



Foreword

Thank you for choosing Low Poly Dungeon Generator! I'm thrilled to have you as a user and sincerely appreciate your purchase. I've put a lot of effort into creating a plugin that enhances your Unity experience by making the creation of low poly dungeons and labyrinths quick and easy.

If you encounter any issues or have any questions, please don't hesitate to reach out to me via email at 'info.past12pm@gmail.com'. I'm committed to providing excellent support and am eager to resolve any concerns you may have before you leave a review.

Your feedback is invaluable and helps me improve my assets and offer a better experience for all users.

Additionally, I invite you to check out my other assets available in the Unity Asset Store. I offer a variety of tools and utilities designed to enhance your development process and bring your creative visions to life.

Thank you once again for your support, and happy developing!

1. Introduction
2. Getting Started
 - 2.1 Installation Instructions
 - 2.2 Opening the Plugin:
 - 2.3 Workspace Customization:
3. Usage Guide
 - 3.1 Getting Started
4. Features
5. Creating and customizing
 - 5.1 Modules
 - 5.2 Script adjustments
 - 5.3 Shader
6. Support

1. Introduction

The Low Poly Dungeon Generator is a versatile and efficient tool designed for generating fully randomized dungeons within seconds. With modular prefabs and intuitive controls, this tool provides endless possibilities for crafting unique and engaging dungeon layouts. Whether you're designing a roguelike game, a level for exploration, or a procedural adventure map, this generator streamlines the creation process, offering flexibility and ease of use.

The tool is particularly useful in scenarios where time efficiency and variety are critical. With its ability to generate dungeons using pre-designed modules and configurable spawn rates, it eliminates repetitive design tasks while ensuring high-quality results. It also allows users to incorporate their own modular designs, seamlessly integrating them into the system with clear guidelines and compatibility.

2. Getting Started

2.1 Installation Instructions

Download and Import:

Download Low Poly Dungeon Generator from the Unity Asset Store. Once downloaded, Unity will automatically import the asset into your project.

Accessing the Plugin:

After importing, you can find Low Poly Dungeon Generator under the "Tools" menu in the top bar of Unity.

2.2 Opening the Plugin:

Navigate to "Tools" in the top bar, then select Low Poly Dungeon Generator to open the plugin window.

2.3 Workspace Customization:

The plugin window functions like any typical Unity window. You can drag and drop it to any location within your Unity workspace to fit your workflow preferences.

Initial Configuration:

The plugin comes with preset configurations which build a foundation you can build upon, allowing you to start generating immediately.

Self-Explanatory Interface:

The interface is designed to be intuitive and user-friendly. You can begin generating and customizing right away without further configuration.

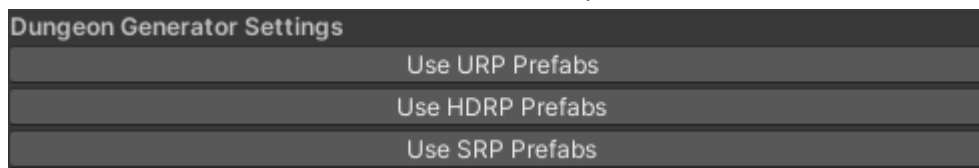
3. Usage Guide

To get the best results with the Low Poly Dungeon Generator, it's essential to follow some best practices and understand how the script operates.

3.1 Getting Started

1. Prepare Your Workspace

Import the tool into your Unity project. By default, the tool checks for the RP you are using and adjusts its models it will be using initially. If it is not finding the right RP, you are able to choose the correct RP at the top of the tool.



2. Consistency

Ensure your prefabs adhere to the modular naming conventions and setup guidelines (discussed in "5. Creating and customizing").

Explore the provided example prefabs depending on your used RP to understand their structure and integration in these paths:

"Assets/Low Poly Dungeon Generator/Prefabs/HDRP/Pathbuilding"

"Assets/Low Poly Dungeon Generator/Prefabs/URP/Pathbuilding"

"Assets/Low Poly Dungeon Generator/Prefabs/SRP/Pathbuilding"

3. Generate a Dungeon

Navigate to the editor window of the tool.

Adjust settings like dungeon size, module spawn rates, and prefab options (details in the "4. Features" section below).

Click the "Generate" button at the bottom to create a fully randomized dungeon.

4. Iterate and Customize

If the generated layout doesn't meet your needs, regenerate the dungeon or tweak specific settings for better results.

Review the placement of modules, ensuring they align with your intended gameplay flow and aesthetic.

4. Features

The Low Poly Dungeon Generator includes several configurable options in the editor window to streamline the dungeon creation process. Here's an overview of the main features:

1. Dungeon Size

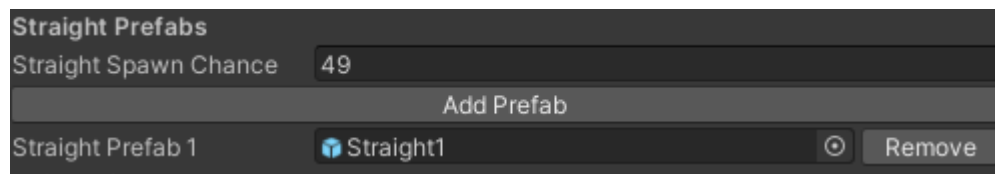
Defines the maximum length and breadth of the generated dungeon.

2. Starting Height

Define the starting height of the dungeon generation process. The dungeon will stay on the same height if no "Stair" prefabs are being spawned.

