FPS Microgame

Level Assets Manual

Technical breakdown of the assets and how they can be used in your level

Compatible with Unity 2020 LTS
Version 2.0

The Grid

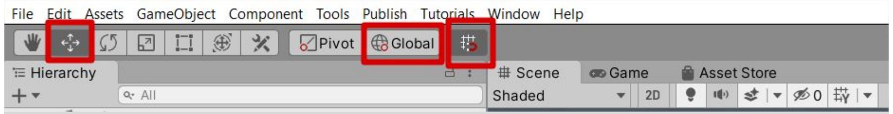
What is the Grid?

In order to use the level assets effectively, you need to know how to use the basic functions* of the the **Grid**.

The **Grid** is a Unity Editor feature that enables you to build up your level with speed and precision by providing an adjustable grid — you can think of it as “graph paper” in a 3D environment.

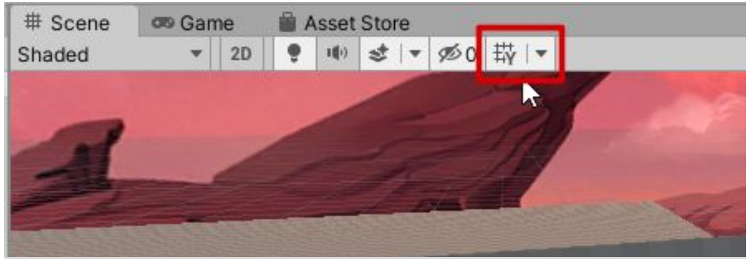
You will use it to:

- Align your assets in the scene by *snapping* them to the grid
- Build “grey box levels” with ProBuilder
- Eliminate mesh gaps and light leaks


File Edit Assets GameObject Component Tools Publish Tutorials Window Help
Pivot
Global
Hierarchy
All
Scene
Game
Asset Store
Shaded
2D
0

The Grid’s **Snap toggle** is available when the **Move** tool is enabled, and the Tool Handle is set to **Global.**

Note how the **Snap toggle** (grid and magnet icon) is toggled ON (down state) in the image.



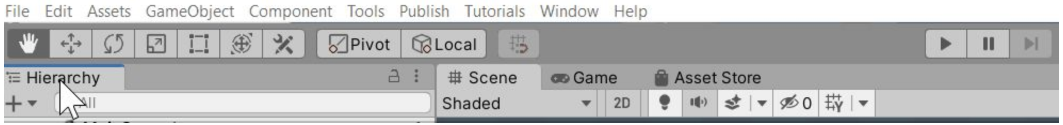
The **Grid** visibility toggle is located in the **Scene** window

*: This document only covers some basic functions of the **Grid**. To explore more options of the Grid, see the complete documentation here: https://docs.unity3d.com/Manual/GridSnapping.html

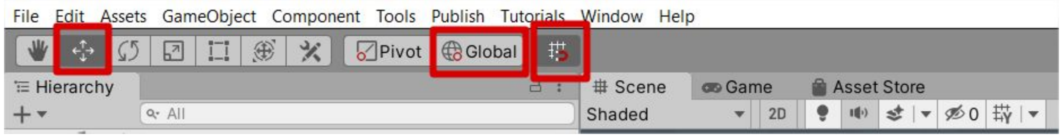
Grid Basics — Turn on Snapping

Turn on the ability to have your assets “snap to the grid”

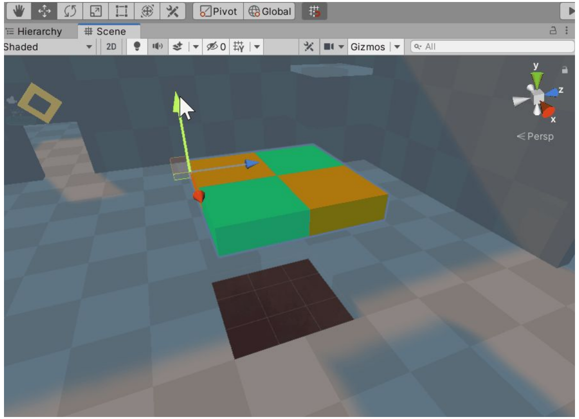
- Select the **Move** tool
- Set the Tool Handle to **Global**
- Set the **Snap toggle** to **ON** (down state)


