Player-Driven
Procedural
Texturing

SIGGRAPH 2007

Henry Goffin, Grue, Chris Hecker,
Ocean Quigley, Shalin Shodhan,
Andrew Willmott

Maxis, Electronic Arts
Spore: Player-Created Stuff

• Want players to be able to create key parts of their game

• Pollinate player-created things via servers, so your game is made of both your own creations and others’

• Richer experience, less art work(!)
Game World
Game World
Player-Created Stuff: Creatures
Player-Created Stuff: Buildings
Player-Created Stuff: Cars
Player-Created Stuff: Boats
Player-Created Stuff: Planes
Player-Created Stuff: Hybrids
Problem Area

• We let players create their own creatures, huts, buildings, cars, boats, planes...

• How do we texture them?

• Once we’re done, how do we turn this into a game model?
Previous Work

- **SSX**: Swap in different player meshes, accessories
- **Sims 2 Bodyshop**: Facial morphs, Select clothing: top and bottom texture pages
- **Need for Speed vehicles**: Decals, morphs on many parts
- Many more
Texturing: Player Control

• Want satisfying player input

• Not too detailed
  – Too tedious for the majority

• Not too simplistic
  – Everyone’s model looks the same
  – Want to go beyond overall texture layer selection
Two Solutions

• Creature/Flora/Cell Texturing
  – Full brush-driven 3D texture painting
  – Driven by our effects system rather than a human
  – Variety by layering different parameterized scripts

• “Mineral” Texturing
  – Repeating textures
  – Paint regions
  – Procedural UV’ing
1: Skinpaint

- Brushes: diffuse, spec, alpha, bump map
  - Mesh is uv mapped, for any point on mesh, brush can be splatted into destination texture

- Brush selection and position controlled by effect system
  - “Particles” can be moved over surface using frame adjustment, affected by skeleton
  - Library of effect scripts
Skinpaint: Early Prototype
2: Procedural UV’ing

- Model parts deform (Rigblocks)
- Model parts tagged with regions
  - Use for material and functional areas too
- Regions tagged with uv’ing type
  - Boxmap
  - Cylinder, sphere, planar, disc
- Applied in Vertex Shader
- Textures parameterized by two colours
Paint & Proc UV Demo

Demo
Player-Created Models In-Game
Problems!

- Player-created model is not suitable for game use
  - Too many meshes (can be many parts)
  - Too many materials and textures
    (Parts x regions = a lot of batches)
  - Efficient rendering on GPU requires minimizing batch count.
  - No LOD!
Solution #1: Splatter

- Generate unique UVs for ‘editor’ model as second uv set.

- Render model with clipPosition = uv2
  - “Splats” source textures into a single texture sheet
  - Allows resampling of high-res editor textures into a known-size texture, constant for all models
Unique UVs: Charting

- Use face clustering to generate charts quickly
  - Generates ‘flattish’ chunks of geometry for charting

- Optionally run LSCM relaxation on these, otherwise planar project

- Use horizon-map-style packer (similar to Levy et al.)
Solution #2: Geometry Pipeline

• Single texture page means we can weld all meshes together

• Generate LODs by simplification
  – Vertex decimation, faster than edge-based simplification
  – Has to handle bones
  – Has to handle mesh discontinuities due to normals, tangent spaces, texture coordinates
  – Must minimize discontinuities in output mesh
Problem with approaches that don’t involve an entire-skin texture: no dark map.

We generate as a post-pass using GPU (accumulate shadow passes for model)

Visual glue that holds everything together
Baking Details

- Desirable that we can do this while game is running
  - No blocking!

- Various chunks written as background jobs, controlled by job manager
  - Background load all source assets
  - Bake, then cache results to disk
  - Graphics assets need to be created in main thread
Questions?