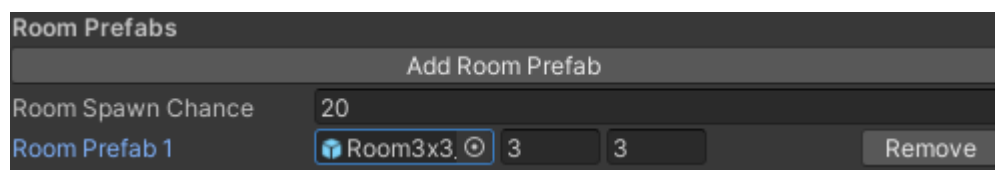
3. Spawn Chances and prefab arrays

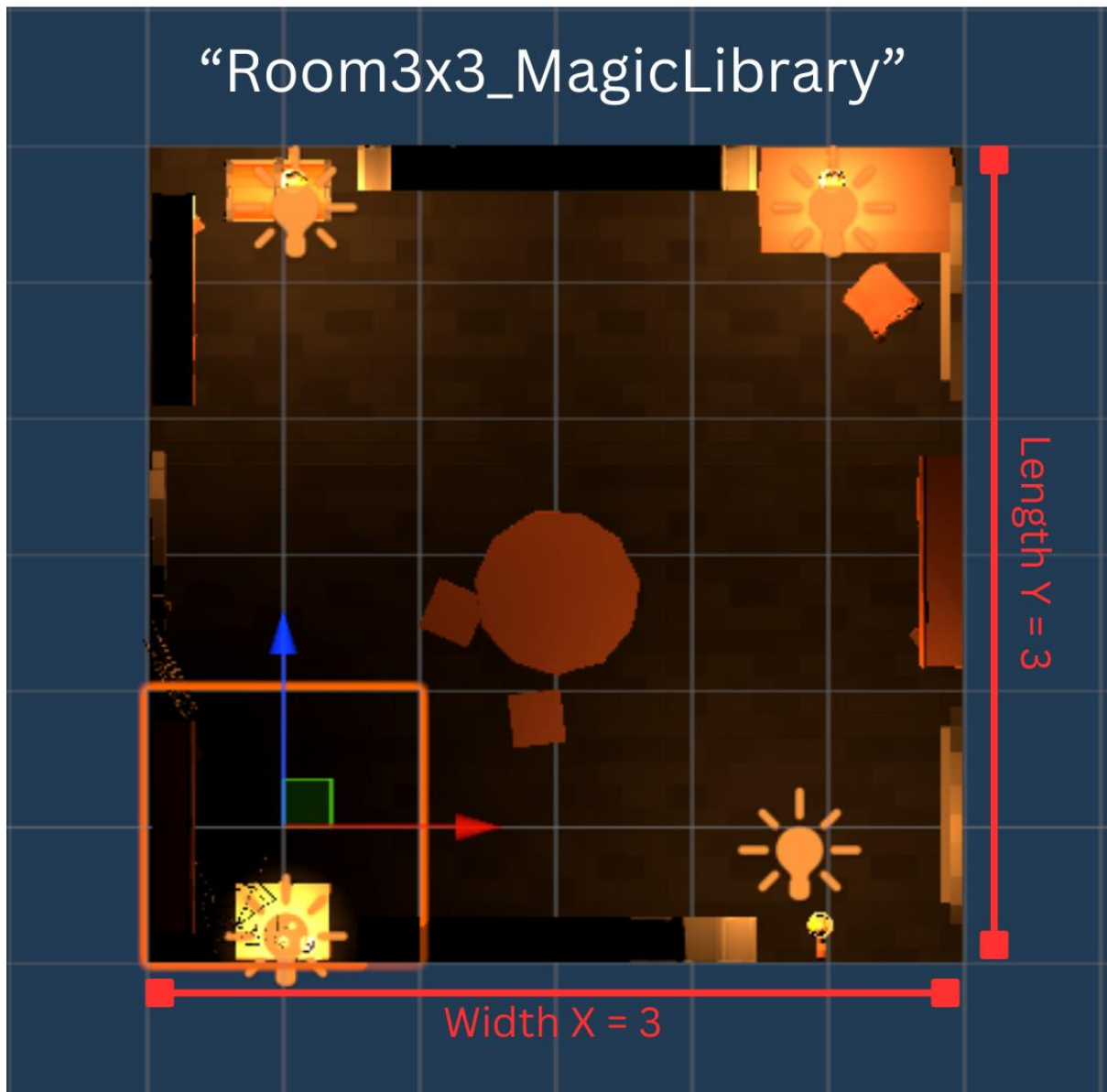
- Adjust probabilities for different module types (e.g., straight paths, rooms, stairs).
- Add new prefab slot "Add Prefab" to the specific array which will then be put into the pool of randomized prefab picking. These have to follow the setup guidelines. More under "5. Creating and customizing".
- Select specific prefabs for inclusion in the dungeon (e.g., Straight, Left, Right, Splits and Room prefabs).
- Remove a prefab with the "Remove" button and exclude it from the dungeon generation process.



4. Room Sizes

To ensure smooth generation of paths, the sizes of the rooms in length and width must be specified next to the added room module. This helps the tool calculate occupied cells and prevent overlapping modules.





5. Creating and customizing

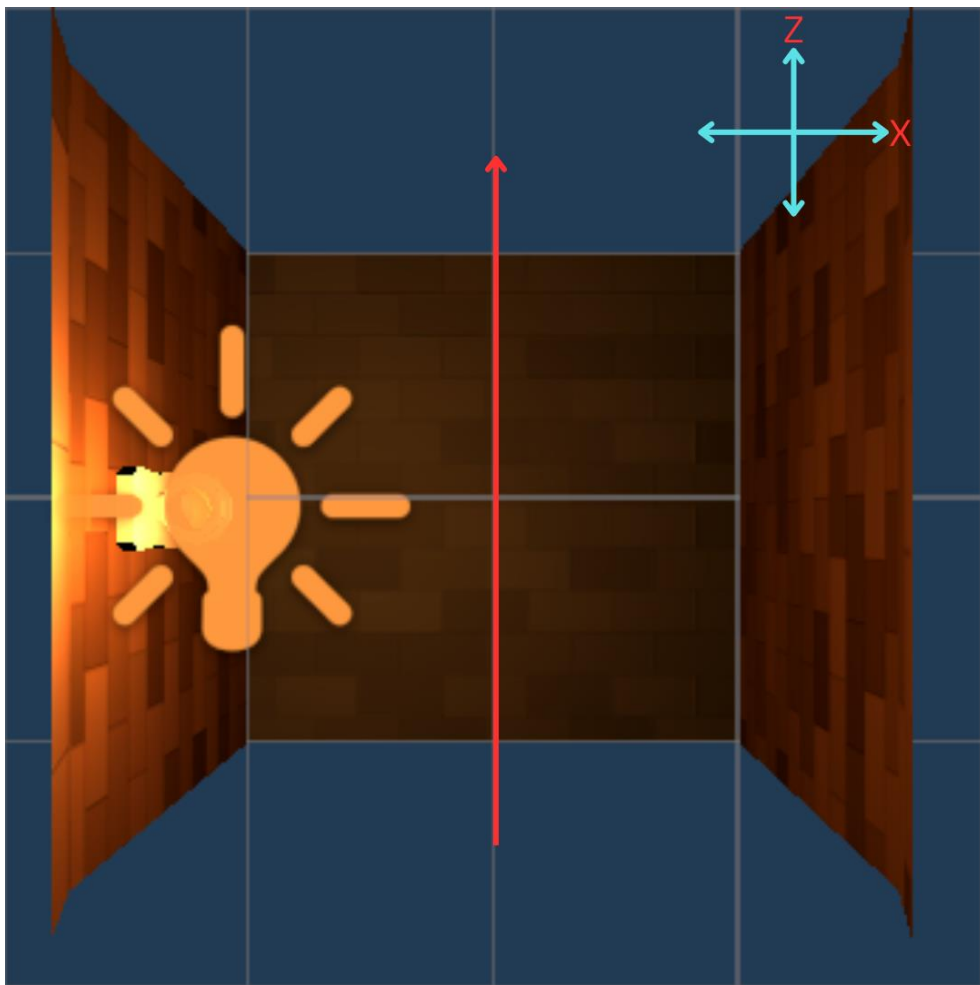
The **Low Poly Dungeon Generator** allows users to create and customize their own modules to seamlessly integrate with the tool. This customization ensures that generated dungeons align with your unique gameplay and artistic vision. Additionally, users can modify the script to customize spawn weights, placement rules, and dungeon dimensions. These adjustments enable control over the frequency of specific modules, the overall dungeon layout, and the logic used to position prefabs, offering both creative freedom and precise control.