File Edit Assets GameObject Component Tools Publish Tutorials Window Help
Pivot
Local
Hierarchy
Scene
Game
Asset Store
Shaded
2D

It should look like this:

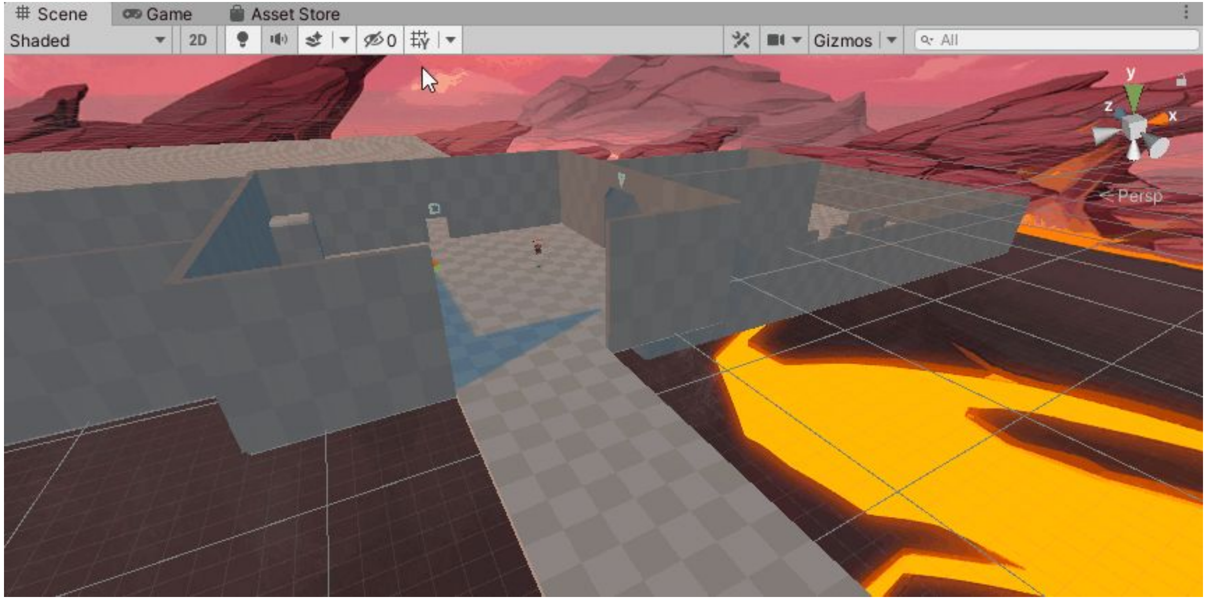

File Edit Assets GameObject Component Tools Publish Tutorials Window Help
Pivot
Global
Hierarchy
All
Scene
Game
Asset Store
Shaded
2D

Test it out by moving some of your assets in the level.
You should see them ‘pop’ in the appropriate increments



Grid Basics — Turn on Grid visibility

- The **Grid visibility** UI is located in the **Scene** window
- Clicking the **Grid** button, will toggle the Grid visual in the scene *on* or *off*

