CS334 – Fundamentals of Computer Graphics

Prior Final Projects

Daniel G. Aliaga

Garden Pavilions

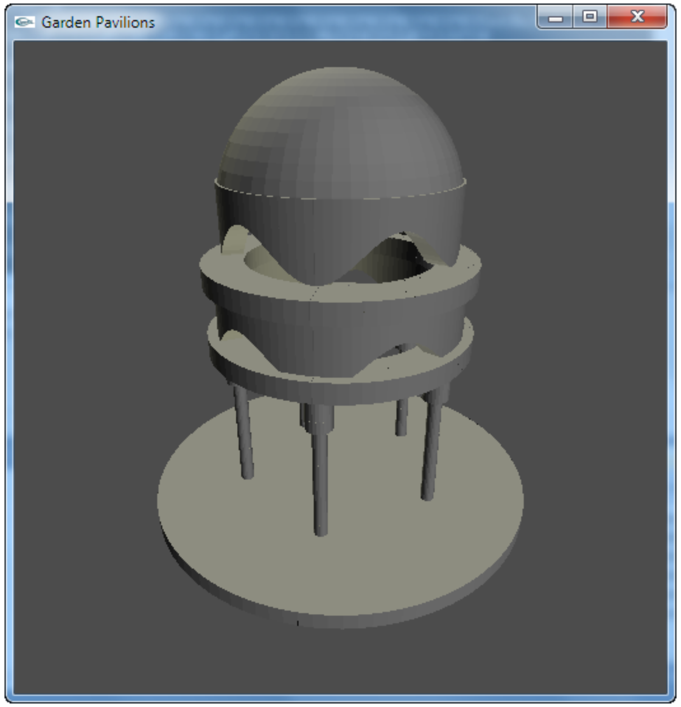
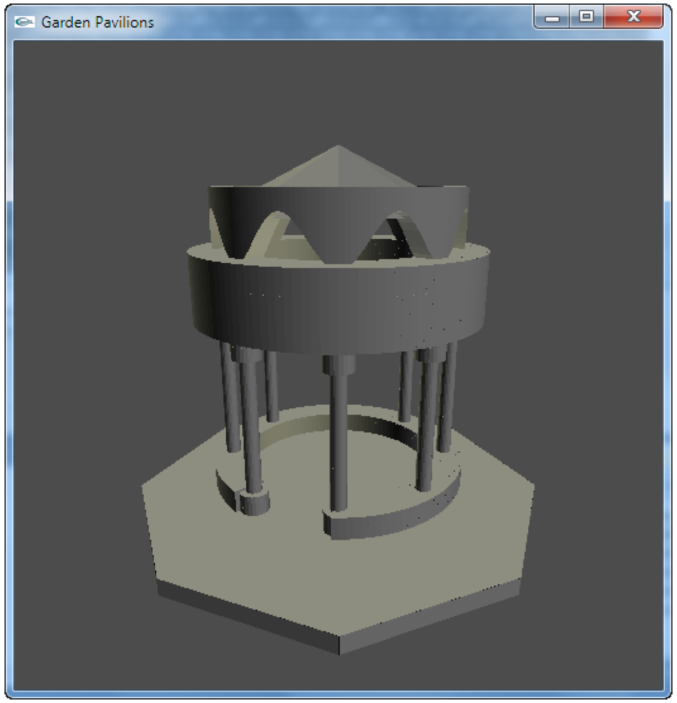
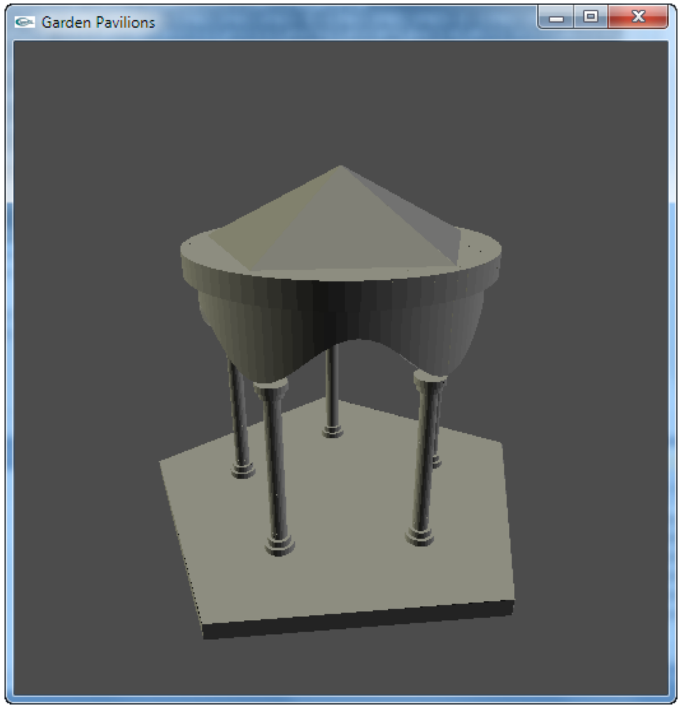
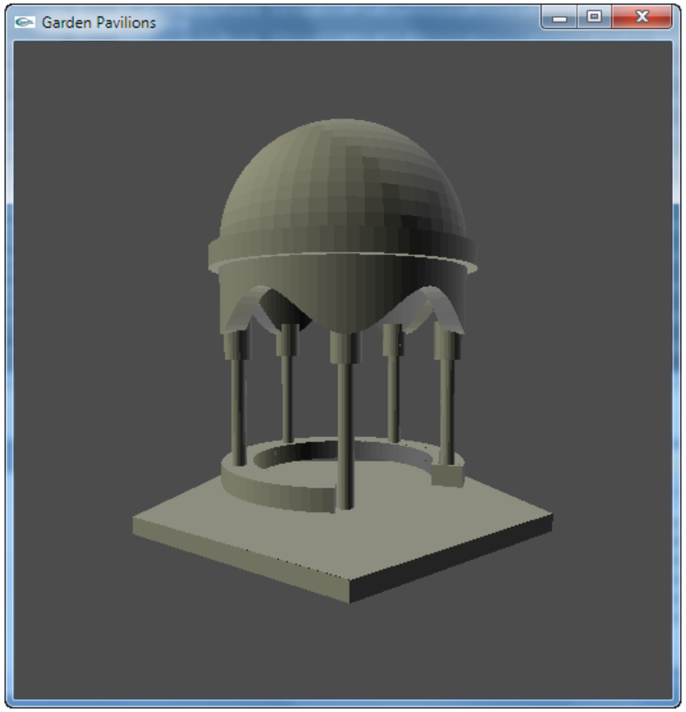
K. K.

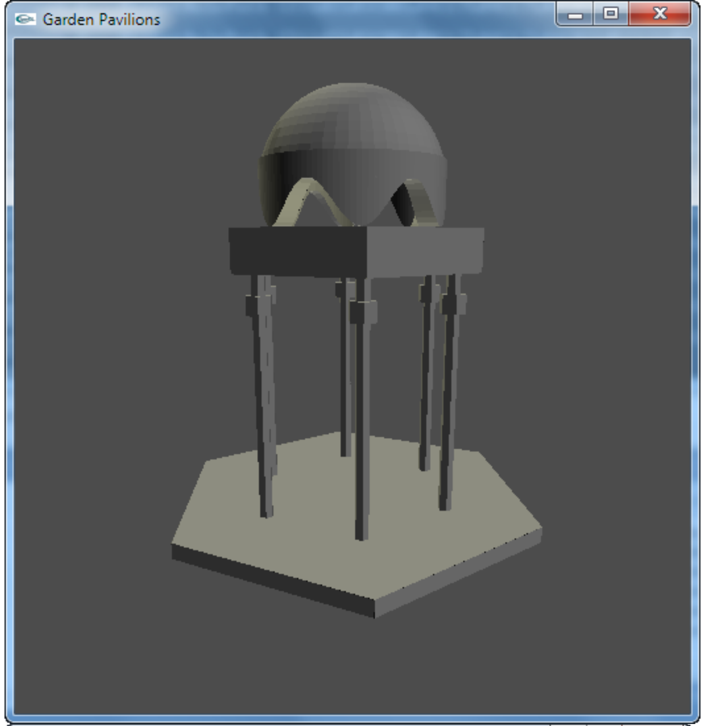
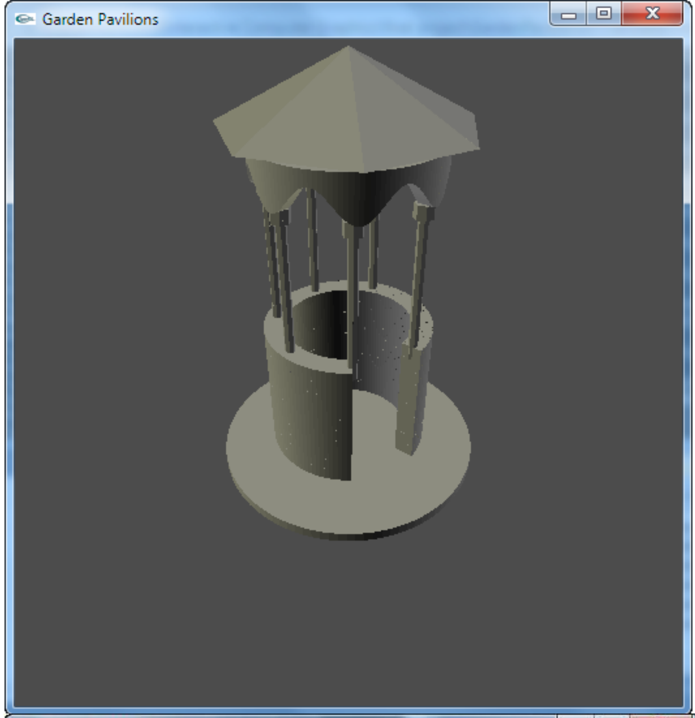
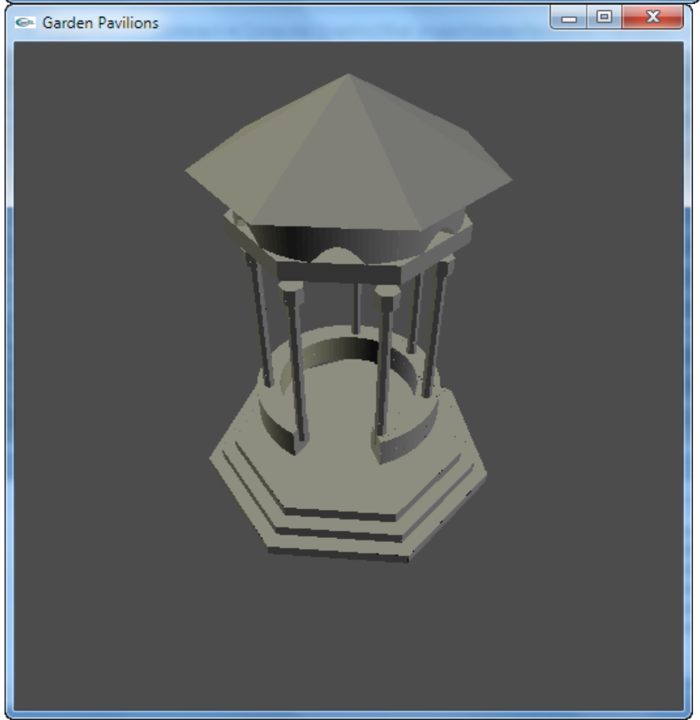
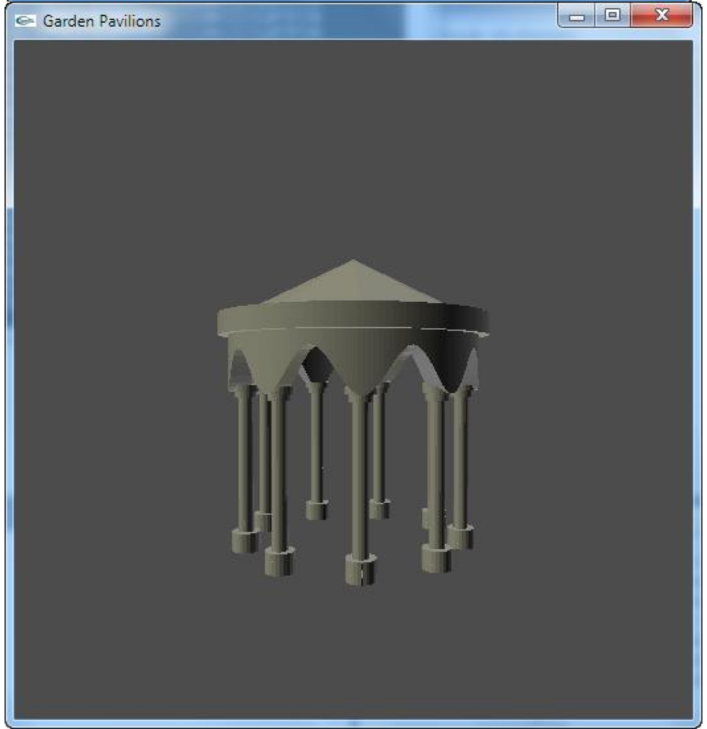
Purdue University

Motivation

Hua Liu, Qing Wang, Wei Hua , Dong Zhou, and Hujun Bao,2005.
Building Chinese Ancient Architectures in Seconds

- Goal:
  - Procedurally generate model of garden pavilions using constructive grammar

