Procedural Contents Generation

History and techniques used in the modern video-game industry.

05.2015 / Davide Aversa

Introduction

PCG: Definition and Basic Concepts

What is PCG?

“PCG is a family of techniques, algorithms and procedures used for generating contents in an automatic way rather than manually.”

What is PCG?

PCG IS NOT RANDOM

What is PCG?

PCG IS A WAY TO OVERCOME THE LIMITED HUMAN IMMAGINATION

Why PCG?

- Help Designer at Design Phase.
- Generate Contents at Runtime

History of PCG in games

- **1952**: One of Alan Turing’s final projects was a computer-based, automated love-letter generator, which some have identified as the first known work of new media art. It was programmed by Christopher Strachey in 1952 for the Manchester Mark I computer.

Example:

Darling Sweetheart

You are my avid fellow feeling. My affection curiously clings to your passionate wish. My liking yearns for your heart. You are my wistful sympathy: my tender liking.

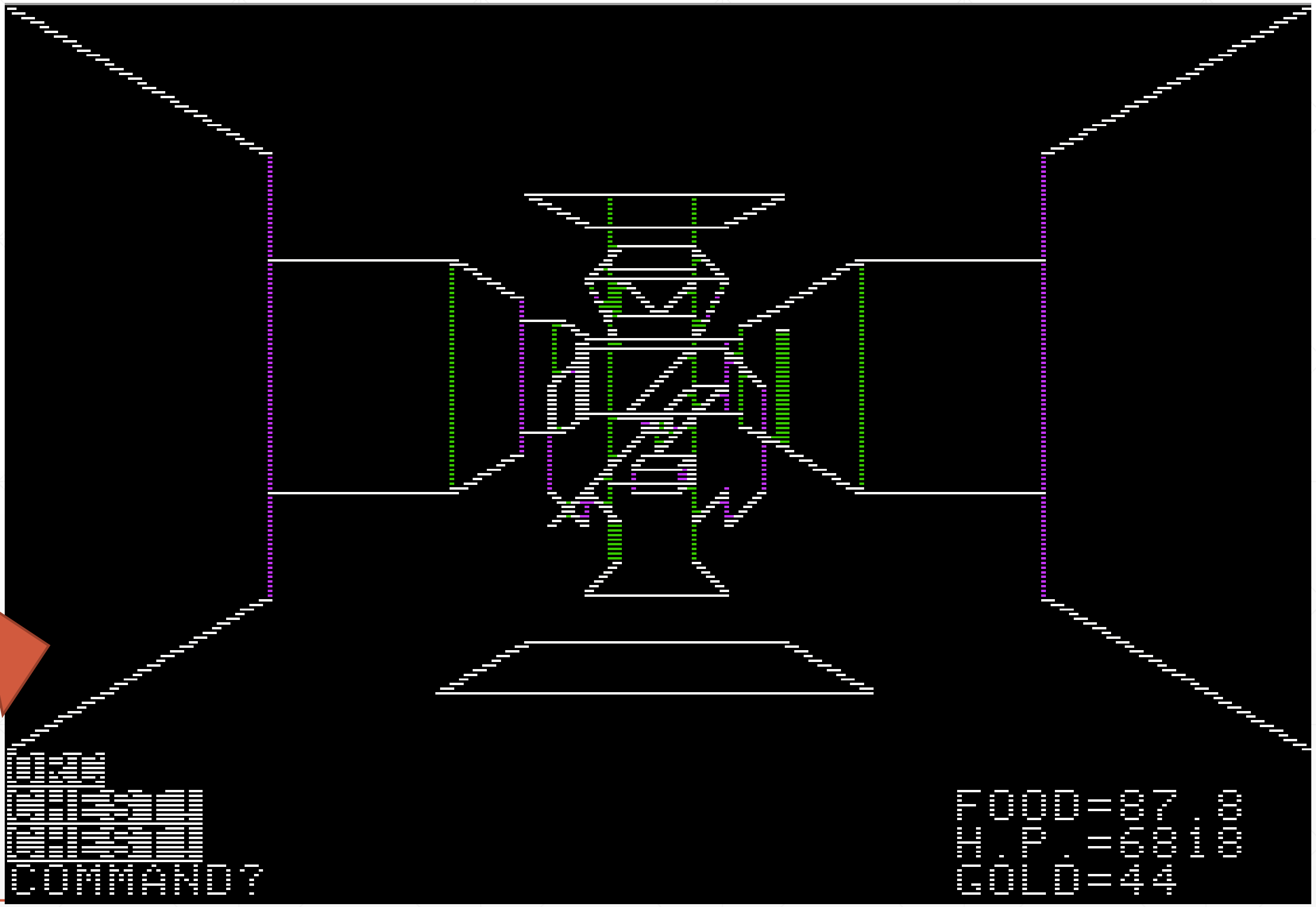
Yours beautifully

History of PCG in games

PCG was born as a way to **compress data**. There was literally no enough space to store pre-made artworks and data.

- **1980**: Richard Garriott's **Akalabeth** is the (maybe) first game to use a seed to generate the game world.

SO COOL!

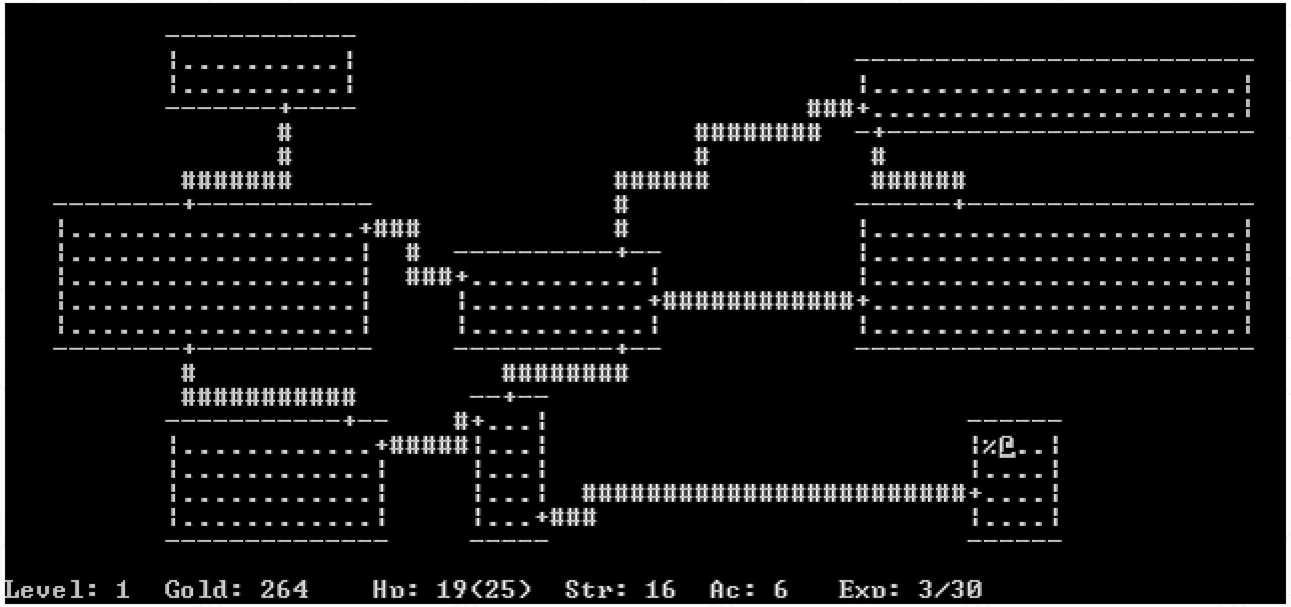


```
COMMAND?
```

```
FOOD=87.8
H.P.=6818
GOLD=44
```