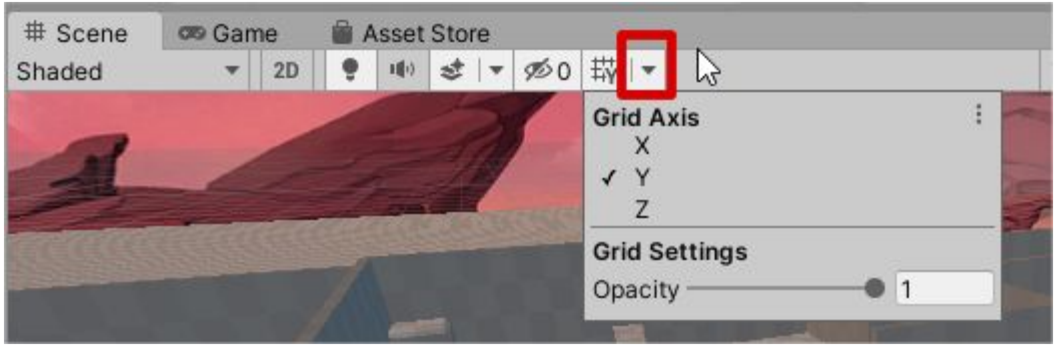


Grid Basics — Visibility settings

- Clicking the **down arrow** button to the right of the Grid button will reveal some basic options:

    - **Grid Axis** allows you to see the grid should on a given axis (X, Y or Z)

    - **Grid Settings > Opacity** allows you to set the opacity of the grid lines

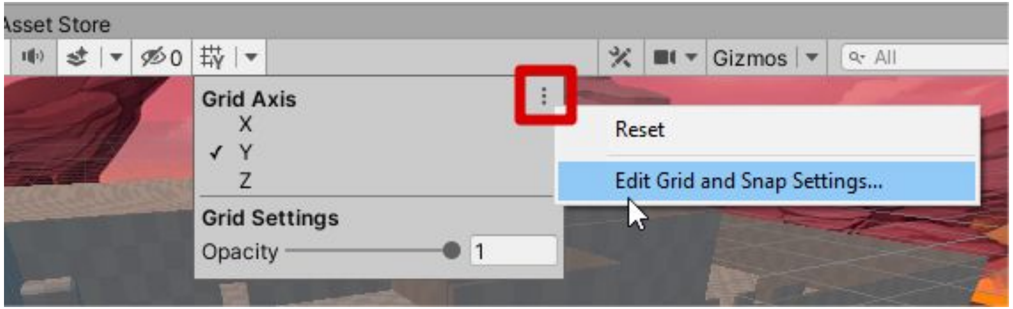


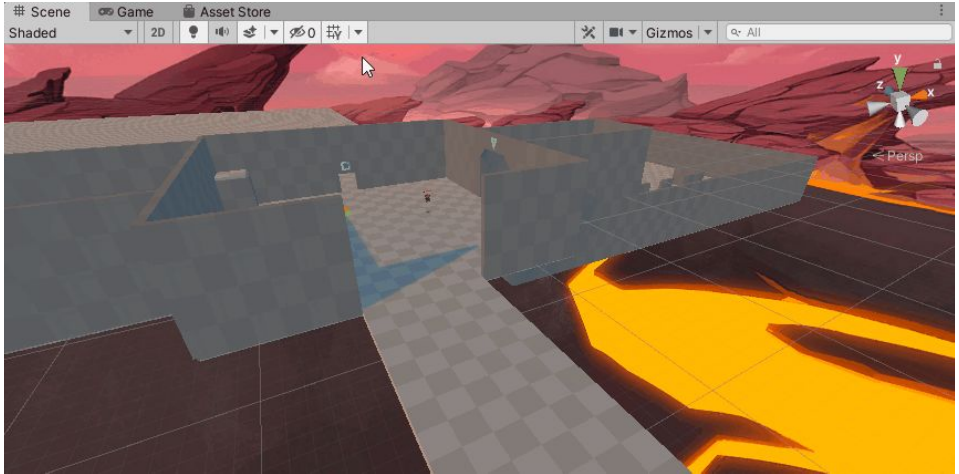
Grid Axis
X
✓ Y
Z

Grid Settings
Opacity 1

Grid and Snap Settings — Opening the window

- Clicking the **“three dots”** button to the right of the Grid’s drop down menu will expose additional options:
    - **Reset** will reset the settings in the drop down menu to the *default* settings (Grid Axis: Y; Grid Opacity: 0.5)
    - **Edit Grid and Snap Settings…** will open the **Grid and Snap** window