5.1 Modules

5.1.1 Different module types

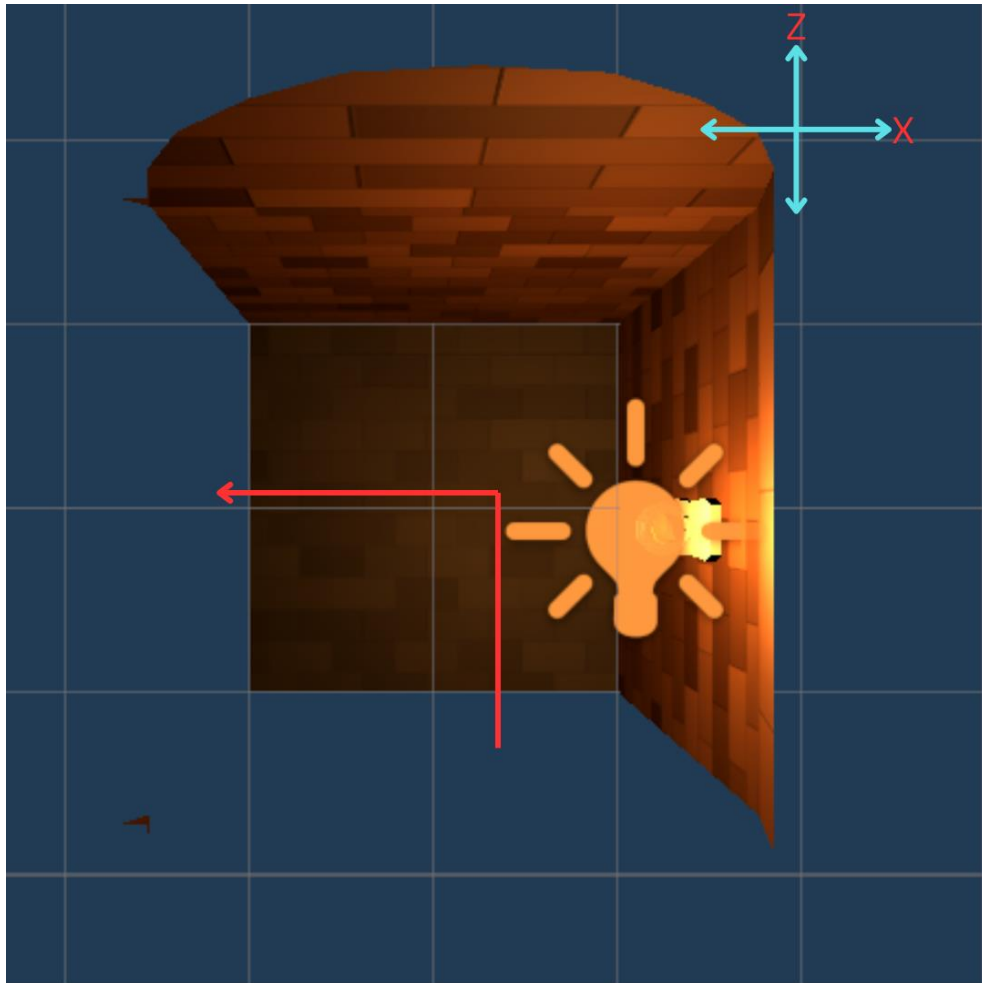
1. Straight Modules

Straight modules serve as the backbone of your dungeon, providing simple linear pathways. They are used to create hallways or continuous corridors, and their straightforward design ensures that the dungeon flows seamlessly from one point to another. These modules are perfect for connecting other module types and maintaining structural balance.



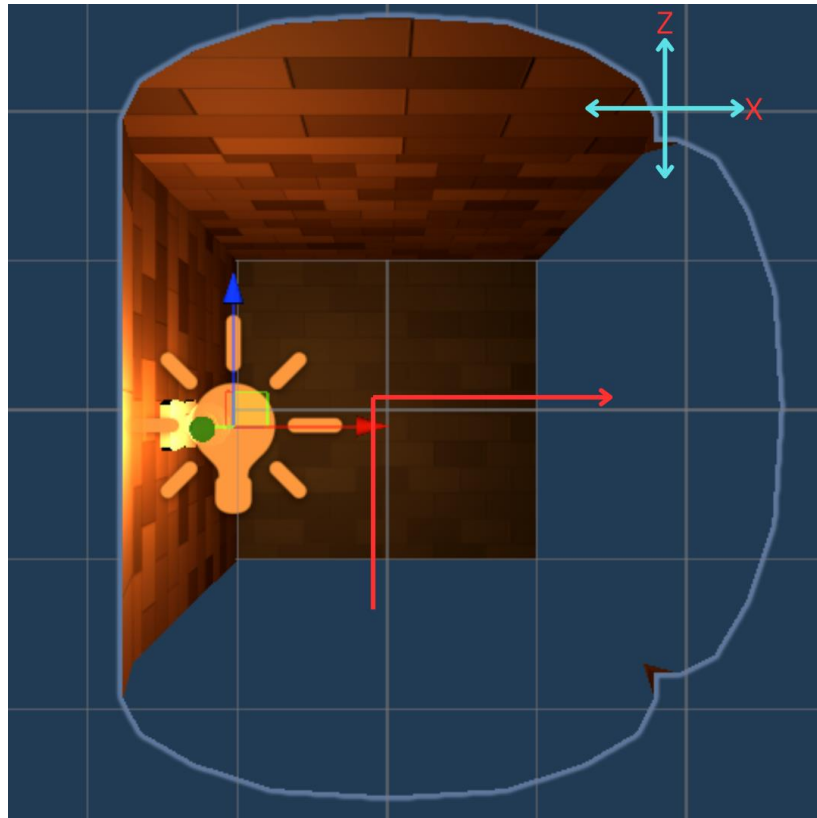
2. Left Modules

Left modules introduce a 90-degree turn to the dungeon's layout, bending pathways to the left. These modules help break up the monotony of straight corridors and provide opportunities for more dynamic, maze-like structures.