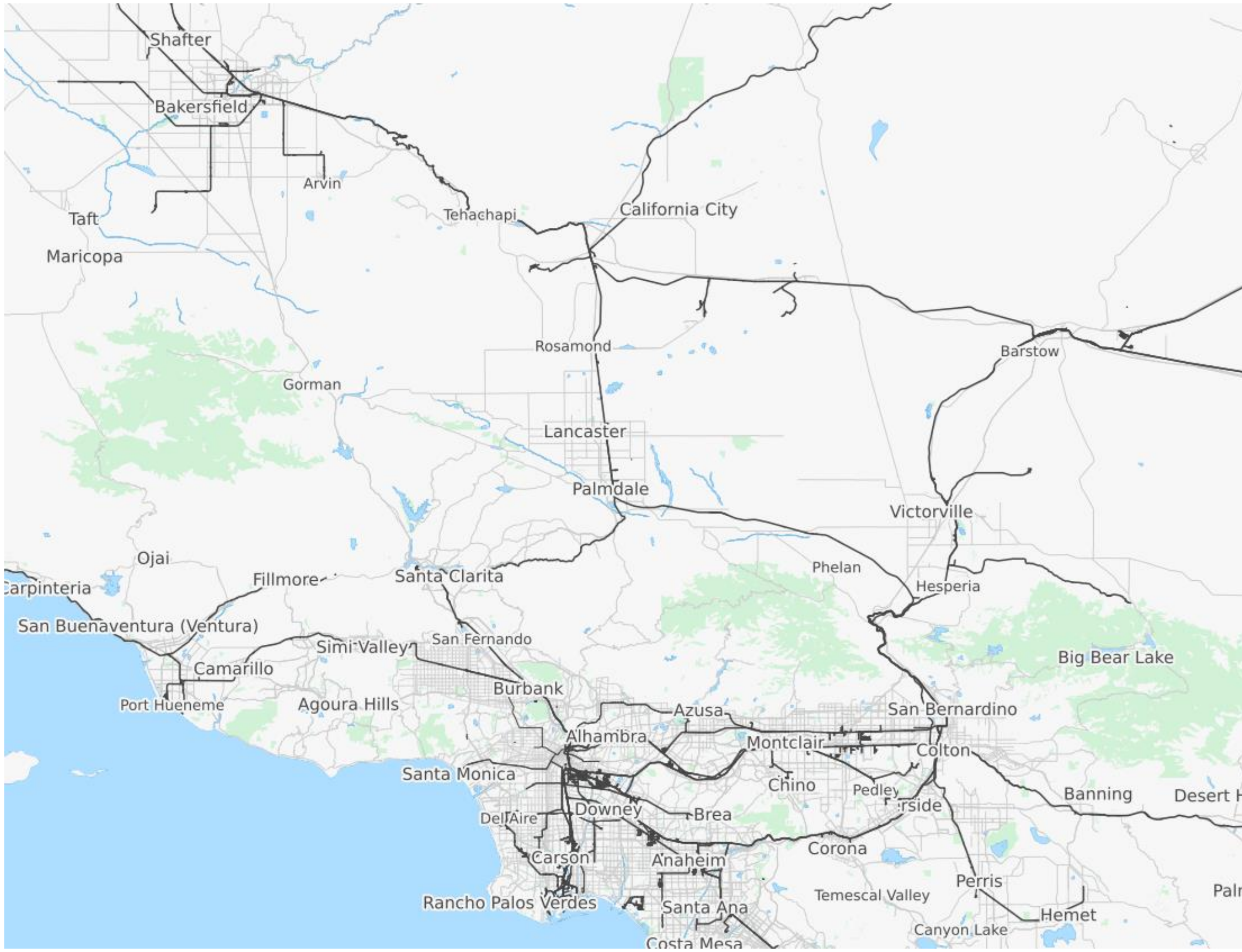

Garden Pavilions

Garden Pavilions

Garden Pavilions

Garden Pavilions

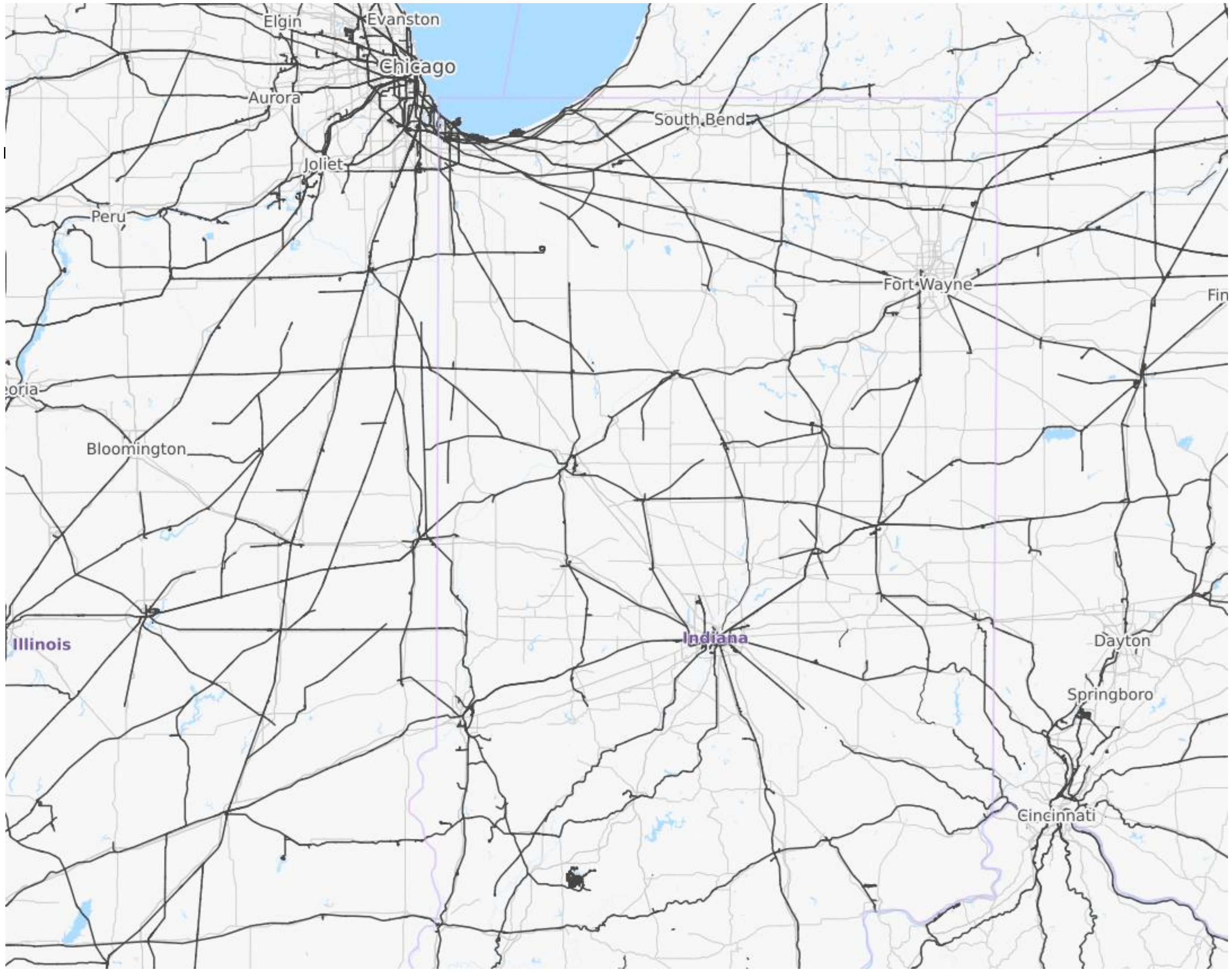

Garden Pavilions

Garden Pavilions

Garden Pavilions

Garden Pavilions

Procedurally Generated Transport Networks

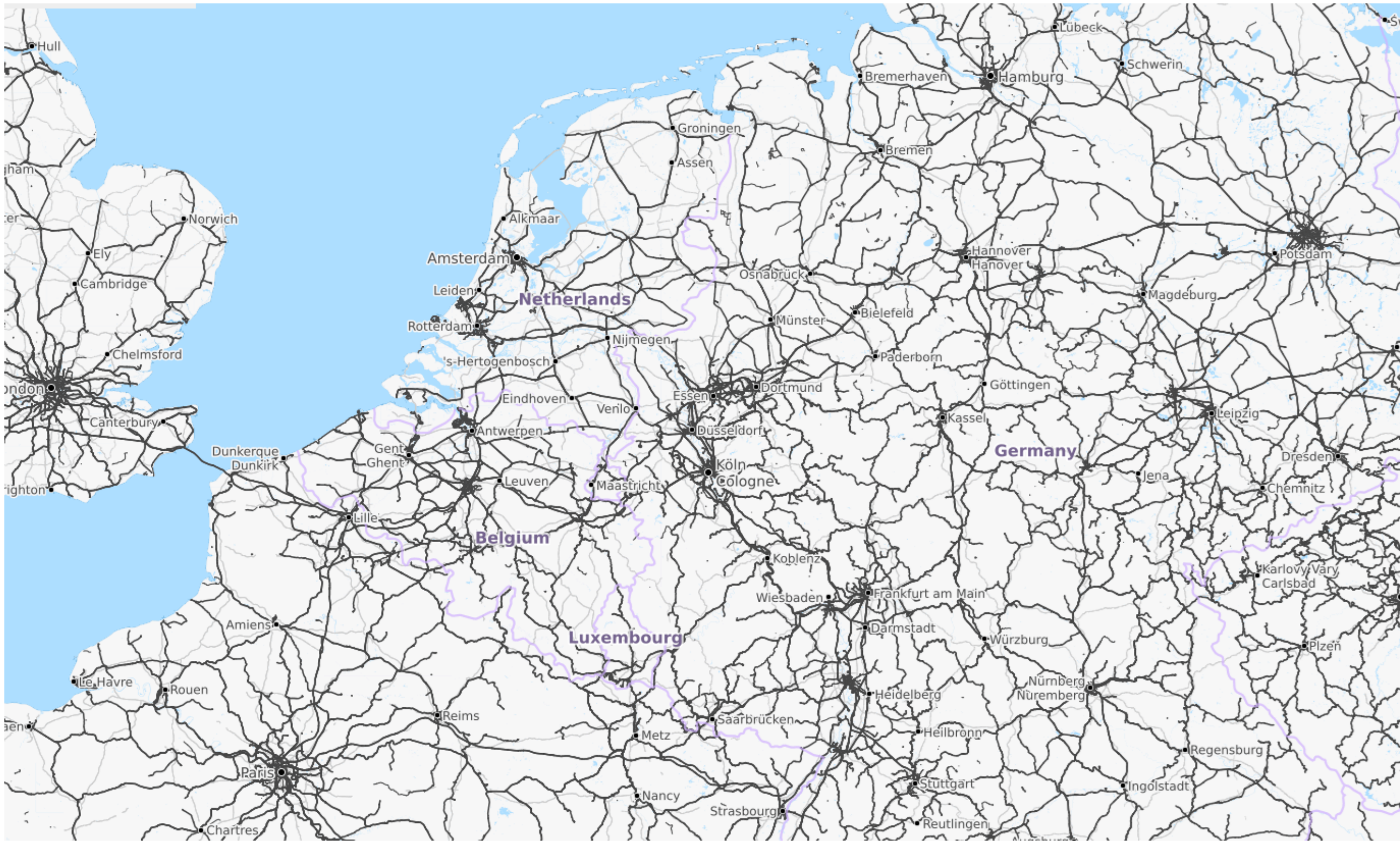
R. D.

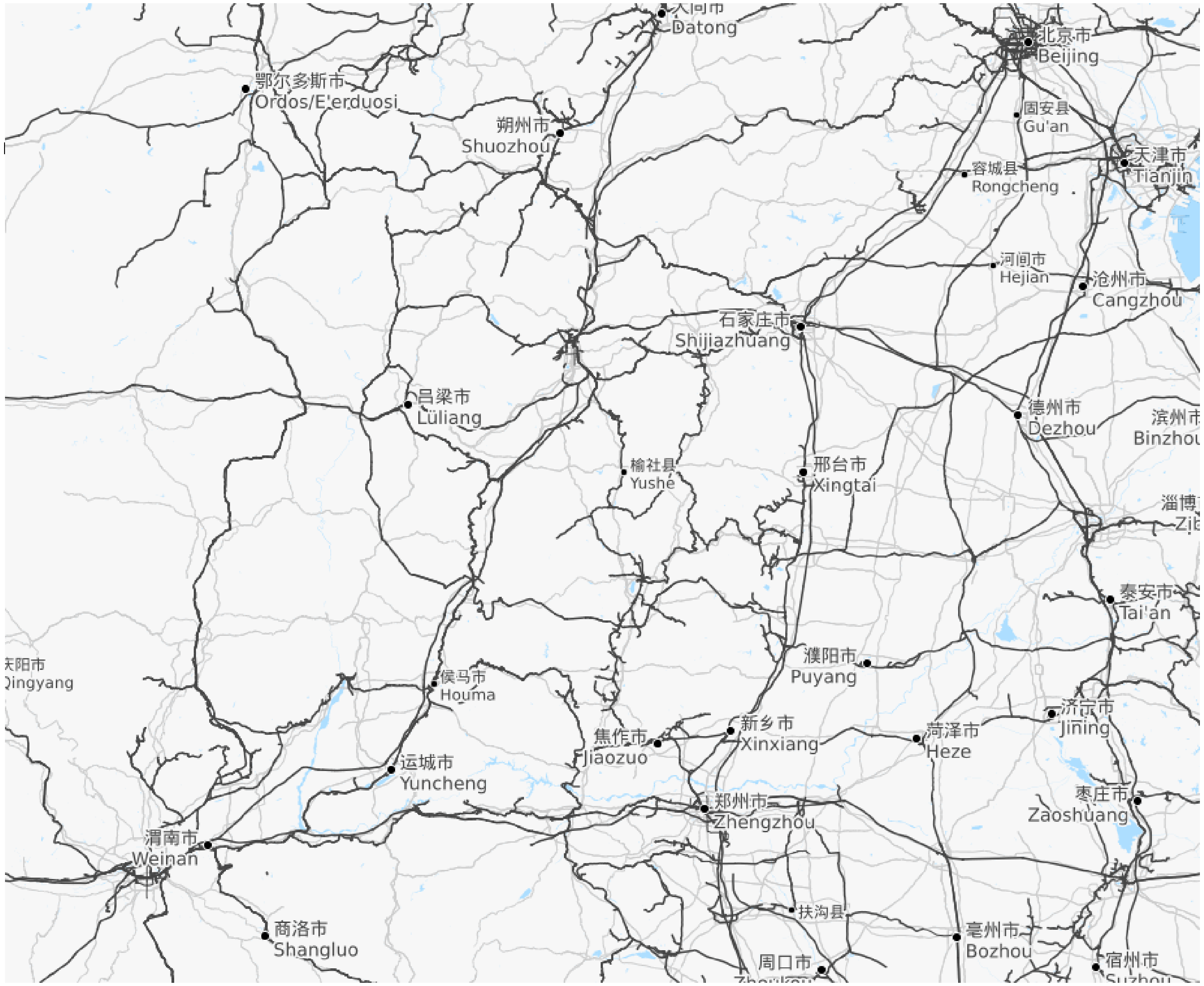
Transport Network Types - Gridded




Elgin
Evanston
Chicago
Aurora
South Bend
Joliet
Peru
Fort Wayne
Fin
eoria
Bloomington
Illinois
Indiana
Dayton
Springboro
Cincinnati

Transport Network Types - Interconnected

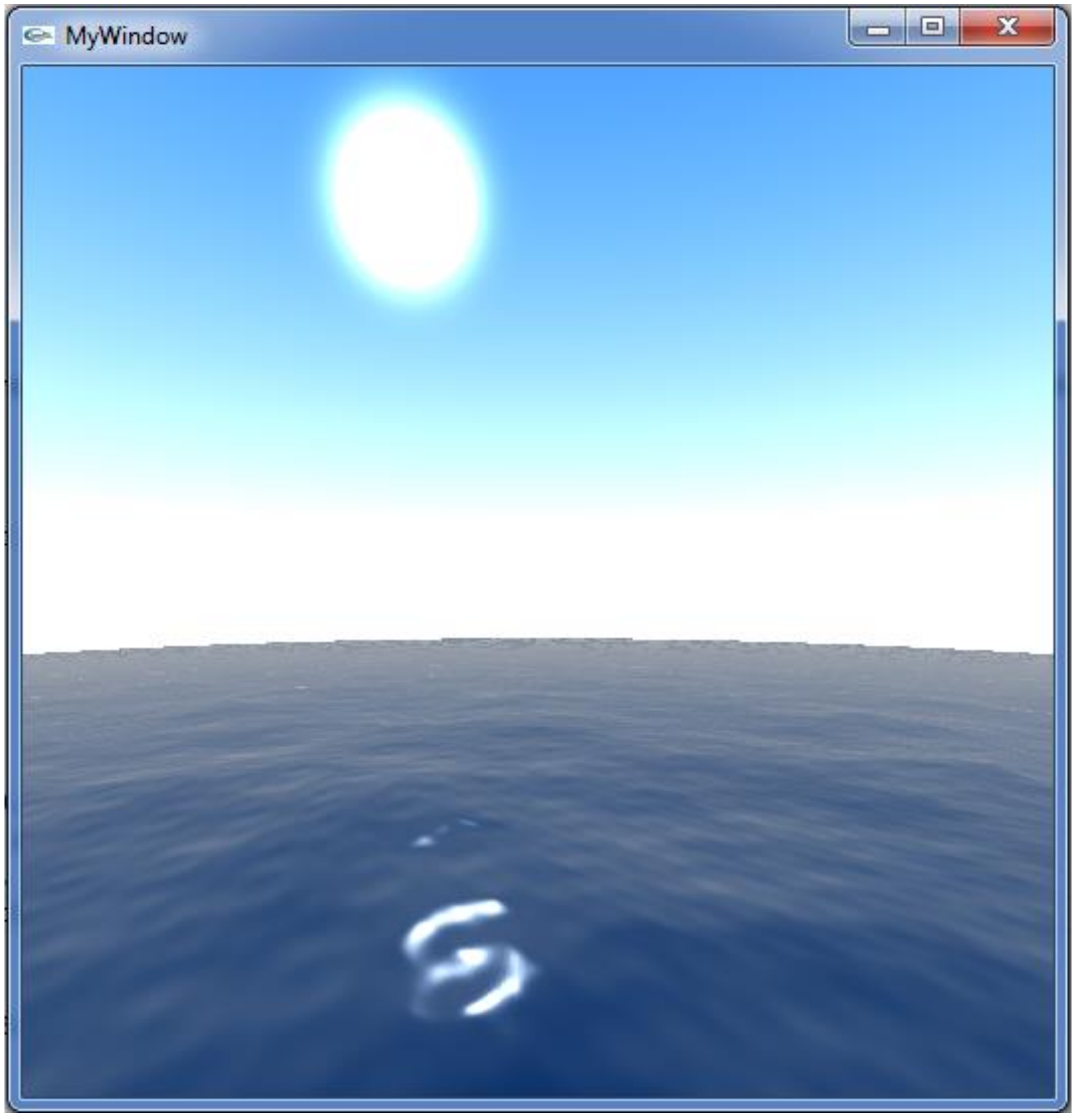



大同市
Datong
北京市
Beijing
鄂尔多斯市
Ordos/E'erduosi
朔州市
Shuozhou
固安县
Gu'an
天津市
Tianjin
容城县
Rongcheng
河间市
Hejian
沧州市
Cangzhou
石家庄市
Shijiazhuang
吕梁市
Lüliang
德州市
Dezhou
滨州市
Binzhou
榆社县
Yushe
邢台市
Xingtai
淄博
Zib
泰安市
Tai'an
庆阳市
Qingyang
侯马市
Houma
濮阳市
Puyang
济宁市
Jining
新乡市
Xinxiang
焦作市
Jiaozuo
菏泽市
Heze
运城市
Yuncheng
枣庄市
Zaozhuang
郑州市
Zhengzhou
渭南市
Weinan
扶沟县
商洛市
Shangluo
周口市
Zhoukou
亳州市
Bozhou
宿州市
Suzhou

3-D Water Simulation

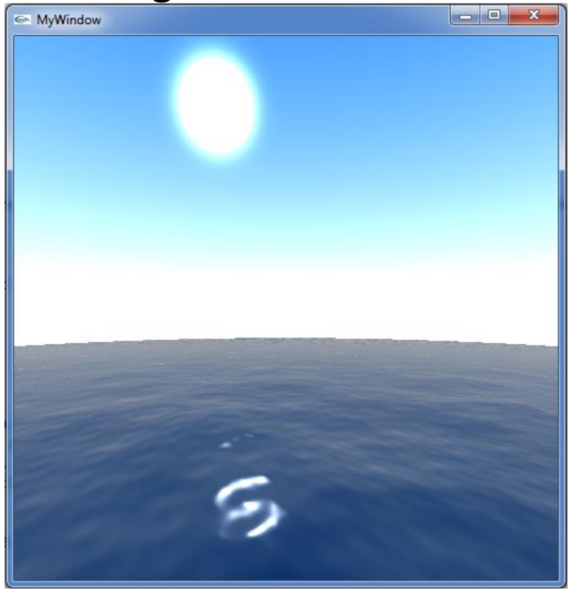
Z. J.