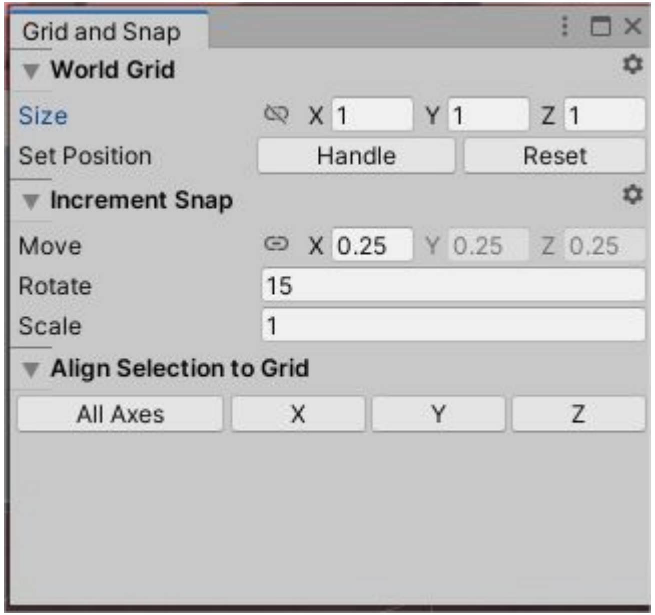
Grid and Snap Settings

- By default the Grid and Snap settings are as displayed in this image:

  - **World Grid > Size** is set to “1” meter
    - With Grid Snapping ON, objects will move by the unit (in meters) set in the fields

  - **Increment Snap > Move** is set to “0.25” meters
    - With holding the “Ctrl” + **Move** tool, objects will move by the unit (in meters) set in the Move fields

  - **Align Selection to Grid** buttons will align a selected asset to the grid along a given axis (**X**, **Y**, or **Z**); or in the case of **All Axes,** the closest grid point



Grid and Snap

World Grid

Size X 1 Y 1 Z 1

Set Position Handle Reset

Increment Snap

Move X 0.25 Y 0.25 Z 0.25

Rotate 15

Scale 1

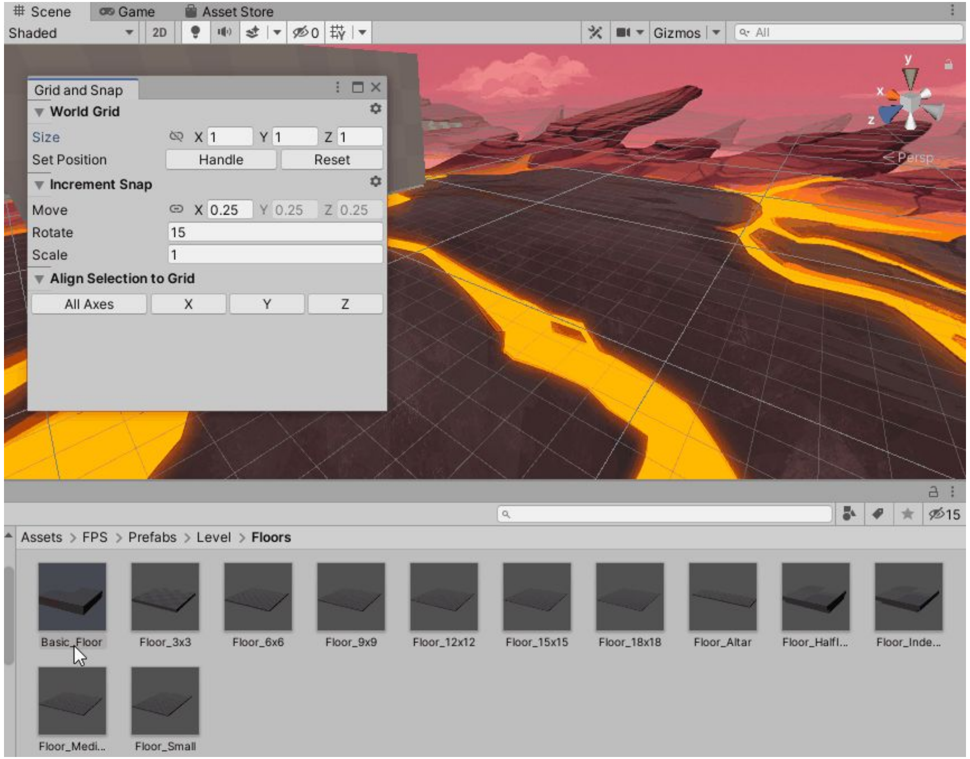
Align Selection to Grid

How to *align* an asset to the grid

In this demonstration, a “Basic_Floor” prefab is dragged into the scene, and then aligned to the closed grid point.

