

# Cavern Collapse **Editor** Tutorial

Welcome! Here I'll be covering everything you'll need to know when using the **dataset editor** for the game.



## What is a **sample**?

A sample is a file containing a **one screen level** that is created and edited in the Dataset Editor. The game includes a set of these by default, which cannot be edited, and also allows you to build your own which can be used in level generation.

## What is a **dataset**?

A dataset is simply a **folder containing a group of samples**. The game includes a default dataset which

cannot be edited and it also allows you to build your own datasets. You might want to have separate datasets to experiment with different build styles and see how it affects the levels that get generated for you in-game.

## What can the editor do?

- Place tile (**Left click**)
- Delete tile (**Right click**)
- Switch block types (**A/D / Click button**)
- Cycle tile variants (**Left click on tile to cycle**)
  
- Undo (**Click button / Ctrl + Z**)
- Redo (**Click button / Ctrl + Y**)
  
- Save sample (**Click button / Ctrl + S**)
- Delete sample/dataset (**Hold button down**)
  
- Load dataset (**Click on “dataset: [ dataset x ]”**)
- Load sample (**Click on “sample: [ x / x ]”**)
  
- New sample (**Click “new” button next to sample**)
- New dataset (**Click “new” button next to dataset**)

## General tips for building samples?

1. Sketch out a general **outline** using solid walls.
2. Add in some **one-way platforms** and **ladders** in interesting places.
3. Place **coins** and **spikes** as desired.

4. Decorate with **vines** from the ceilings, **torches** on the ground, **signs** and various **foliage**.
5. Go through the sample again, **adding variation** to the insides of walls by **cycling** to their other variant, and doing the same for **foliage** and **signs**, etc.
6. You could even create little pockets in walls by deleting them and adding little enclosed rooms, **it's up to you!**

## Why does a **procedurally generated** game need a level editor?

Unlike many other procedurally generated games, Cavern Collapse uses a unique algorithm called **Wave Function Collapse** to generate its stages.

This algorithm **analyses examples** of the desired outcome (the samples this editor allows you to create) and produces **rules** and **probabilities** about the way that tiles **connect** to generate a stage. By giving you as the player access to this algorithm, I hope to provide a unique experience that allows you to collaborate with the level generator.

## One last thing...

There are lots of **tile combinations**, so the more samples you create, the better the end result will be! The algorithm gets a better understanding of the way that you design levels with more **data**.