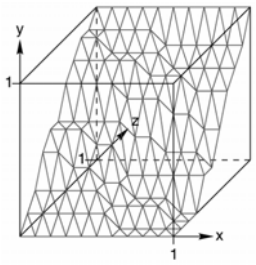
- The maximum height is one.
- You have to scale it.

Height Field

```
height_field{
     [HF_TYPE]
     "filename"
     [HF_MODIFIER...]
     [OBJECT_MODIFIER...]
   }
HF_TYPE:
   gif | tga | pot | png | pgm | ppm | jpeg | tiff |
   sys

HF_MODIFIER:
   hierarchy [Boolean] | smooth | water_level Level
```

Height Field



Height Field

- The un-scaled height field size will always be 1 by 1 by 1.
- Higher resolution →smaller triangles

Example

Example

```
height_field{
  tga "MountLo1.tga"
  translate<-0.5,0,-0.5>
  scale<50,7,50>
  texture{
    pigment{color rgb <0.85,0.6,0.4>}
    normal {bumps 0.75 scale 0.025  }
    finish {ambient 0.1 diffuse 0.8 }
  }
}
```

How to get an image?

- Paint one
- Use application software to create one
- Use procedural texture

“baking” a texture

converting a procedural texture to an image

- costs memory – usually large file size
- saves computation in the long run
- use
  - orthographic camera
  - only ambient coloring
  - perspective and light interaction will come later
  - high quality settings
    - image size
    - color representation
    - anti-aliasing

baking in POV-Ray

- see heightFieldCreate.pov
- global settings – 16 bit grayscale

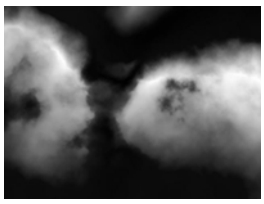
```
global_settings {
  //assumed_gamma 2.2
  hf_gray_16
}
```

- add render setting for increased accuracy

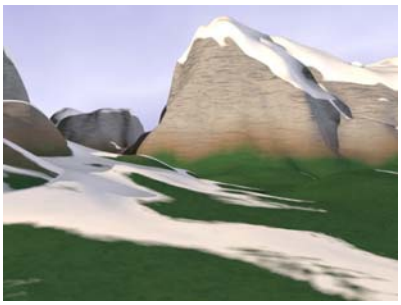
```
[1024x1024, AA 0.1]
+FT
Width=1024
Height=1024
Antialias=On
Antialias_Threshold=0.1
```

- or even higher, 2048x2048
- note: high frequency or quick changes will look BAD if you are close!
  - may need to use more than one heightfield

heightfield



Eric Sedgwick



Eric Sedgwick

Snow?

Eric Sedgwick

Slope pattern

```
pigment {
   slope {
      <Direction> [, Lo_slope, Hi_slope ]
           [ altitude <Altitude> [, Lo_alt, Hi_alt ]] }
      [PIGMENT_MODIFIERS...] }
```

Computes pattern value based on **slope**

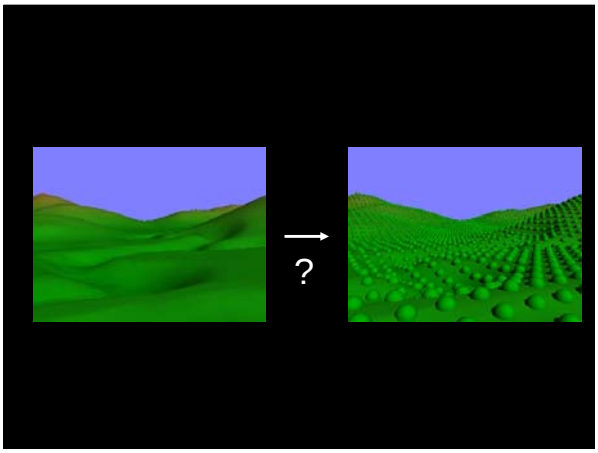
Slope pattern

```
pigment {
   slope {
     <direction>}
     [PIGMENT_MODIFIERS...] }
```

surface normal opposite direction of <direction> : slope = 0.0
surface normal perpendicular to <direction> : slope = 0.5
surface normal parallel to <direction> : slope = 1.0.

