Procedural Generation

Concepts and Examples

Operational Definition

Algorithmically generating instances of some kind of content by coming up with very specific answers to:

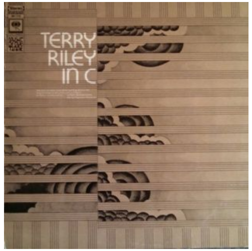
1. **Represent content?**
2. **Generate content?**
3. **‘Render’ content?**

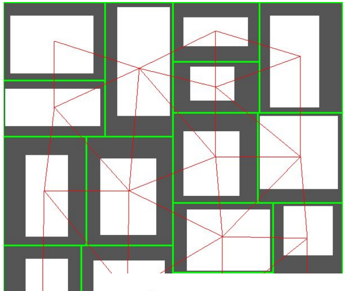
Settlers of Catan Board State

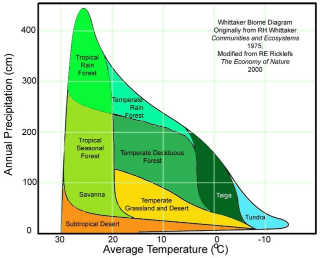
1. *Represent*
   a. Sequence of tiles, numbers
2. *Generate*
   a. Randomly shuffle tiles
   b. Deterministically order numbers
3. *Render*
   a. Place tiles and number sequence down according to consistent pattern.

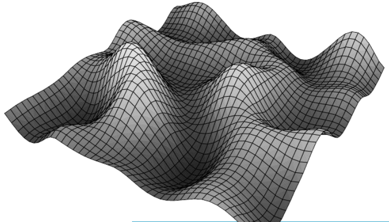
Structure

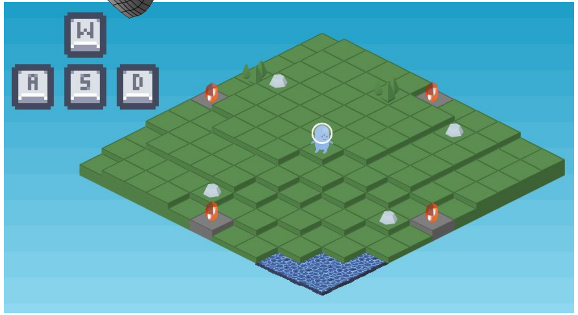
- Examples of ‘Kinds’ of Content
  - Answers to the 3 questions.
  - Kinds
    - 2D Images
    - 3D shapes
    - Audio
    - ‘Levels’
    - Systems
- Demo Implementation
  - Download, run, **modify** an example implementation
  - Level Generation Methods









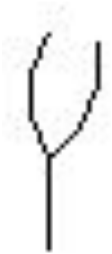


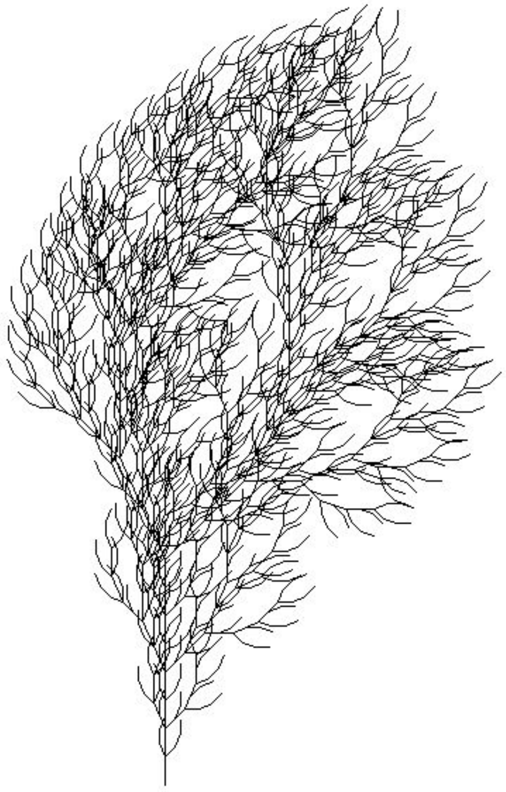
2D/3D Objects

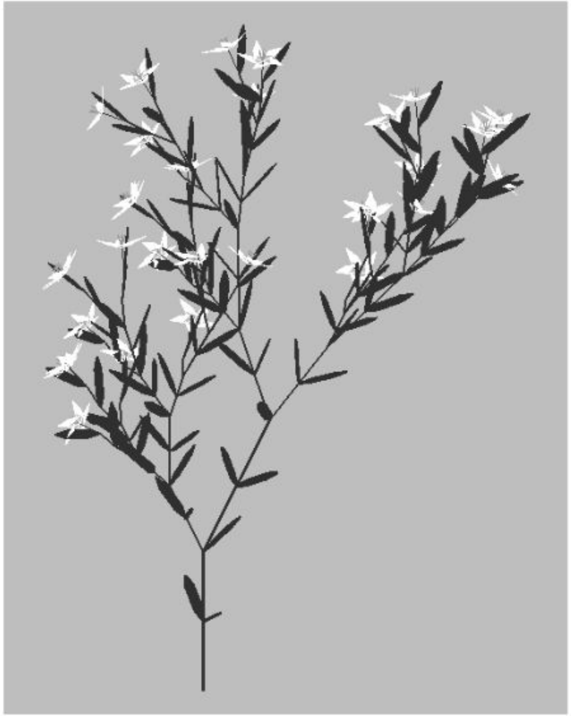
L-Systems

1. *Represent*
   a. Define an ‘Atom’
   b. Define a Replacement Rule
2. *Generate*
   a. Apply N-many steps of the replacement rule
   b. Could apply multiple, stochastically.
3. *Render*
   a. Interpret the Atom symbols as commands for a drawing method.
   b. Could mean interpreting “control symbols”
      i. Brackets - “start position stack”
      ii. Angles/Parameters

```
F -> FF-[-F+F+F]+[+F-F-F]

  F    = “move pen forward”

-   +  = Change angle of movement

[   ]  = Push/Pop position of pen
```



