The Venice Simulation

Eisaku Imura ❖ Jonathan Chuang ❖ Matt Loyola

OVERVIEW

- 3D Simulation of Venice, Italy
- Textured city geometry
- Day/night cycle
- Cloud system/simulation
- Wave simulation

The City

Jonathan Chuang

Accurate building data from OSM.

Earclipping & shoelace formula for building obj geometry & normals.

Procedural building textures & materials in tandem with Eisaku.

Generating *calli* & *ponte* procedurally.






The City

Jonathan Chuang

Accurate building data from OSM.

Earclipping & shoelace formula for building obj geometry & normals.

Procedural building textures & materials in tandem with Eisaku.

Generating *calli* & *ponte* procedurally.

The City

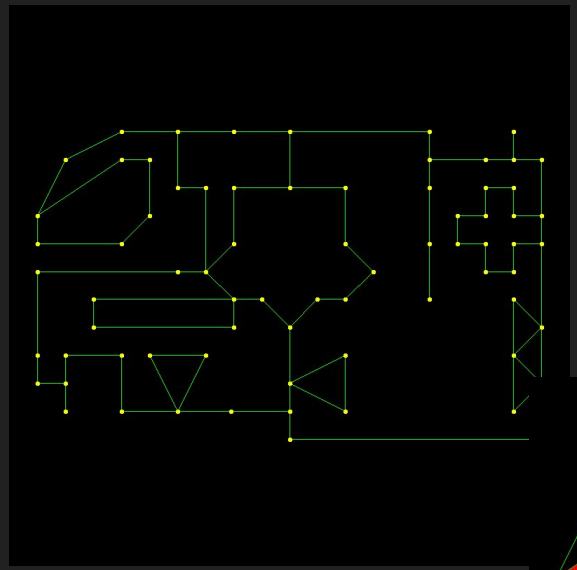
Jonathan Chuang

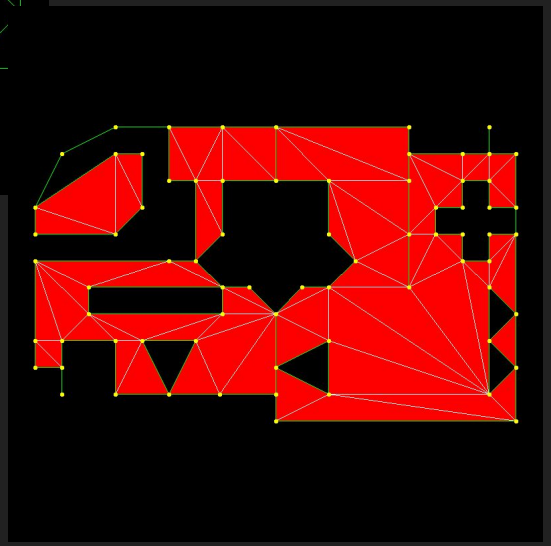
Accurate building data from OSM.

Earclipping & shoelace formula for building obj geometry & normals.

Procedural building textures & materials in tandem with Eisaku.

Generating *calli* & *ponte* procedurally [reach goal].





The City

Jonathan Chuang

Accurate building data from OSM.

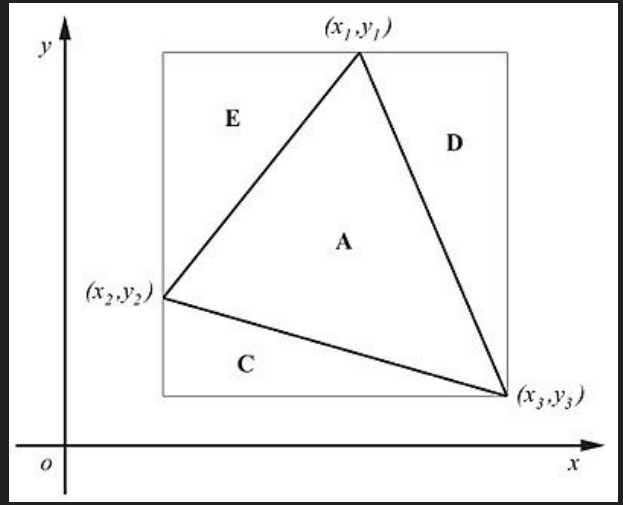
Earclipping & **shoelace formula** for building obj geometry & normals.

Procedural building textures & materials in tandem with Eisaku.