Try it out:

- Drag and drop “Basic_Floor” prefab into the scene
- Click **All Axes** and the asset will align itself to the closest grid point



Grid and Snap

World Grid

Size X 1 Y 1 Z 1

Set Position: Handle | Reset

Increment Snap

Move X 0.25 Y 0.25 Z 0.25

Rotate 15

Scale 1

Align Selection to Grid

All Axes | X | Y | Z

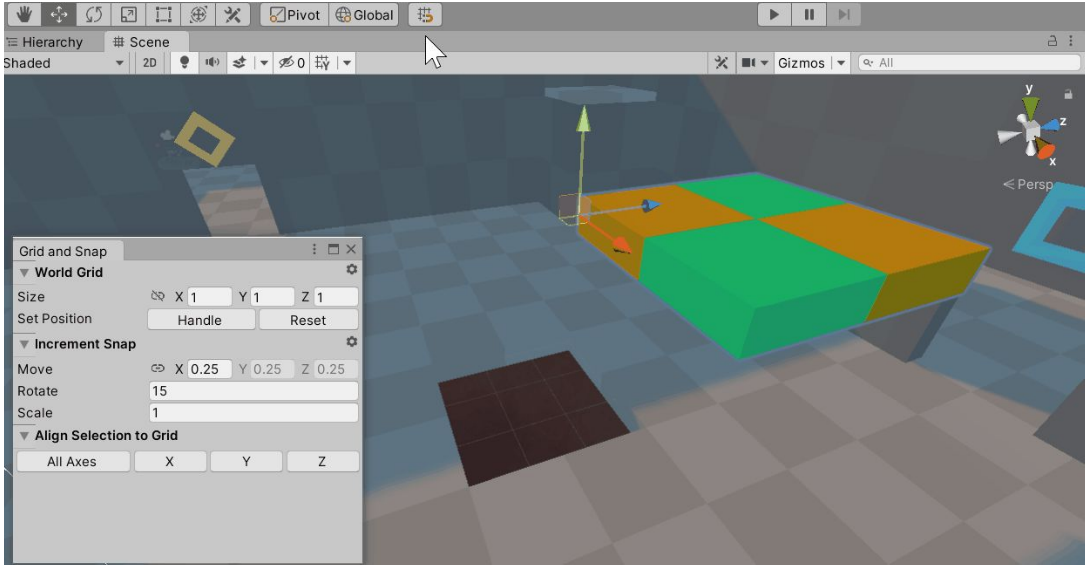
Assets > FPS > Prefabs > Level > Floors

How to *snap* an asset the grid

In this demonstration, a modified “Basic_Floor” prefab is aligned, and moved into position

Try it out:

- With **Grid snapping OFF**, Move an asset from the FPS Microgame level out of position
- In the **Grid and Snap** window, click **All Axes** to align the asset to the closest grid point
- Turn **Grid snapping ON**
- Select the **Move** tool
- Then move the asset into its spot