History of PCG in games

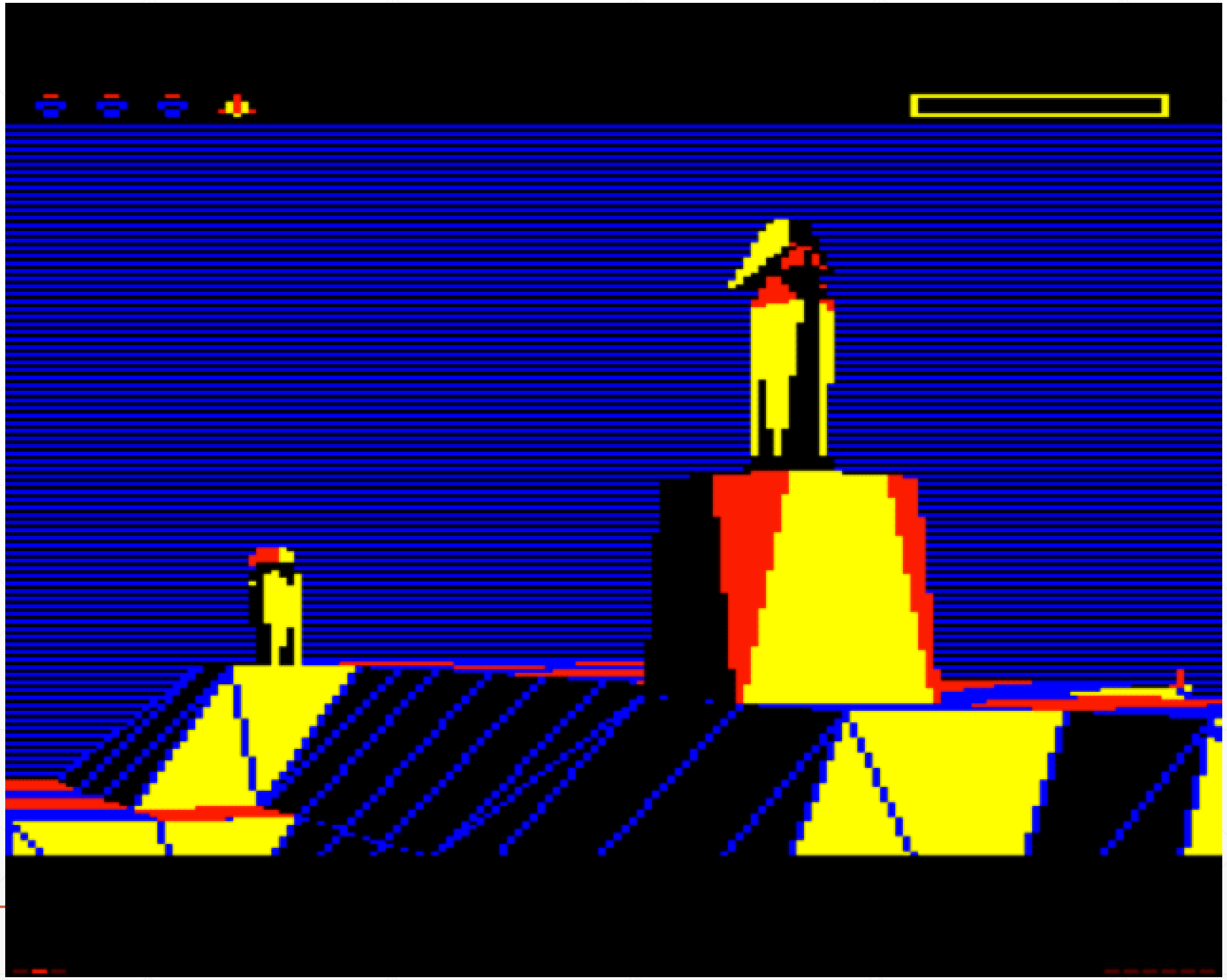
- **1980**: **Rogue**, the real ancestor of the rogue-like games used PCG to create a fully replayable game experience.
- Rogue, and all the rogue-like games from then, are one of the main game genres to use PCG in modern videogame.



```
Level: 1   Gold: 264    Hp: 19(25)  Str: 16   Ac: 6   Exp: 3/30
```

History of PCG in games

- **1985**: **The Sentinel** had 10,000 different levels stored in only 48 and 64 kilobytes.

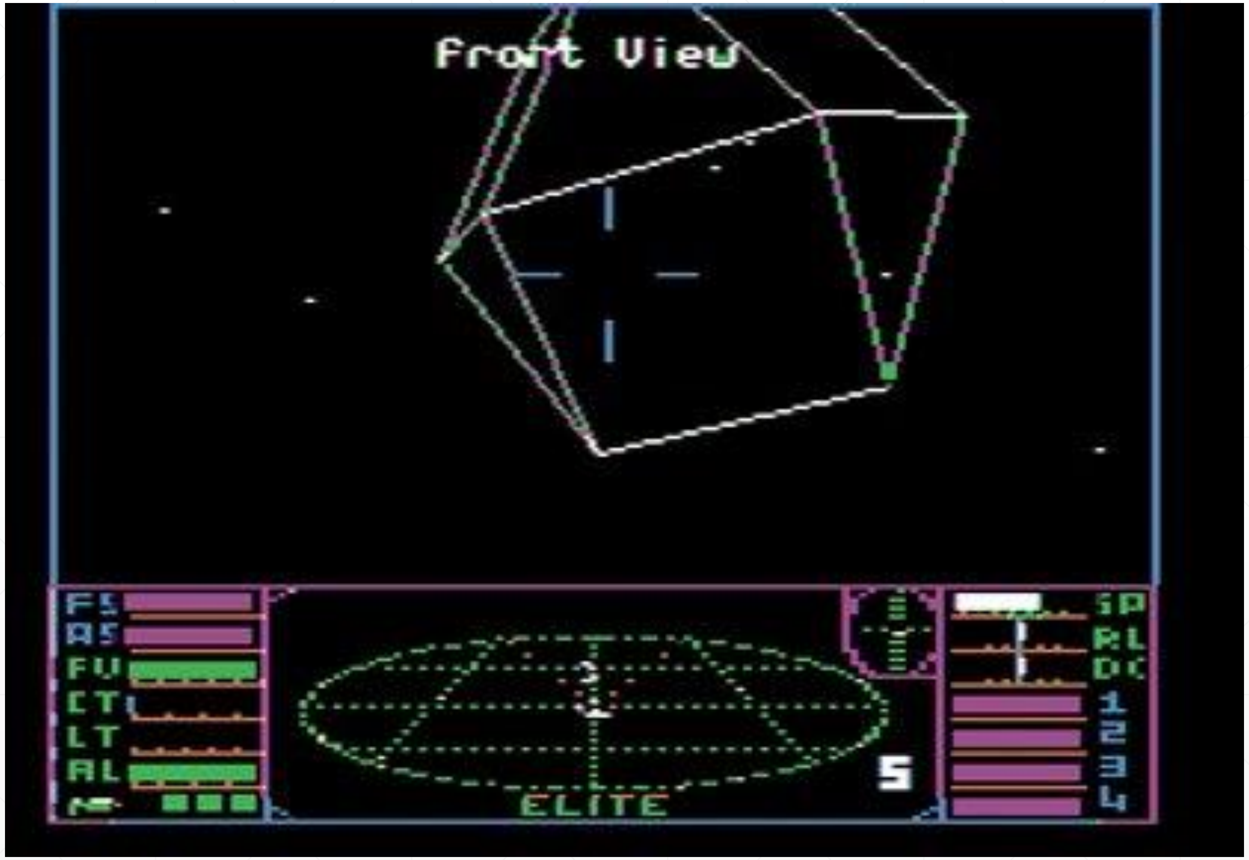


History of PCG in games

- **1985**: **Elite** use PCG to generate an universe with 8 galaxies with 256 solar system each. Each solar system has from 1 to 12 planets, each with a space station in its orbit, a proper names, a personal terrain , prices of commodities, and name and local details.
- Everything have to be contained in a 32Kb home PC. Both code and "contents".
- **SO EVERYTHING IS PCG**

History of PCG in games

- The seed of the universe is **hardcoded**.
- The seed number in the released game is **4096**
- However, there was some problems. Some solar system was poorly connected and the random name generator sometimes used profanity to name planets or space stations. :)



History of PCG in games

- The original algorithm is open source and can be found here:
  - http://www.iancgbell.clara.net/elite/text/
- A detailed explanation of the algorithm is given here:
  - http://www.gamasutra.com/view/feature/3377/algorithms_for_an_infinite_universe.php