Demonstration

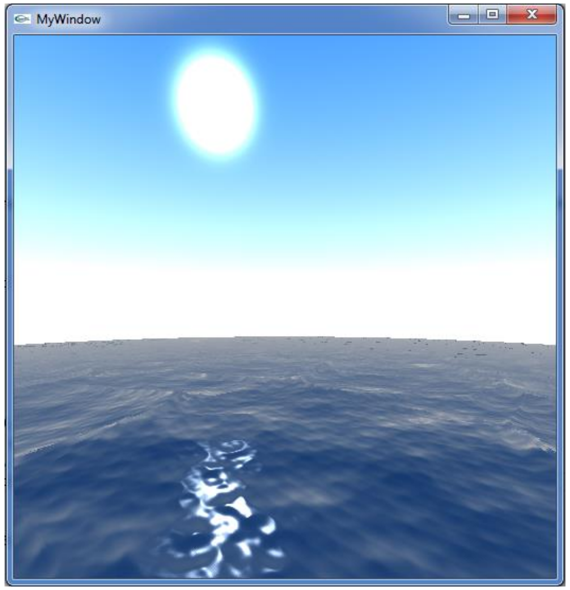


Wave Intensity

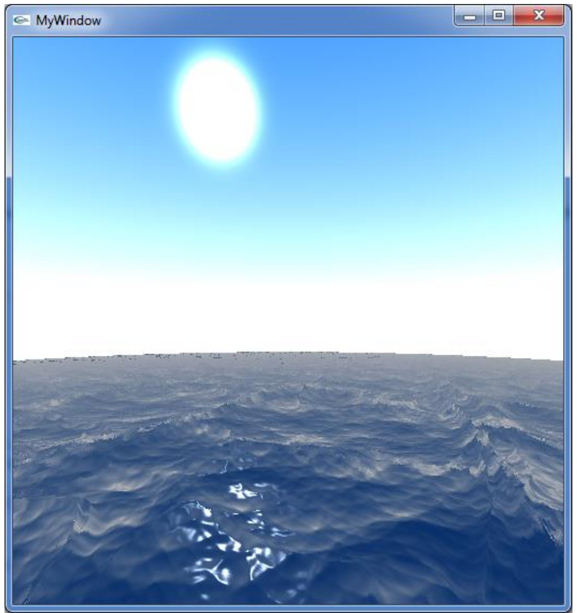
Light Waves



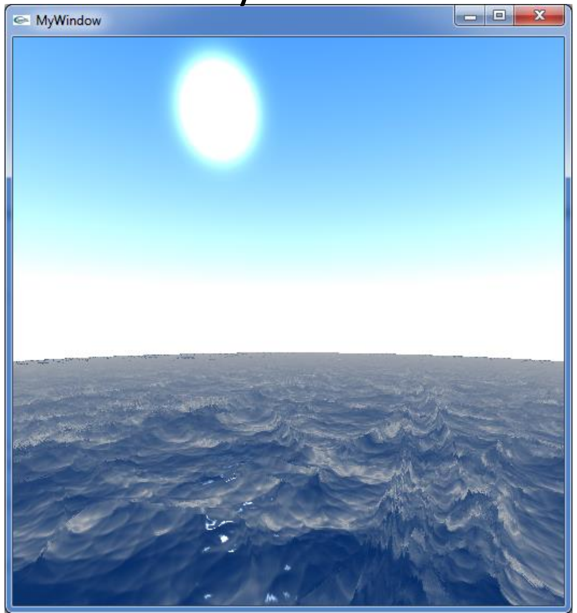
Small Waves



Medium Waves



Heavy Waves



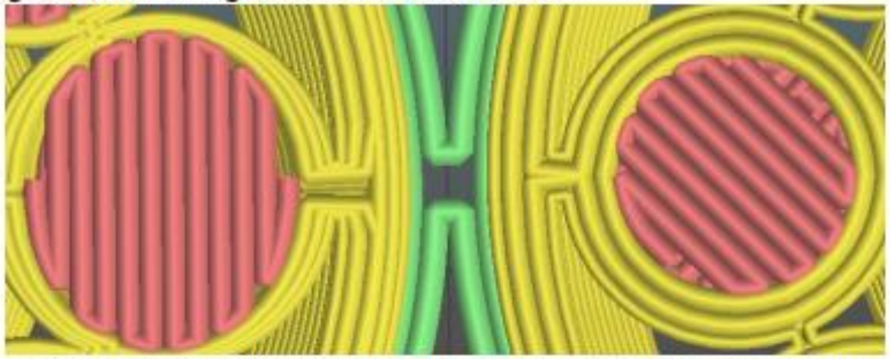
Alternative Nozzle Geometry Slicer Toolkit or… ANGST

A. P.

Exactly What I Did:

alter all print paths on perimeter

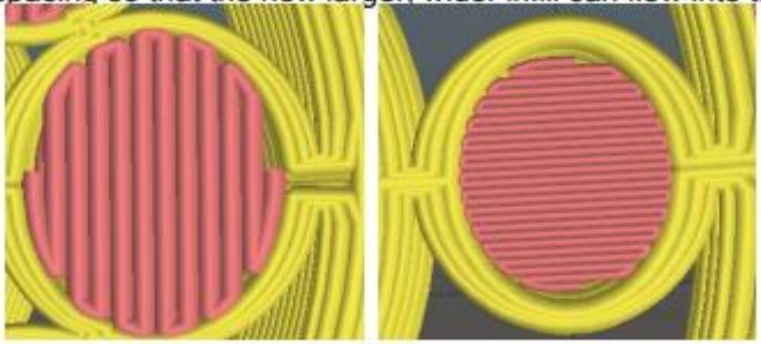
- Multiply the normal of all line segments by the nozzle shape (x,y)
- Shift the edge location along the new 'normal' vector


Shape with compensated geometry to stretch the part so it prints circular (left) vs traditional circular part (right)

Exactly What I Did:

Alter the infill so it fills in the thickest lines possible

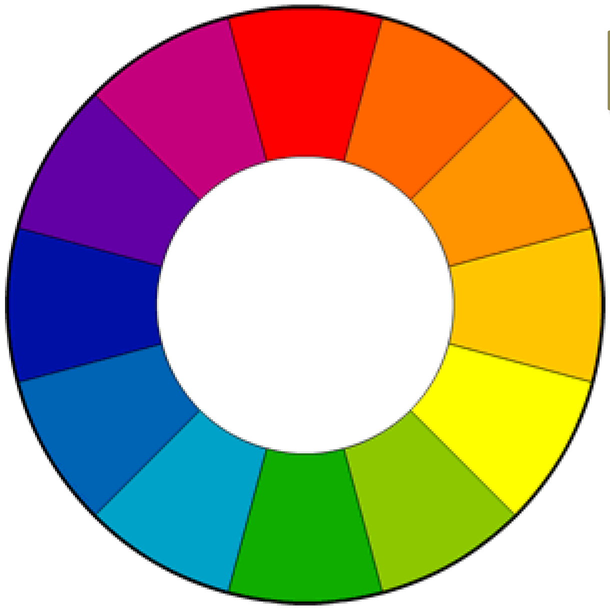
- Aggressively aligns the infill to be perpendicular with the nozzle widest access
- Expands the infill spacing so that the now larger, wider infill can flow into the space.

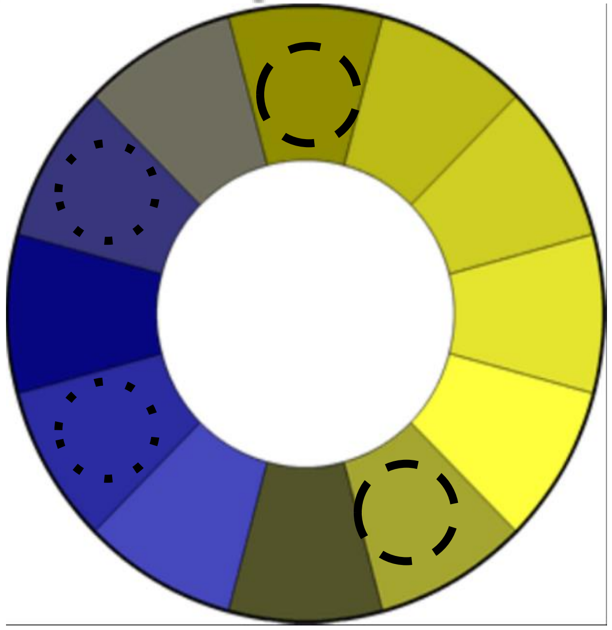

Infill with infill aligned with the nozzle width (left) with the dumb infill, printed denser due to the need to fill in the space.

Real-Time Assistance for Colorblind Gaming

R. V.

Goal: Make differences along the red-green color axis perceptible by h color





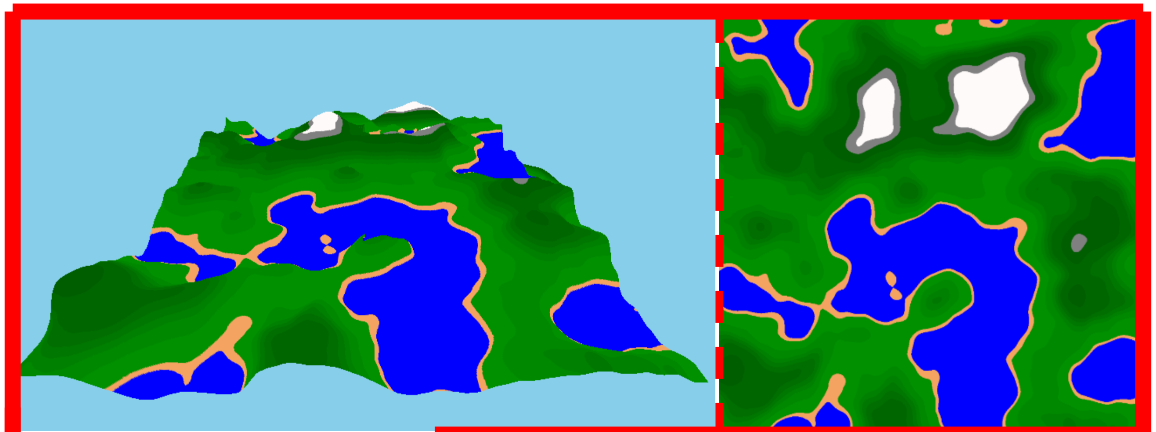
Procedurally Generated Terrain

B. W.

Motivation

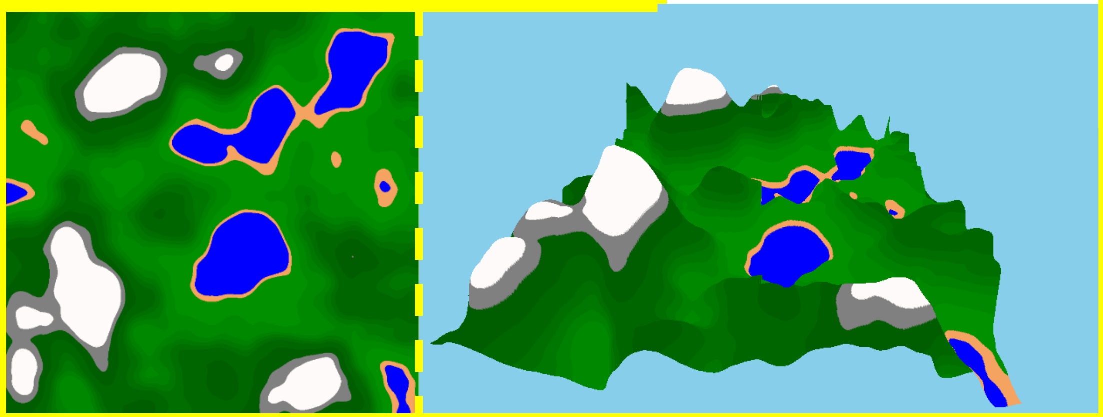
- I have been interested in procedural generation since before I was a computer science student
- Procedural generation is almost making an artist out of a computer program
- An interest in video games and their immersive environments at a young age pushed me to explore this topic
  - Minecraft
  - Terraria
  - The Binding of Isaac
  - Spelunky
  - Sid Meier's Civilization
  - Diablo

Examples of my Terrain

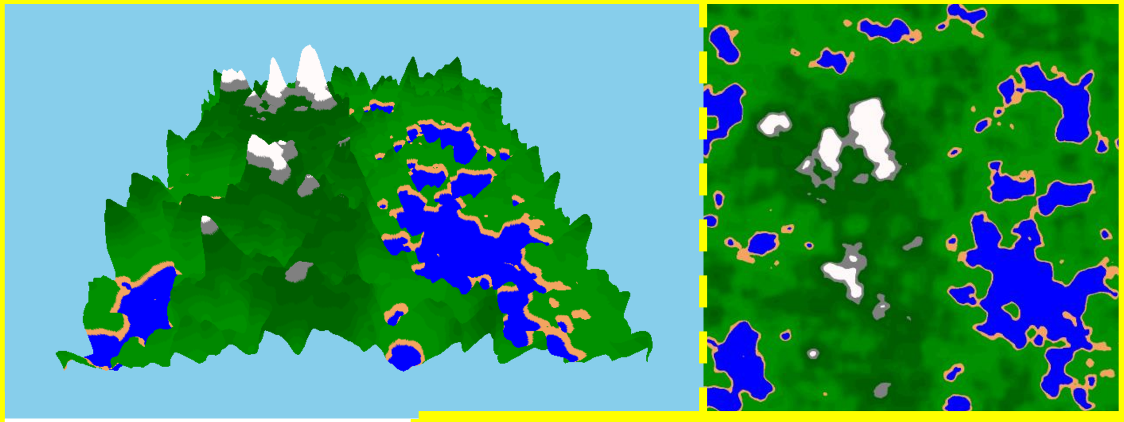


- Intensity = 100
- Roughness = 30

- Intensity = 200
- Roughness = 30

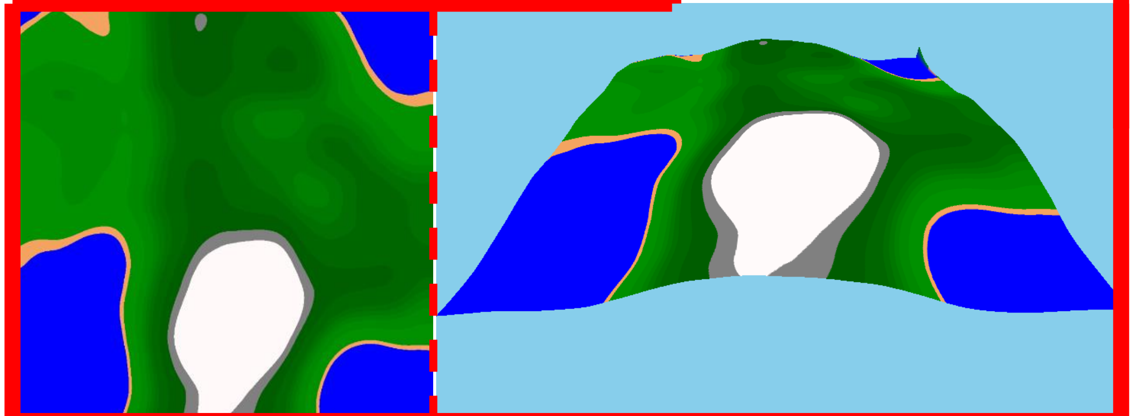


Examples of my Terrain (extremes)