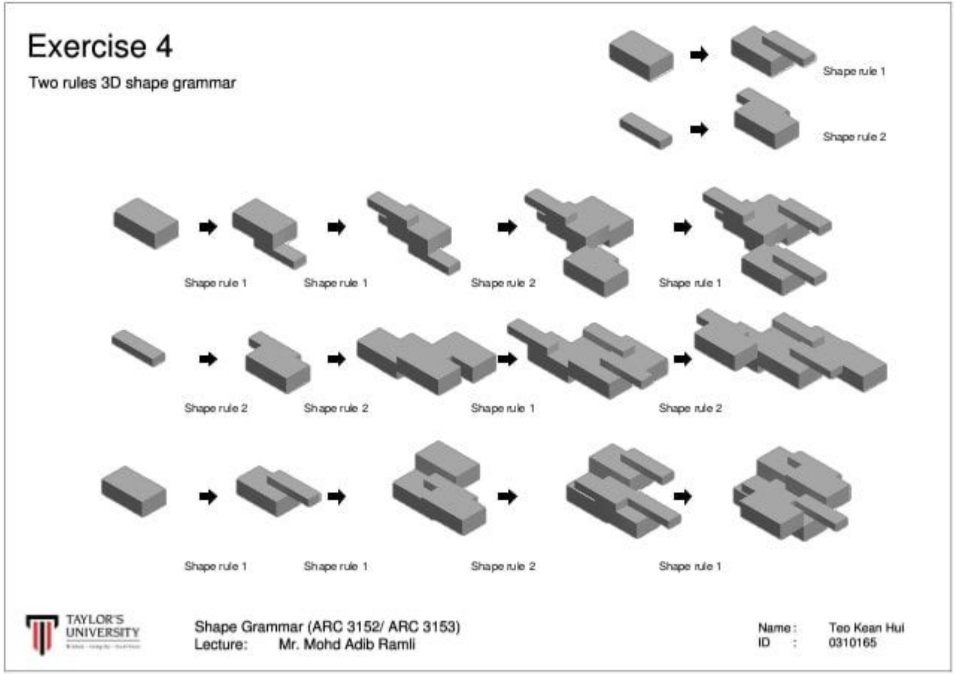




$$\begin{aligned}
\omega :&\quad \textbf{plant}\\
p_1 :&\quad \textbf{plant} \rightarrow \textbf{stem} + [\,\textbf{plant} + \textbf{flower}]\, -- \,//\\
&\qquad\qquad\qquad [\,--\,\textbf{leaf}\,]\;\textbf{internode}\,[\,+\,+\,\textbf{leaf}\,]\,-\\
&\qquad\qquad\qquad [\,\textbf{plant flower}\,]\,+\,+\,\textbf{plant flower}\\
p_2 :&\quad \textbf{internode} \rightarrow F\;\textbf{seg}\,[//\,\&\,\&\;\textbf{leaf}\,]\,[//\,\wedge\wedge\;\textbf{leaf}\,]\,F\;\textbf{seg}\\
p_3 :&\quad \textbf{seg} \rightarrow \textbf{seg}\;F\;\textbf{seg}\\
p_4 :&\quad \textbf{leaf} \rightarrow [\text{'}\,\{\,+f - ff - f + \,|\, +f - ff - f\,\}\,]\\
p_5 :&\quad \textbf{flower} \rightarrow [\,\&\,\&\,\&\;\textbf{pedicel}\;\text{`}\;/\;\textbf{wedge}\;////\;\textbf{wedge}\;////\\
&\qquad\qquad\qquad \textbf{wedge}\;////\;\textbf{wedge}\;////\;\textbf{wedge}\,]\\
p_6 :&\quad \textbf{pedicel} \rightarrow FF\\
p_7 :&\quad \textbf{wedge} \rightarrow [\text{`}\,\wedge\,F\,]\,[\,\{\,\&\,\&\,\&\,\& -f + f \,|\, -f + f\,\}\,]
\end{aligned}$$