History of PCG in games

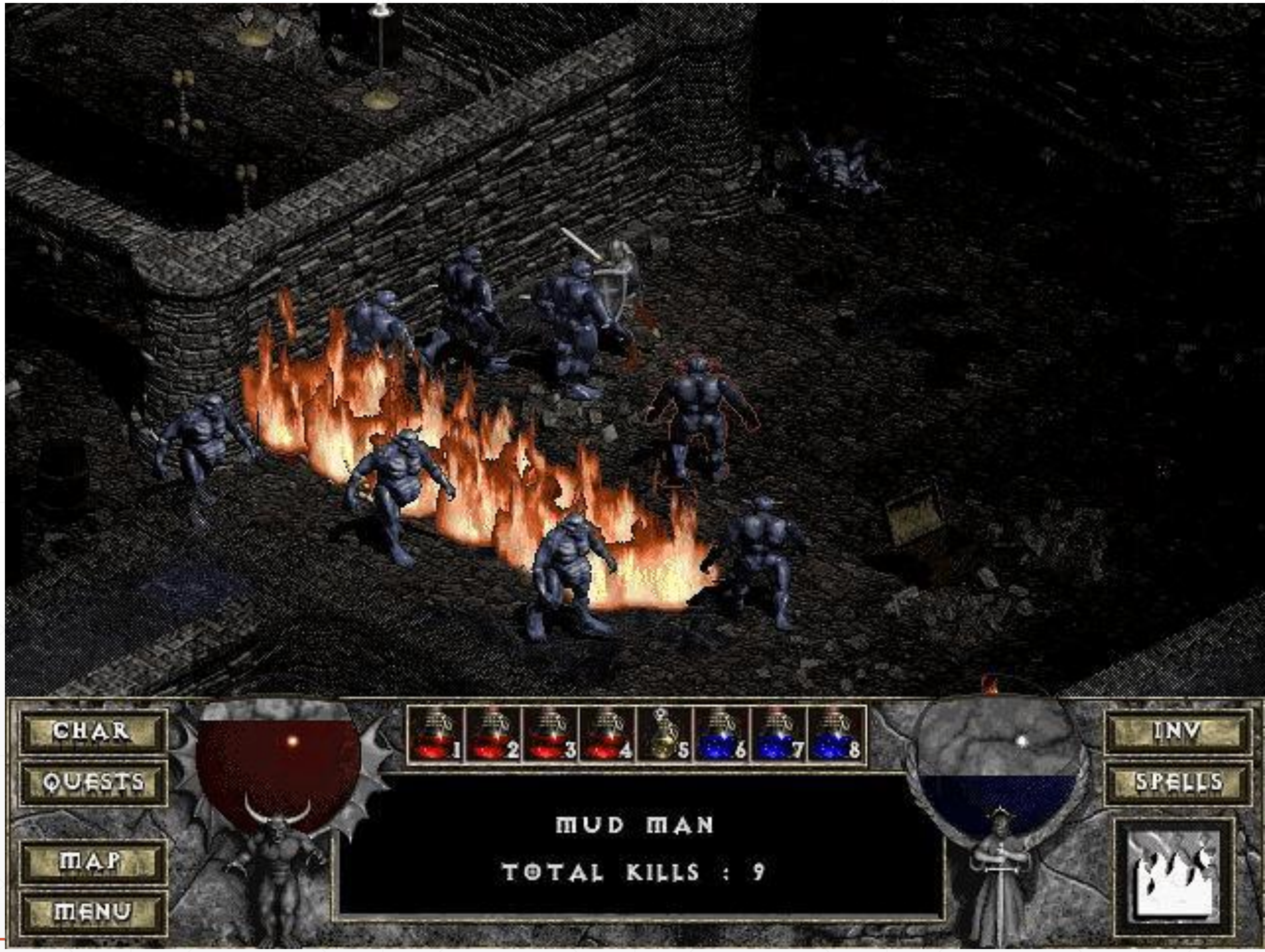
- By the way, the new version of Elite, **Elite Dangerous** use PCG to generate 1:1 replica of the Milk Way with more than **400 billion star systems.**
  - If we assume 1Kb per star system (very optimistic), without PCG the full system will occupy more than 400 Terabyte.

History of PCG in games

- Then the CD comes and developers became able to store thousands of times as much data than was possible in the early 80s, using procedural generation to build large worlds became unnecessary.
- So nothing happens, until mid **90s**. At this time PCG has three main purposes:
  1. Automatize designer into producing assets (for instance trees, rocks, foliage, and so on)
  2. Increase the amount of contents in a game in a way that is not possible to do manually.
  3. Provide improved replayability of a game!

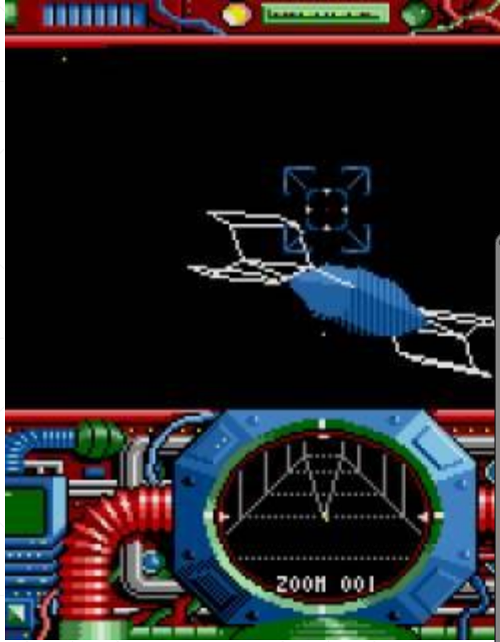
History of PCG in games

- **1995**: **Diablo** was one of the first games to introduce PCG and rogue-like games into the modern era of the videogame industry.
- Introduced
  - **Random dungeon layouts.**
  - **Random item generation.** This was the new things in commercial videogames.



History of PCG in games

- After 1996, real-time PCG was almost only confined into RPG, space games or obscure rogue-like game (not really mass-popular at that time).



Federation of Free Traders (1998)

Race: Cnidarian Nebula Dweller

Description: Large hemispherical bioluminescent invertebrate floater. Average thinker and physically quite weak.

Urluquai are natural space navigators due to their undersea heritage and their innate ability to levitate.

Since the gelatinous Urluquai have no speech, they communicate with each other using complex patterns of bioluminescent light and color.

Motives: Feed on fear and death.
Demeanor: Malevolent and thoroughly evil.

Special Ability: Enhanced Spatial Relations Sense

Home Colony: Urlu

Cnidarian Cloud
Urlu
Races

Mines: 10
Prod : 0

Surface: 801 Iron
Crust: 2070 Uranium
Deep Crust: 4864 Uranium
Mantle: 8752 Iridium
Core:12928 Uranium

Strange Adventures in Infinite Space (2002)

Dark Clouds (2000)

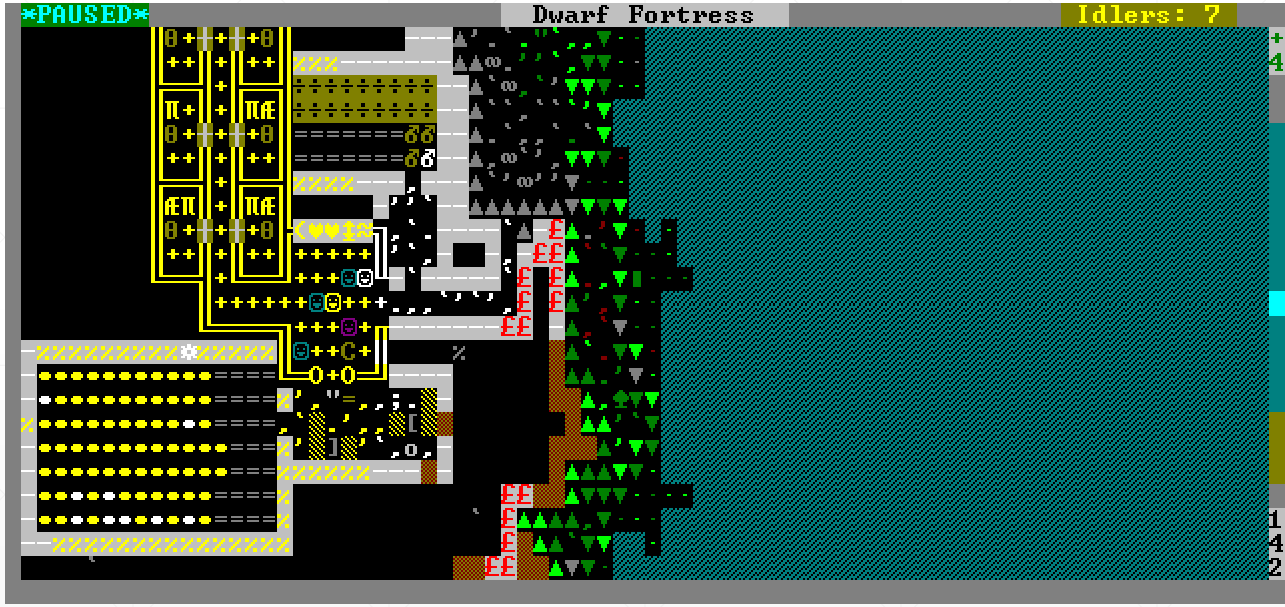
History of PCG in games

- Commercial design PCG based tools was instead starting its business!
- For instance, SpeedTree, the most famous PCG tools for trees and other organic assets, starts in 2002.

speedtree

History of PCG in games

- **2006**: **Dwarf’s Fortress.** Meanwhile, some obscure developer was implementing a state-of-the art fantasy world generator.
- The DF engine can generate:
  - A full who takes into account weather, biomes, geological distribution of materials, plate tectonics, wind and water erosion.
  - A full history for the world with population, races, cities who can rise and fall.
  - Poetry, monsters, animals, events, cities and A LOT MORE.



History of PCG in games

Dwarf’s Fortress was (and is) a niche game. Inspired by that game Markus Persson (Notch) decided to develop **Minecraft** (2009)

Minecraft is considered the root of the modern hype for PCG in the game industry.

History of PCG in games

Since then, PCG is now present in almost every non-FPS game released!

Modern PCG

How PCG is used nowadays!

