

Bears Game Procedural Level System

Bears is a tactics-style strategy game focusing on a fast pace and chaotic, unpredictable gameplay. Level design in the game generally focuses on physics-like objects in the world, but the multiplayer demands a procedural approach.

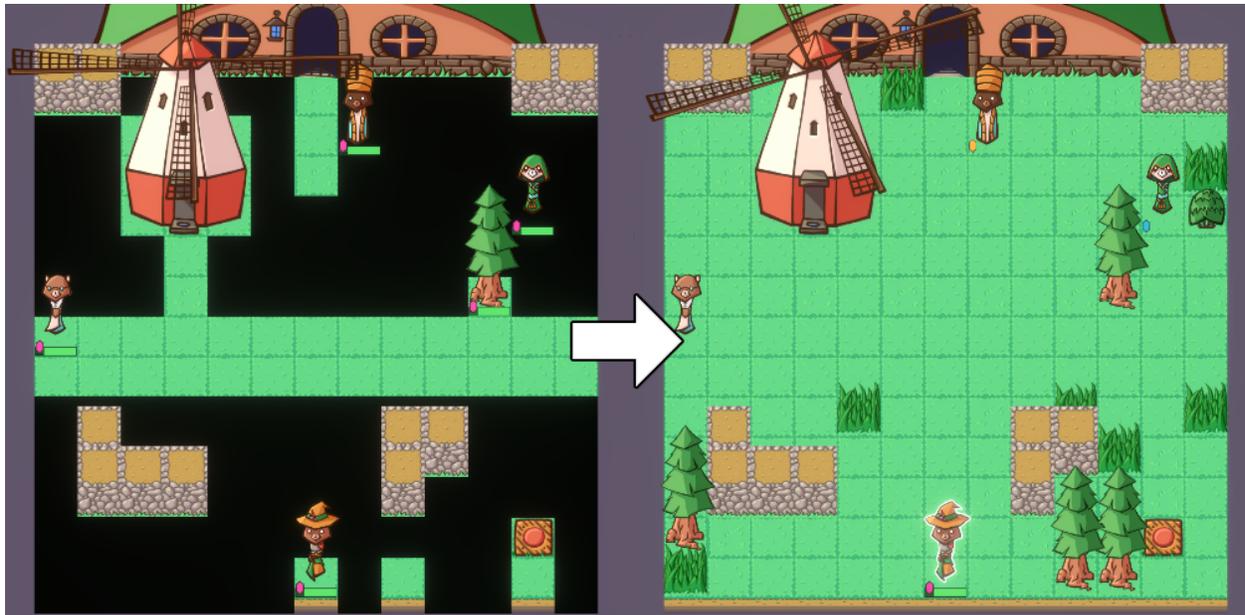
Design Goals:

- Create a flat, predictable landscape which can be filled with many bears on many teams with minimal disruptive obstacles, but enough to offer variety.
- Make it feel like a forest, lots of trees but ample room to walk between them.
- Allow the system to work around pre-placed objects, so if a character or obstacle is placed by hand in the Unity editor, the generator will work around them.
- Create a system flexible enough to be used across multiple biomes.



Filling In The Gaps

One of the design goals of this procgen system was to make maps that could be partially procedural. If the design of the level demanded assets in particular locations, such as enemies with particular spawns, or an object which you must interact with, the procedural system could work around those requirements. In this image, the black squares are procedurally replaced while the non-black squares are unchanged. The system also recognizes when an obstacle, like a bear, is placed onto a tile and will not try to place a tree or a bush in those locations.



Intelligent Placement

Tiles in the procedural system also respond to other nearby tiles. Trees are less likely to spawn next to other trees, which keeps the forest floor walkable. Each tile has a set of values that determine the likelihood they will spawn next to other tile types, or even to occupied tiles.

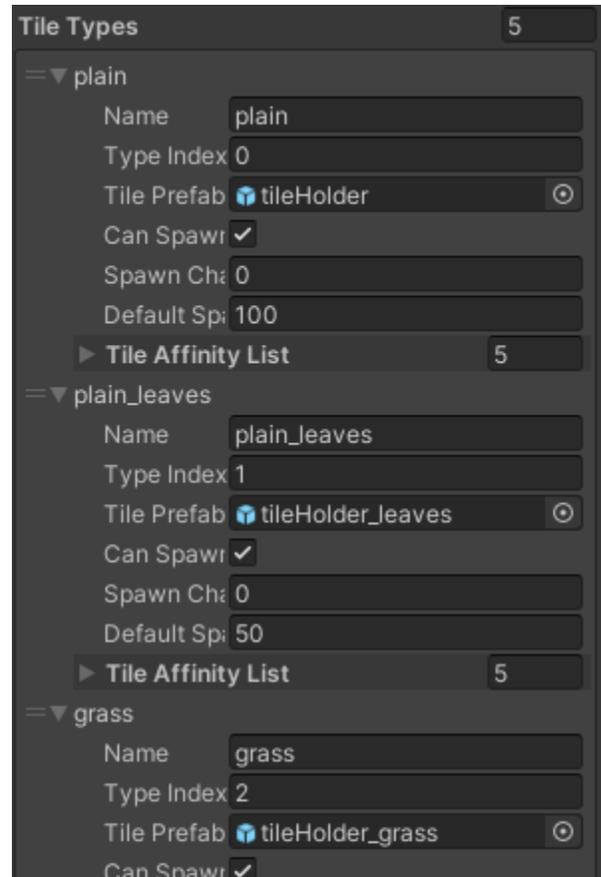
Similarly, procedurally-placed enemies will spawn with specific distances to other bears on their own team, or to bears on other teams. This also applies to procedurally-placed interactable objects, such as the various buttons and other victory conditions which might want to be placed randomly, but done so with awareness of not being too close to any particular bear.

Multiple Biomes

The tile system is robust through multiple biomes as a result of tiles spawning from a list of possible tile types.

On the image to the right, you can see the variables which define each particular tile in the forest tile set. This same list can be easily duplicated to set up a new biome made of entirely different tiles with entirely different properties. Each list is then pointed to as the target list for this current level, making swapping incredibly simple.

Placement of an object in the index can be maintained manually, if swapping becomes frequently desired, as each tile is referenced by their index for standard use. In other words, if I put the Forest Tree as the 5th item on the list, then I can also define the Desert Tree as the 5th item and then if I copy a map between biomes, the type of object present in that biome will remain consistent.



The obvious weakness of this system is that everything must be placed in the list somewhat manually, since indexes are easy things to change, particularly on longer lists. Fortunately, this list of items remains relatively short, as it only ever includes environment objects like plants. If, for example, we want to put down a fence and a path and a house, those objects would be placed manually. There would never be, in this game, a desire to randomly place those types of objects.