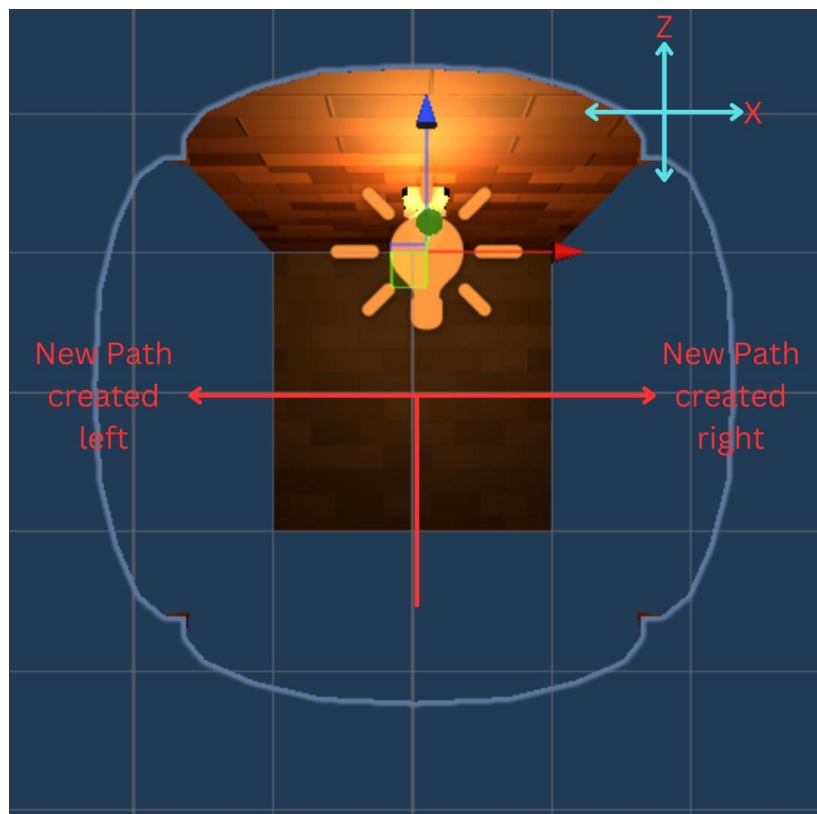
3. Right Modules

Similar to left modules, right modules add a 90-degree turn, but this time to the right. By combining left and right modules, users can create winding paths and intricate layouts that keep players engaged and encourage exploration.



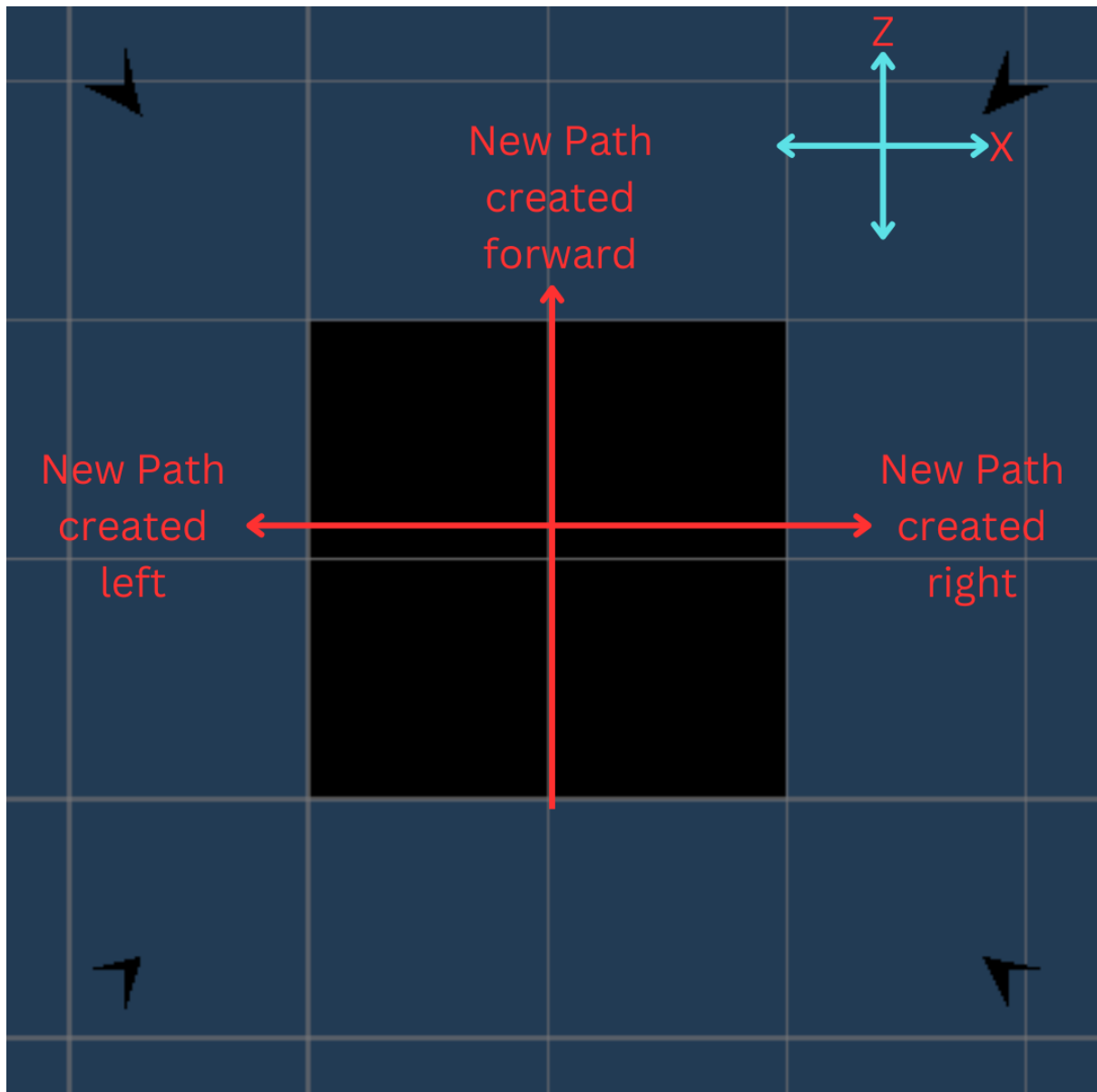
4. Double Split Modules

Double split modules create a branching point with two separate paths emerging from a single connection. These modules are ideal for adding decision-making moments in the dungeon, giving players a sense of choice and variety as they explore.