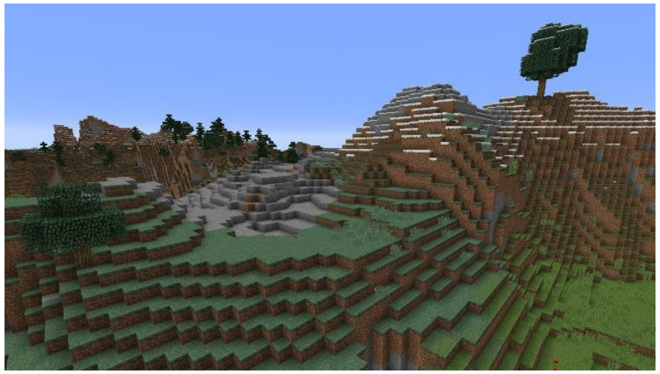


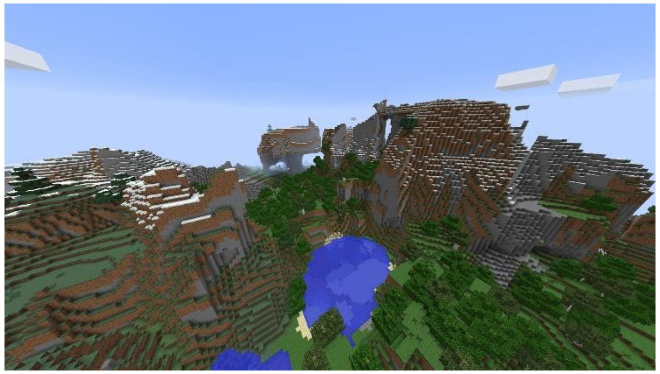
- Intensity = 100
- Roughness = 100

- Intensity = 100
- Roughness = 10



Examples of this in Games



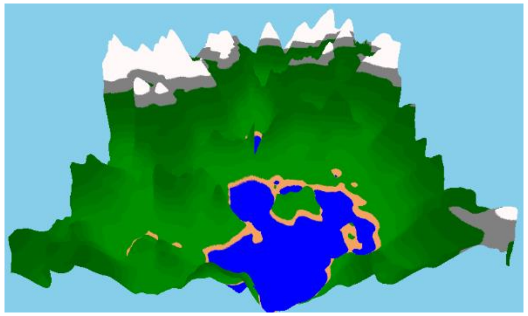


Similar to average values in terrain

Roughness ~50

Similar to higher values in terrain generator

Intensity ~200, Roughness ~50

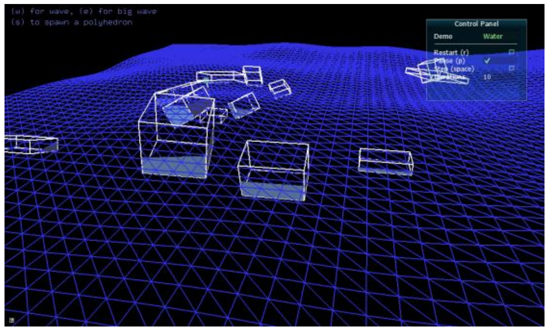


Final Presentation: Water & Buoyancy

E. Z.

Motivation

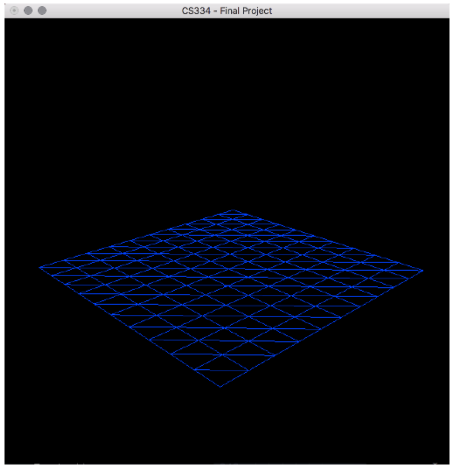
- Improve upon Assignment 2
- 3D simulation of water

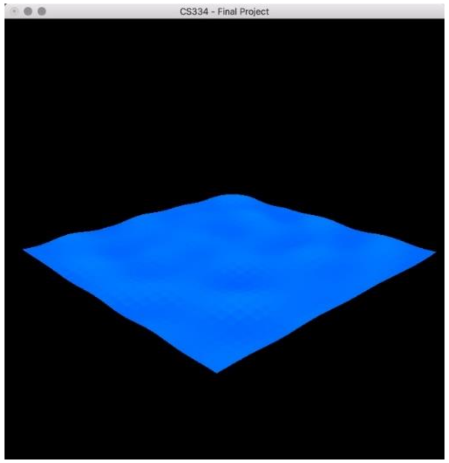


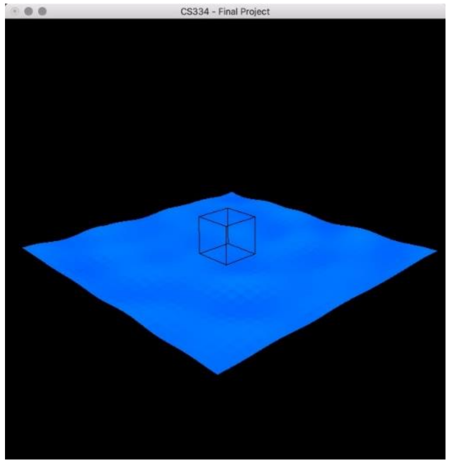
(w) for wave, (e) for big wave
(s) to spawn a polyhedron

Approach

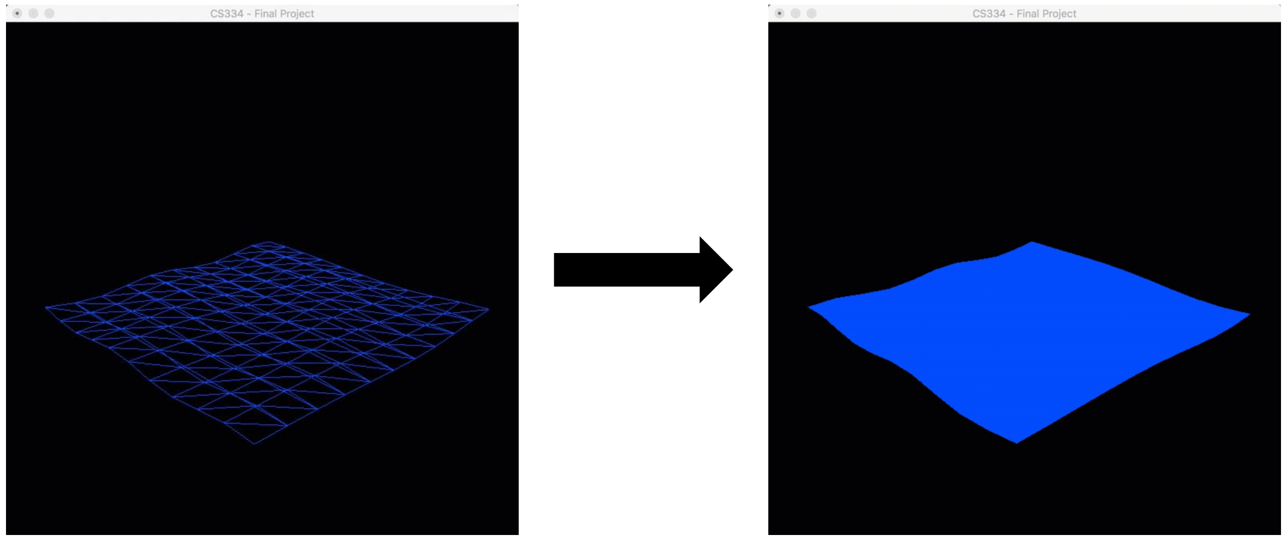
- 3D representation of waves
- Sum of sine waves
-



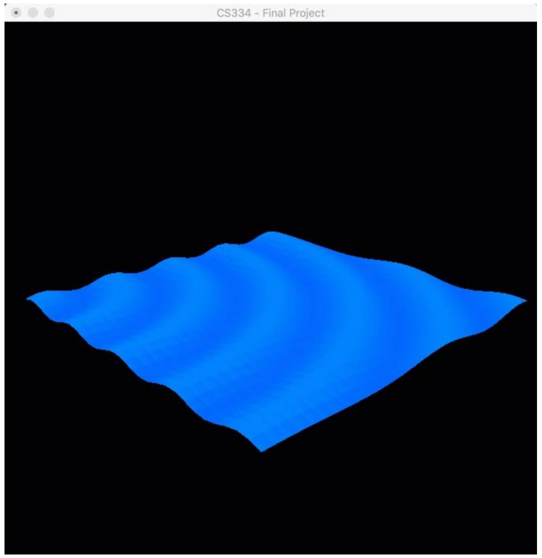


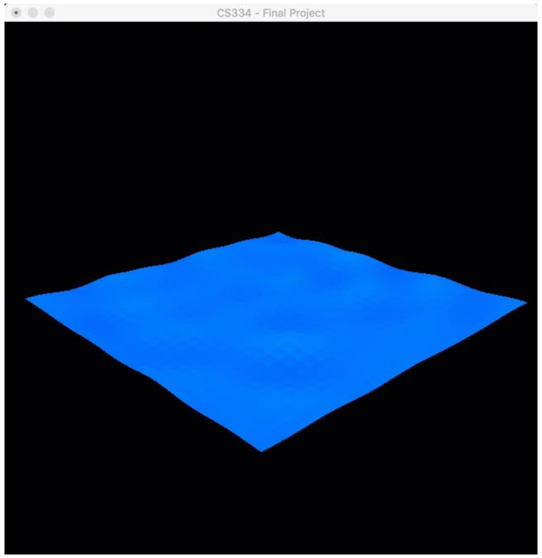


Approach – 3D wave representation



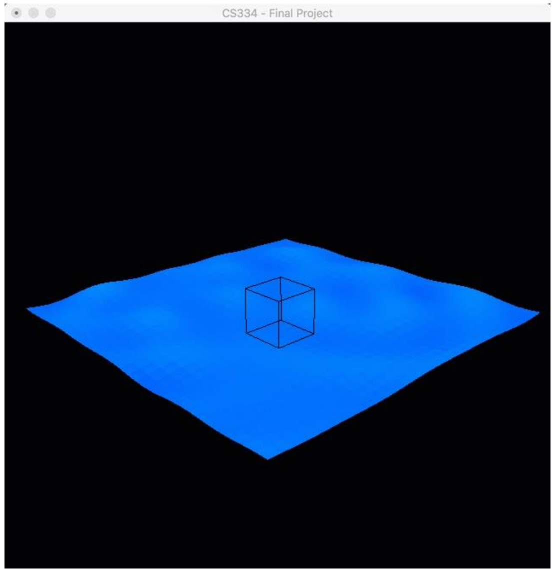
Approach – sum of sine waves





- One sine wave from the origin (left corner)
- Four sine waves from various starting points