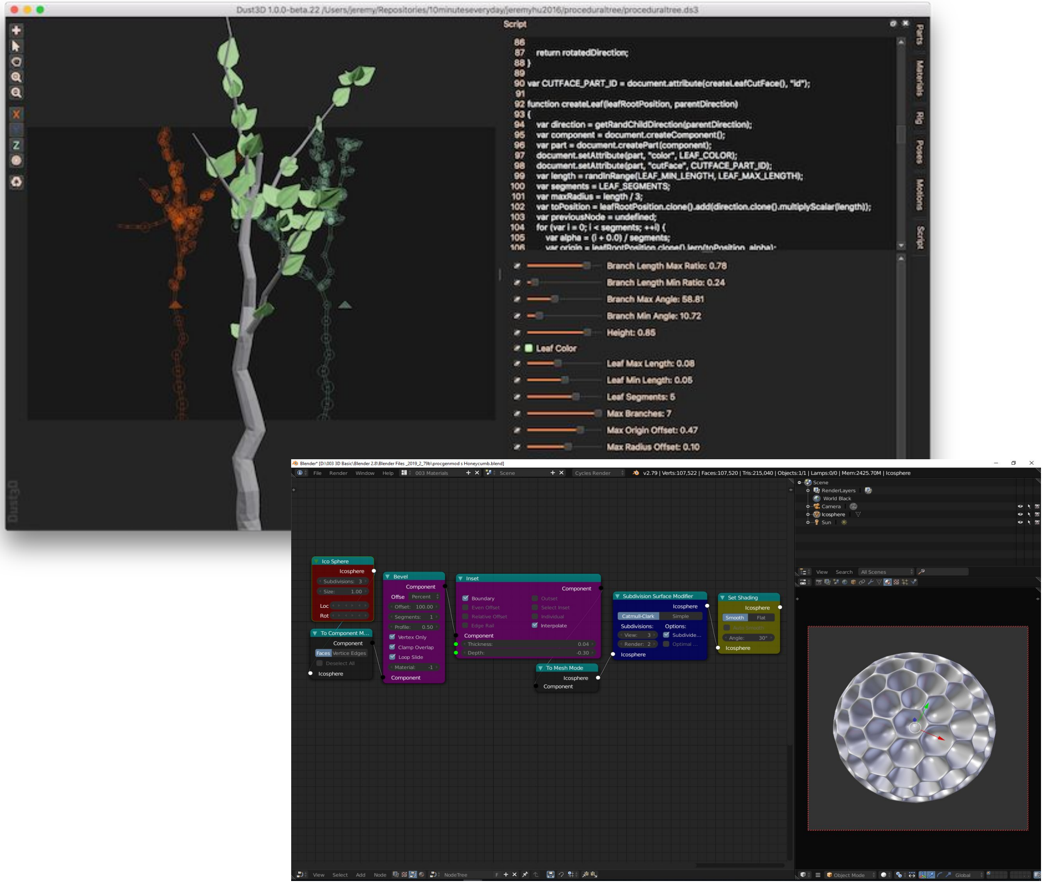
Shape Grammars

1. *Represent*
   a. Sets of Structures Representing Shapes
   b. Along with attributes, describing a pos
2. *Generate*
   a. Apply replacement rules from the shape grammar
3. *Render*
   a. Interpret the structures and attributes to render a full 3D Object.



Procedural Modelling

1. *Represent*
    a. Programmatically organized sets of instructions
2. *Generate*
    a. Stochastically or deterministically follow parametric instructions
3. *Render*
    a. Allow the engine/framework to do its magic.



Dust3D 1.0.0-beta.22 /Users/jeremy/Repositories/10minuteseveryday/jeremyhu2016/proceduraltree/proceduraltree.ds3

Script

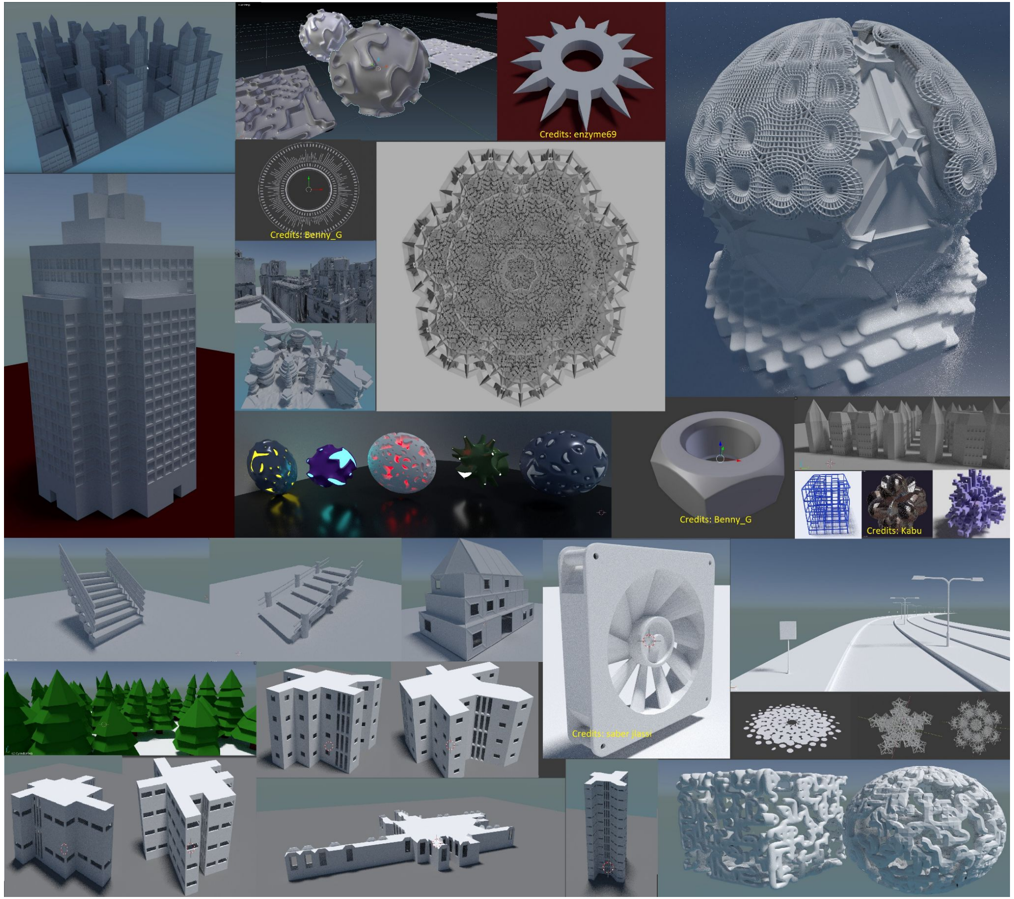
```
    return rotatedDirection;
}

var CUTFACE_PART_ID = document.attribute(createLeafCutFace(), "id");

function createLeaf(leafRootPosition, parentDirection)
{
    var direction = getRandChildDirection(parentDirection);
    var component = document.createComponent();
    var part = document.createPart(component);
    document.setAttribute(part, "color", LEAF_COLOR);
    document.setAttribute(part, "cutFace", CUTFACE_PART_ID);
    var length = randInRange(LEAF_MIN_LENGTH, LEAF_MAX_LENGTH);
    var segments = LEAF_SEGMENTS;
    var maxRadius = length / 3;
    var toPosition = leafRootPosition.clone().add(direction.clone().multiplyScalar(length));
    var previousNode = undefined;
    for (var i = 0; i < segments; ++i) {
        var alpha = (i + 0.0) / segments;
        var origin = leafRootPosition.clone().lerp(toPosition, alpha);
```