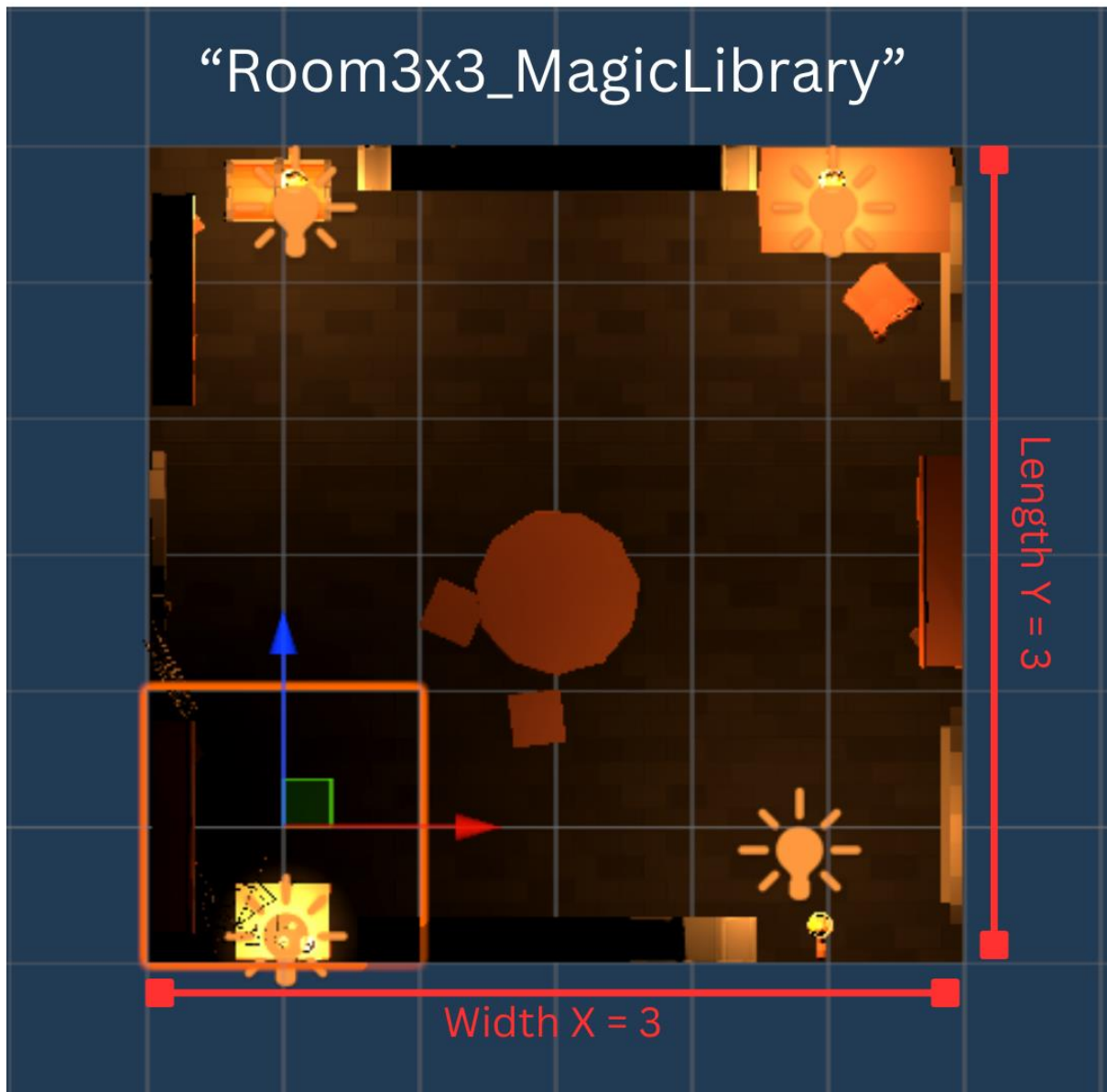
5. Triple Split Modules

Triple split modules provide three branching paths from a single origin, offering even greater complexity and opportunities for exploration. These modules are particularly effective for introducing highly dynamic layouts and encouraging players to thoroughly investigate their surroundings.