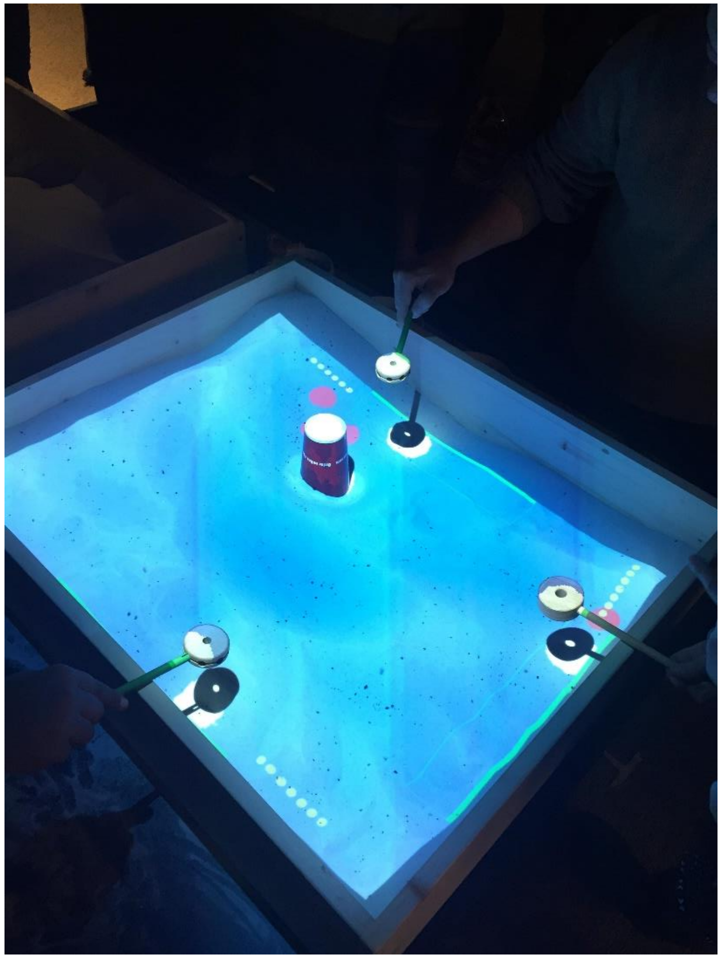
Approach - buoyancy

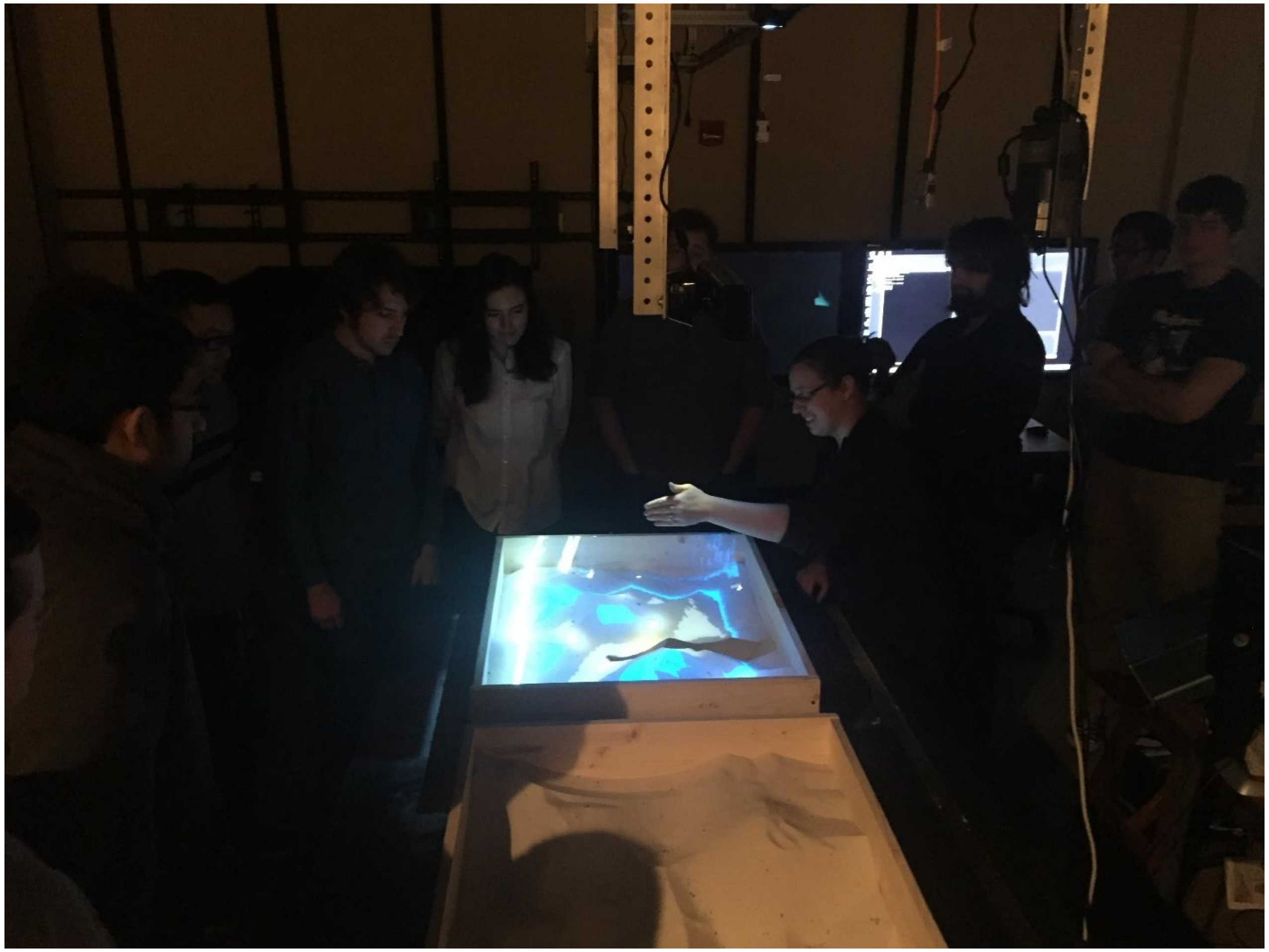


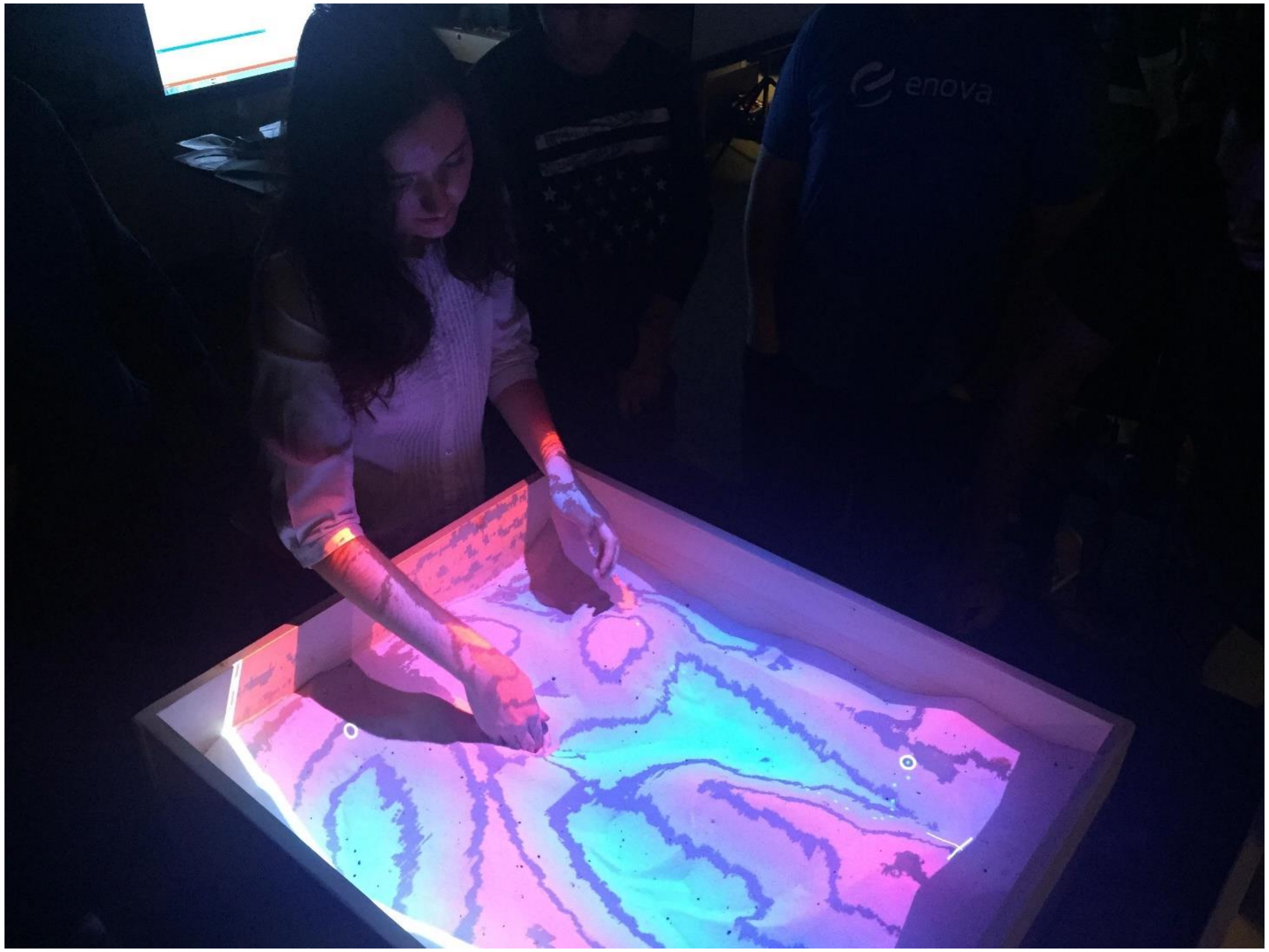
Rigid Bodies

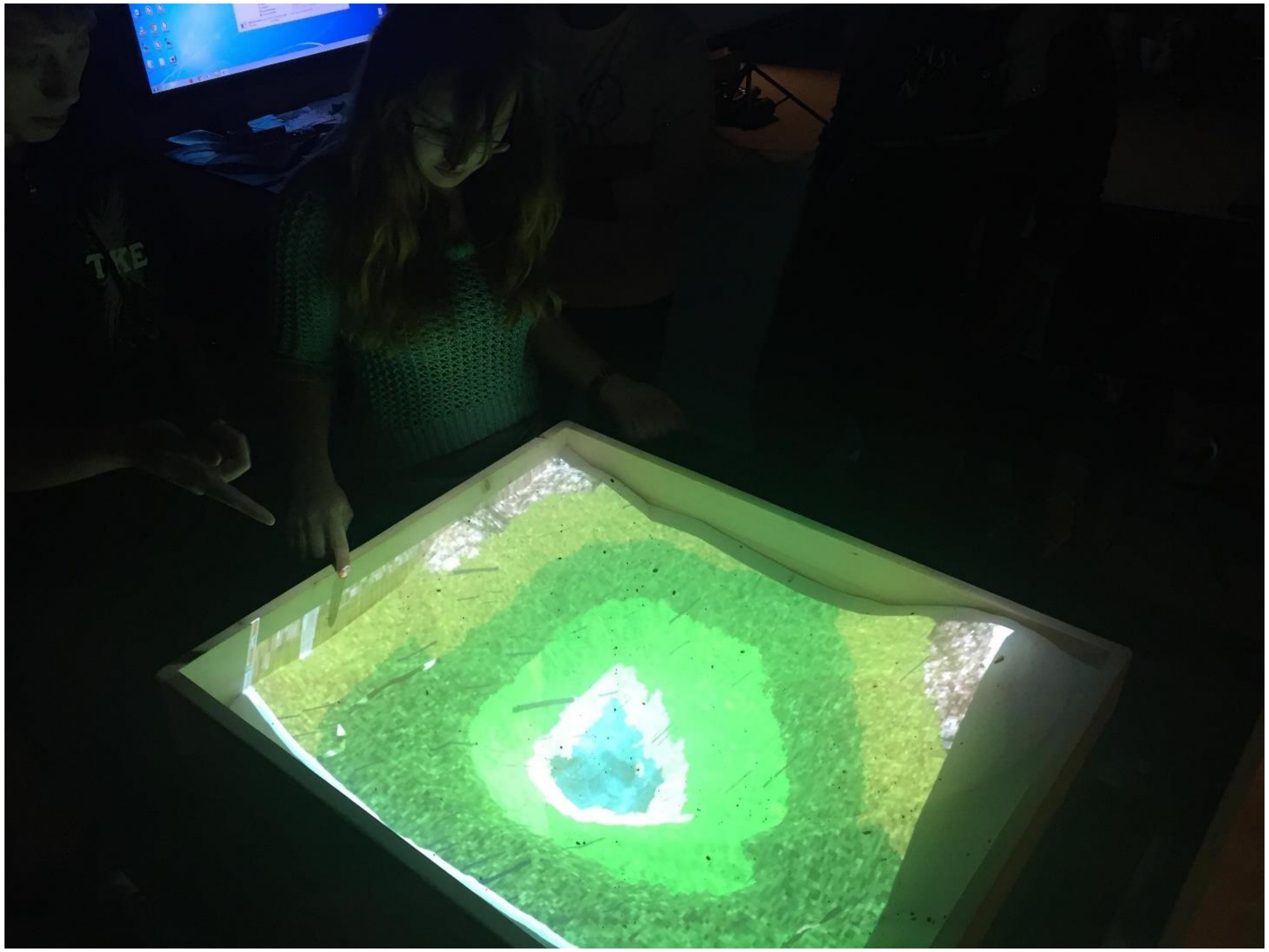
Zachary Neumann

SandBox Misc…

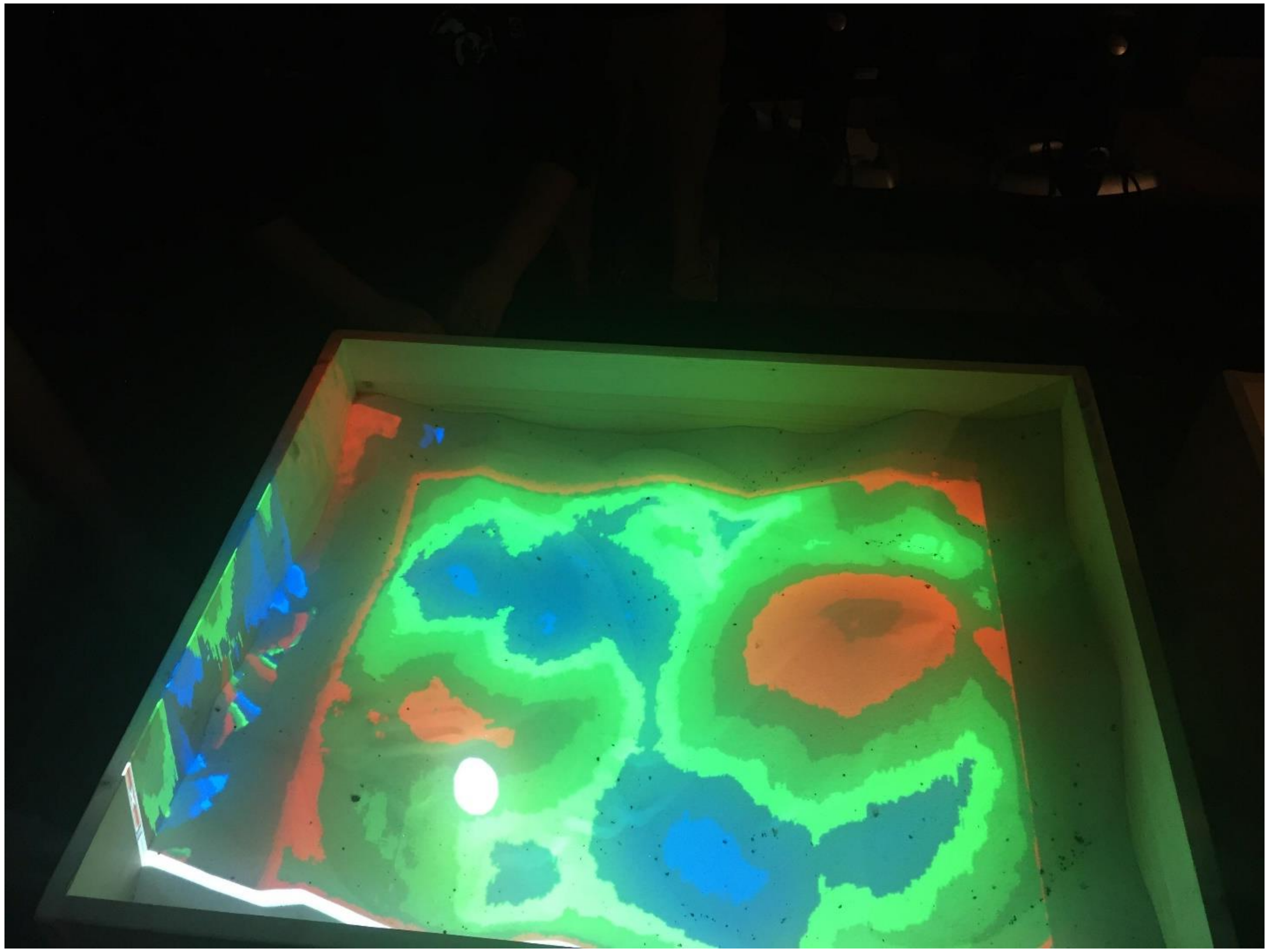





enova



Believe
Beauty

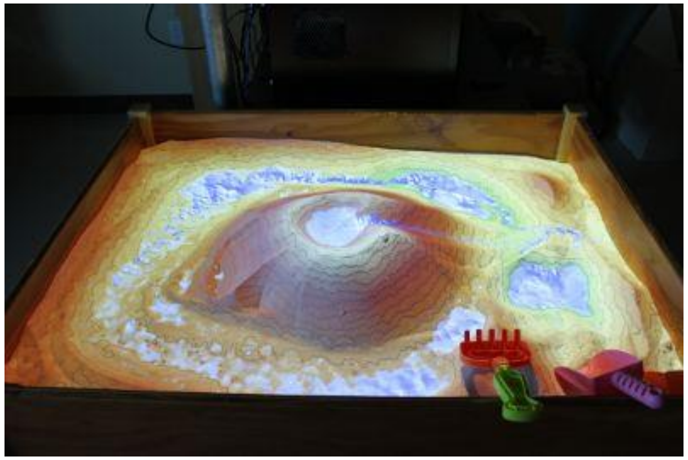


Virtual Sandbox & Game

Y. L.

Motivation

Virtual Sandbox



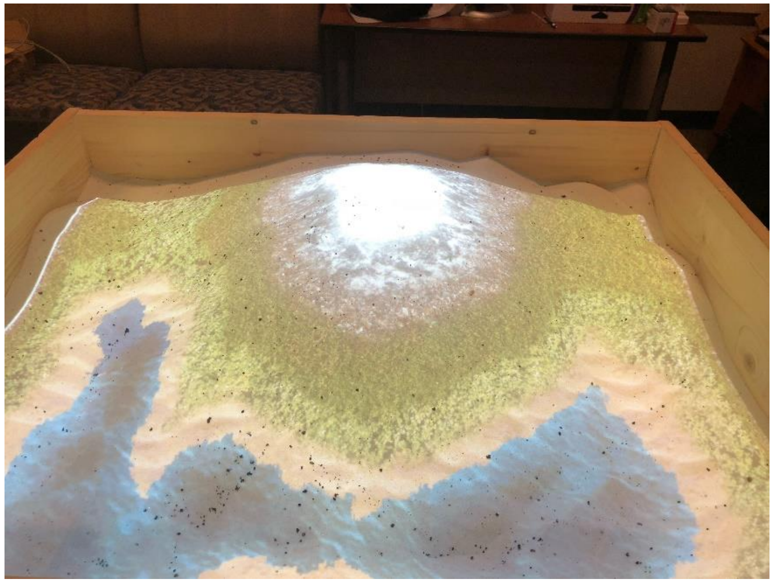
Treasure Hunt

SandBOX Observation

- Layout

Snow >> Rock >> Tundra >> Forest >> Grass >> Sand >>Water

- Mix texture based on height between sections



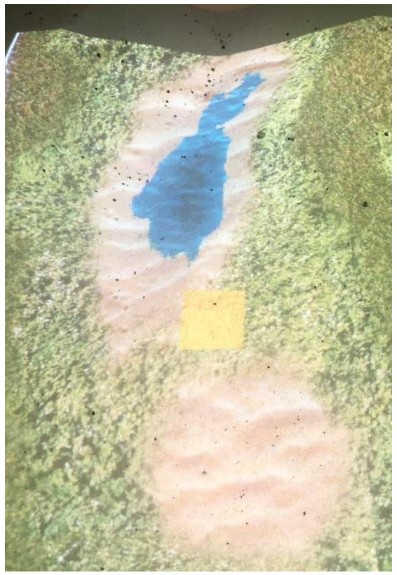
SandBOX APPROACH

- Load Image
- Generate Texture
- Bind Texture with sampler
- Modify vertex and fragment shader

GAME OBSERVATION

Hide

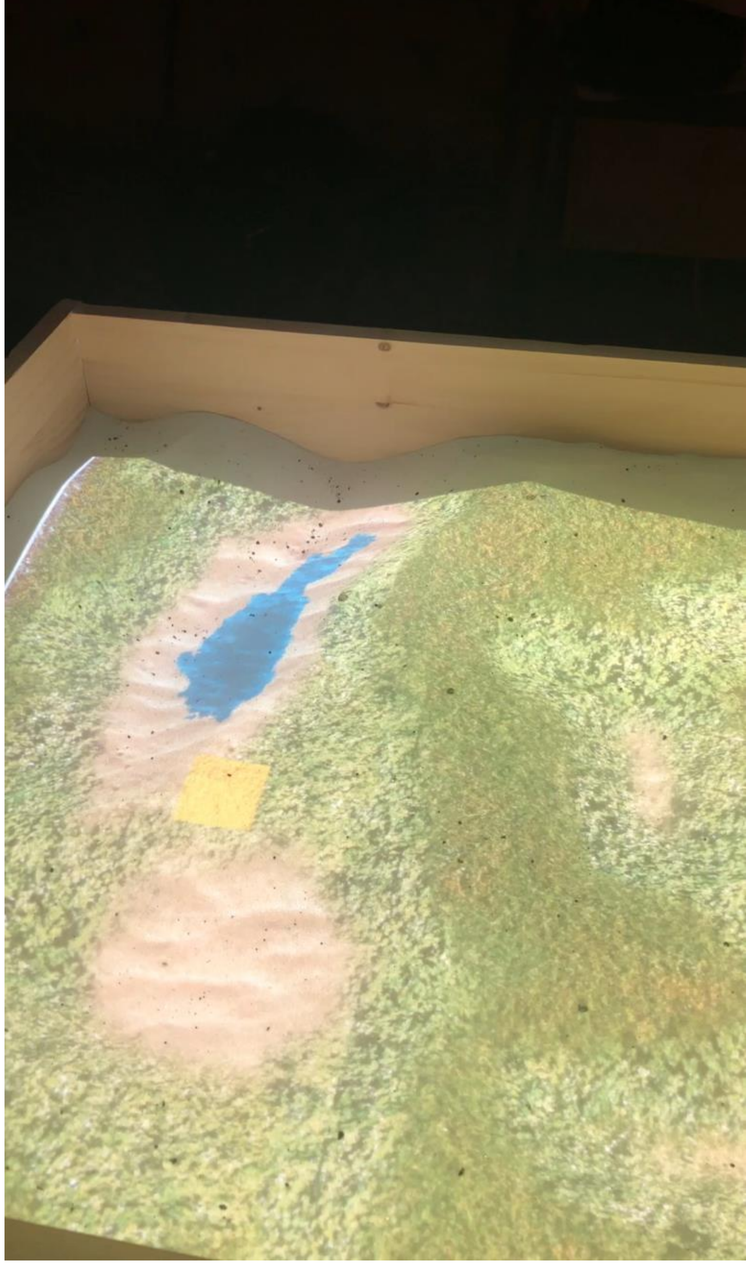
Show



GAME DEMO

Mechanism of Game:
Use high location hand wave to update location randomly

Demo:
Set the gold block always visible

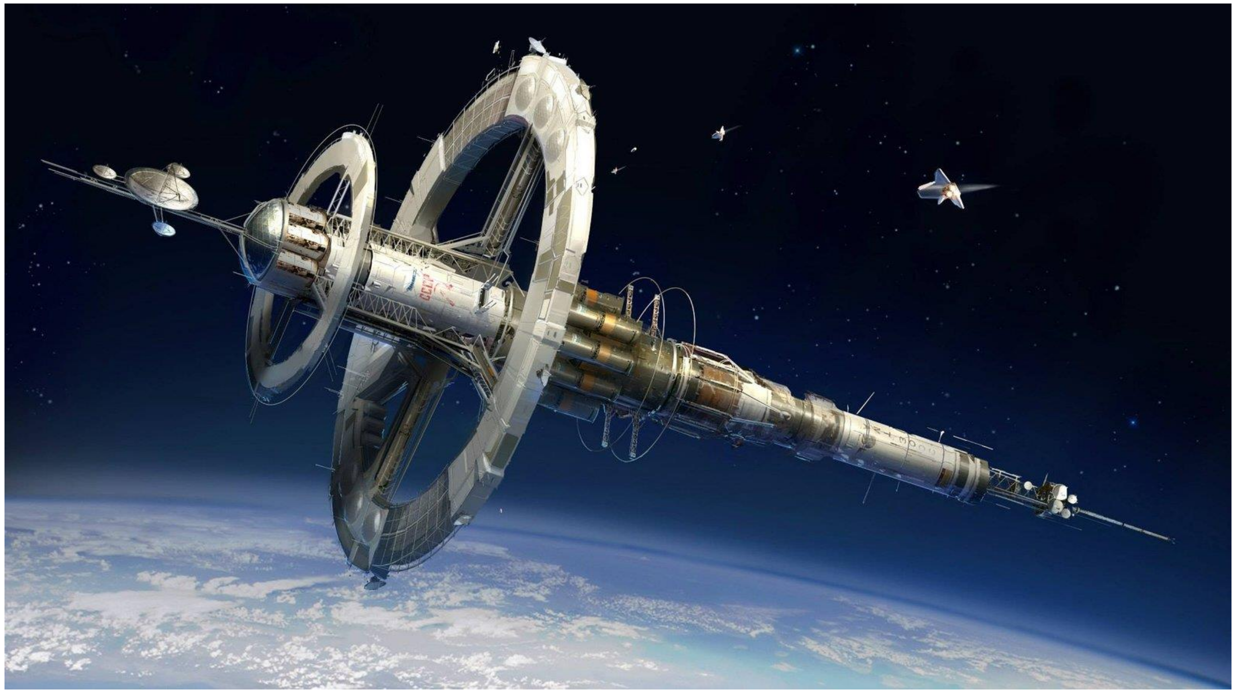


Proposals (Fast Forwards)

