

Introduction

In this tile-laying game, players must strategically place procedurally generated tiles onto a hexagonal grid, ensuring that each tile connects appropriately to its neighbors; for instance, a forest tile must connect to another forest tile.

We were initially inspired by the game Dorfromantik, which all of our team members deeply enjoy, particularly the satisfying experience of placing tiles to construct a town. Motivated by this, we've decided to develop a planetary version. Looking ahead, we plan to enrich the game by introducing a variety of new tiles, including animals, boats, and rivers.

Goal

We want to let the player create a procedural planet by putting tiles. At first, the planet is covered by ocean. Players will be provided with tiles that are associated with specific environments (such as forests, deserts, mountains, cities, etc.) Therefore, the player has the freedom to craft their own unique planet.

When the player places a tile, we fill it with Wave Function Collapse to ensure the most realistic appearance is achieved.



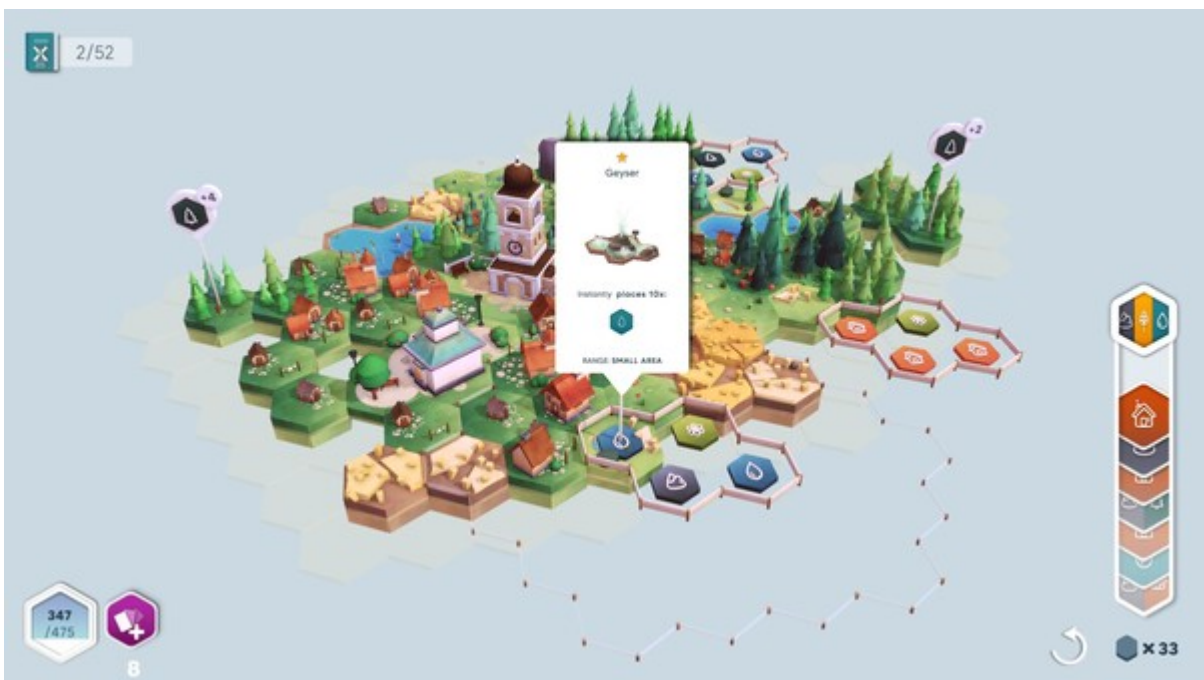
Inspiration/reference:

Our inspiration is drawn from a game known as Dorfromantik, which is a tile-laying game offering an escape from reality. We are quite fond of its mechanics and aim to create a planetary version of it.

[Dorfromantik](#):



Pan'orama:



Specification:

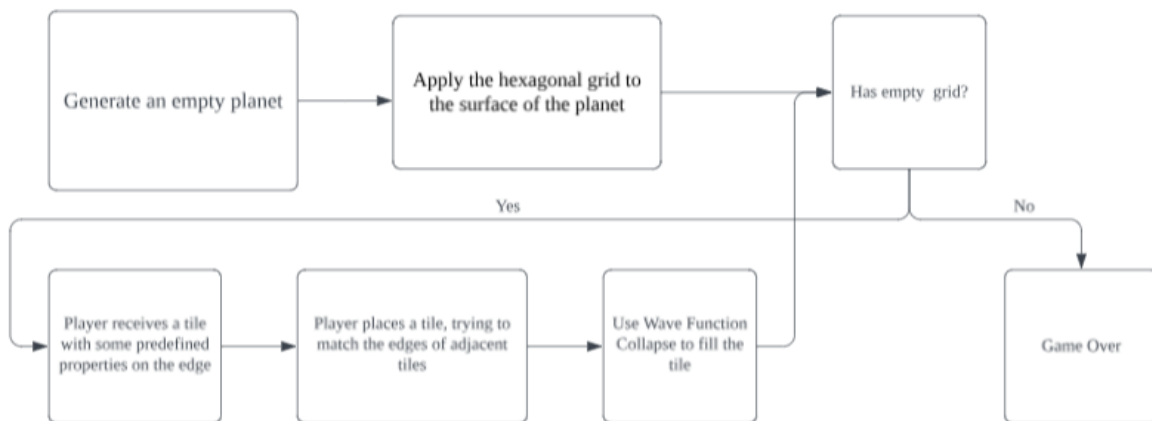
Procedural part:

1. Edge matching with wave function.
2. Procedural properties and elements for planets.

Techniques:

1. Use Wave Function Collapse to fill the grid with the attention of adjacent grids.
Reference: Stalberg, O. "[Wave Function Collapse in Bad North](#)." (2016).
2. The initial planet covered by the ocean is a perfect sphere rendered with an implicit surface.
3. Use noise function to generate clouds.

Design:



Timeline:

Milestone 1: Construct basic game level features and assets, develop the mapping of regular polygons on the surface of the planet.

Milestone 2: Enrich the integrity of the game level, implement Wave Function Collapse and disinform for regular polygons matching.

Milestone 3: Improve the game level, create procedural planets based on different properties like proportion of ocean, proportion of land, etc.

	Milestone 1	Milestone 2	Milestone 3
Alex Fu	Finish the framework of Wave Function Collapse	Adding attention to the adjacent tiles to the wave function. Make tiles seamless	Polish the procedural generation

Jiefu Ling	Design the Shader for the Planet and Get Modeling Prefabs of House, Trees, Ice.	Design atmosphere shader (cloud)	Create diverse planet surface shader
Xiaoyue Ma	Implement the mapping of regular polygons on the planet	Create procedural terrain on the planet	Create procedural diverse planets