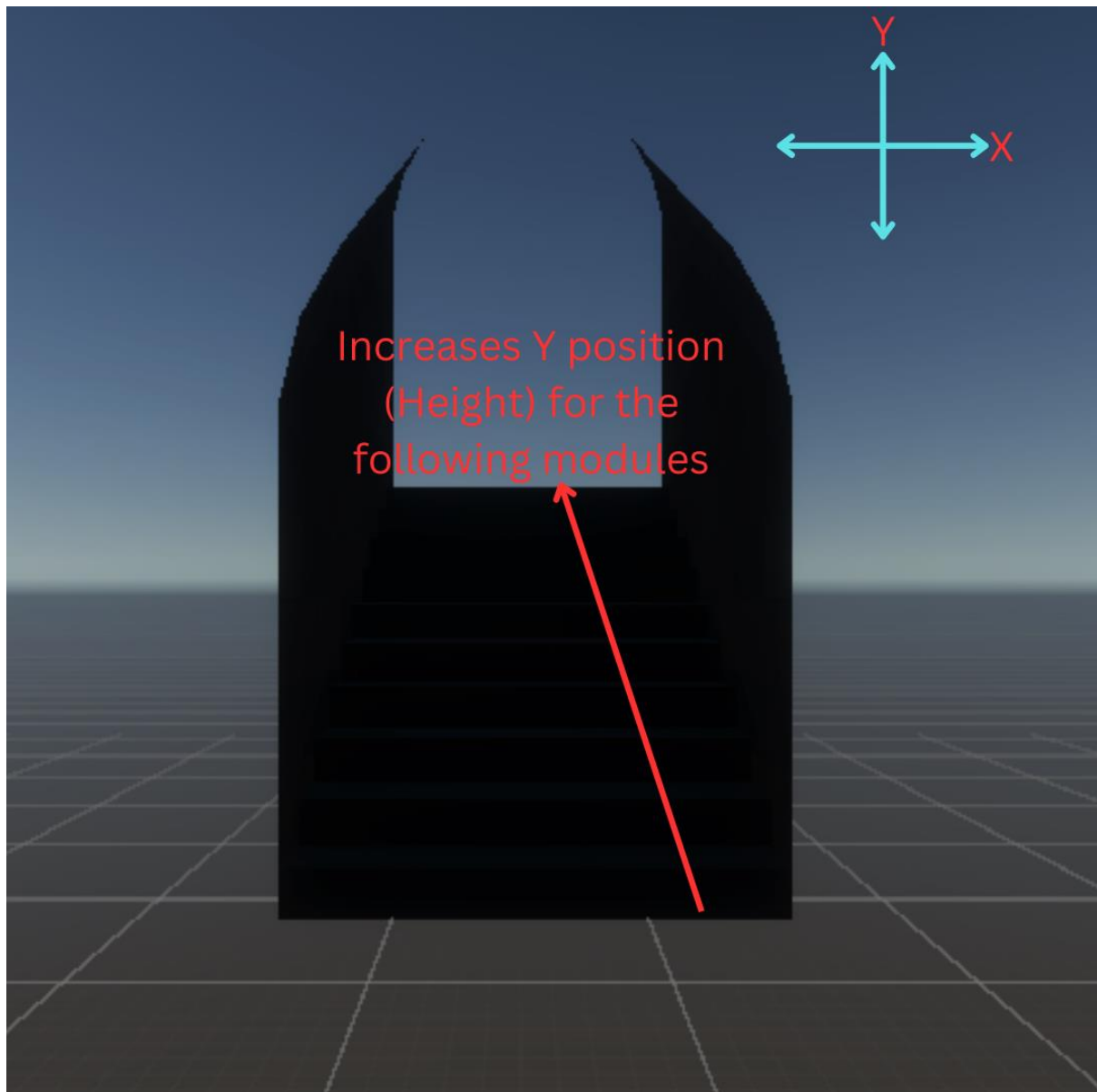
6. Room Modules

Room modules act as focal points within the dungeon, often serving as hubs, encounter areas, or zones for treasure and events. Their larger size and customizable shapes make them versatile, allowing for a wide range of gameplay scenarios. Users can define their dimensions and internal layouts to suit their needs.



7. Stair Module

The stair module introduces verticality to the dungeon by connecting different height levels. This module allows for multi-floor dungeons, enhancing exploration with a sense of depth and variety. Stairs can also signify transitions to new areas or difficulty levels.



5.1.2 Creating your own modules – setup instructions

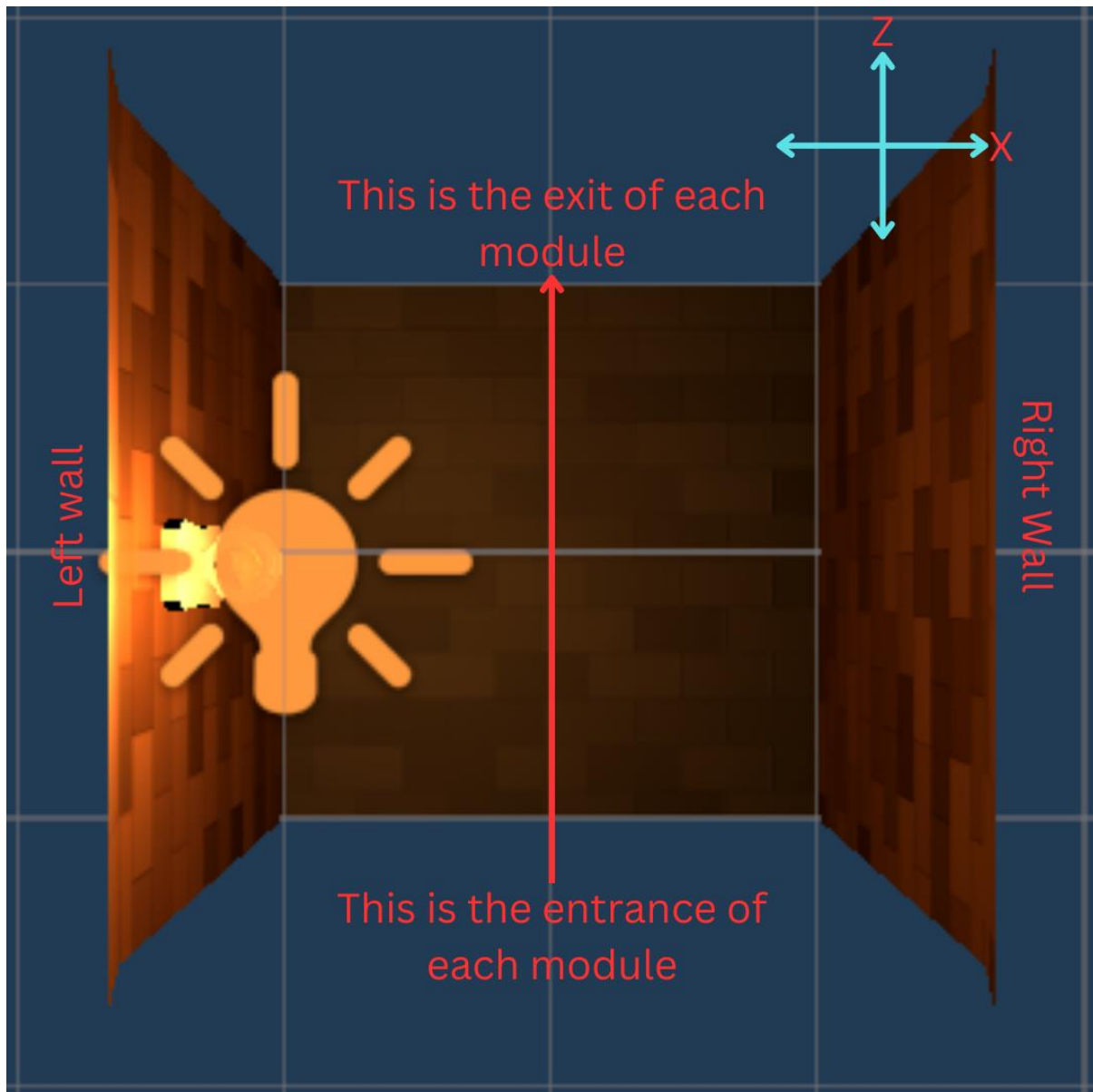
Pathway modules

When creating your own pathbuilding modules, it is highly recommended that you begin by copying and modifying the existing prefabs included in the tool and tweak them around. This allows you to gain an understanding of how they are structured and ensures consistency in your designs. If you wish to create your own Straight, Left, Right, or Split modules from scratch, here are the key considerations:

1. Module Dimensions and Orientation

Each module must adhere to a grid size of 1x1, meaning the width and length of the module are exactly one grid unit.

The module must be oriented so that the entrance is aligned with the negative Z-axis. This ensures proper alignment when the modules are placed in sequence during dungeon generation.



2. Floor and Wall Setup

Walls are constructed using planes rather than full rectangles with width. This approach has a major advantage:

Planes use fewer polygons, as they consist of only one face instead of six, which is the case with a rectangle. This significantly reduces the polygon count for large dungeons, making them much more efficient, particularly for mobile games or large-scale environments.