Procedural Modeling with sci-fi aesthetics

B. A.

Inspiration



▪http://www.deviantart.com/art/FUSE-Soviet-Space-Station-375498745

Inspiration

- Mass Effect 2 - Bioware
- http://www.gamebreaker.tv/news-main/pc-2/mass-effect-4-survey-leak-gives-possible-look-at-game-details/

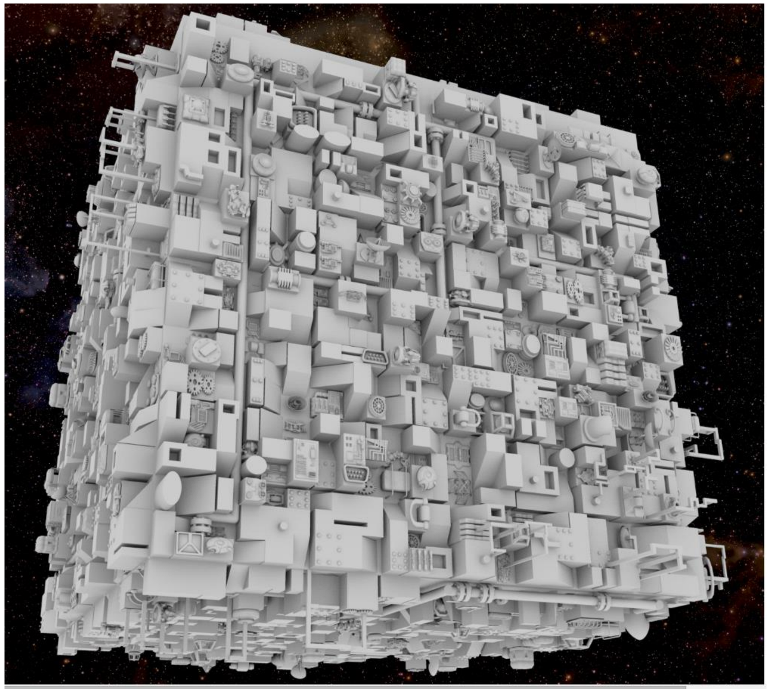
Project Goals

- Explore generation of different space stations styles
  - Perhaps 'Borg' vs 'Watchtower'
- Focus will be on geometric modelling as

▪Procedural generation of surface detail for science fiction spaceships
▪-Kinnear, Kaplan 2010

Implementation

- C++ and OpenGL w/GLSL shaders, extending the framework built from course assignments
- Split/shape grammar rules guided by input file


▪Procedural generation of surface detail for science fiction spaceshi
▪-Kinnear, Kaplan 2010

Demo

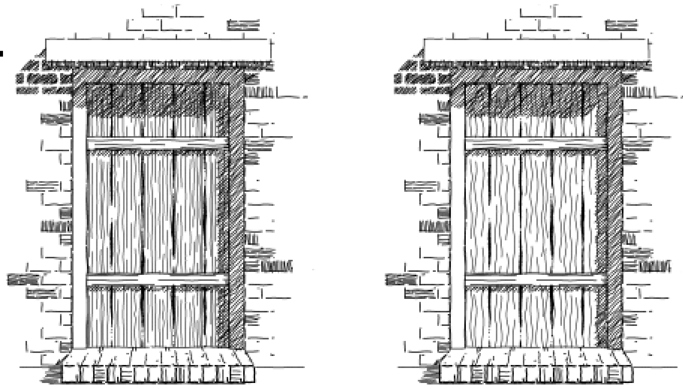
- Camera explorable OpenGL scene to simplify viewing results produced with parameters from input file

Pen-and-Ink Non-Photorealistic Renderer

S. K.

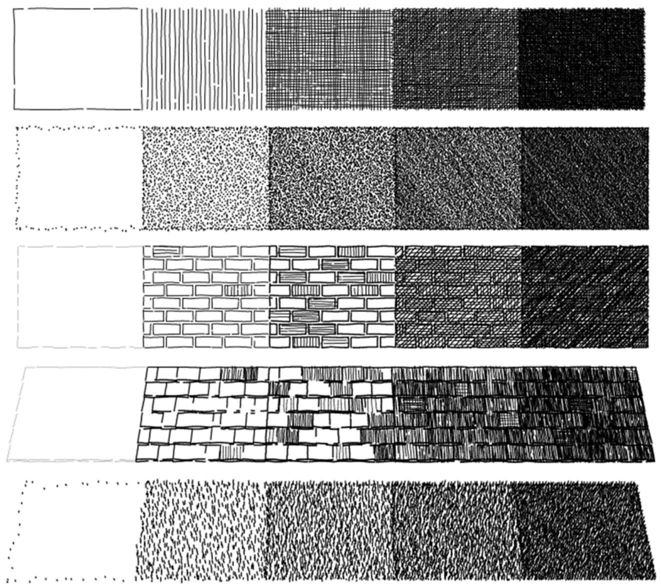
Pen-and-Ink NPR

- Develop and implement a pen-and-ink renderer using precreated tonal maps
- Building off of the work done in Computer Generated Pen- by Winkenbach et



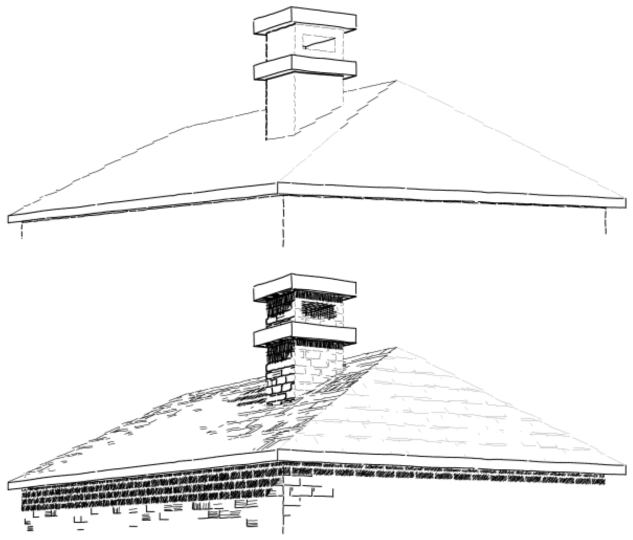
Example Precreated Tonal Maps

- Assign tone dependent on intensity of light
- Assign texture from user input



Outlines

- Boundary outlines express texture
  - Each stroke texture has corresponding boundary outline
- Minimize interior outlines
  - Only drawn when the tones of two neighboring faces are not very distinguishable
- Accent outline
  - Used for shadowing and relief



Demo

- Real-time rendering of several sample objects and manipulating view point
- Real-time rendering of an environment from multiple view points.

Image Sources

- All images retrieved from:

  ftp://ftp.cs.washington.edu/tr/1994/01/UW-CSE-94-01-08b.d/UW-CSE-94-01-08b.pdf

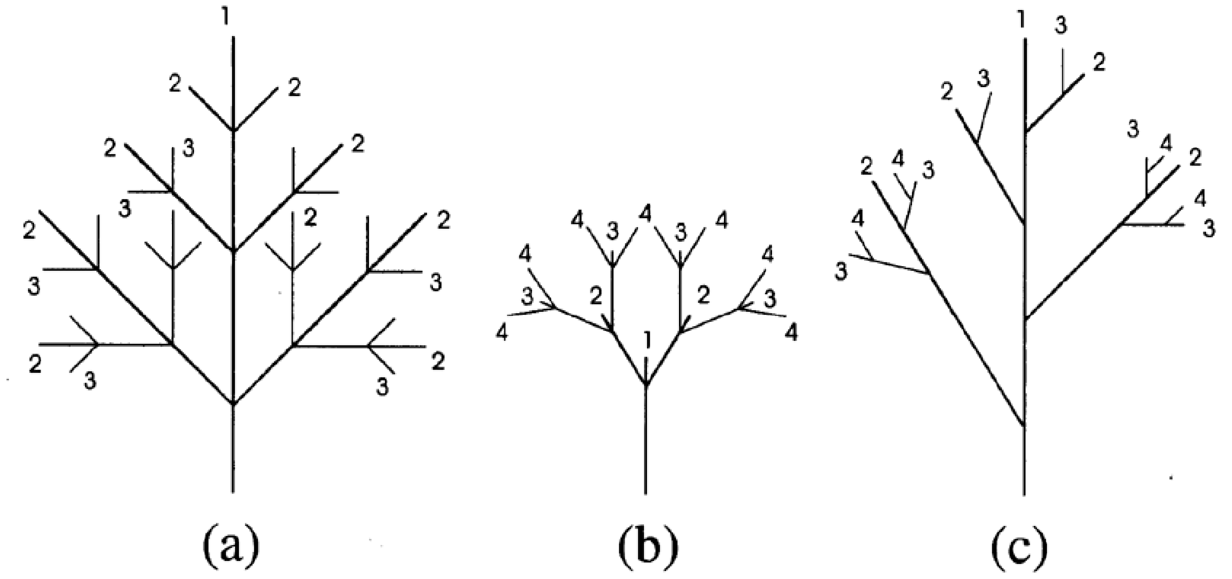
Terrain Procedural Modeling

D. K.

Terrain Procedural Modeling

What Am I Going To Do?

1. Procedural Modeling to Create a River



What Am I Going To Do?

2. Based on river location generate the terrain
   1. Generate down-sloping terrain from mouth of river
   2. Near river add more foliage / green terrain
   3. Away from river procedurally generate terrain

Demo

Procedural Modeling and Fluid Dynamics Simulation

Terrain and flora with water generated through SPH

I. O. and W. B.

Overview

- A 3D Generated Scene/Game
- This will be split into three main parts
  - Terrain generation
  - Plant generation
  - SPH water simulation

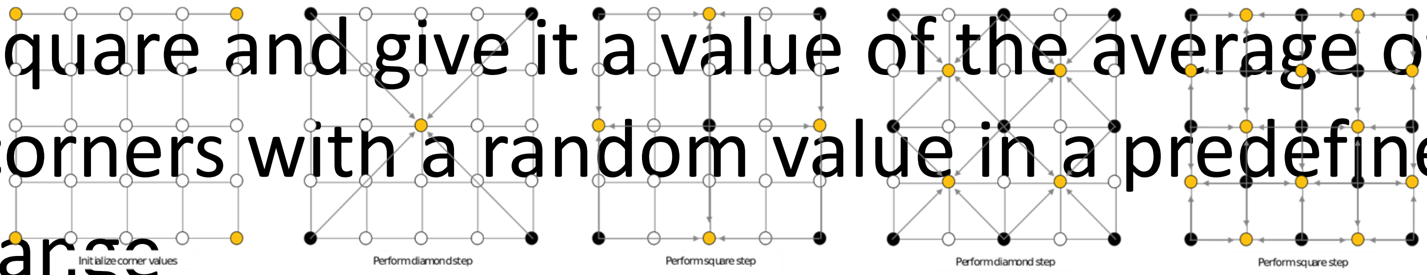
Main Aspects

- Procedurally generated environment
- Realistic fluid flow dynamics.
- User can create/interact with water.

Terrain Generation

- We will be using the **Diamond Square Algorithm** to generate random terrain
- The following is constantly looped:

  - **Diamond Step**: make a midpoint for every square and give it a value of the average of the corners with a random value in a predefined range

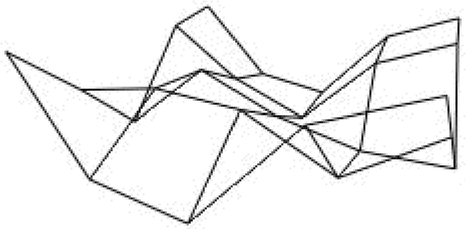
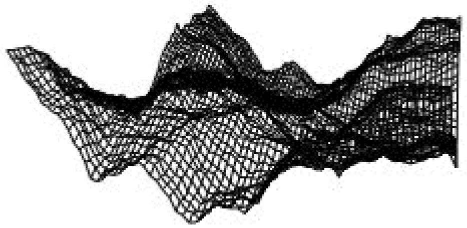
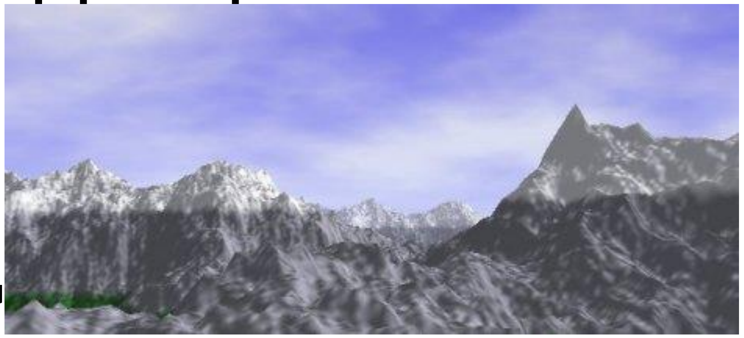


  - **Square Step**: make a midpoint for every diamond and give it a value of the average of the corners with a value in a certain range[1][2]

Terrain Generation

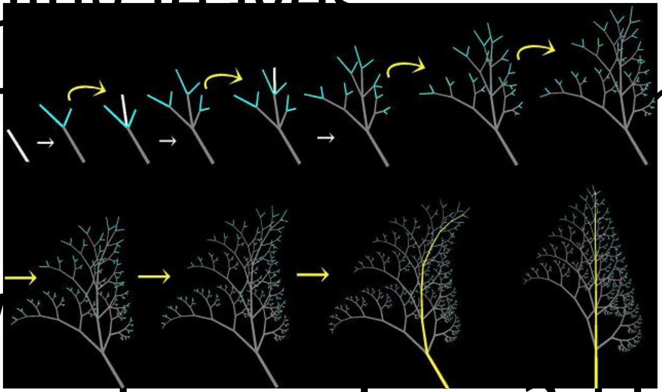
- Texture and colours will be added depending on the heights to make a more varied landscape, such as snow capped peaks
- Mountains could be f

added value at certain poi

Flora Generation

- We will use L System grammars to generate a variety of plants.[3]
- Tree will be generated by altering trunk size and adding leaves
- Differen napped on to the object
- Plants v at certain heights
- Plants can be made in 3d by adding a random direction for which the branches should stem from[4] [5]

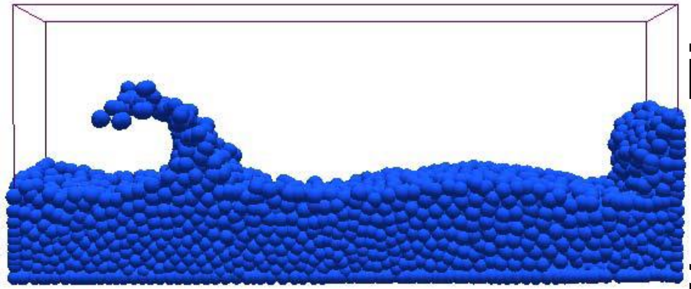
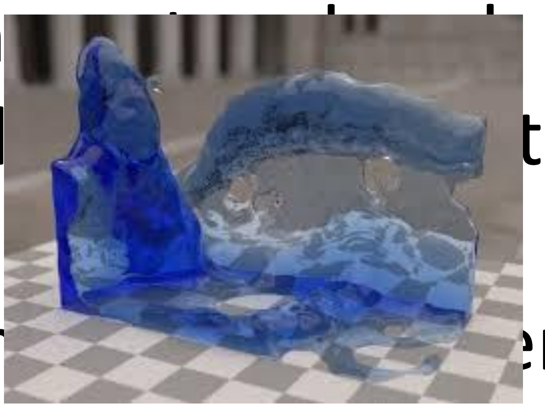


Fluid Simulation

- Lagrangian Method
  - coordinates move with the fluid
  - lower accuracy, but faster
  - prefered method for real-time fluid simulation
- Eulerian Method
  - grid-based
  - more consistent, but slower[6]

Smooth particle Hydrodynamics

- The concept for this method of water simulation is to constantly keep track of thousands of particles, while monitoring how they react to the environ … other
- … ionall … t great
- … s ther … er the top of the individual particles [7]

Smooth particle Hydrodynamics

- Each particle has a set of values stored such as position, velocity, density, mass and pressure and they values are used to calculate a new position after altering the acceleration at every iteration
- Basically calculating the change in velocity for every particle at every iteration in time: $\partial V/\partial t = A^{pressure} + A^{viscosity} + A^{gravity} + A^{external}$ [6]
- Every iteration the density and the pressure needs to be recalculated in order to find the

Team Dynamics

We will be using the agile development technique of pair programming so that we both get the experience in working through the different areas of this project. Also we will be able to program more efficiently as errors, tiredness and complications in code will be less of a problem.