Slope pattern

```
texture {
    pigment {
      slope { y altitude .65*y}
       pigment_map {
        [.573 Clear]
        [.58 White]
      }
    }
}
```

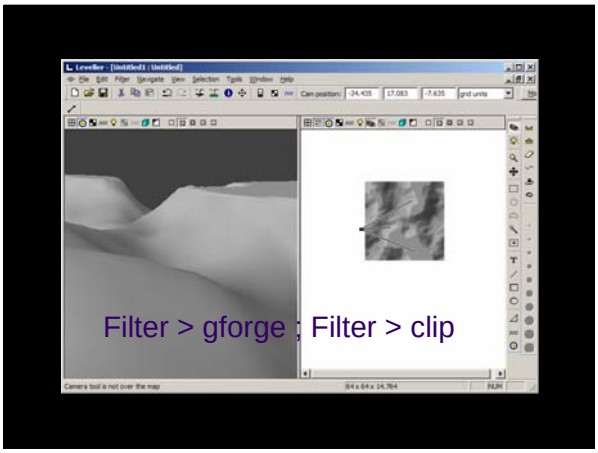


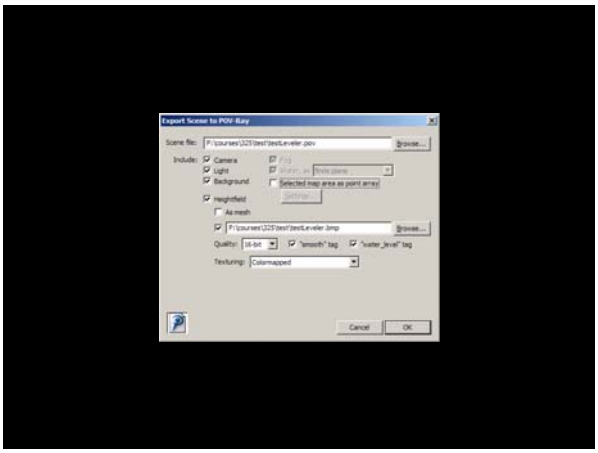
Leveler

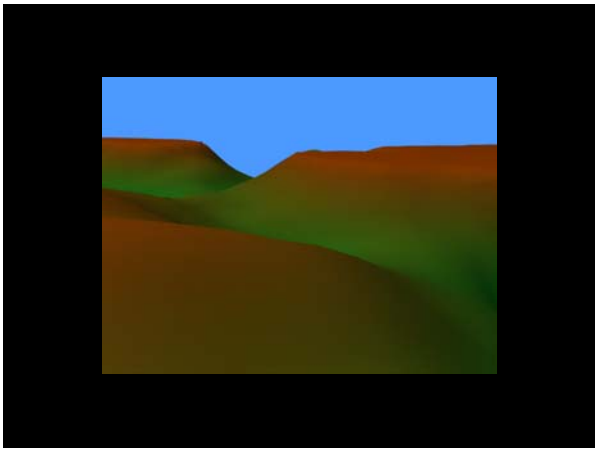
- Wysiwyg interface for terrain creation
- Export POV, positions
- www.daylongraphics.com

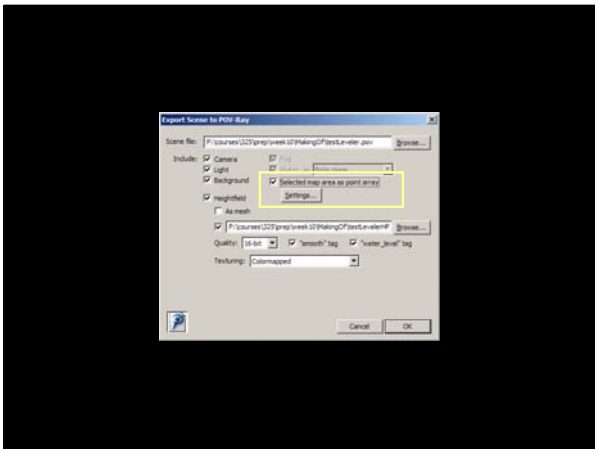
Process

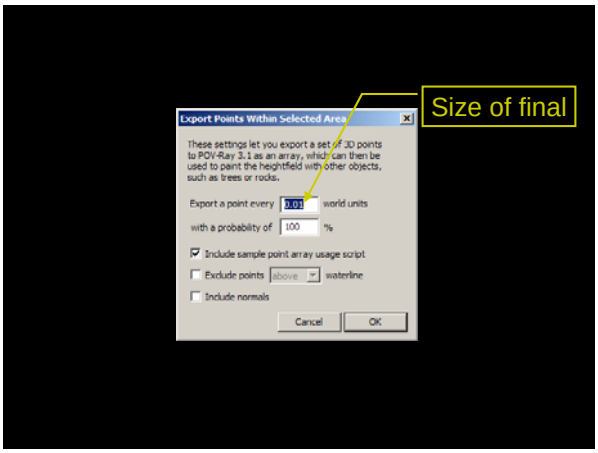
- Create terrain
- Select areas
- Export POV-Ray points