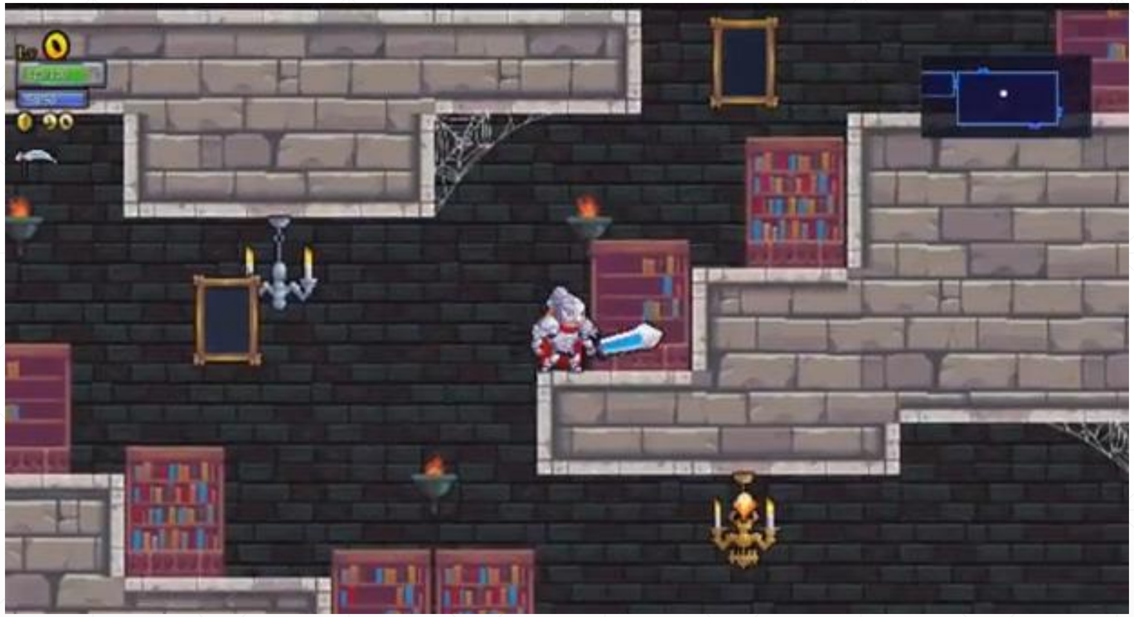
Types of PCG

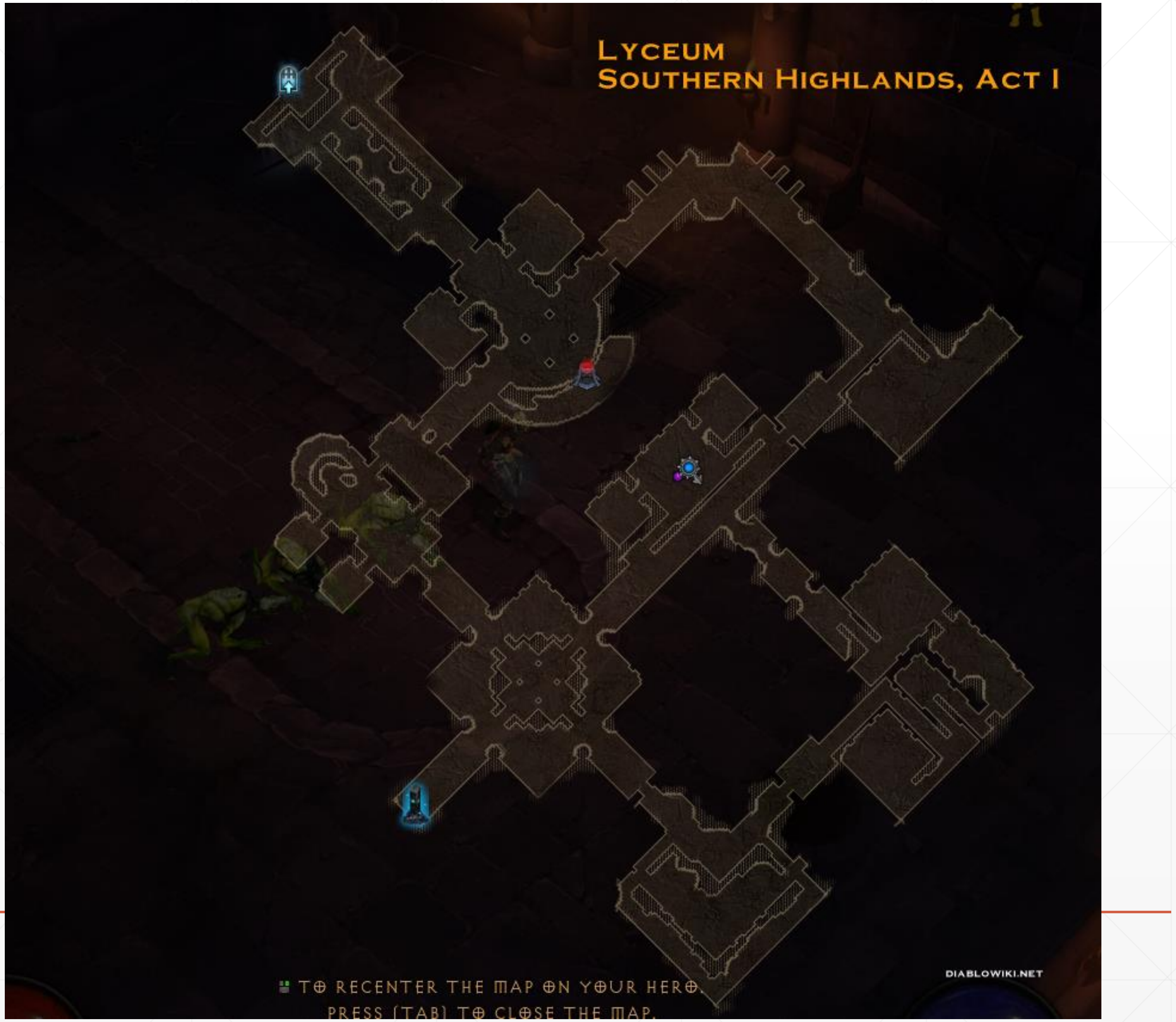
- There are six main application of PCG
  - **Runtime Random Level Generation**
  - **Design Of Level Content**
  - **Instancing Of In-Game Entities**
  - **User Mediated Content**
  - **Dynamic Systems**
  - **Procedural Puzzles And Plot Generation**

Runtime Random Level Generation

- **It is generation of game levels/world while the game is being played or loaded**. In this category fall all the map generation techniques. It is probably the most famous type of PCG techniques in games.
- **Notable Games**: Elite, Minecraft, Spelunky, Diablo, and a lot more.

Runtime Random Level Generation



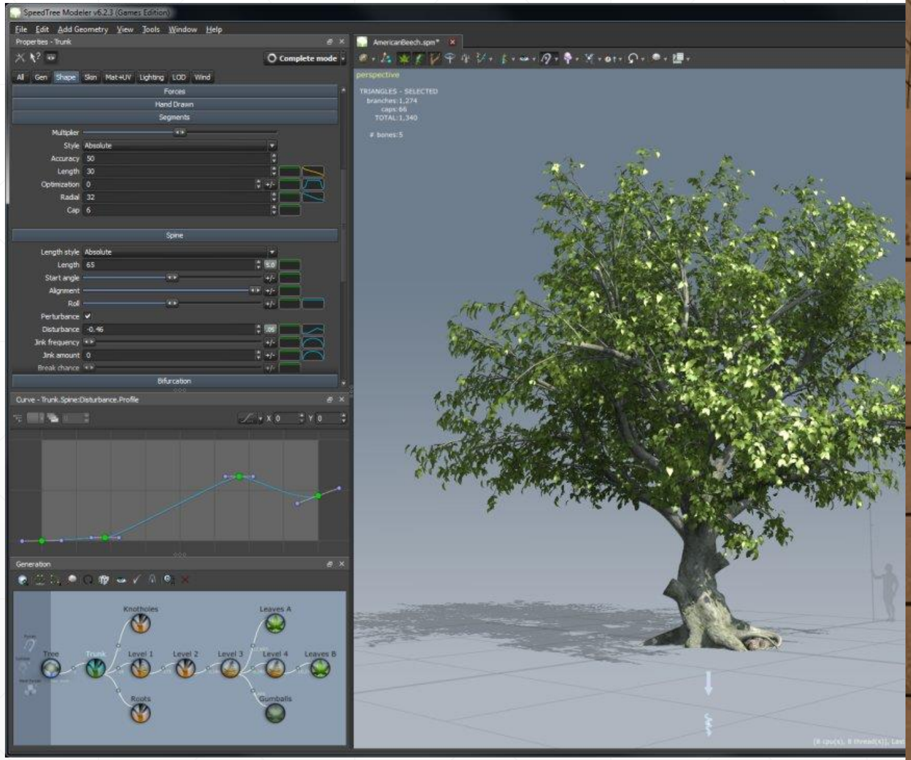


TO RECENTER THE MAP ON YOUR HERO.
PRESS [TAB] TO CLOSE THE MAP.

Design Of Level Content

- It is the use of PCG techniques **at design or build time**. In other world, the goal of PCG is to help the designer into creating contents.
- It is usually hidden to the final player.
- A common example of these techniques is the use of fractal *height map* to create landscapes and terrains which will than populated with other objects by hand.
- **Notable Games**: almost every game since 2002.