Grid and Snap

World Grid

Size X 1 Y 1 Z 1

Set Position: Handle | Reset

Increment Snap

Move X 0.25 Y 0.25 Z 0.25

Rotate 15

Scale 1

Align Selection to Grid

All Axes | X | Y | Z

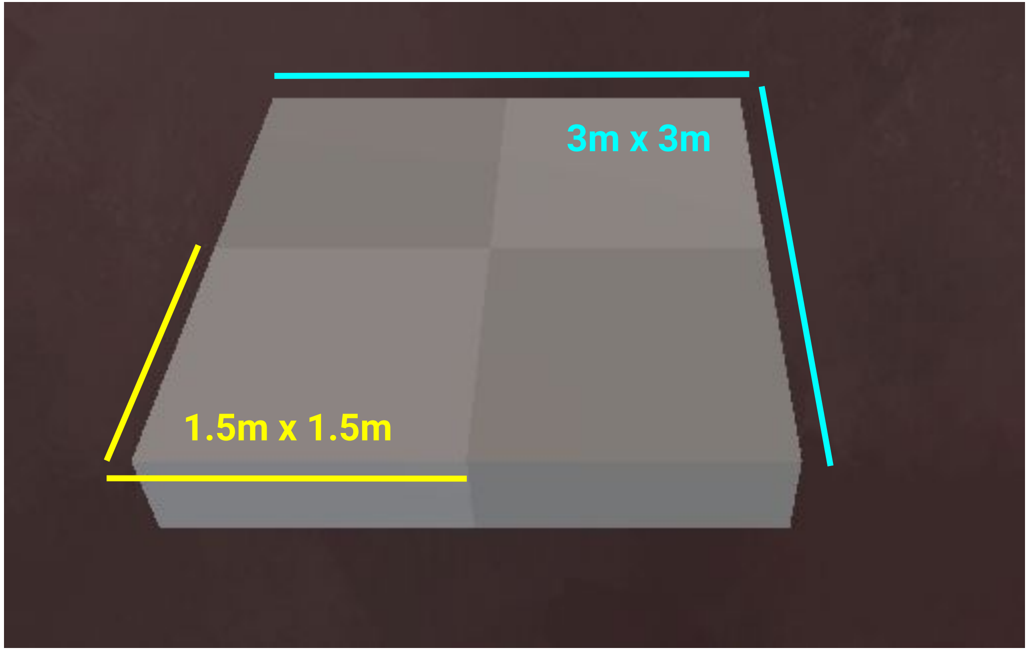
Floors

Basic_Floor

The *Basic_Floor* prefab is comprised of 4 cubes.

The cube's dimensions are 1.5m x 0.5m x 1.5m

Putting together 4 cubes in a 2 by 2 fashion allows the floor to be exactly 3m x 0.5 x 3m

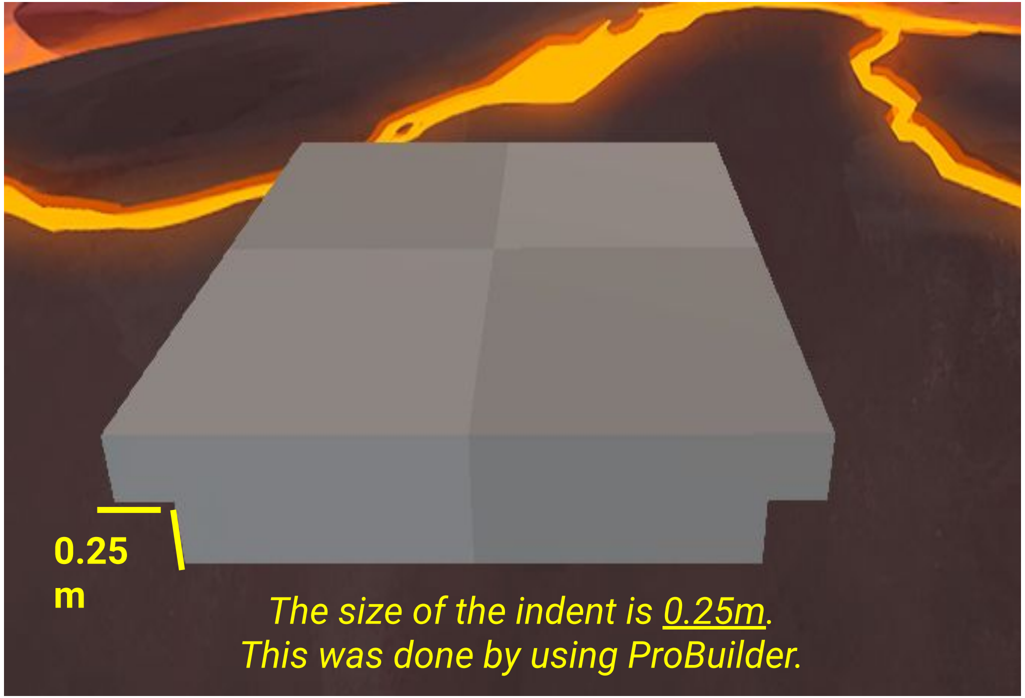


Floor_HalfIndented

The following prefab is constructed by adjusting the *Basic_Floor* prefab.

