



# LEVEL GENERATOR 2D

# Level Generator 2D

## Quick Start



---

# 1. Add a Level Controller to your Level

Add a **Level Controller** to your **Level's** GameObject. Doing so will automatically add a **Level** component to the GameObject. Alternatively, a **Level** can be added manually, or will automatically be created when a **Room** is made using the menu (*GameObject/Level Generator 2D/Room*, or *Ctrl-Shift-R*).

There are three options for **Level Controllers**:

The **SimpleLevelController** is the simplest version. It only contains the bare minimum to run the **Level Generator 2D**. Attaching this component will have your **Level** automatically generate based on the Camera's position.

The **LevelController** is the standard version. It provides added functionality in the inspector to adjust as needed. With this controller, the random seed used, the loading rules, and the folders loaded from can be changed.

The third option is to create your own **Level Controller**. For larger games, this is probably the best option. There is an extensive [API](#) available to customize the controller any way you want. It is recommended to follow the tutorial for the standard **LevelController** to get a better idea of how the **Level Generator 2D** works.

# 2. Create a Room

A **Room** can be created by going into the menu (*GameObject/Level Generator 2D/Room*), or using the hotkey (*Ctrl-Shift-R*).

The **Level Generator 2D** using its own coordinate system, called global points. The **Room's** global point is its position within that coordinate system. This is what you can use to change a **Room's** position.

Its size determines how big you want the **Room** to be.

GameObjects can be saved to a **Room** by being their child. Locking the children will force all of its children to remain within the bounds of the **Room** in the editor.

The weight determines how likely this **Room** is to appear over other **Rooms**. Prioritized folders will always take priority over **Room** weight.

---

---

**Rooms** have **Doors** which are used to connect them to other **Rooms**. **Doors** should be placed at entrances and exits of the **Room**.

If you want this **Room** to be loaded by the **Level Controller** opposed to being placed manually, place it in the Resources folder as a prefab. Subfolders may matter, depending on if you are using the priority folders from the standard **LevelController**. Delete the **Room** if you do not want it loaded in a static location.

### 3. Add Spawners to your Room (optional)

**Spawners** allow **GameObjects** to randomly be spawned within a **Room**. To add a **Spawner**, select the **Room** and in the menu go to *GameObject/Level Generator 2D/Spawner* (or *Ctrl-Shift-W*). A **Spawner** will randomly spawn the specified **GameObjects** at the position.

The spawn chance determines the likelihood that a **GameObject** will spawn. Spawn on start will cause a **Spawner** to trigger when play is pressed, opposed to triggering the **Spawner** from code. The spawn count allows for a random number of spawns chances between that range.

Add a **GameObject** to the **Spawner** to allow that **GameObject** to be spawned there. Multiple **GameObjects** can be added to a single **Spawner**, where the weight determines the likelihood of a particular **GameObject** being spawned more commonly than another **GameObject**.

### 4. Repeat Steps 2 and 3

For as many **Rooms** and **Spawners** as you want, repeat steps 2 and 3, placing the **Rooms** in the Resources folder. After you are finished, make sure you have a **Room** in the Scene from which other **Rooms** can be loaded.

---

---

## 5. Press Play

If you have a **Level**, a **Level Controller**, a starting **Room**, and other **Rooms** in your Resources folder, you are finished! Just press play, and the **Rooms** should be loaded during runtime and pseudo-randomly placed.

- Extra Help

Check out our [video tutorials](#) or contact us directly at [levelgenerator2d@evorlor.com](mailto:levelgenerator2d@evorlor.com) for support or feature requests!