Branch Length Max Ratio: 0.78
Branch Length Min Ratio: 0.24
Branch Max Angle: 58.81
Branch Min Angle: 10.72
Height: 0.85
🟩 Leaf Color
Leaf Max Length: 0.08
Leaf Min Length: 0.05
Leaf Segments: 5
Max Branches: 7
Max Origin Offset: 0.47
Max Radius Offset: 0.10

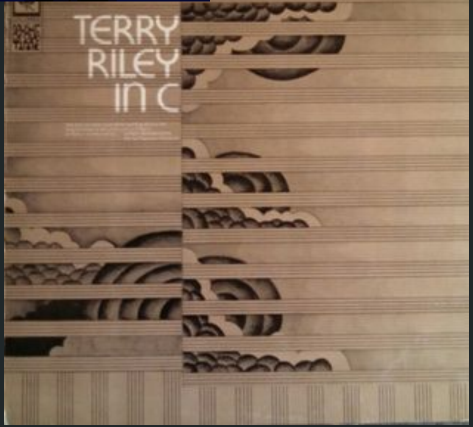
v2.79 | Verts:107,522 | Faces:107,520 | Tris:215,040 | Objects:1/1 | Lamps:0/0 | Mem:2425.70M | Icosphere

Ico Sphere: Subdivisions: 3, Size: 1.00
To Component M...: Faces
Bevel: Offse Percent, Offset: 100.00, Segments: 1, Profile: 0.50, ☑ Vertex Only, ☑ Clamp Overlap, ☑ Loop Slide, Material: -1
Inset: ☑ Boundary, ☐ Even Offset, ☐ Relative Offset, ☐ Edge Rail, ☐ Outset, ☐ Select Inset, ☐ Individual, ☑ Interpolate, Thickness: 0.04, Depth: -0.30
Subdivision Surface Modifier: Catmull-Clark, Subdivisions: View: 3, Render: 2, Options: ☑ Subdivide..., ☐ Optimal ...
Set Shading: Smooth, ☐ Auto Smooth, Angle: 30°

“Artist In a Box”



Audio



Musical Phrases

1. *Represent*
   a. Sets of phrases, voices
2. *Generate*
   a. Choose duration, other parameters
3. *Render*
   a. Play!
   b. Can make choices based on player actions.

iMUSE SYSTEM

- Lucas Arts action game music system
- Composed phrases based on elements of current scene

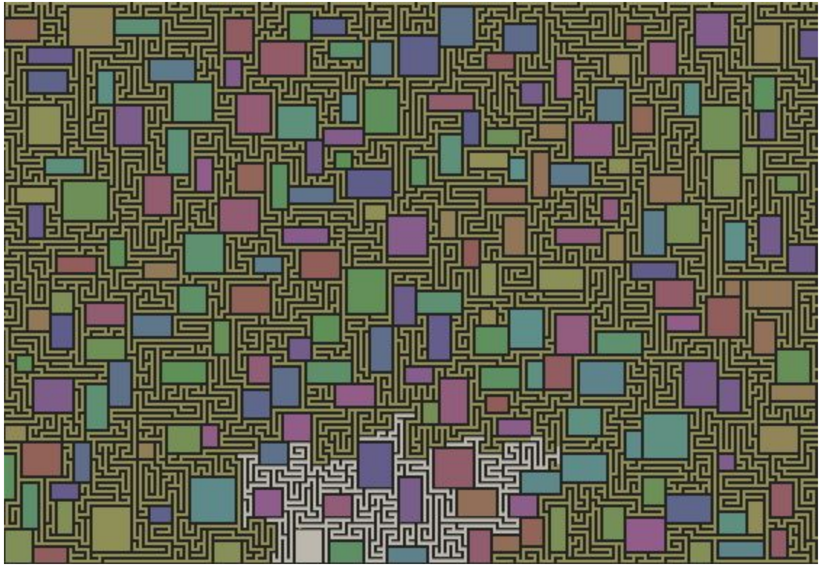
“In C, Terry Riley”

- Procedure for an arbitrary large set of artists to generate a composition in real time.
- 53 Phrases
  - Half a beat to 32 beats
  - Numbered
- Each artist controls;
  - Which phrase
  - Repeats
- Encouraged to stay within some number of phrases of one another