Design Of Level Content



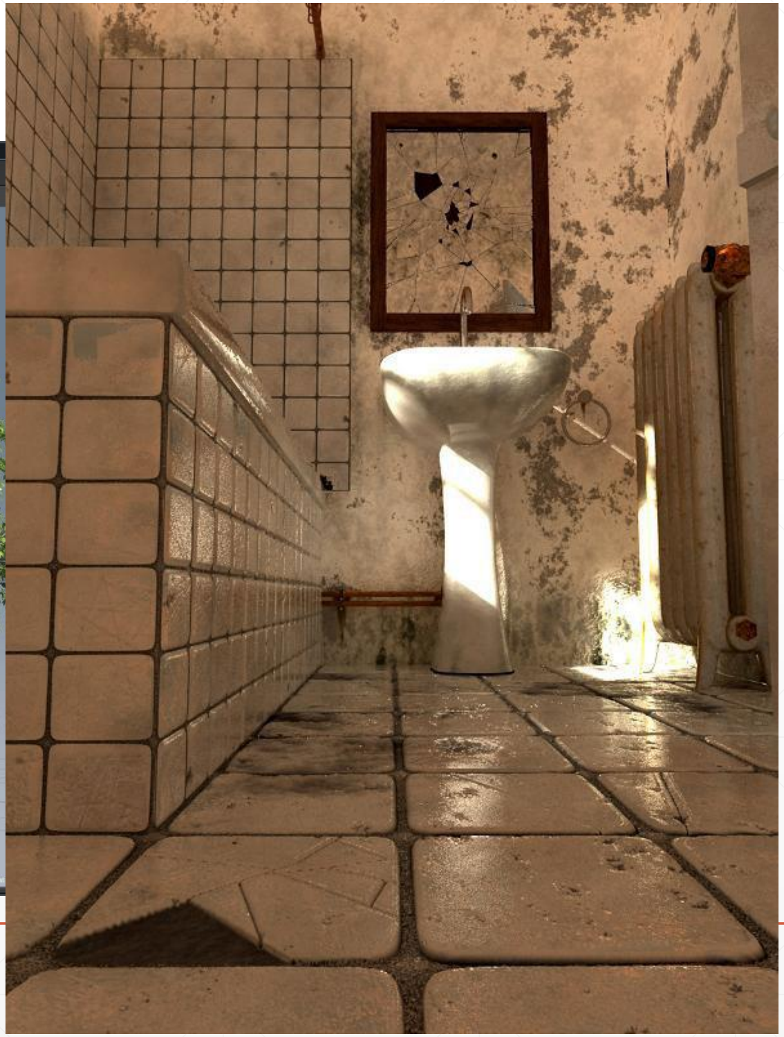
TRIANGLES - SELECTED
branches: 1,274
caps: 66
TOTAL: 1,340

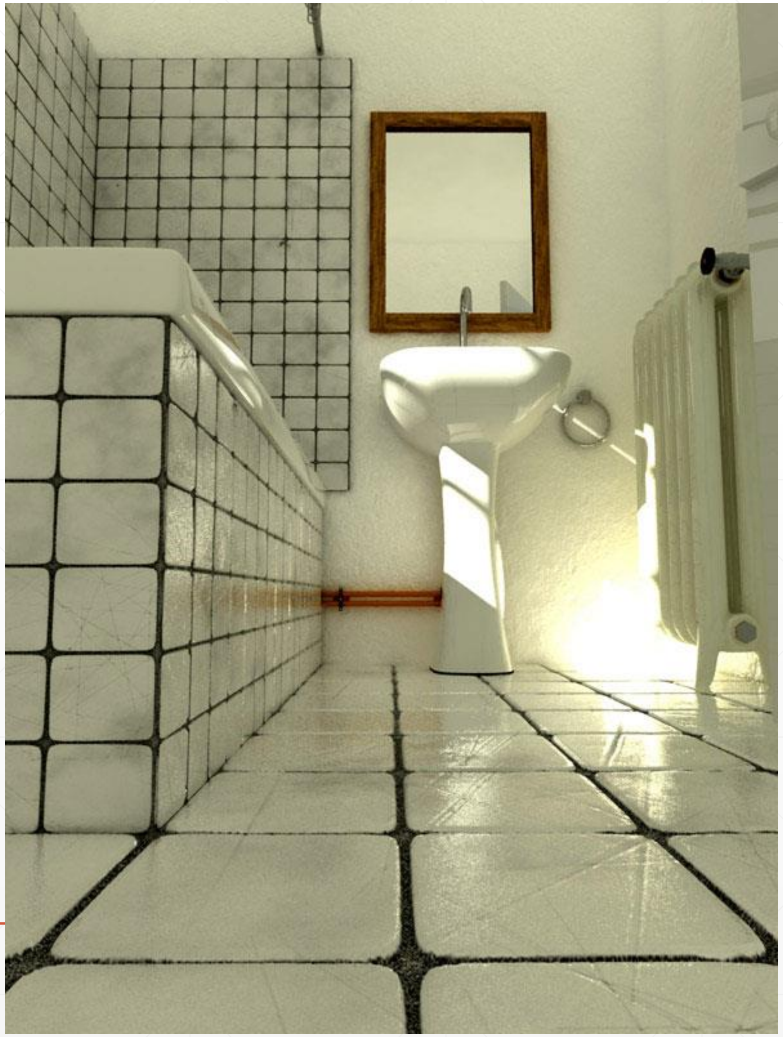
bones: 5

Multiplier
Style: Absolute
Accuracy: 50
Length: 30
Optimization: 0
Radial: 32
Cap: 6

Length style: Absolute
Length: 65
Start angle
Alignment
Roll
Perturbance: ✓
Disturbance: -0.46
Jink frequency
Jink amount: 0
Break chance

X 0 Y 0

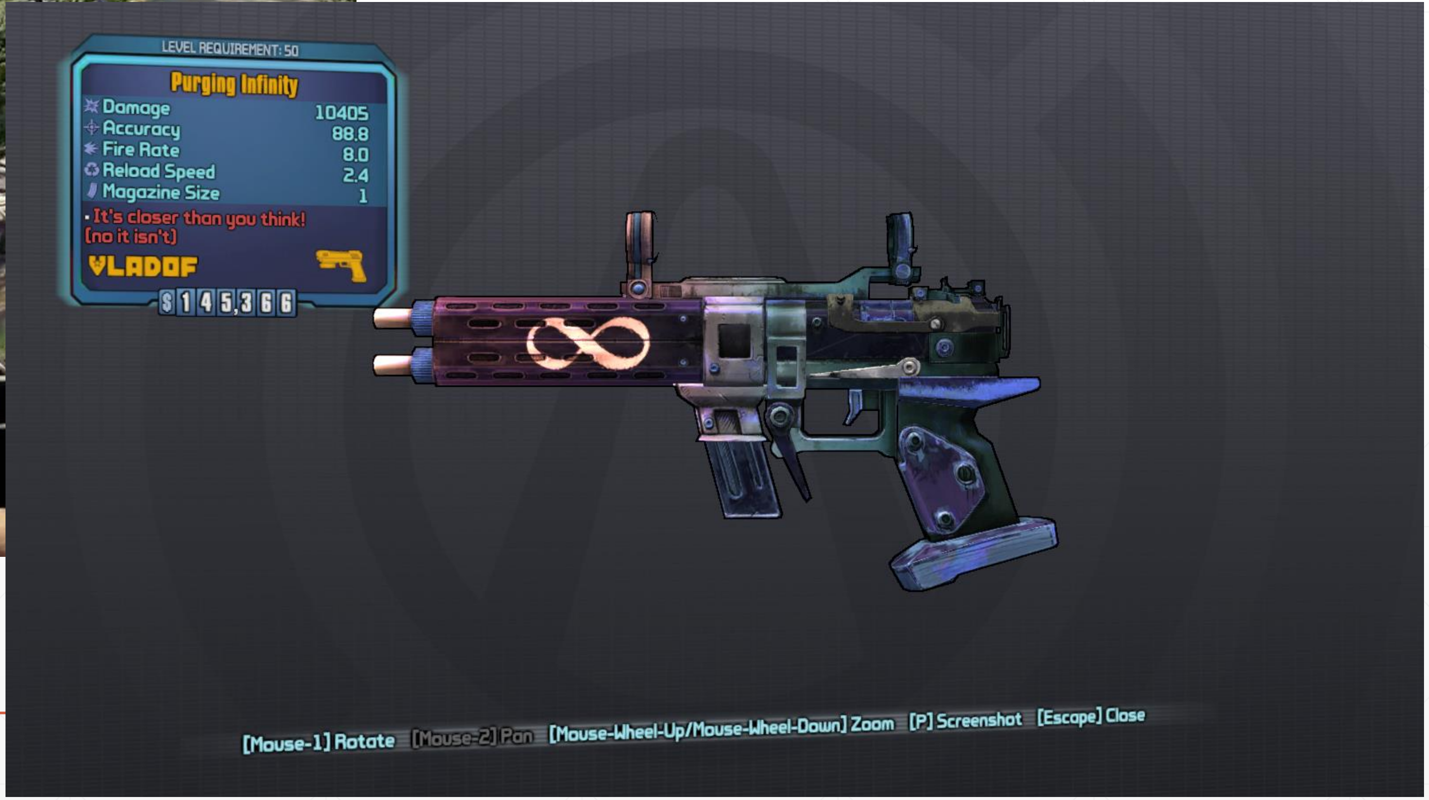




Instancing Of In-Game Entities

- These techniques are, in some sense, orthogonal to the previous category.
- Instead of generating the world, they instantiate the game objects (such as, trees, monsters, characters, item, treasures and so on).
- **Notable Games:** **Left for Dead**. It is used to instantiate dynamically the enemies into a static environment in order to provide variety and some adaptation capability to the game.

