

Procedural Cave Generator

If you have any questions, or suggestions for improvements, please email robert.wells@gandhigames.co.uk.

Attribution

Sprites by IMakeGames and available from <http://opengameart.org/content/lo-res-2d-platformer-graphics>

Public Variables

Adjust the caves properties in the GridManager class attached to the Environment object in the sample scene. Play around with these properties to control how the cave looks. It can take some trial and error before you create a cave that suits your purposes. Press the 'Generate' and 'Delete' buttons to create/destroy the grid in the editor.

- GridSize: the size and position of the cave.
- NumberOfTransitionSteps: during a transition walls are converted to floor tiles based on the 'WallsToFloorConversion' and floor tiles are converted to walls based on the 'FloorToWallConversion' variable.
- ChanceToBecomeWall: during the initialisation of the cave, all tiles are set to floor tiles. The tiles are then iterated over, and if a random number (between 0 and 1) is less than this variable then the tile is converted into a wall tile. A higher numbers means it more likely for the cave to have more wall tiles.
- FloorsToWallConversion and WallsToFloorConversion: used during a transition step to determine how often a wall is converted to a floor/background tile and vice versa. They define the number of neighbouring wall tiles that should be present for the tile to be converted. For example, if a wall tile has less than the number of neighbouring wall tiles defined by WallsToFloorConversion then it is converted to a background/floor tile.

There are a number of TexturePack classes also attached to the Environment object. One of these packs is chosen at random when generating a new level.

How It Works

Environments are created based on the principle of Cellular Automaton (see http://en.wikipedia.org/wiki/Cellular_automaton for more information)

Cave Generation (GridManager class):

1. Each tile is initialised to a wall or background tile based on the 'ChanceToBecomeWall' variable.
2. A number of transition steps are performed (the number is decided by 'NumberOfTransitionSteps'). In each transition every tile is checked:
 - If tile is a wall tile and has more than n number of neighbours that are also wall tiles then it is converted to a background tile. The number is defined by the variable 'WallsToBackgroundConversion'.
 - If tile is a background tile and has more than n number of neighbours that are wall tiles then it is converted to a wall tile. The number is defined by the variable 'BackgroundToWallConversion'.
3. After the transition steps it is likely that the environment will consist of a number of disconnected caverns. If the variable 'IsConnectedEnvironment' is true then the caverns will be connected by creating paths between them using A* search algorithm.

Key Classes

GridManager: Creates the environment. You can change the size and position of the grid (as well as the variables discussed in the previous section).

DetailsManager: Adds small details to floor tiles in the environment after it has been created. Only adds details if TexturePack has sprites in its 'Details' array. You can configure the maximum number of details that can be added and the chance of them being added to a floor tile.

TexturePack: Contains the tile and detail sprites.

PathManager: Used for path finding, GetShortestPath returns a list of nodes that will guide a character from one position to another.

ObjectPool: Prevents the creation and destruction of objects by disabling/enabling them instead, which is a lot quicker! see <https://unity3d.com/learn/tutorials/modules/beginner/live-training-archive/object-pooling> for a great introduction.

Adding your own texture packs

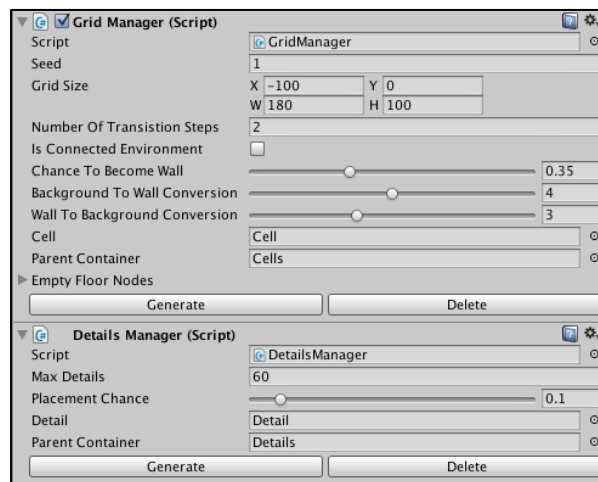
When adding your own tiles they'll be two situations:

1. All tiles will be the same size (ideal).
2. Tiles will differ slightly in size.

The cave generator will check the size of the first tile and then use that in all its calculations. If your tiles are slightly different sizes then manually overriding this can help prevent issues e.g. lines appearing between the tiles.

In the Utilities class in the Awake function on line 36, the tile size is being set manually (as the tiles in the example scene differ in size slightly). In your project if all tiles are the same size then comment out that line and their size will be calculated automatically. If they differ in size then change the values on line 36, which can usually be found with trial and error, if the tiles are too far apart then smaller numbers are required and vice versa.

Generate Environment In Editor



You can generate and destroy the environment in the editor using Generate and Delete buttons on the GridManager and DetailsManager class in the inspector (shown above). Using this method you can update the environment and then press the Generate button to see the changes in the editor.