Levels

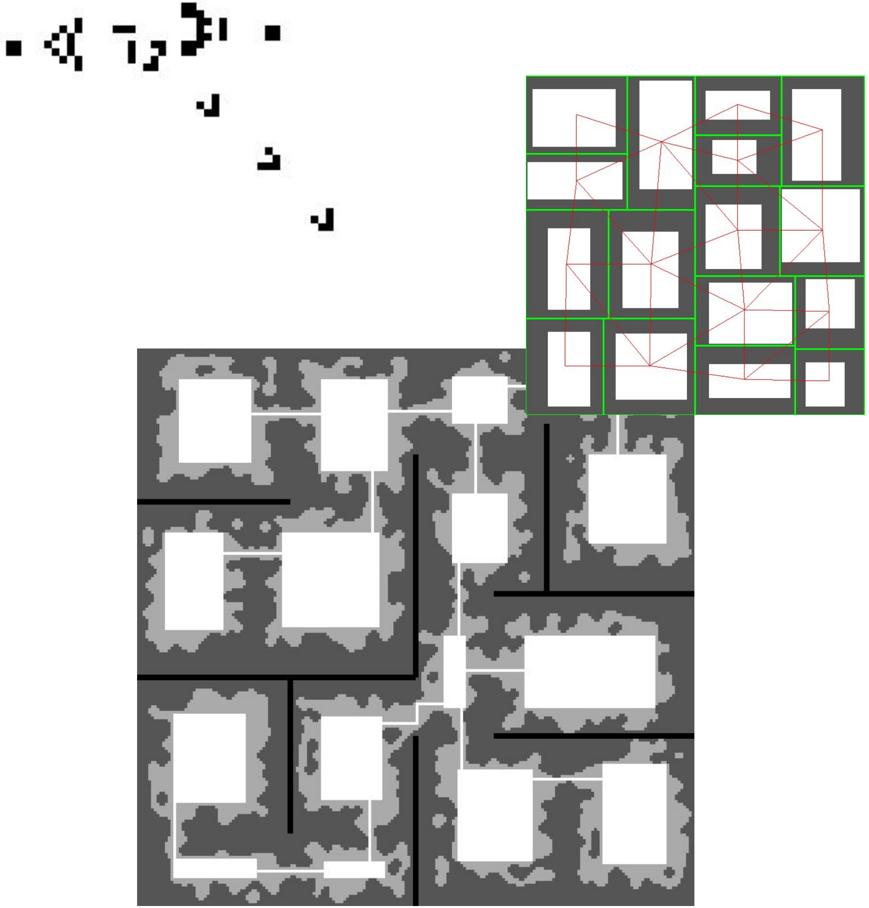
Rooms and Mazes

1. *Represent*
   a. Grid of cell values indicating status as room/walkway.
2. *Generate*
   a. Place Rectangular Rooms
   b. Flood Fill a Maze
3. *Render*
   a. Interpret Cell values as tiles.



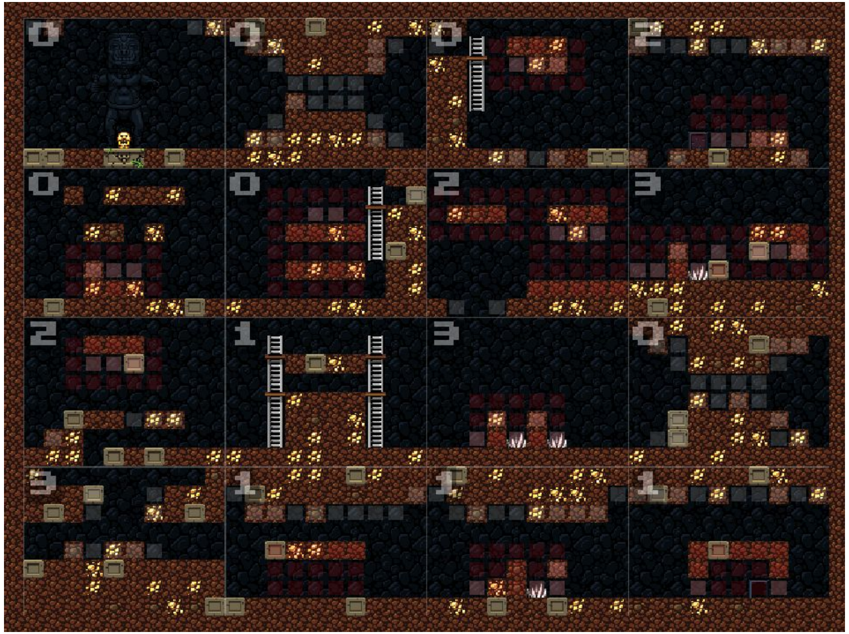
Caves - BSP+CA

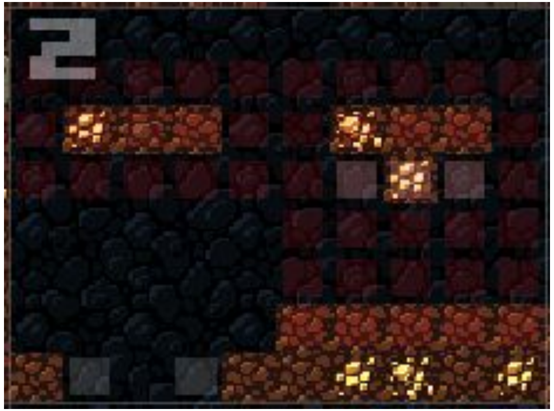
1. *Represent*
   a. Grid of cell values
   b. Choice of CA rules
2. *Generate*
   a. Binary Space Partitioning for ‘rough outline’
   b. Cellular Automata rules to create organic detail
3. *Render*
   a. Interpret cell values as tiles.



‘Tile’ Maps - Spelunky, Diablo

1. *Represent*
   a. Coarse grid of template cells/tiles
   b. Adjacency and Path Rules
2. *Generate*
   a. Fill in coarse grid, while following adj. rules, and pathing goals.
3. *Render*
   a. Fill in the random elements of the templates, and generate graphics.