References

[1]: Procedurally generated terrain tutorial through fractal : http://www.gameprogrammer.com/fractal.html
[2]: Diamond Square image: https://en.wikipedia.org/wiki/Diamond-square_algorithm#/media/File:Diamond_Square.svg
[3]: Plant generation: https://www.cs.purdue.edu/homes/aliaga/cs334-15fall/lectures/lec-proc-modeling.pdf
[4]: L Systems: http://algorithmicbotany.org/papers/abop/abop-ch1.pdf
[5]: Tree generated with L Systems: http://www.ms.is.ritsumei.ac.jp/profile/staff/ijiri/ProjSketchLsystem/index.html
[6]: SPH simulation, section 5: http://cg.informatik.uni-freiburg.de/intern/seminar/gridFluids_fluid-EulerParticle.pdf
[7] SPH: http://www.glowinggoo.com/sph/bin/kelager.06.pdf

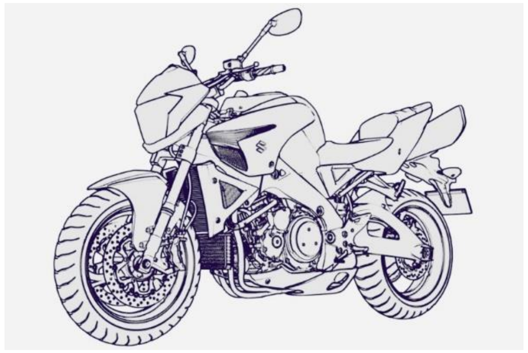
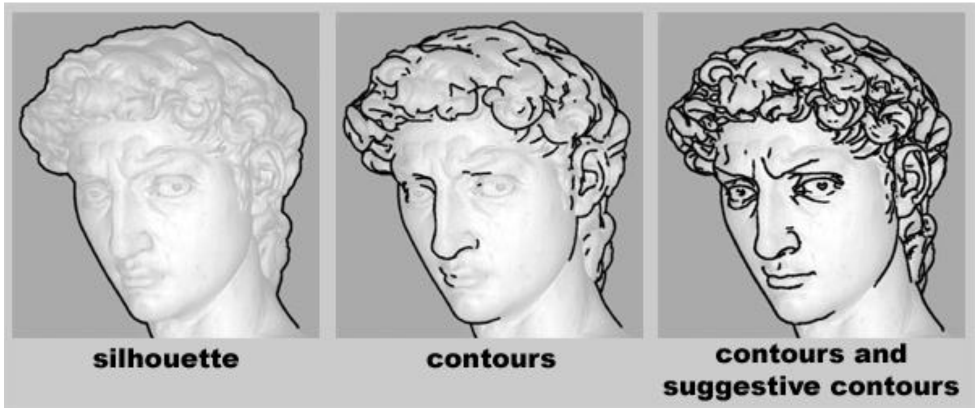
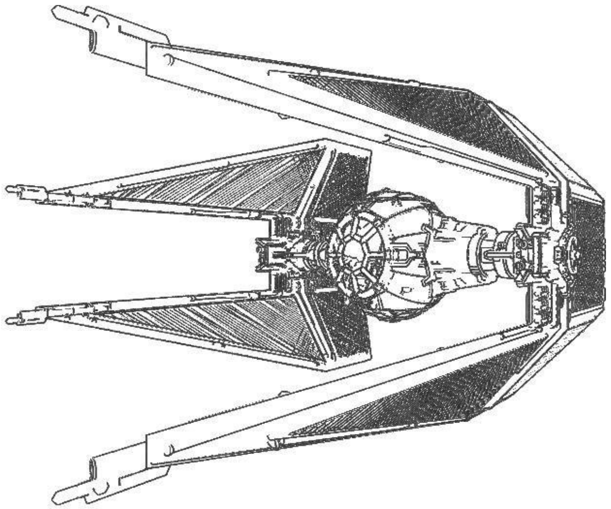
Non-Photorealistic Rendering

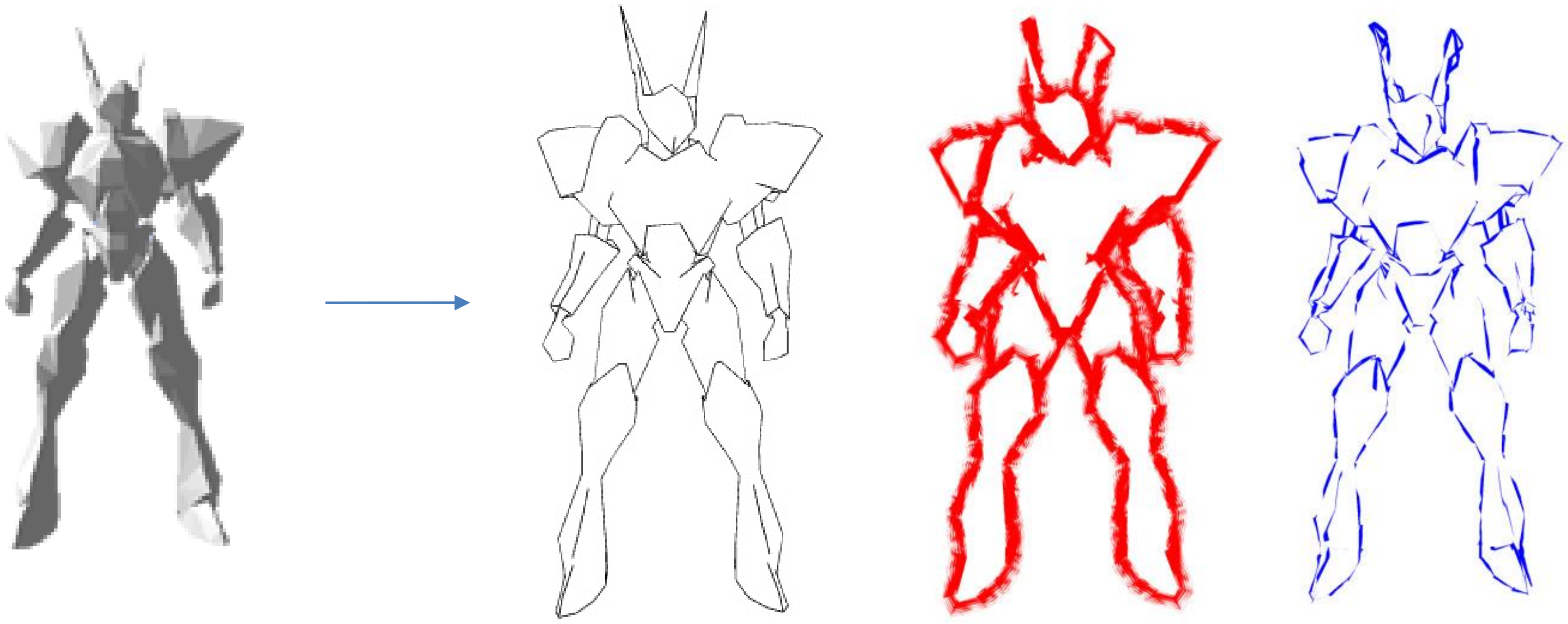
Contour / Silhouette

A. S.

Introduction

- The silhouette is the simplest form of line art and is used in cartoons, technical illustrations, architectural design and medical atlases.
- Given E as the eye vector, a point on a surface (u, v) with surface normal N is a silhouette point if $E \cdot N = 0$.
- Used for realistic rendering and interactive techniques.
- Object space algorithms involve computations in three dimensions and produce a list of silhouette edges or curves for a given viewpoint.



silhouette
contours
contours and suggestive contours


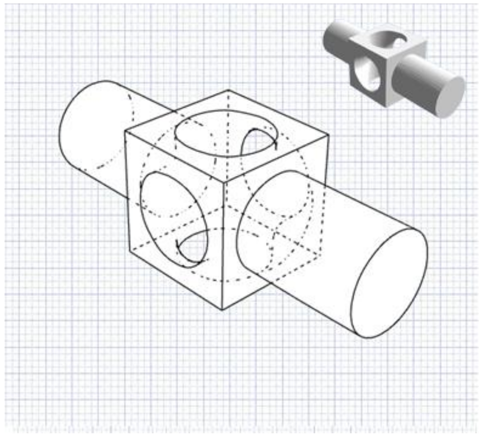


Object – Space Silhouette Algorithms

- Existing Algorithms such as:
  - Brute Force
  - Edge Buffer
  - Probabilistic
  - Gauss Map Arc Hierarchy

Edge Buffer

- Iteration over polygons.
- Create a polygon list.
  - Have a front flag and a back flag.
- Create an edge list.
- Edges sharing exactly one front and back facing polygon



Procedural Modeling of Skyscrapers

G. H.

■How?

Input:

