Random map generation

A presentation to learn about the random generation of maps and how to apply them

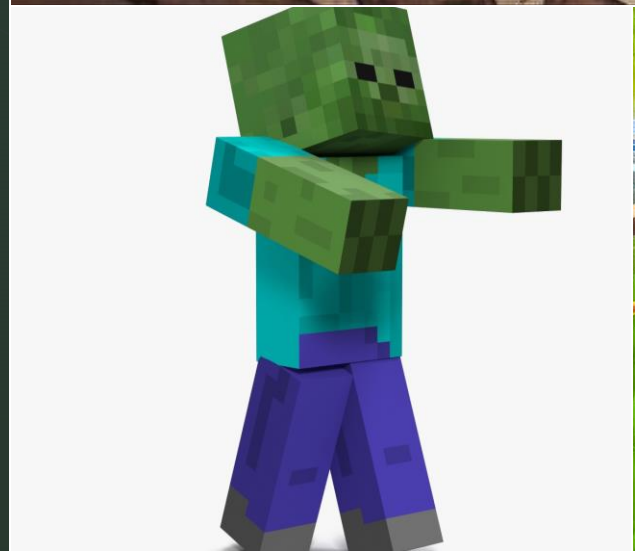
Jordi Espriu

Introduction

- Procedural generation vs random generation.
- Procedural generation: The game itself will create original content for the player to explore or use
- Random generation: The game will build the content on predifined elements that the developers hard coded

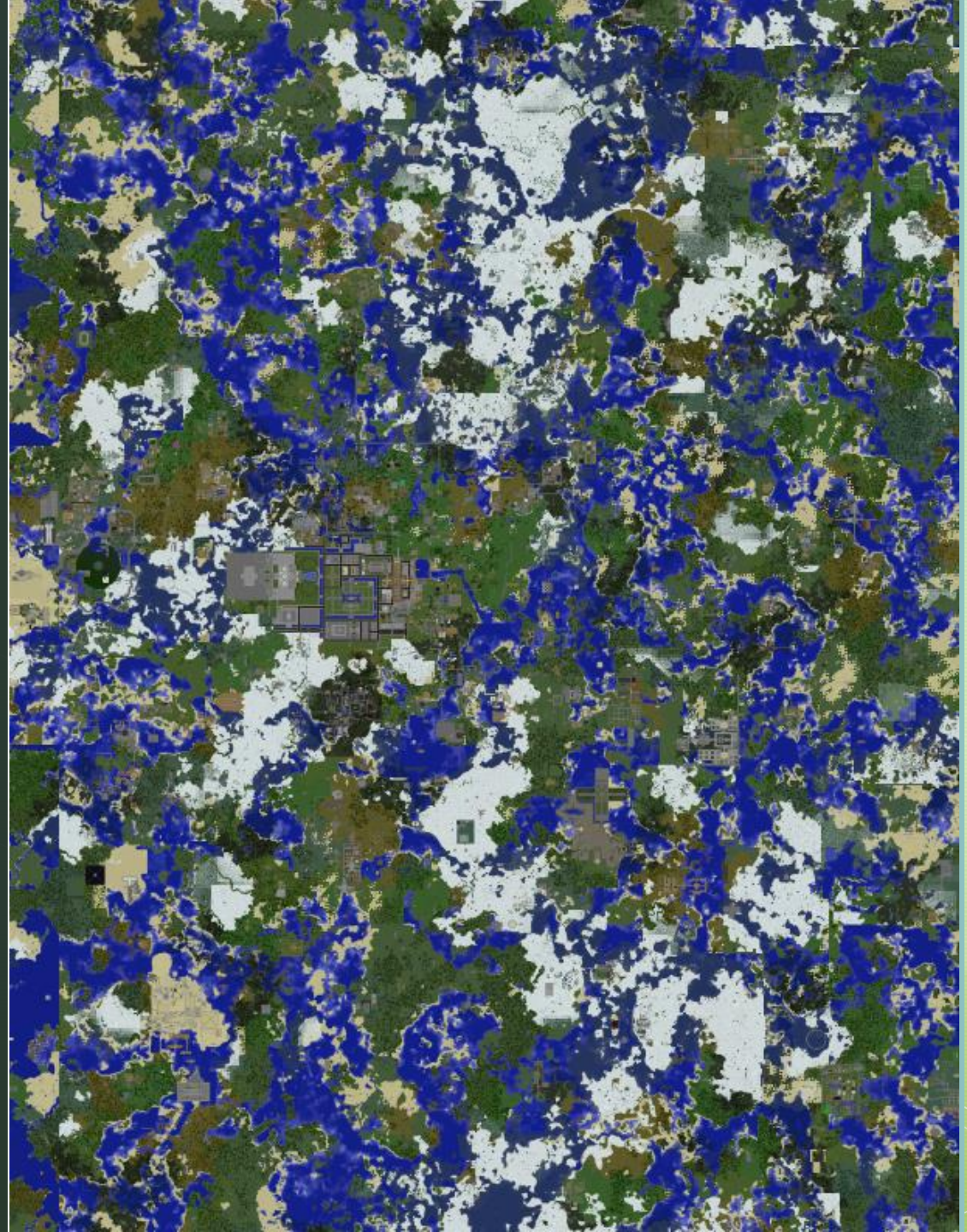
Procedural generation: Applications

- -Terrain/level creation
- -Enemies/spawning
- -Loot/weapons/gear
- -Models/textures/animations
- -Sound/music
- -Story/quests/dialog



Full randomness

- I don't think this has been hardcoded...