Instancing Of In-Game Entities


LEVEL REQUIREMENT: 50

Purging Infinity

Damage	10405
Accuracy	88.8
Fire Rate	8.0
Reload Speed	2.4
Magazine Size	1

- It's closer than you think! (no it isn't)

VLADOF

$145,366

[Mouse-1] Rotate [Mouse-2] Pan [Mouse-Wheel-Up/Mouse-Wheel-Down] Zoom [P] Screenshot [Escape] Close

User Mediated Content

- This is a borderline technique.
- It uses the users themselves as a source of new and “PCG” contents. This can be also be mixed with other PCG techniques (hence the word “mediated” in the name).
- Users can manually change PCG parameters to generate personalized contents, share these parameters, and so on.
- **Notable Games**: **Spore**, **X-COM** and **Dryad**, a tool for tree and organic offline generation.

Dynamic Systems

- This is **PCG applied to agent behaviors**.
- For instance, dynamic systems such as weather, and group and crowd behaviour, can be modelled using PCG techniques.
- In short: PCG applied to agent-s behaviors.
- **Notable Games:** **S.T.A.L.K.E.R.: The Shadow of Chernobyl**, contains one thousand **non-scripted** characters.

Dynamic Systems

Procedural Puzzles And Plot Generation

- PCG can be also applied to plot, story, quests and puzzle generation.
- This is a more difficult field because it often requires some **Natural Language Processing** ability. However, many games are starting to integrate automated quests into their code and gameplay.
- Puzzles can be extended by making multiple parts of the dependency graph randomly placed (e.g. moving the key that opens the door to a random accessible location) or by changing the shape of the dependency graph completely.
- **Notable Games: Apophenia**, indie-fully PCG puzzle game. **Mount and Blade**, for the PCG plot (thanks to community mods).

Cutting-Edge PCG

There are a lot of other PCG field expanding right now. This field are not still fully absorbed by the game industry.

- **Experience-Driven Procedural Content Generation**
  - Use player experience as a source for PCG algorithms.
- **Search-based procedural content generation**
  - Use evolutionary technique to produce more controllable PCG contents.
- **Procedurally Artistic-Contents Generation**
  - Visual art, music, poetry, architectures and so on.
- **Procedurally Generated Gameplay**

AND A LOT MORE

PCG in practice

Some Real-World algorithm for PCG

Teleological vs. Ontogenetic

- **Teleological Algorithms**: creates an accurate physical model of the environment and the process that creates the thing generated, and then simply runs the simulation, and the results should emerge as they do in nature. **Usually used in offline applications.**

T

- **Ontogenetic Algorithms**: ontogenetic approach observes the end results of this process and then attempts to directly reproduce those results by ad hoc algorithms. **Usually used in online applications.**

O

O

Mazes

Mazes are one of the eldest form of PCG. Studied in mathematics for several decades.

There are **A LOT** of algorithms:

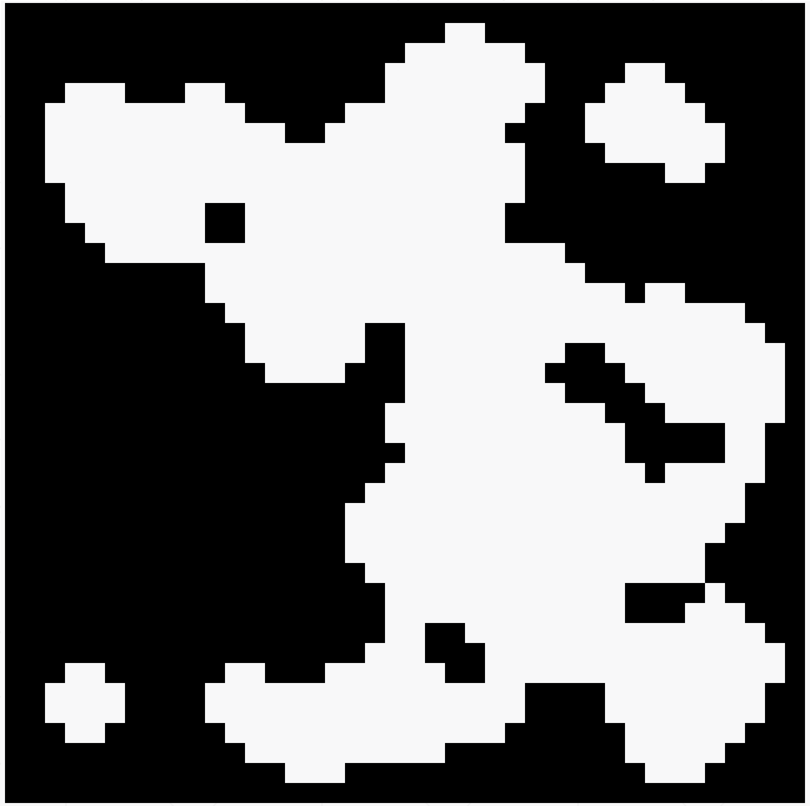
- **Prim’s Algorithm:** http://bl.ocks.org/mbostock/11159599
- **Random Traversal:** http://bl.ocks.org/mbostock/70a28267db0354261476
- **Randomized Depth-First:** http://bl.ocks.org/mbostock/1ef3b1fb9eb35ca8ffff
- **Wilson’s Algorithm:** http://bl.ocks.org/mbostock/11357811
- And others….

O

Cellular Automata

A cellular automata is a grid of cells, each one having a state, and a rule for determining what state a cell transitions to based on the state of it and its neighborhood.

Used for cave/natural-like environments.



Cellular Automata

```
function cellularLogic(r,c,clean) {
    var numWalls = countAround(r, c, 1, 1);
    var numWalls2 = countAround(r, c, 2, 2);
    if (isWall(r,c)) {
        if (numWalls >= 3) { return 1; } return 0; }
       else {
           if (!clean) { if (numWalls >= 5 || numWalls2 <= 2) { return 1; } }
           else { if (numWalls >= 5) { return 1; } } } return 0; }
```

O

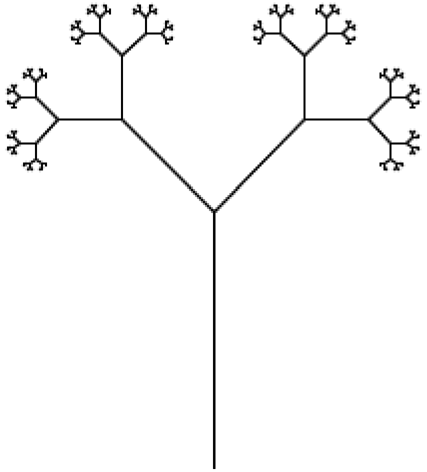
L-System

An L-System is a parallel rewriting system and a type of formal grammar.

- Introduced in 1968 by **Aristid Lindenmayer**, a botanist. Was used to describe the behavior of plant cells and to model the growth processes of plant development.

L-System: Example

variables : 0, 1
constants: [,]
axiom : 0
rules : (1 → 11), (0 → 1[0]0)

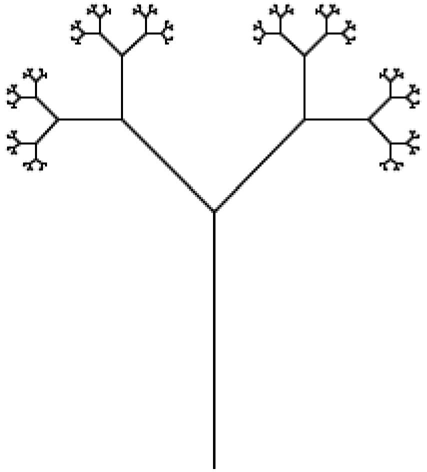


The shape is built by recursively feeding the axiom through the production rules.

axiom:	**0**
1st recursion:	1[0]0
2nd recursion:	11[1[0]0]1[0]0
3rd recursion:	1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

O

L-System: Example



At the end we add a **semantic** to each symbol. For instance:

- 0: draw a line segment ending in a leaf
- 1: draw a line segment
- [: push position and angle, turn left 45 degrees
-]: pop position and angle, turn right 45 degrees

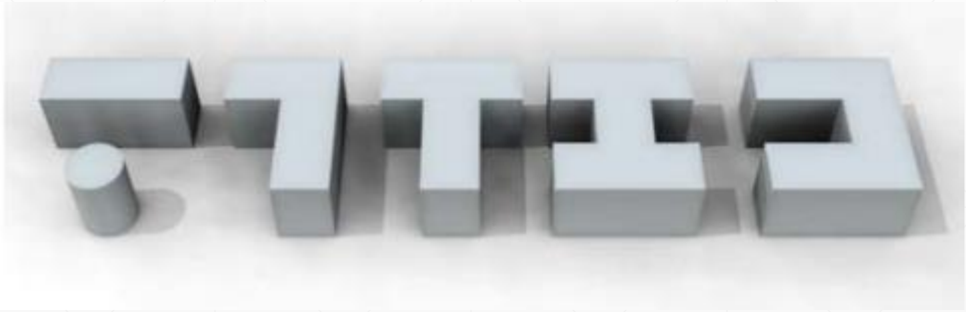
TRY THIS 11[1[0]0]1[0]0

L-System Use Case: PCG Buildings

L-Sys has been proven useful for PCG buildings.