The *Floor_HalfIndented* prefab uses notches on both sides of the floor to help avoid “Z-Fighting”.

Z-Fighting occurs when there’s geometry fighting to be on the same position as one another.



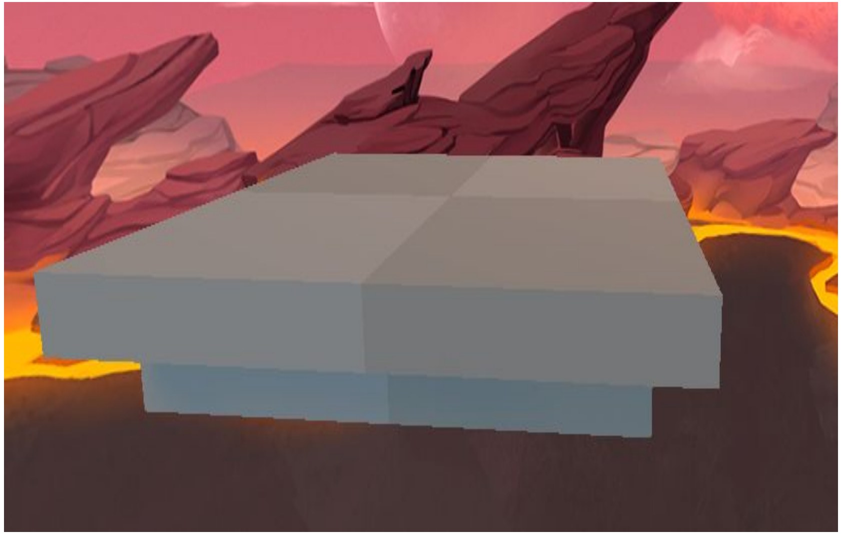
The size of the indent is 0.25m.
This was done by using ProBuilder.

Floor_Indented

Similar to the *Floor_HalfIndented* this prefab expands the notches on all 4 sides of the object.

This was done to further eliminate objects from Z-Fighting with one another.

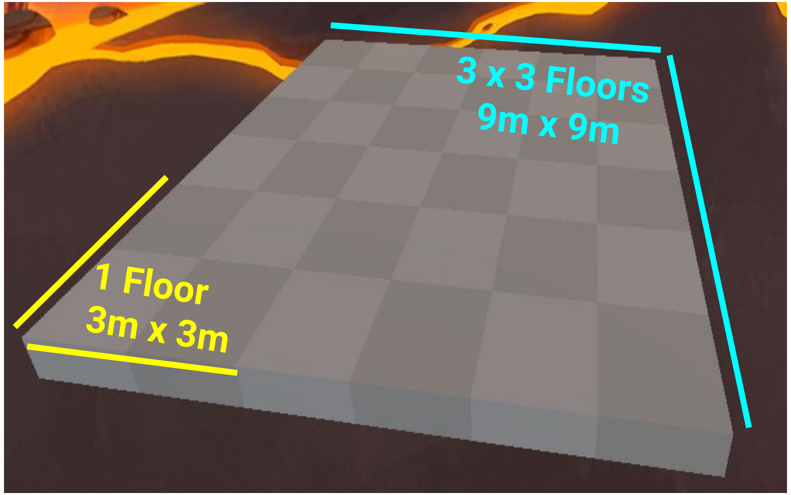
Z-Fighting occurs when there's geometry fighting to be on the same position as one another.



Floor_#x#

The *Floor_#x#* prefabs use the *Basic_Floor* prefab the amount of times indicated in the name.