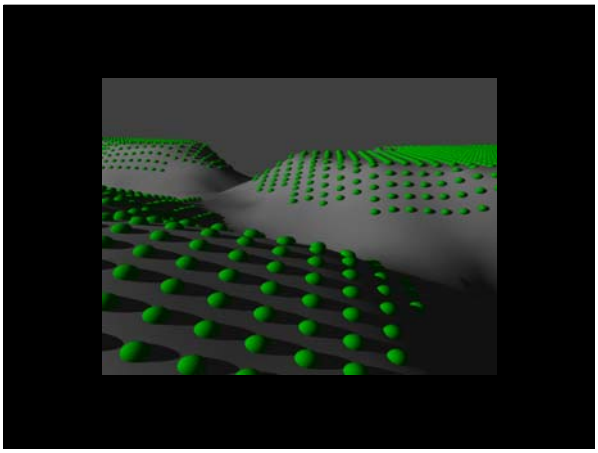

Export Scene to POV-Ray
Scene file:
Browse...
Include:
Camera
Light
Background
Selected map area as point array
Heightfield
Settings...
As mesh
Quality:
16-bit
"smooth" tag
"water_level" tag
Texturing:
Colormapped
Cancel
OK

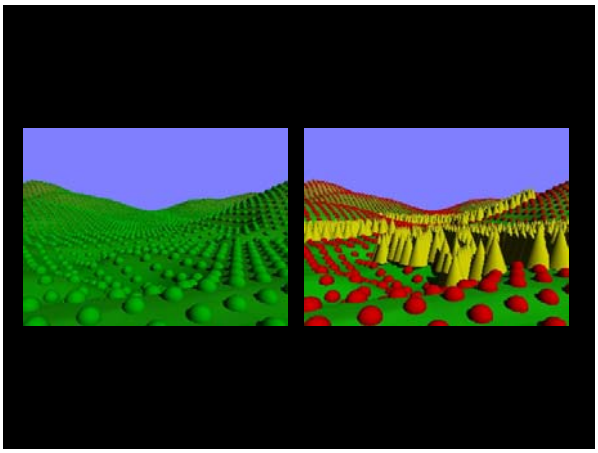


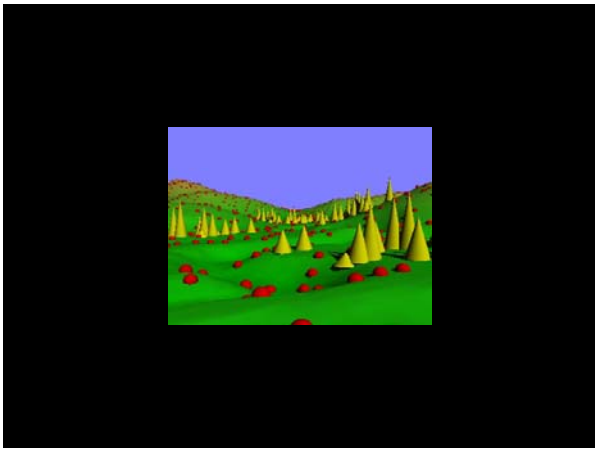

Export Scene to POV-Ray
Scene file:
Browse...
Include:
Camera
Light
Background
Selected map area as point array
Settings...
Heightfield
As mesh
Quality:
16-bit
"smooth" tag
"water_level" tag
Texturing:
Colormapped
Cancel
OK