Müller, Pascal, et al. *Procedural modeling of buildings*. Vol. 25. No. 3. ACM, 2006.

Use basic building block (L, H, U and T) and use grammars to generate complex shapes (rotation, splitting, scaling, roof selection and a lot of things).

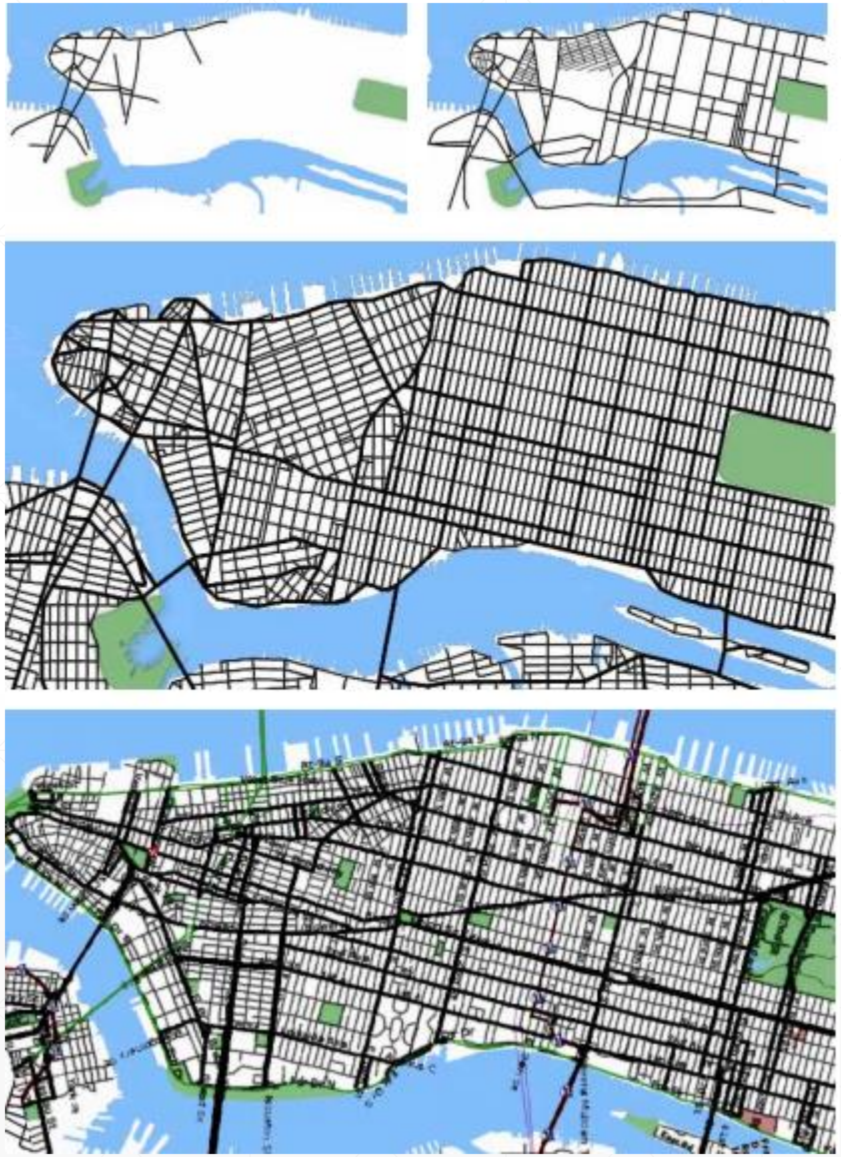


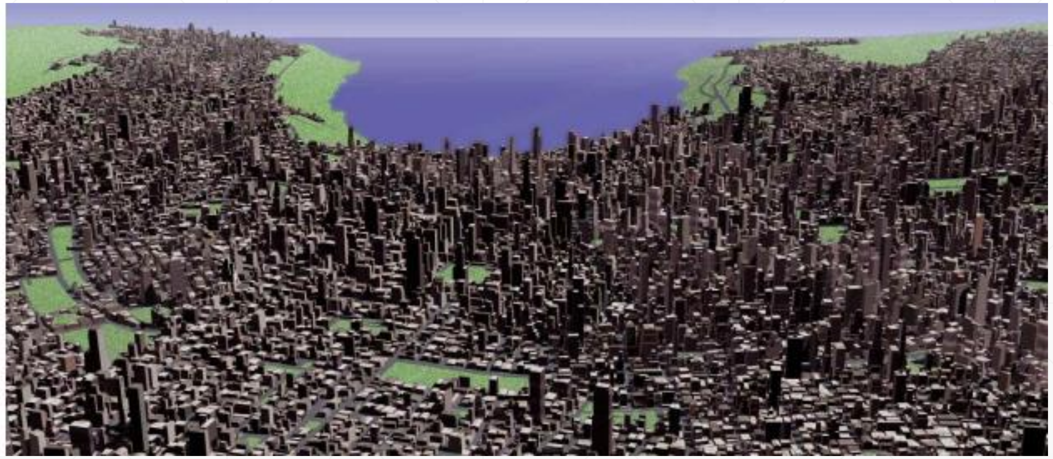
O

L-System Use Case: PCG Cities

L-Sys has been proven useful also for cities!

Parish, Yoav IH, and Pascal Müller. "Procedural modeling of cities." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.