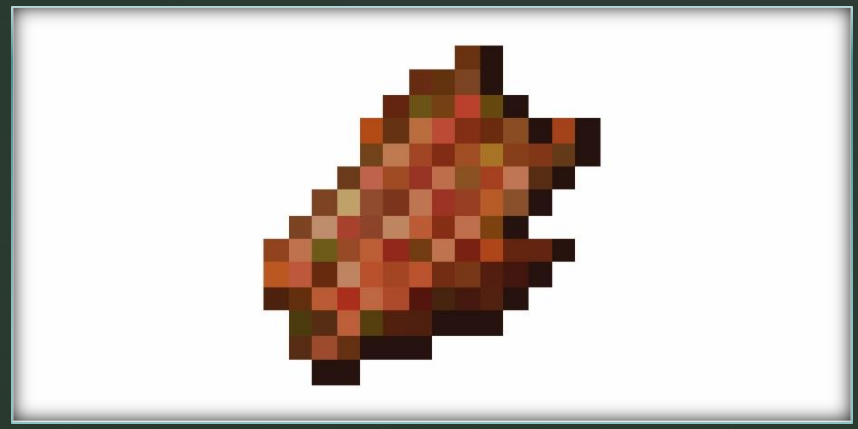
Guided randomness

- Can you spot the difference...?

Seeds

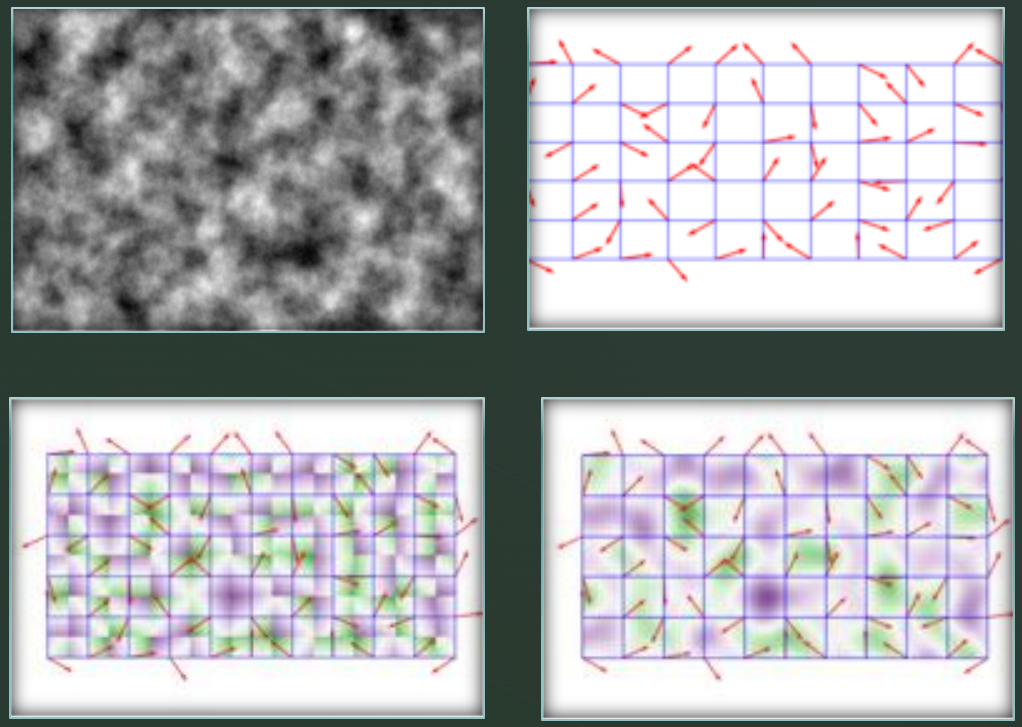

Create New World
Seed for the world generator
-7865816549737130316
Leave blank for a random seed
ate Structures: ON
World
, dungeons etc
ow Cheats: OFF
Bonus
ds like /gamemode, /experience

Random generation: Pre-generated features



Terrain generation: Perlin noise

- Perlin noise is a type of gradient noise developed by Ken Perlin in 1983 as a result of his frustration with the "machine-like" look of computer-generated imagery (CGI) at the time.
- https://www.youtube.com/watch?v=-POwgollFeY&t=175s



Designing of a random map

- First of all you need a design stage to decide what will be random and what will not, this is basically because randomness is a really cool feature but could be deadly when used wrong.

Implementing a random map in C++

- Random library from c++

```
#include <iostream>
#include <string>
#include <random>
#include <cassert>
```

TODO 1

- Create an enum class named "Tile" with the following: Unused, DirtWall, DirtFloor, Corridor, Door, UpStairs, DownStairs.

TODO 2

- Create an enum class called "Direction" with the following: North, South, East, West.

TODO 3

- Fill this function that checks the limit of the X axis of the map

```
bool IsXInBounds(int x) const
```

- Fill this function that checks the limit of the Y axis of the map

```
bool IsYInBounds(int y) const
```

TODO 4

- Fill this function that generates random numbers.Use the following function from the random library: uniform_int_distribution
- PD: typedef std::mt19937 RngT;

```
int GetRandomInt(RngT& rng, int min, int max) const
```

TODO 5

- Set the xLength and the yLength of the room for the following function, using the GetRandomInt function is necessary

```
bool MakeRoom(Map& map, RngT& rng, int x, int y, int xMaxLength, int yMaxLength, Direction direction) const
```

TODO 6

- Create the room with the function SetCells (from the class Map)
- PD: the room is formed by dirtfloor and dirtwalls.

TODO 7

- Make a variable called "Chance" that gets a random number between 0 and 100

TODO 8

- Make one room in the middle to start things off.

MUST SEE

- https://www.youtube.com/watch?v=C9RyEiEzMiU&t=2320s