Height/number of

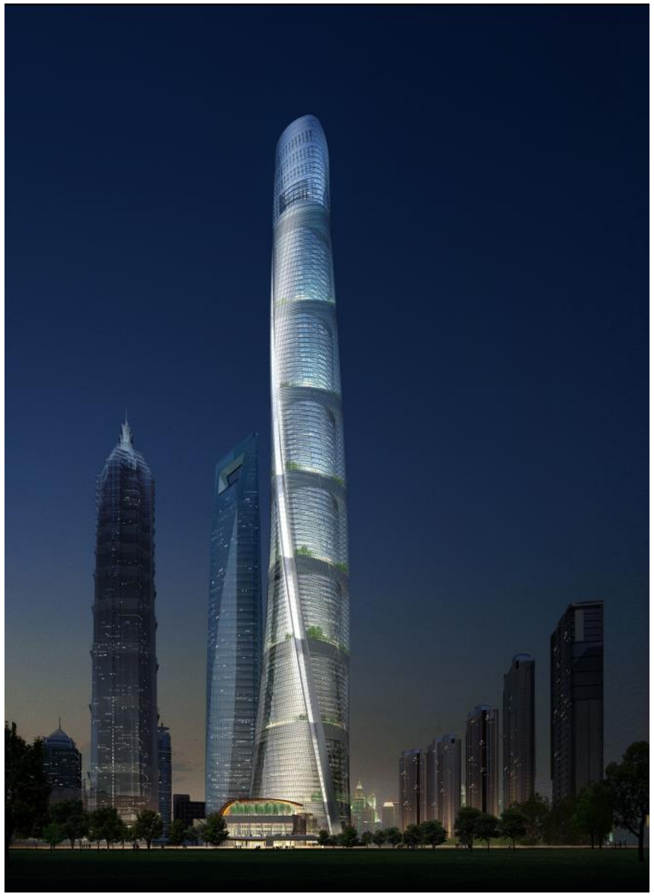
Length and width of

▪Building Generation

Types of Buildings

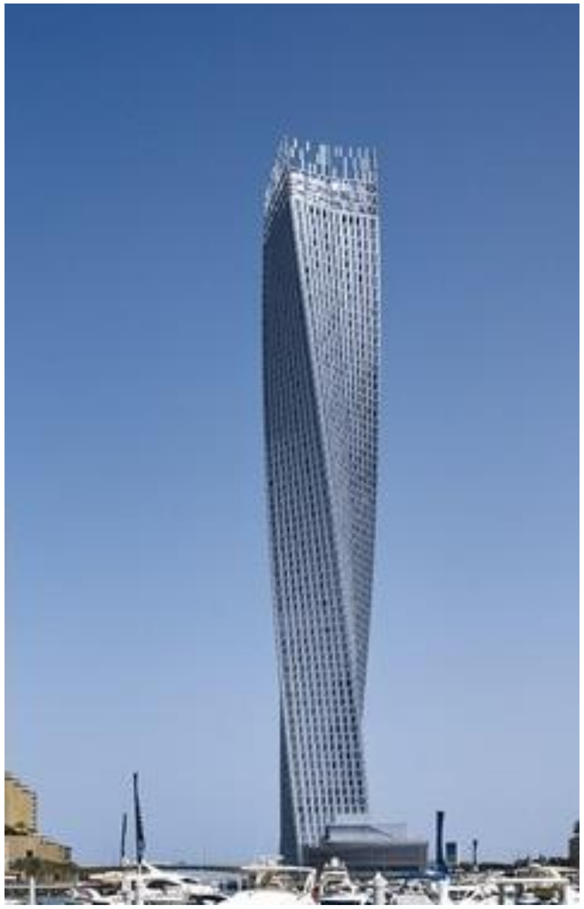
Curved

▪- Rectangular

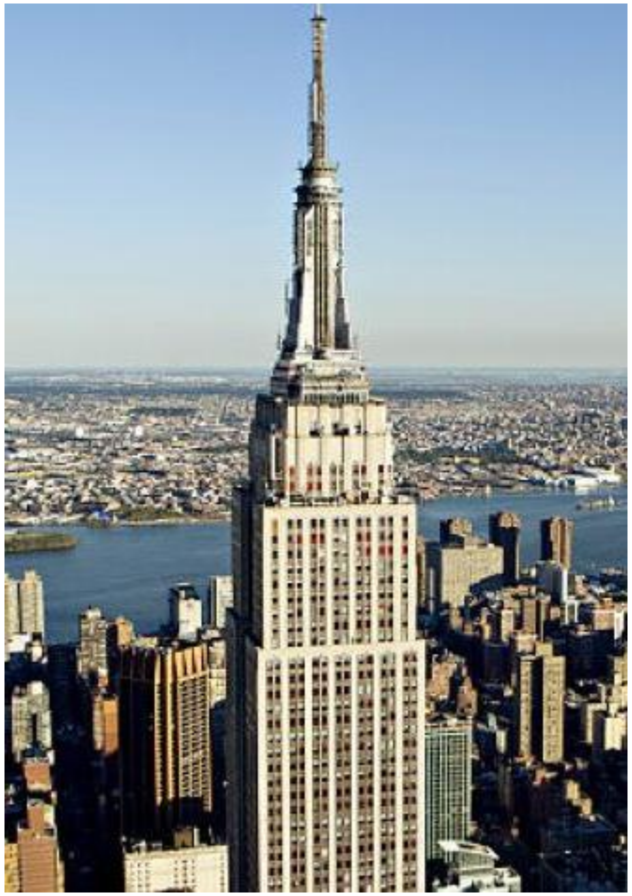


▪Building Variation

Twist



Tiered



■Generating Building Textures

Randomly generate textures

Wi

Colo

Door

Bum

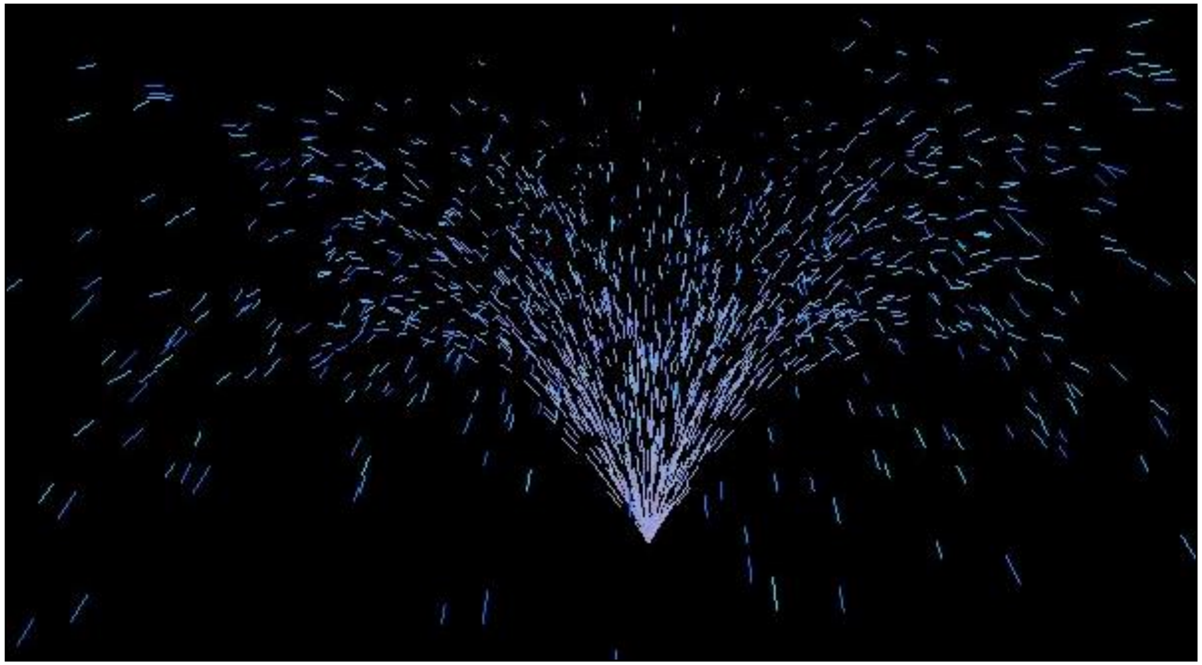
■Demo

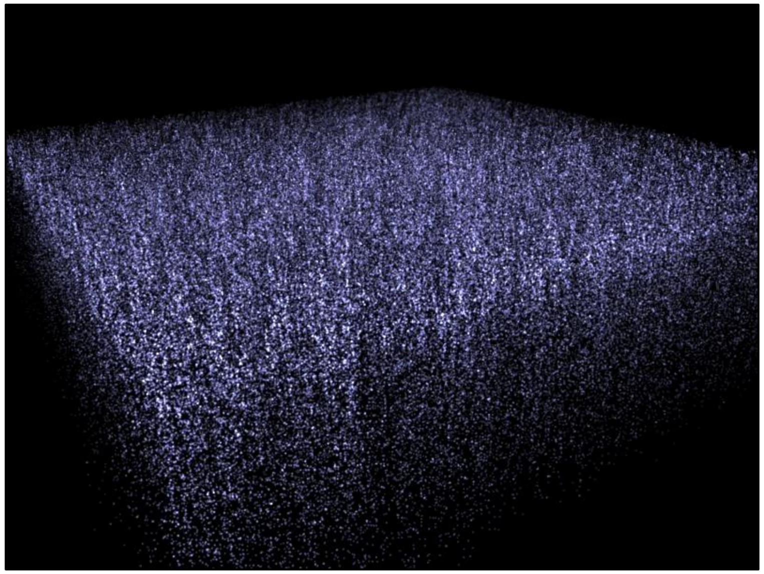
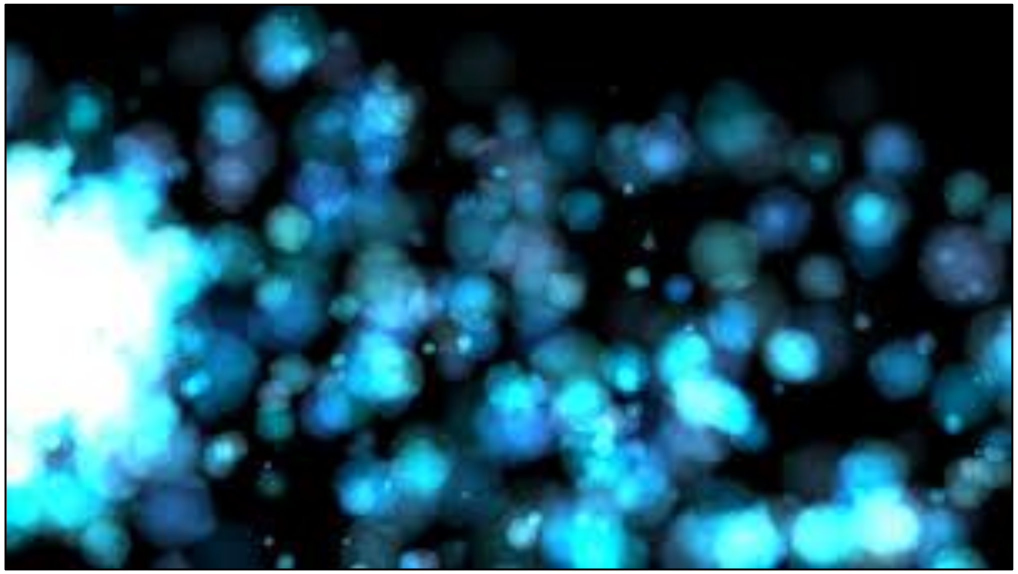
Vary input

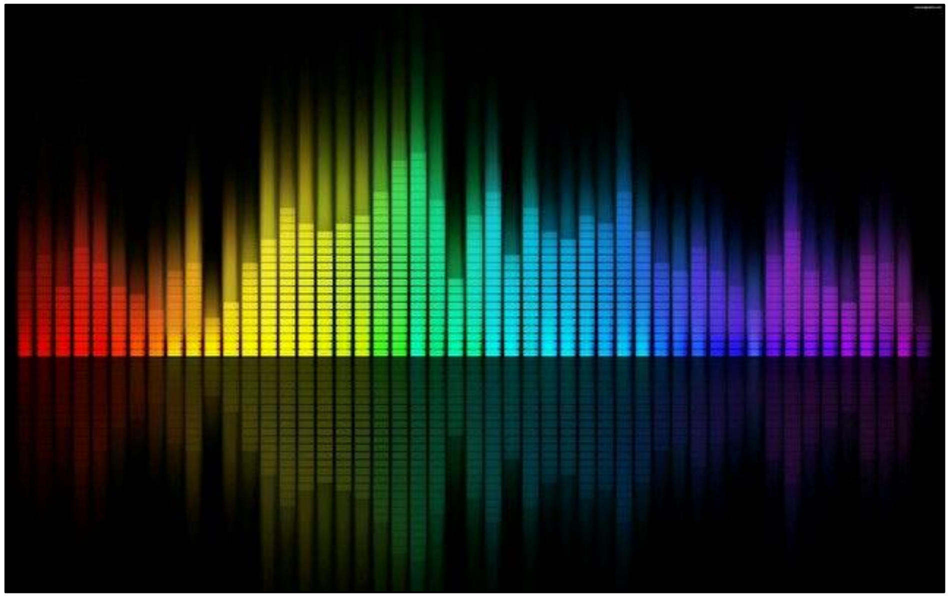
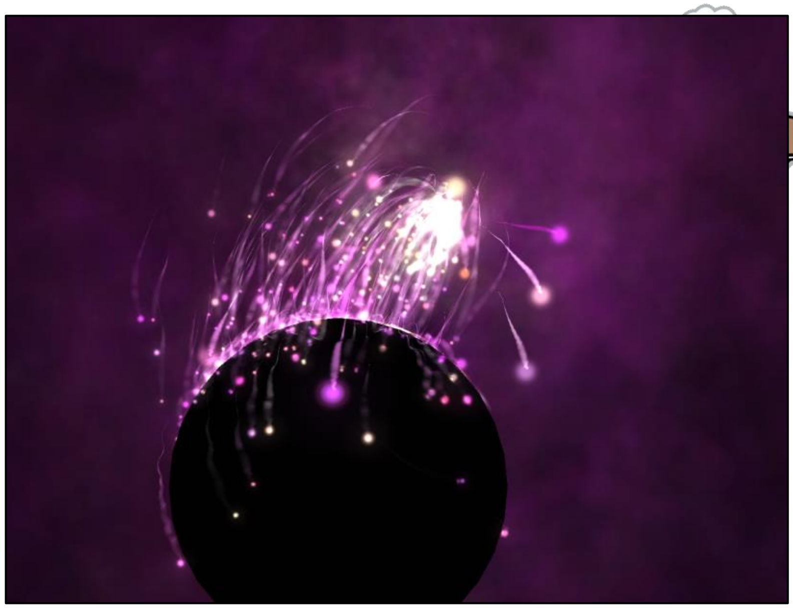
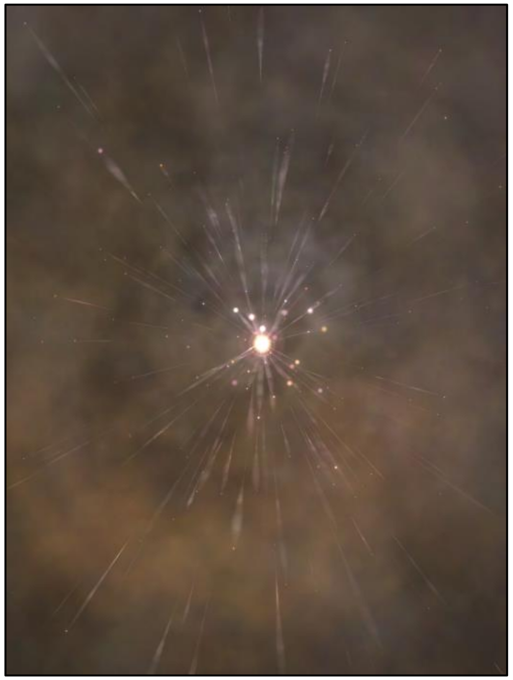
Generate multiple buildings per input

Particle Audio Visualizer

N. S.



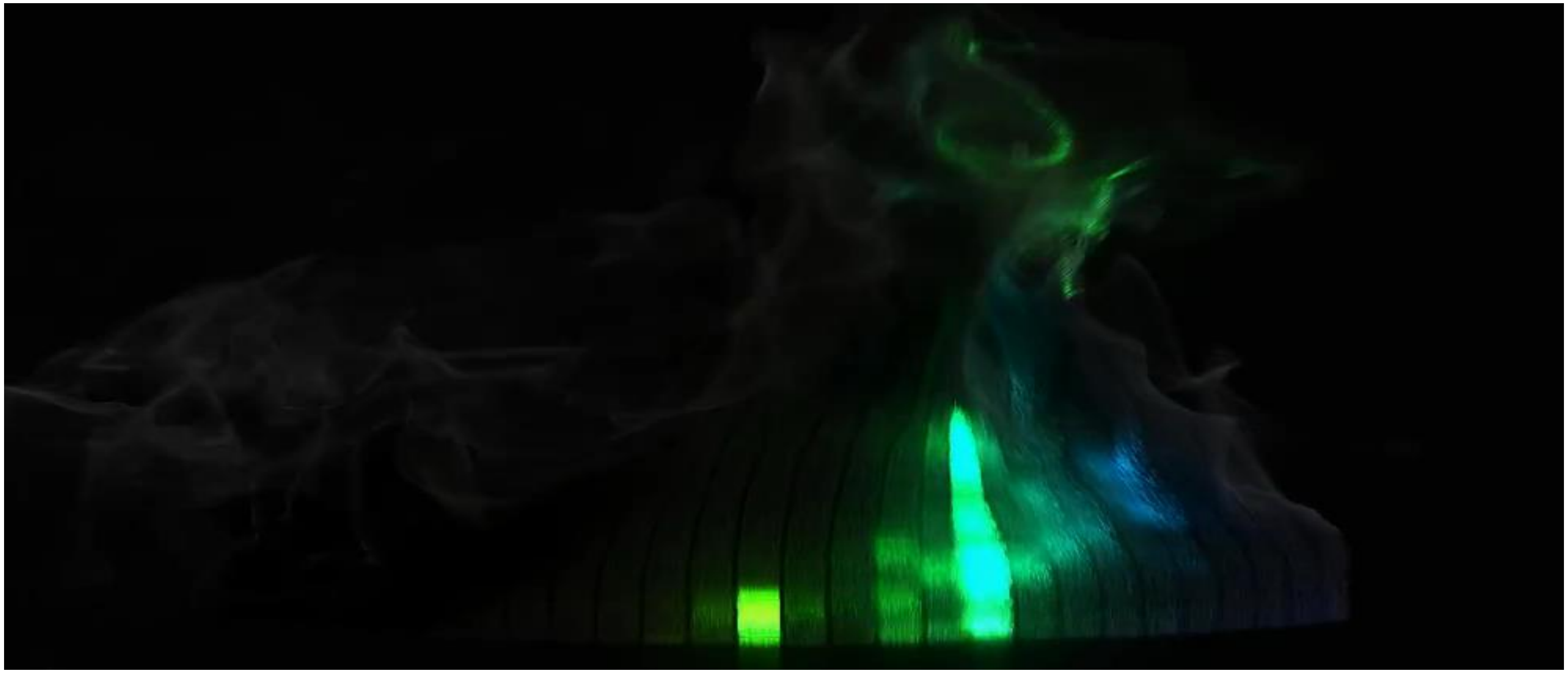
How?

- Web Graphics Library (WebGL)
- Web Audio API
  - Audio loading & playing
  - FFT Analyzer
- System of Particles
- Frequency Bins
- Lighting, Velocity, Fade

Demo

- Live on web page
- Vote on song(s) via piazza

Example



source

PM-Plants

M. J.

What: Modeling Plants 2D/3D

Grammar defined as a tuple:

(V, A, P)

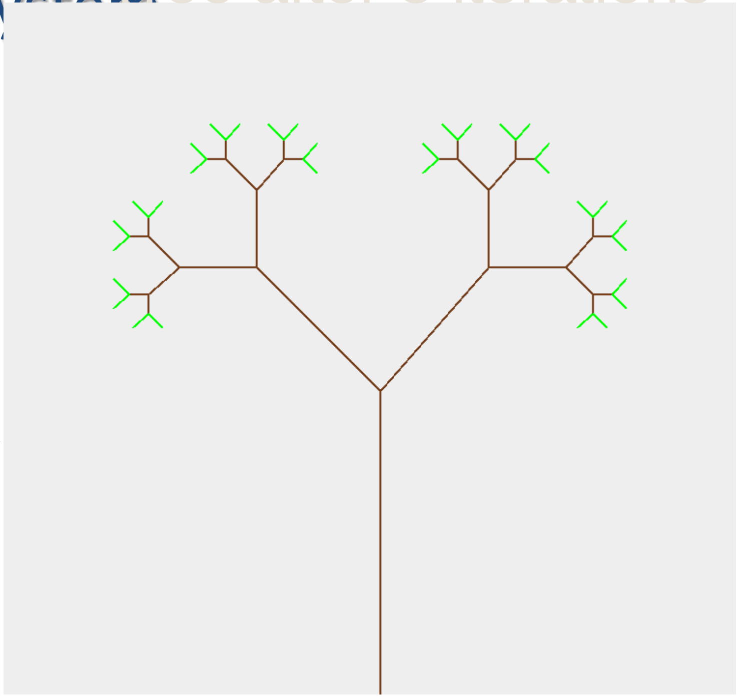
V = variables and constants

A = axiom

P = production rules

Branching L-System

Tree after 5 iterations



<u>Grammar to produce this colored plant:</u>

variables : 0, 1

constants: [,]

axiom : 0

rules : (1 → 11), (0 → 1[0]0)

<u>Drawing Rules:</u>

0 : draw leaf, green color

1 : draw branch, brown color

[: push position to stack and turn left 45 deg ,] : pop position from stack

Iteration 5

1111111111111111[11111111[1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0]1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0]11111111[1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0]1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

Stochastic L-System

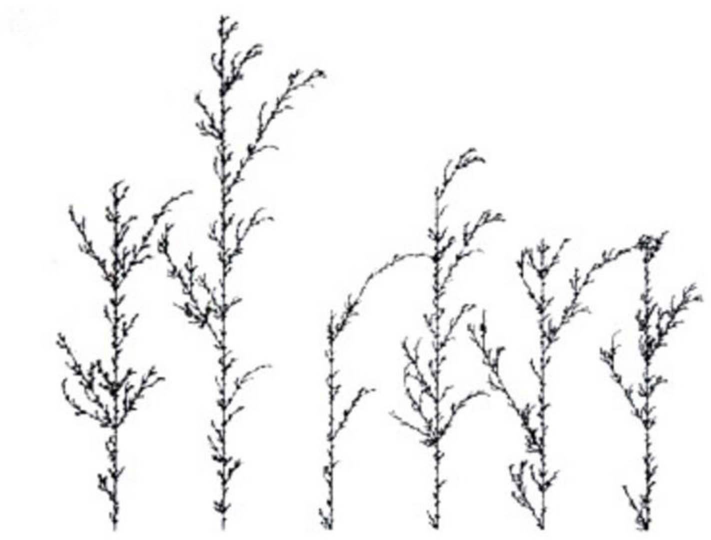
Probability based gener
of leaves and branche

1 -> 11, P = .33

1 -> [1]0 , P = .33

1-> 1[01]0 , P = .33

Many variations to create
complexity



3D L-System

Have grammar that manipulates segments in 3d space. Pitch, Roll, Yaw, increase size, decrease size, change col

iterations=7, initial angle=22.5

axiom: A

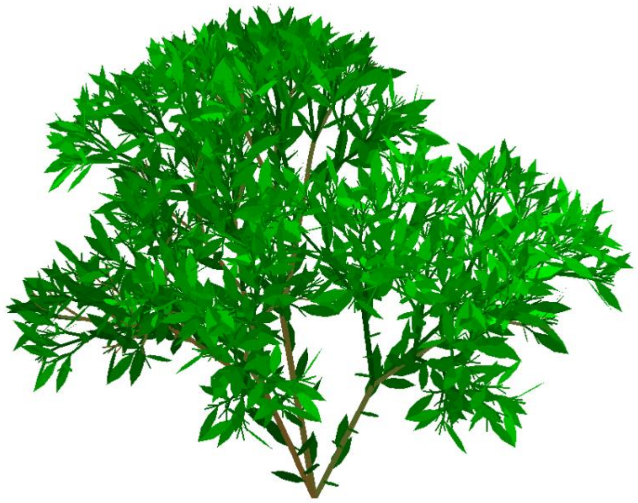
Rules:

A → [&FL!A]/////’[&FL!A]///////’[&FL!A]

F → S ///// F

S → F L

L → [’’’∧∧{-f+f+f-|-f+f+f}]



How and Demo

Parser to understand the grammar

Interpreter compute iterations. Draws segments based on orientation from grammar

OpenGL/C++ program to draw based on produced string

Demo will consist of rendering variety of 3d/2d plants using various L-Systems