Creating Spherical Worlds

Kate Compton, James Grieve, Ed Goldman, Ocean Quigley, Christian Stratton, Eric Todd, Andrew Willmott

Maxis, Electronic Arts
Background

• Spore based on “powers of 10”
  – Cell life (2D world)
  – Planet: creatures, tribes, civilisations
  – Solar System
  – Interstellar
  – Galaxy

• Want seamless transitions

  planets need to be spherical
Planet Constraints

• Need to have lots (millions? billions?)
  – many more than we can manually author

• Need to be playable

• Must look good

• Need to be fast to generate
  – We can’t store all these planets
  – Would like to transmit them at some point

• Need to support terraforming
  – Player modification of planet to support life
Areas of Interest

- Parameterization
  - How do we store planet representation over surface? How do we store game data?
- Generating Heightfields
  - What are the operations? How can we make it fast?
- Texturing
  - Must be heightfield driven
- Authoring
  - Variety, art control
Parameterization

- Possible approaches:
  - Longitude/latitude (pole cap)
  - Gnomic
  - Freeform 3D: Sparse Voxel
  - Charts
    - Regular: cubemap, diamond, duodecahedron ...
    - On-the-fly (Voronoï-style)
    - Orthographic projection
    - Perspective projection
Parameterization Goals

- Minimize distortion and discontinuities
- Efficient (heightfield) storage
- Fast mapping from \((x,y,z)\) to \((u,v)\) and back
- Wrapping between charts
- Rectangular area splatting
- Efficient normal map generation
Parameterization: Cube Maps

- Chose cube maps as the best compromise
Parameterization: Cube Maps

- Chose cube maps as the best compromise
- Faces are grids
  - Familiar from previous games
- Distortion at corners
  - But not too bad, much better than pole distortion
- Face wrapping is tractable
  - Pick right face mappings -> simple permutation rules
- Projective projection
  - Lines map to great circles on sphere: very useful!
Colour Map
Normal Map
Normal Map

- Derived from height map
  - Large source of CPU time early on
- Standard DDF to find ‘flat’ normal map
  - Can then use Jacobian to warp to spherical form

\[
J(s, t, h) = \begin{pmatrix}
\frac{h}{w}(1 - \frac{s^2}{w^2}) & -\frac{sth}{w^3} & -\frac{sh}{w^3} \\
-\frac{sth}{w^3} & \frac{h}{w}(1 - \frac{t^2}{w^2}) & -\frac{th}{w^3} \\
\frac{s}{w} & \frac{t}{w} & \frac{1}{w}
\end{pmatrix}
\]

\[
w = \sqrt{\left(s^2 + t^2 + 1\right)}
\]
Generating Height Fields

- Brush system that operates on the sphere
- Brushes are 2D textured rects
- Several different brush operations
  - Conditionally raise or lower terrain
- Applied on GPU, after clipping brush footprint to faces
Controlling Terrain Brushes

• Use our effects system, Swarm, to run brushes over the surface

• Controlled by:
  – Particle systems (spawning other particle systems)
  – Randomized parameter ranges, random walks
  – Terrain forces
  – Force/control operates in the tangent plane
Texturing

• Derive Control Map from height field
  – Filter: water level, gradient, curvature
  – Combine according to tech artist formula

• Blends source textures to form base colour
  – Blends detail maps on the fly

• Planets have type, atmosphere, temperature
  – Control colour ramps, and atmosphere/fogging
Terraforming

Atmosphere

Temperature
Authoring

- Concept Sketches
PLAYABLE: Yes

TAXONOMY CATEGORY: Storybook

Based on the floor of an ancient forest, this planet has landforms that appear to be giant roots covered in moss and various fungii-looking rocks.

PARTICLE EFFECTS

CLOUD PATTERNS

LOOPBOX PARTICLES

WATER

SKY

LIVE TERRAIN

BEACH

CLIFF

DEAD TERRAIN

SCRIPT TO GO UNDER MASSIVE OBJECT

LAND SCRIPT

POND SCRIPT

CIVILIZATION VIEW
Inspired by crab shells, this planet is made mostly of small strips of land that randomly connect to each other and to a main section where there is more room for cities.
PLAYABLE: Yes
TAXONOMY CATEGORY: Storybook

moss scattered randomly near base of big rocks

Small rocks clustered together
Authoring

- Originally one mega effects script
  - random selection between various child effects
- Difficult to control
  - Hard to get art-directed
- Introduced a top layer with more control: *terrain scripts*
- Each script produces a particular kind of planet
The Result
Authoring: Planet Editor

Demo
Questions?