Use the Blender standard plane (can be found at this path: Assets/Low Poly Dungeon Generator/Models/Pathbuilding/Floor.fbx) for the floor and walls, ensuring consistent sizing and alignment.

The walls are constructed using a consistent game object and are positioned based on the module type:

For Straight modules, walls are placed along the left and right edges of the module.

For other module types (Left, Right, Splits), wall placement depends on the intended shape and path direction, ensuring they guide players effectively.

For example, walls should not be placed at the positive Z-axis for Straight modules, as this is where the path continues forward.

3. Material Rendering and Grid Alignment

Since modules are aligned on a grid, it's crucial to ensure that wall materials only render the front face (disable double-sided rendering).

Double-sided materials can cause z-fighting, a visual artifact where Unity struggles to determine which overlapping surface to display, resulting in flickering walls.

To address this limitation:

Duplicate each wall game object and rotate the duplicate 180 degrees on the Y-axis. This creates a back-facing wall to prevent visibility issues when other paths or rooms intersect with the backside of the wall.

4. Ceiling Setup

Add a ceiling to your module depending on its design and function. The ceiling ensures the dungeon feels enclosed and immersive, matching the style of the other prefabs in the tool.

Room modules

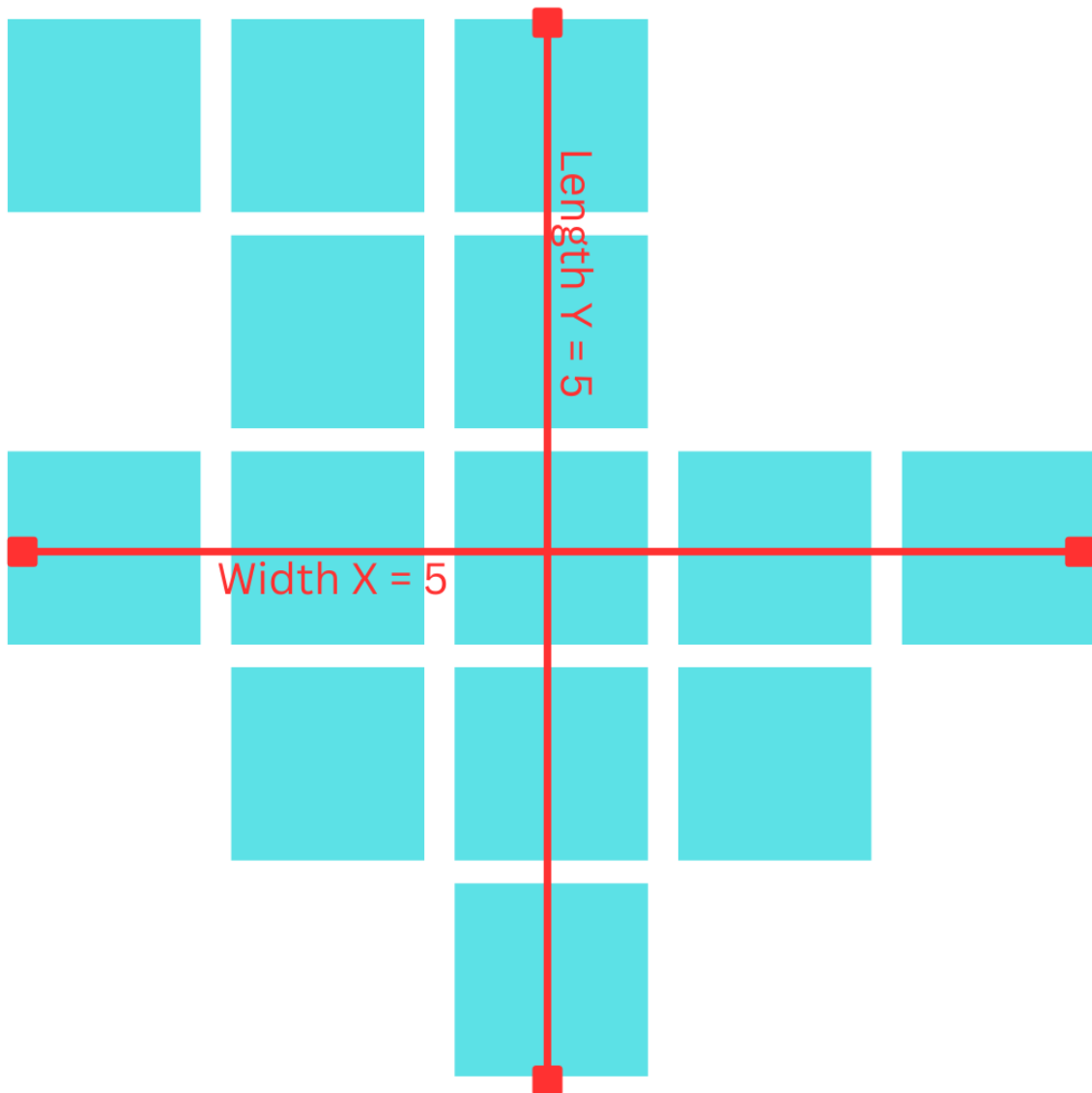
1. Room Sizes and Grid Logic

Each room module must have its width and length specified in the editor window when adding it as a room prefab. This ensures that the generator understands how many grid cells the room occupies.

Rooms can have flexible dimensions and do not have to be square. For example, a room can be 3x3, 5x7, or even 9x20.

Regardless of the shape of the actual room design, you must input the furthest width and length values into the editor. For example:

A 5x5 room prefab that only partially uses the grid space must still be marked as 5x5 in the editor to ensure accurate placement.

