Procedural Low-Poly Map Generation

Sam Donow and Nigel Munoz
CS371 F2014 Midterm
Williams College
Goals: Low Poly, Infinite, Procedural

--All Algorithm inputs are random.
--Someone within the world cannot notice seams in an infinite world
--Low-poly art style; A low-density heightfield
Random Terrain is made with Noise

Random seeds are coordinates at lattice points


--Two separate noise terms used explicitly instead of octaves to give more control over distribution.
--Diagram: Higher order terms in large boxes, lower in small boxes
--The seed to the noise function is the normalized coordinates of lattice points, which allows for consistency between regenerations
“Infinite” Terrain

- Moves with camera while remaining consistent

When the player reaches the end of the world, we regenerate terrain with them at the center, so that they cannot see the end of the world; also, we make the lattice point nearest them as the center, allowing consistency between regeneration.
Water Fills in valleys and has Waves

--Waves movement created in vertex shader

The vertex shader has a complex chain of sines and cosines dependent on time and position in order to compute the motion of water
Trees with Random Height and Design

--Trees only grow from land
--Consistent with terrain

Random height for trees within bounds
Random whether trees have one or two pyramids at the top
Analogous colors are good. Needed varying shades of green and blue for the color palette. This is because when low-poly flat shading is implementing, we need various colors to display, to make differences between adjacent faces clear.
Fog: Fading Reduces Seams

-A pixel shader lerps between scene and skybox or a solid color, weighted by depth.

Fog makes it harder to see past the edges of the world to our shader. Depth is obtained from the depth buffer rendered by G3D. We lerp to a solid color on scenes where the skybox is too varied for lerping to it to appear as fog (e.g., a night sky with stars).
Flat Shading Reveals Polygons

- Smooth Shading: interpolate normals over tri
- Flat shading: hold normal constant on tri

Flat shading is just a matter of adding a single line to a shader, it is a very simple feature.
But, as the above images show, they greatly improve the quality of an image, especially relative to the art style.
Basic Physics added to the world in order to ensure compatibility with making a game, which is likely the eventual use of our Terrain Generator.

The only Collision detection necessary is sphere-mesh, with the meshes stored in a collision hierarchy to make collision detection as efficient as possible: as we are doing nothing fancy, a bounding sphere is more than adequate for calculating intersections as someone moves around the scene.
A real time demo will be shown during the presentation, it shows off our results graphically, while also showing that the basic physics engine does, in fact, work
At this point, we showed our finalized video which is on display in the same gallery as this pdf