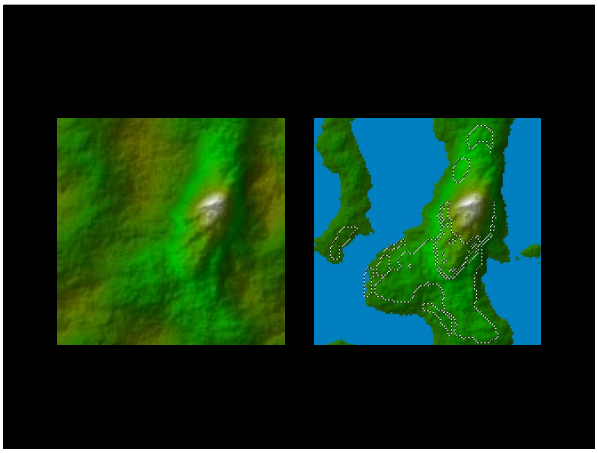


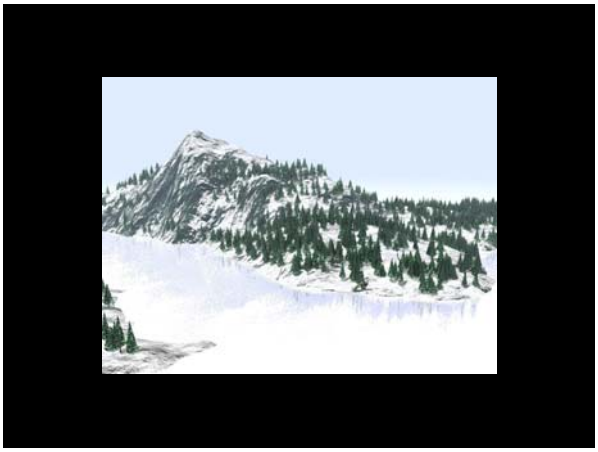

```
#declare HF_pts = 2384;
#declare HF_pts_array = array[HF_pts]
{
      <-31.5,12.216,25.5>,
      <-31.5,12.51,24.75>,
      <-31.5,12.595,24>, ...
}
#declare N = 0;
#while(N < HF_pts)
  sphere { < 0, 0, 0 > 0.2
           pigment { Green }
           translate HF_pts_array[N]
  }
  #declare N = N + 1;
#end
```











Walk cycle in POV

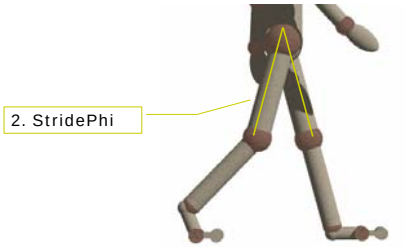
- Theresa Willis `http://www.jbarchuk.com/twillis`
- Specifies human proportions
- Declare key motion parameters
- Specifies type of motion
- Declare a figure

Proportions

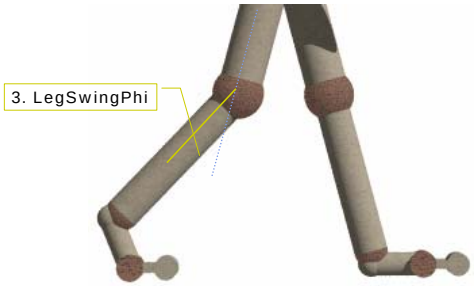
```
#declare HeadRho = .130  ;
#declare NeckRho = .066  ;
#declare TrunkRho = .306 ;
#declare lClavicleRho = .125  ;
#declare rClavicleRho = .125  ;
#declare lArmRho = .18 ;
#declare rArmRho = .18 ;
#declare lHandRho = .054  ;
#declare rHandRho = .054  ;
#declare lFingerRho = .054 ;
#declare rFingerRho = .054 ;
...
```

Declare parameters

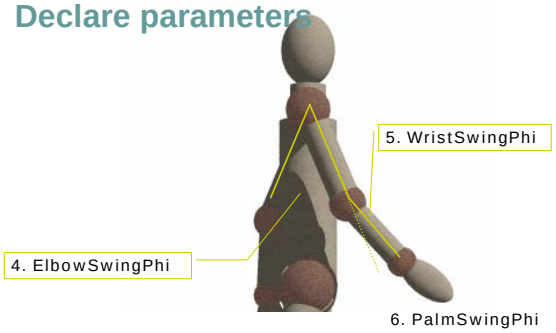
1. steps



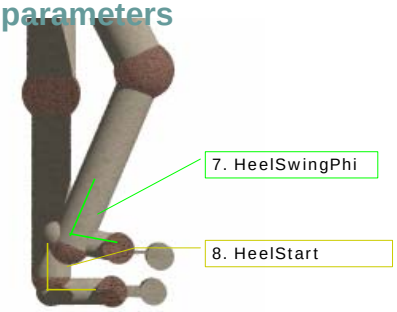
Declare parameters



Declare parameters



Declare parameters



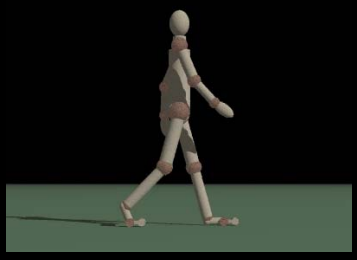
Specifies motion

- walking.inc
- standing.inc
- sitting.inc

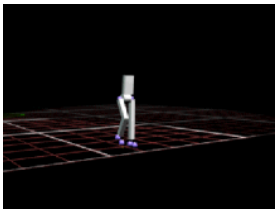
Create figure

```
lBall       Ball of left foot
lHeel       Left Heel
lAnkle      Left Ankle
lKnee       Left Knee
lHip        Left Hip
Center      Center of Pelvis
rBall       Ball of right foot
rHeel       Right Heel
rAnkle      Right Ankle
rKnee       Right Knee
...
```

Results



Using AI to walk

Simulate evolution:

http://www.naturalmotion.com