# Level Editor Guide
# Quick version

FYM team

## Introduction

Ruin Raiders allows people to build their own levels in game. To do so, a complete level editor is offered to build with a simple drag-and-drop tool, inspired by Mario Maker. A person can then save their level and load it to play it. This also allows very fast and efficient testing for the game makers. In a later version, a full GUI will allow people to load, test, distribute and share levels with others, with a full leaderboard (with high scores). For now, this level editor allows one to play custom levels and share them online quickly. A valid level is stored in a json file that can be safely distributed only.

## Saving and loading levels

Every tile has a specific ID that has to be unique. In a later version of the level editor, a check will be done to ensure no two tiles have the same ID. In any event, a player will not have inner access to the game engine and its assets, such as the prefabs.

Levels are stored using a json file. It is possible to edit a level directly. One could choose to open a .json file in a text editor and build the level by typing the corresponding tile ID.

When a level is saved, it is saved in an array of number like this:

**1 0 0 1000 0 50 1000 9**

**100 100 100 100 100 100**

This, for instance, would spawn a player level one, standing on a row of normal tile of tilemap #1. There would be two basic tree background effect in front of him, as well as the first enemy, level 1 (difficulty adjustment).

Spaces mark the end and the beginning of a new entry (a new tile). A return key changes the row. The level will be built according to regular specific cations of the game. Shall some numbers be left empty, they will be replaced by the last element in the row, i.e.:

**1 0 0 0 0 0 0 1**

**100**

would create:

**1 0 0 0 0 0 0 1**

**100 100 100 100 100 100 100 100**

A level must contain the follow elements to be deemed valid:

1. A player start (several different player spawns are possible, see full guide).
2. A level end.

## Number ID list

Every tile is a prefab, and referred to by its ID. Even different tile has a different ID used to save and load levels.

This is a quick explanation of the most commonly used tile ID

### 1 à 100 – Base tiles

0 – Empty element

1- Character spawn #0
2- Level end
3- Blocked door
4- Movable Obstacle
5- Pressure platform (open door with ID3; for multiple doors and pressure platforms, see below)
6- Collectible #1 (coin)
7- Collectible #2 (gem)
8- Key
9- Locked door (openable with ID8; for multiple doors and keys, see below)
10- Level end
11- Kept for developers
12- Kept for developers
13- Kept for developers

14- Kept for developers
15- Kept for developers
16- Kept for developers
17- Kept for developers
18- Kept for developers
19- Kept for developers
20- Locked door #0 (can be unlocked by key #0; more than one key #0 can be collected
...
30- Key #0
...
40- Locked door by movable obstacle #0
...
50- Movable obstacle #0
...
60- Pressure plate #0 (must have movable obstacle #0 on top of it)
...
70- Pressure plate any movable object (any movable obstacle will do)
...
80- Rarer Collectible #0...
...
90- Single use item #0
...

## 100 to 1000: Tilemaps

100- Tilemap 0, tile #0
101- Tilemap 1, tile #1
...
200- Tilemap 1, tile #0
...
1000- Enemy 0, variation 0.
1001- Enemy 0, variation 1 (harder)
...
1100- Enemy 1, variation 0.
1101 – Enemy 1, variation 0 (harder)
...

## 2000 to 9999: Other elements & level modifiers

7000- Misc decoration item number.
...
8000- Empty object. Changes background image to random background #1
...
9000 – Empty object. Other level modifier (music, speed, etc.)

## Final considerations

As one might expect, a lot of those ID are kept empty of purpose. Many of the items listed above will not be implemented in the game and will be kept for an eventual full release and to allow a greater variety of puzzles.

Likewise, it is not advisable to pre-select the difficulty of a monster (1199 would be enemy 1 at its highest difficulty) as it is preferable (in the spirit of the game) to let the game manage the level's difficulty. Eventually, full worlds (and quite possibly level packs) can will be able to be developed and linked with this editor, allowing people to design a complete game.