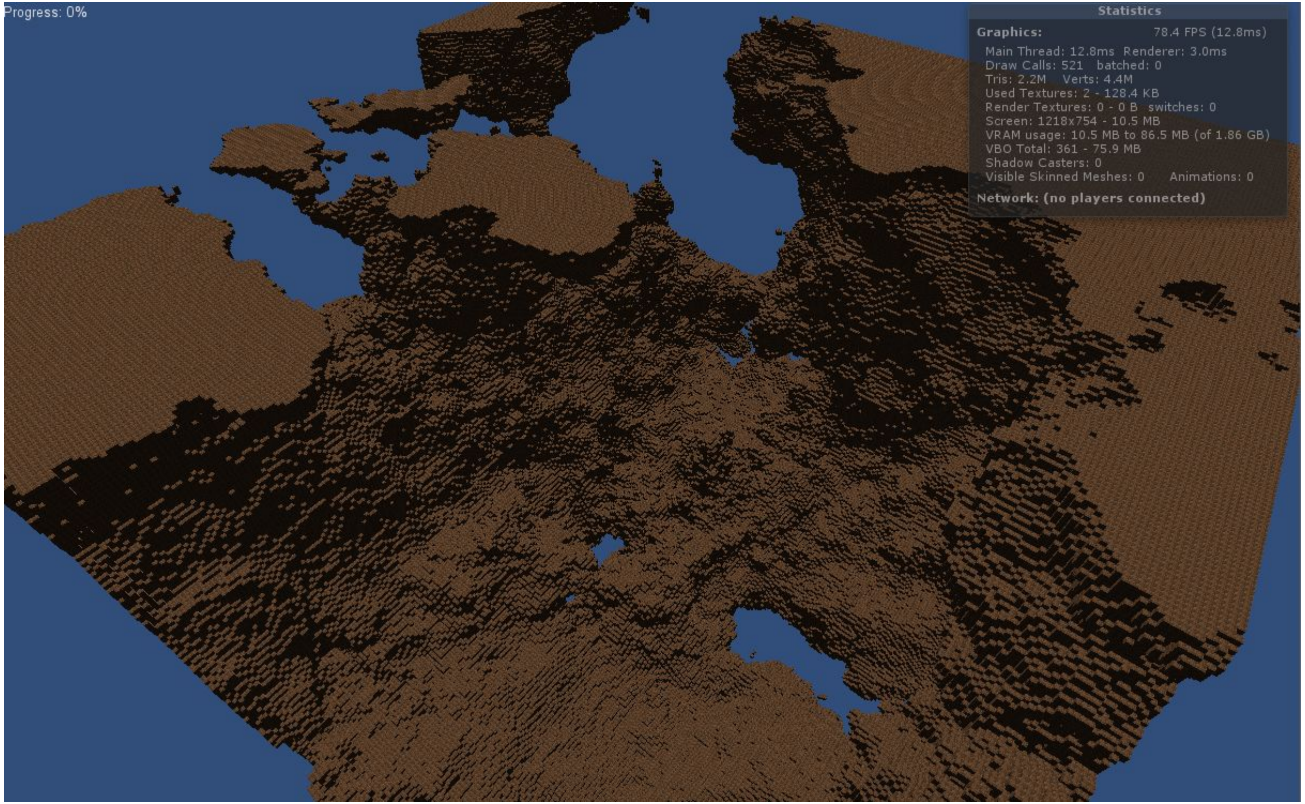




Smooth Surfaces

1. *Represent*
    a. Associate points within a game system as points on a smooth random surface
2. *Generate*
    a. Choose scale, octaves, offsets.
    b. Perlin, Simplex Noise
3. *Render*
    a. Interpret value as height, density, abiotic value.

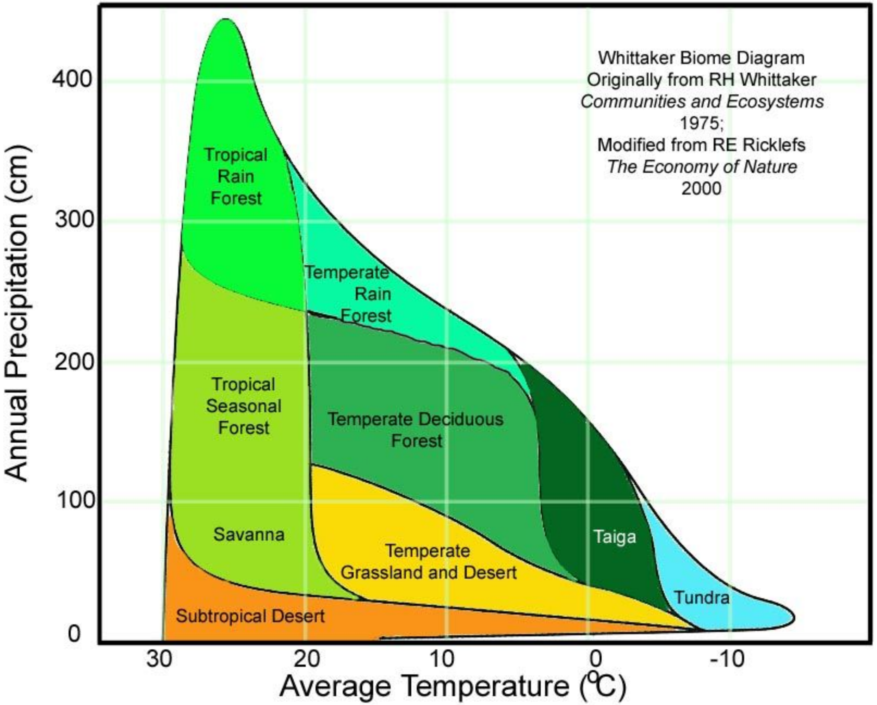
Smooth Surfaces - 3D as Density



```
Progress: 0%
```

```
Statistics
Graphics:                    78.4 FPS (12.8ms)
  Main Thread: 12.8ms  Renderer: 3.0ms
  Draw Calls: 521   batched: 0
  Tris: 2.2M    Verts: 4.4M
  Used Textures: 2 - 128.4 KB
  Render Textures: 0 - 0 B  switches: 0
  Screen: 1218x754 - 10.5 MB
  VRAM usage: 10.5 MB to 86.5 MB (of 1.86 GB)
  VBO Total: 361 - 75.9 MB
  Shadow Casters: 0
  Visible Skinned Meshes: 0      Animations: 0
Network: (no players connected)
```

