

Cave Exploration Starter Kit

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

Attribution

The external resources used in this pack can be found below. They can be used in any game you create; however depending on individual licenses you may need to credit the author(s).

Background Audio by Icydemon: <http://opengameart.org/content/nocturnal>

Torch by Irwin: <http://opengameart.org/content/torch-0>

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

Enemy_3 by Blarget: <http://opengameart.org/content/bgi-asset-dump>

Sound Effects by bart: <http://opengameart.org/content/8-bit-platformer-sfx>

Important Classes

You'll find most of the important classes in the example scene attached to the 'Scripts' gameobject.

Director: entry point into the game.

GridManager: creates the caves, you can set the level size and position in the editor. Also sets the start position (where the player is spawned) and the end position (where the torch is spawned).

EnemySpawner: as the name suggests, spawns the enemies. There are three enemy spawners in the example scene, one for each enemy type. There are four options for enemy placement (selected in the editor for the script):

- Random Background Tile: a random tile is selected, no other constraints.
- Random Floor Tile: a random tile on the floor is selected.
- Random Tile With Rect space around: places enemy on a random tile that has no cave tile or other enemy located in a rectangular space around the tile. Suitable if you want your enemy to move without being obstructed.
- Random Tile With Circle Space Around: same as above, however a circular (rather than rectangular) check is performed.

CollectibleSpawner: spawns collectibles (surprise surprise!). The collectible wall limit defines the number of surrounding cave wall tiles. For example, it is set to 3 in the scene so a collectible will only be spawned in a tile with at least 3 adjacent wall tiles. This generally places them in slightly harder to reach areas.

PathManager: used for pathfinding, the method GetShortestPath returns a list of cells that can be traversed by a character to reach the destination tile.

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.

TexturePack: Contains a list of sprites used to create the cave. If more than one texture pack is present then a pack is chosen at random.

Utilities: contains a number of helper methods used by various scripts. You can toggle 'IsDebug' to turn on/of debug logging.

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 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, they can usually be found with trial and error, if the tiles are too far apart then smaller numbers are required and vice versa.

Random Seed

Tile, collectible, enemy, and player placement all depend on the Seed (a number used by the random generator). The initial seed for the first level is set in the Director script. When the player reaches the end of the level and new seed is generated randomly.