Generating *calli* & *ponte* procedurally.

$$\mathbf{A} = \frac{1}{2}\left|\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n\right|$$

$$= \frac{1}{2}\left|x_1 y_2 + x_2 y_3 + \cdots + x_{n-1} y_n + x_n y_1 - x_2 y_1 - x_3 y_2 - \cdots - x_n y_{n-1} - x_1 y_n\right|$$



$$\frac{1}{2}\left|\sum_{i=1}^{n} \det\begin{pmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{pmatrix}\right|$$

Day/Night Cycle - Eisaku Imura

- Dynamic skydome
- Sunlight
- Moonlight (with dynamic stars?)
- Crepuscular rays (if time allows)
- Different weather (if time allows)
- Adjustable with GUI

Clouds and Water ~ Matt Loyola

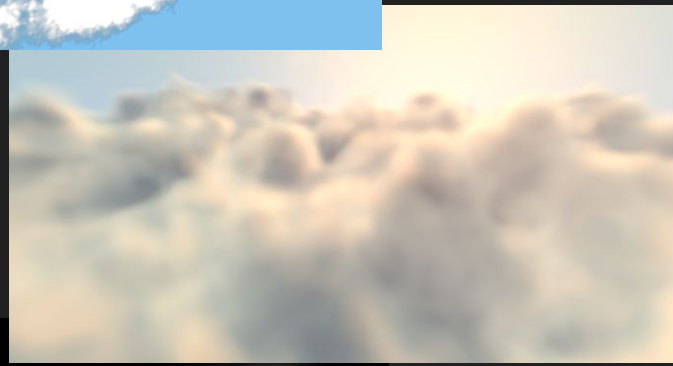
- Goal is to replicate 3D clouds within the skybox/skydome that change shape and consistency over time (like real clouds).
- Regardless of the method of implementation, the cloud system should easily fit in moving across the top of the skybox to aid in the believability of the venice scene.
- Water will be implemented using noise functions to displace the texture on a plane.
  - This will create a wave-like effect.
  - Wave control through GUI.
- Water reflections of the sky and buildings if time permits.

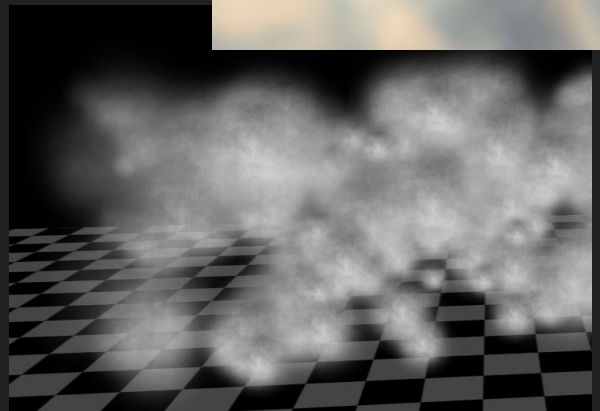
Clouds ~ Examples

- The first two examples to the right utilize various noise functions in the fragment shader to spawn different clouds of varying shapes and densities.
  - The ShaderToy example uses raymarching to calculate the light interaction. Raymarching is used as opposed to raycasting in this instance because we'd want the light from the sun, or any other source, to (scatter) through the cloud object.
- Clouds could also be implemented as a particle system with size-varied sprites (or point sprites) as seen in the third example.

"THREE.Cloud" Example from GitHub. ←

"Clouds" ShaderToy Example →

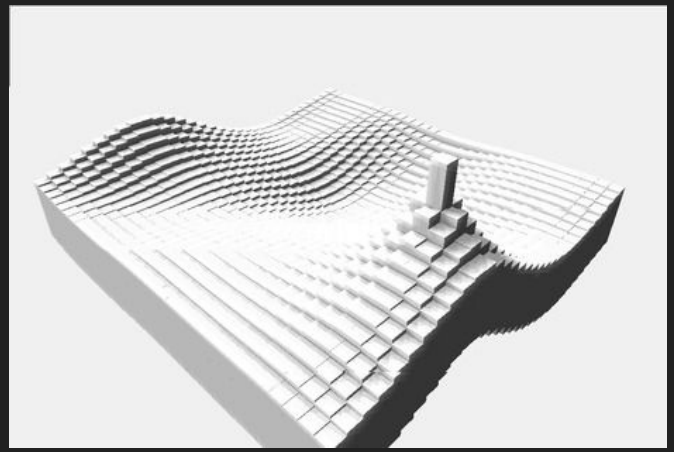




Particle system using cloud sprites. ←

Water ~ Examples

“Zelda Windwaker” Inspired wave shader from the Week 9 Lab →



Voxel Water Displacement ←

“Seascape” from Shadertoy using noise and heightmaps →

The Venice Simulation

Eisaku Imura ❖ Jonathan Chuang ❖ Matt Loyola