Smooth Surfaces - 2D as Abiotic



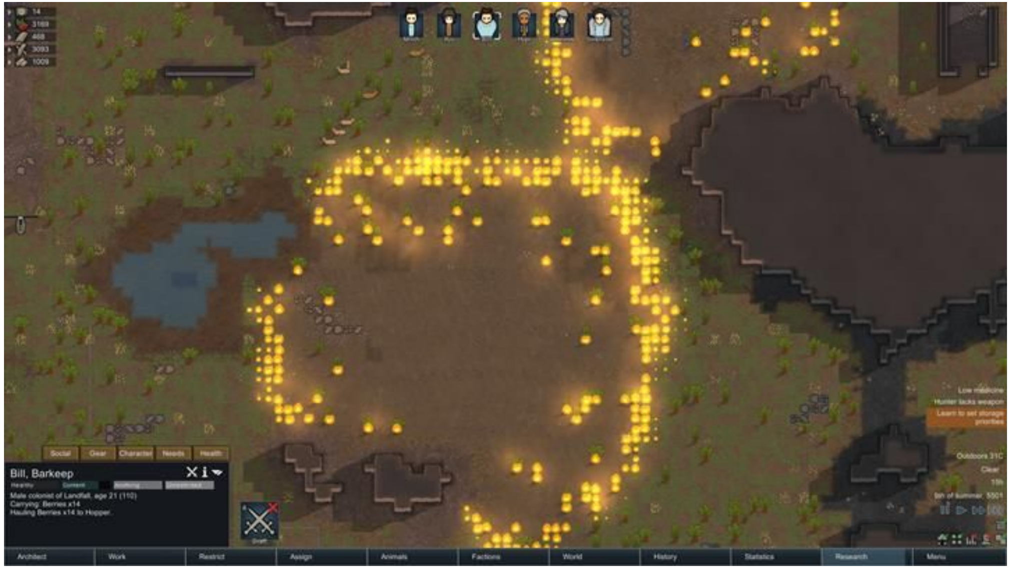
Systems

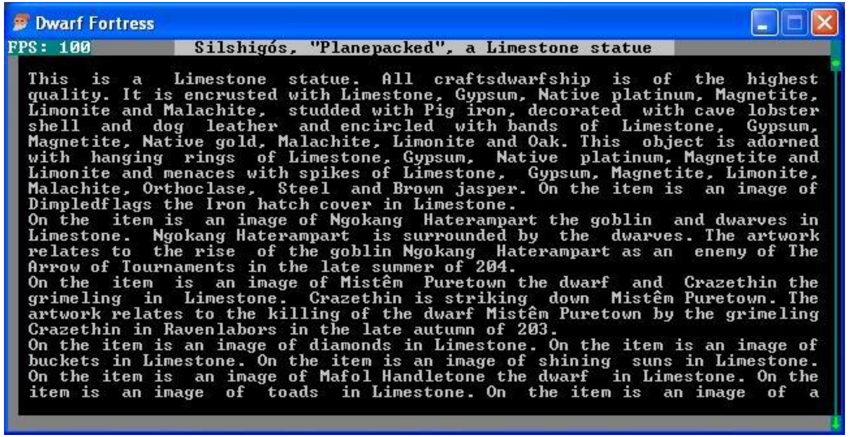
PAUSED
Dwarf Fortress
*PAU
a: View Announcements
b: Site a Building
c: View Civilizations
d: Designations
u: Unit List j: Job Li
k: Look Around
m: Military x: Squads
o: Set Orders and Options
p: Stockpiles i: Zones
q: Set Building Tasks/Pre
r: View Rooms/Buildings
t: View Items in Building
v: View Units h: Hot Ke
l: Artifacts
n: Nobles and Administrat
z: Status V: Visual
Tab: Move this menu/map
?: Help Escape: Option
;: Movies D: Depot Acces
Space: Resume .: One-Step
hide.

RIMWORLD

‘Story Generation’

1. *Represent*
   a. Define collection of interacting systems, simulating elements of the world
2. *Generate*
   a. Random events
   b. Guided Story tellers
3. *Render*
   a. Allow the situation to play out.
   b. Fun!!