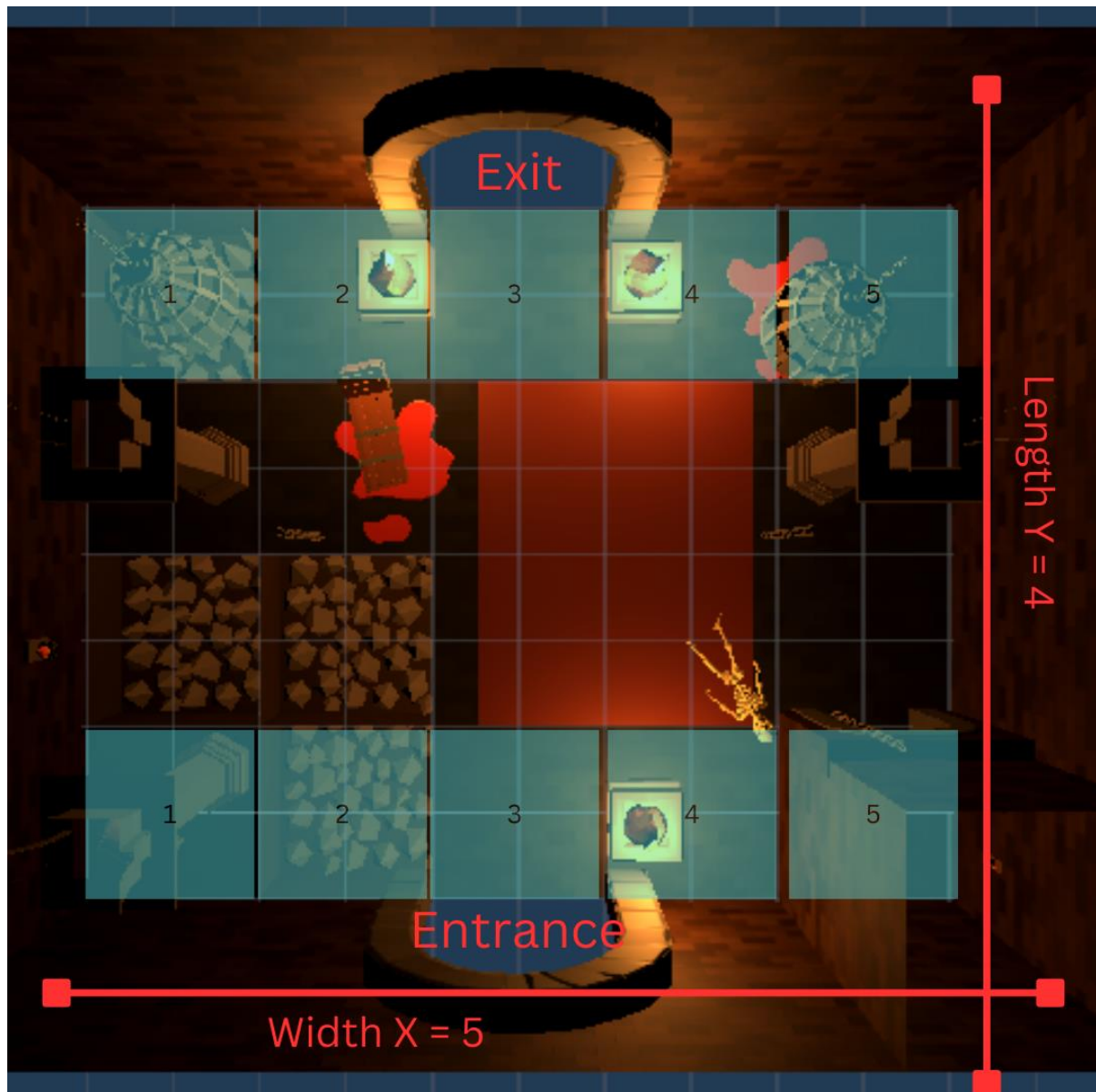


2. Entrance and Exit Placement

The entrance and exit of a room must always be located at the center of the row where the connection occurs.

For example, in a 5x4 room, the entrance would align with the third grid cell (middle cell) of the row. Similarly, the exit would also align with the middle grid cell on the opposite side.

Proper alignment of entrances and exits ensures that paths can seamlessly connect to and from the room, preventing misaligned or blocked corridors.



3. Wall and Ceiling Logic

Walls and ceilings for rooms follow the same guidelines as path modules:

Use planes for walls instead of rectangles with width to save polygons and optimize performance.

Ensure that wall materials only render the front face, and duplicate and rotate walls 180 degrees on the Y-axis to prevent visibility issues when paths or other rooms intersect with the room's walls.

The ceiling should fully cover the room to provide a cohesive and immersive dungeon design.

By adhering to these guidelines, your custom room modules will integrate smoothly with the dungeon generator, occupying the correct grid cells, aligning properly with paths, and maintaining visual and functional consistency with the rest of the dungeon. These considerations help ensure the generated dungeon remains both visually appealing and technically sound.

5.2 Script adjustments

For advanced users who want to further customize the dungeon generation process, the script provides flexibility in key areas. Below are three possible adjustments you can make to tailor the tool to your specific needs.

5.2.1 Maximum Path Length

By default, the generator limits the length of new paths after a split to 10 modules. This ensures that each branch of the dungeon has a manageable size and avoids excessive generation. However, you can change this value if you want shorter or longer paths after a split.

Location in the Script:

Inside the `GenerateDungeon()` method at the end, look for the following line:

```
maxPathLength = 10;
```

This adjustment is useful when you want to control the density and complexity of the dungeon layout.

5.2.2 Grid Cell Size

The current script generates dungeons based on a fixed grid size, with each grid cell representing a 2x2 unit space in Unity's world coordinates. If you want to change the size of grid cells (e.g., to fit larger or smaller prefabs), you'll need to modify how the placement positions are calculated.

Location in the Script:

Inside the PlacePrefab() method, look for the following line:

```
Vector3 worldPosition = new Vector3(placementPosition.x * 2, localHeight, placementPosition.y * 2);
```

How to Adjust:

Modify the * 2 multipliers to reflect your desired grid cell size. For example:

```
Vector3 worldPosition = new Vector3(placementPosition.x * 3, localHeight, placementPosition.y * 3); // For larger 3x3 grid cells  
Vector3 worldPosition = new Vector3(placementPosition.x * 1, localHeight, placementPosition.y * 1); // For smaller 1x1 grid cells
```

This change ensures the modules are placed according to your custom grid size, but keep in mind that prefabs and their dimensions must align with the new grid scale.

5.2.3 Height of “Stairs”

The current implementation increases the height of the dungeon by 2 units when a stair module is placed. If you are using a custom stair model with a different height, you’ll need to adjust this value to match your model’s dimensions.

Location in the Script:

Inside the PlacePrefab() method, look for the following line:

```
localHeight += 2; // Increase the height for stairs
```

This adjustment ensures that the placement of modules after the stairs aligns correctly with the new height.

5.3 Shader

5.3.1 Crystal Shader

The CrystalShader simulates the appearance of a crystal with smooth color transitions and specular highlights. It includes properties for two colors, fade height, and shininess. The shader operates in two passes for forward base and additive lighting. It uses vertex, geometry, and fragment shaders to calculate the fade effect based on the height of vertices, blending between two colors. It also computes lighting, including ambient, diffuse, and specular reflections, and applies fog effects. This creates a crystal-like material with dynamic lighting and smooth color transitions.

6. Support

For assistance with the Low Poly Dungeon Generator, please use the following contact methods:

Email: Directly reach out to me at info.past12pm@gmail.com. Provide detailed information about your issue or inquiry to help me address your needs effectively.

Website Contact: You can also contact me through my website. Visit the contact page for additional ways to get in touch.

I am dedicated to helping you with any questions or issues you may have and will do my best to provide prompt and helpful responses.