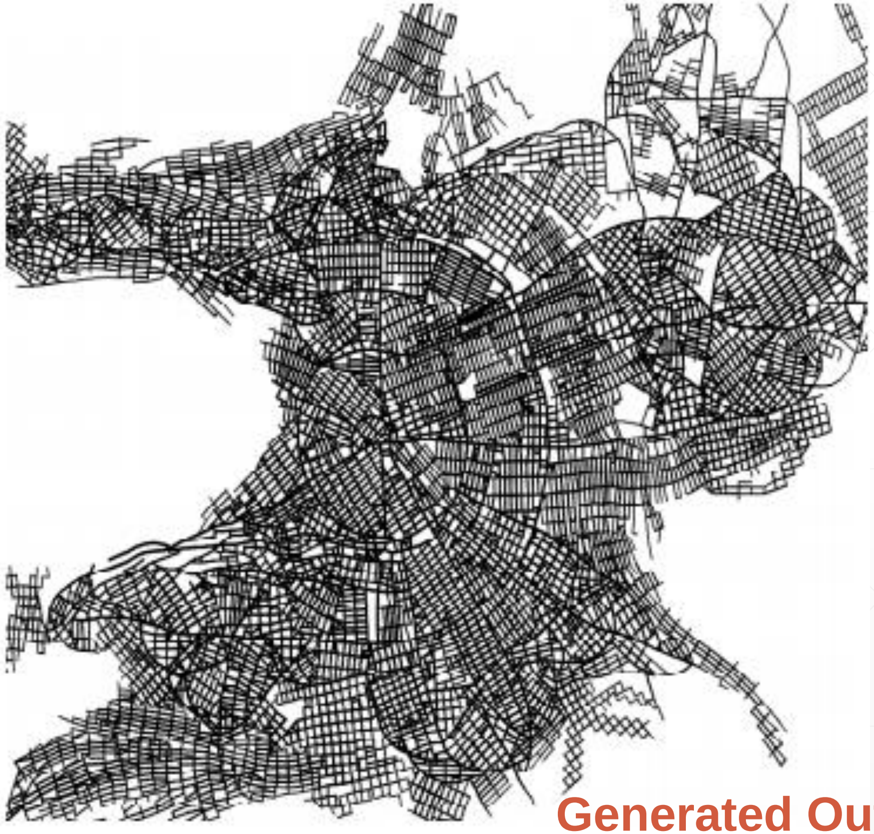




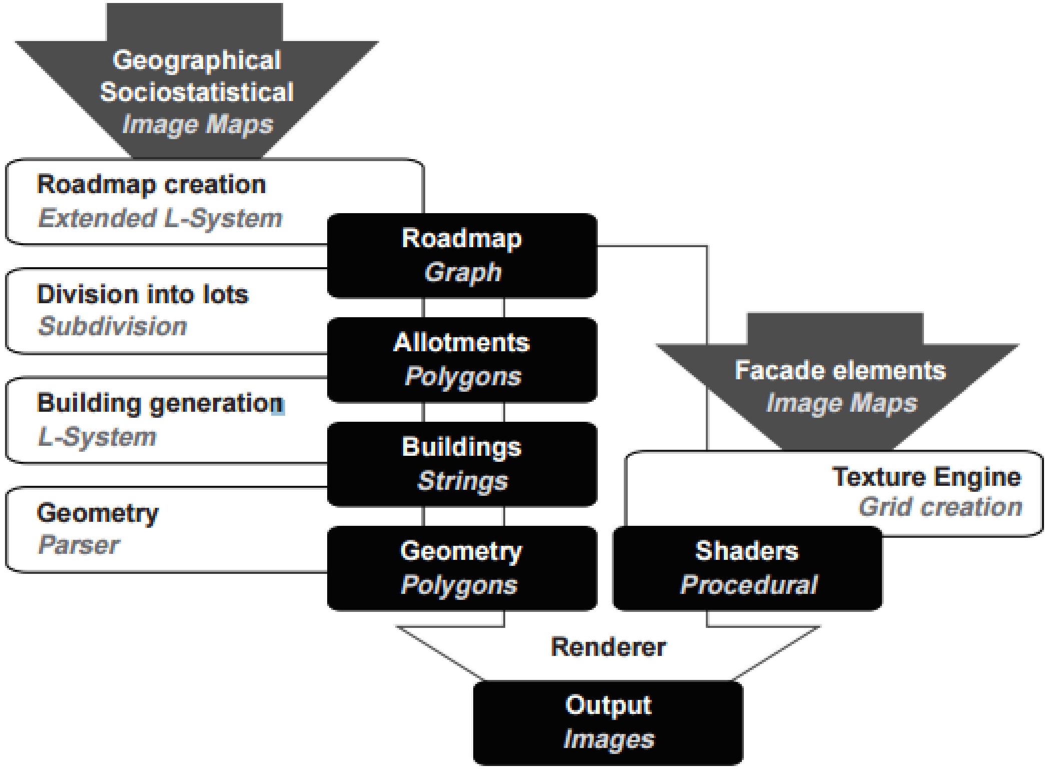
L-System Use Case: PCG Cities

Water
Altitude

Pop.Density



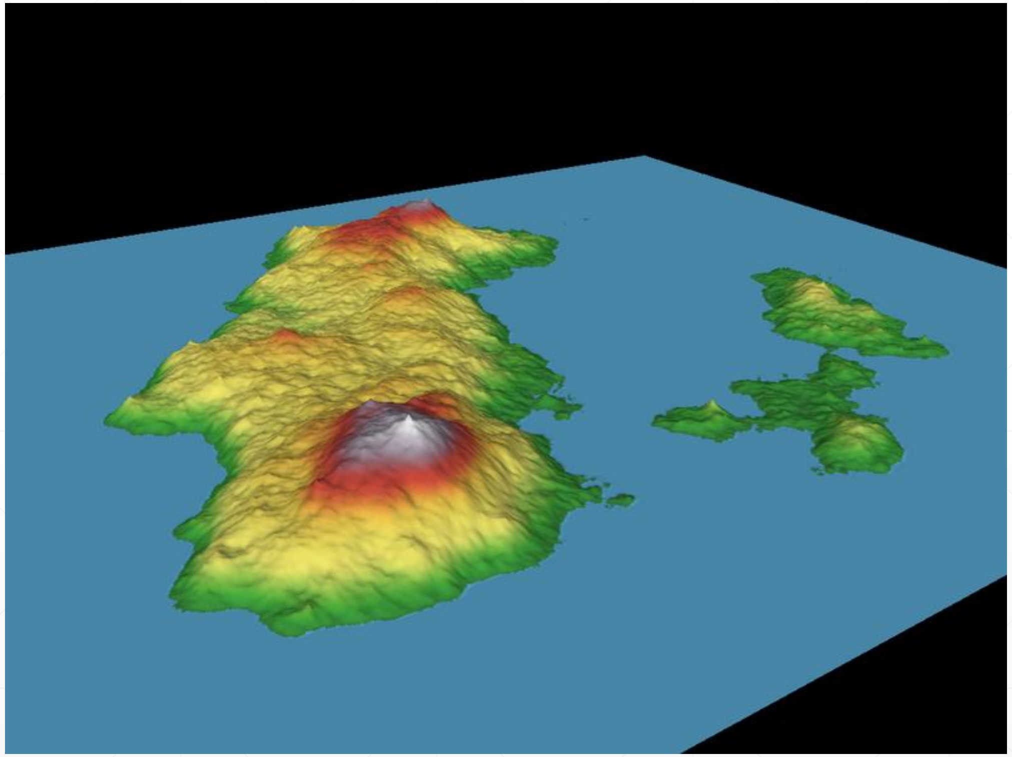
Generated Output



Diamond-square algorithm

Is a common implementation of the **midpoint displacement algorithm**.

1. Assign a height value to each corner of the rectangle (image).
2. Divide the rectangle into 4 subrectangles, and let their height values be the mean values of the corners of the parent rectangle.
3. When computing the middle height, one should add a small error that depends on the size of the rectangle (**roughness**)
4. Iterate and subdivide each rectangle into smaller ones. Eventually, they will be too small to produce a noticeable difference. When this occurs, stop the iteration, and render the pixel with the mean of the height values.



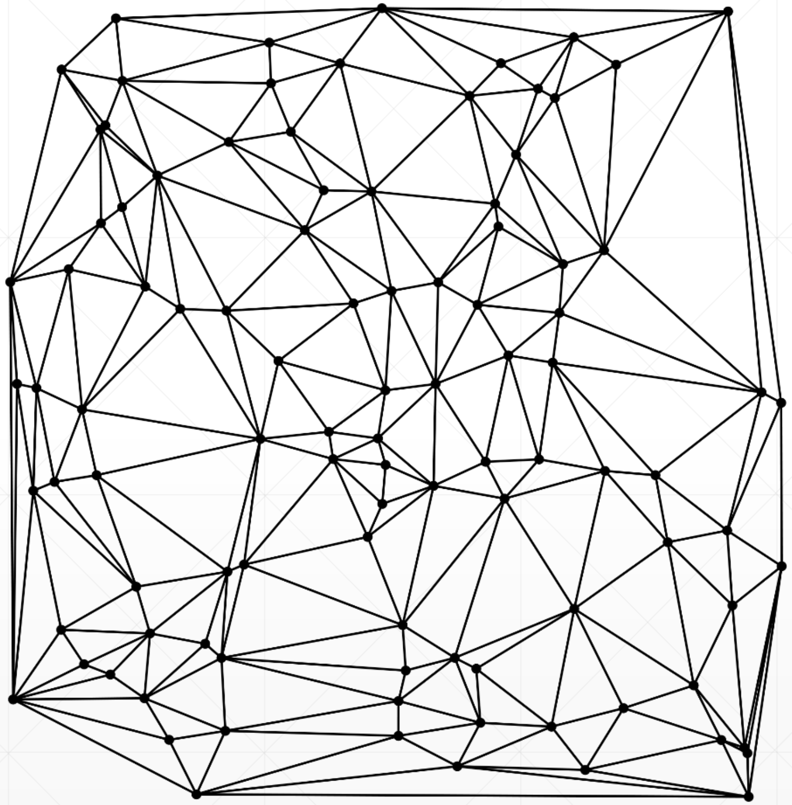
Dungeons

There is an huge amount of techniques to generate dungeons. This can be a presentation by its own. :D

Dungeons: Voronoi-Delaunay Triangulation

Dungeon Generators is another big field of PCG. A common technique to build dungeon-like artificial structures is given by the Voronoi-Delanuay Triangulation.

Example:
http://tinykeep.com/dungen/



Rain-Drop Algorithm

A **rain drop algorithm** is a method of modifying a height field by simulating rain drops falling onto the height field, and then moving from higher points to lower points. The rain drops are used to initially remove height from where they first appear, then deposit additional height at lower or lowest points they find. This emulates the process of erosion.

Other Teleological Algorithms

- **Fire Propagation**: Diffusion of flames in dynamic environments.
- **Artificial Life**: Simulate the environment through the creation of a lot of simple artificial life agents.
- **Fluid Dynamics**: Dynamic of fluids like water and magma (e.g., in Dwarf Fortress the user can create “steam engines” using the in game fluid and thermodynamic simulation.
- **Reaction-Diffusion System**: introduced in 1950 by Alan Turing to explain the morphogenesis of living creatures.

PCG Music

Procedurally generated music is usually implemented using **Hidden Markov Models**.

http://hdl.handle.net/1802/1510

Funny Example:
http://www.fakemusicgenerator.com/

Generates, album, artists, tracks and, obviously, generate MP3.