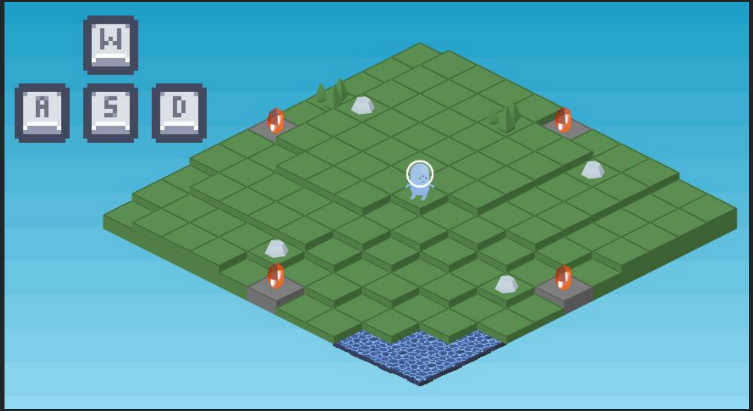


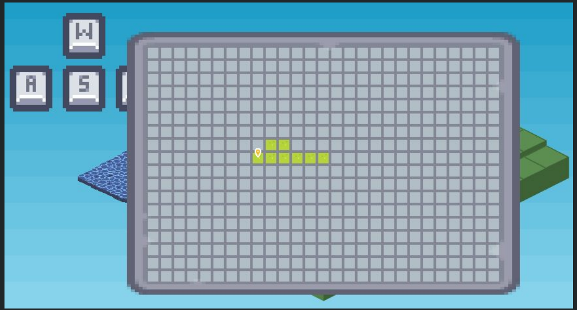


```
FPS: 100        Silshigós, "Planepacked", a Limestone statue

This is a Limestone statue. All craftsdwarfship is of the highest
quality. It is encrusted with Limestone, Gypsum, Native platinum, Magnetite,
Limonite and Malachite, studded with Pig iron, decorated with cave lobster
shell and dog leather and encircled with bands of Limestone, Gypsum,
Magnetite, Native gold, Malachite, Limonite and Oak. This object is adorned
with hanging rings of Limestone, Gypsum, Native platinum, Magnetite and
Limonite and menaces with spikes of Limestone, Gypsum, Magnetite, Limonite,
Malachite, Orthoclase, Steel and Brown jasper. On the item is an image of
Dimpledflags the Iron hatch cover in Limestone.
On the item is an image of Ngokang Haterampart the goblin and dwarves in
Limestone. Ngokang Haterampart is surrounded by the dwarves. The artwork
relates to the rise of the goblin Ngokang Haterampart as an enemy of The
Arrow of Tournaments in the late summer of 204.
On the item is an image of Mistêm Puretown the dwarf and Crazethin the
grimeling in Limestone. Crazethin is striking down Mistêm Puretown. The
artwork relates to the killing of the dwarf Mistêm Puretown by the grimeling
Crazethin in Ravenlabors in the late autumn of 203.
On the item is an image of diamonds in Limestone. On the item is an image of
buckets in Limestone. On the item is an image of shining suns in Limestone.
On the item is an image of Mafol Handletone the dwarf in Limestone. On the
item is an image of toads in Limestone. On the item is an image of a
```

Demo

1. Dev Overview
2. Quick Code Structure
3. Level Generation





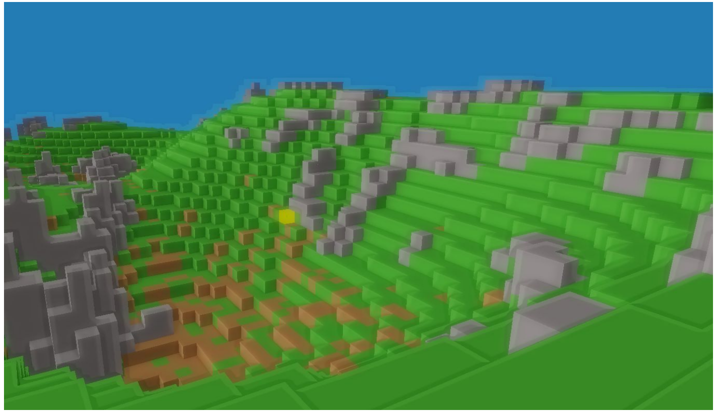
JS Dev Environment in 30 Seconds

- *How are you managing dependencies?*
  - npm
- *How are you ‘modularizing’?*
  - ES6-ish
- *How do you turning multiple files/modules into a single ‘compiled’ (bundled) file?*
  - webpack
  - babel
- *How are you serving the project, locally?*
  - webpack-dev-server
  - with a ‘hand-made’ static directory/index.html

Setup

- github.com/wpower12/procgengame
- npm install
- webpack serve

Overview

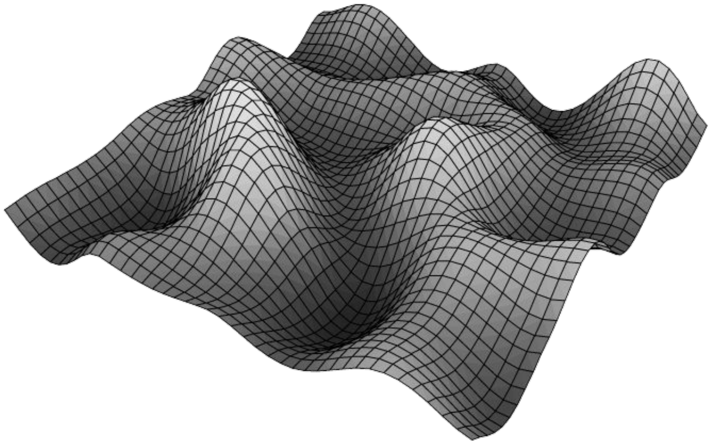


- Phaser3
  - Scenes
    - Lifecycle methods to help organize
      - Loading of assets
      - Updating game state
      - Scene transitions
  - Isometric Plugin
  - Animations, Tweening, Events, Input
- Noise Libraries
  - simplex-noise
  - seedrandom
- KenneyNL Assets

Level Generation

- Representation
- Chunkify World
- Height
  - Simplex Noise Surface
- Doo-Dads
  - Consistently Seeded PRNG
  - “Hash” chunk location
- Portal Placement

```
levelcell = {
    number: height,
    string: tile,
    string: top,
    bool:   portal,
    string: facing
}
```



Improvements/Projects/Goals

- Rare Spawns
- “Biomes”
- Sockets.io
  - Multiplayer?
  - Persistence?
- “Wildlife”/NPCs