For example, the *Floor_3x3* prefab uses 9 *Basic_Floor* prefabs to create a single floor prefab.

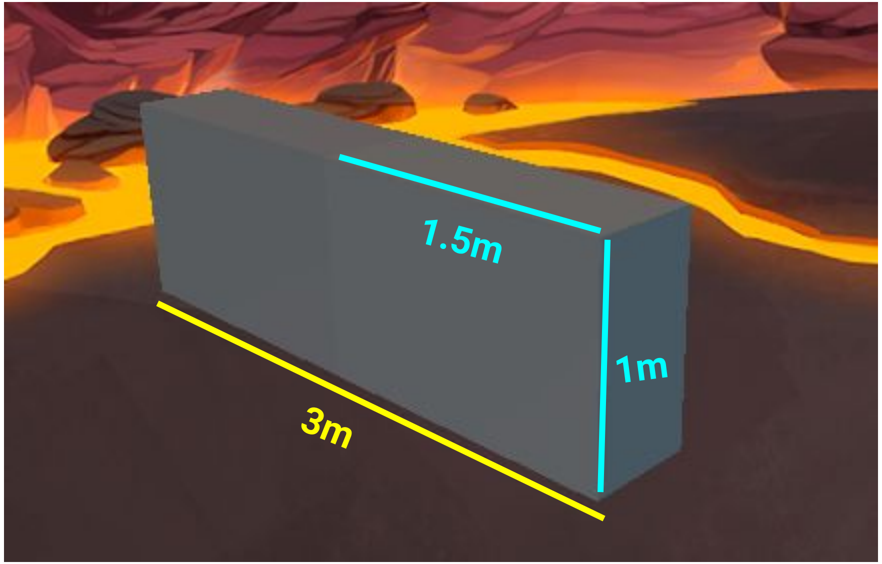


Walls

Wall_1m

The *Wall_1m* prefab uses 2 cubes to create a checkered pattern with a 1m height.

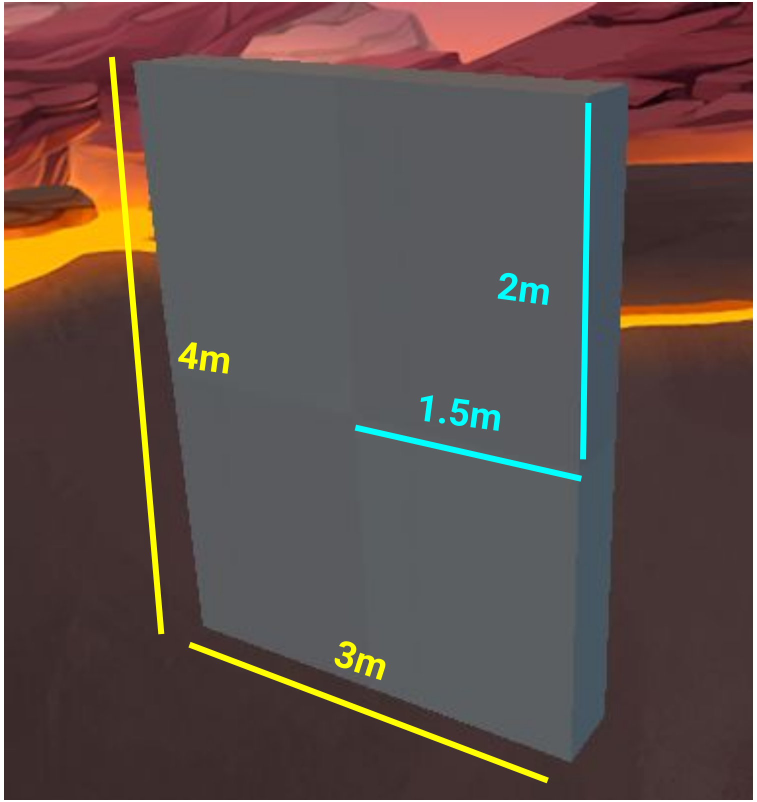
This smaller wall makes it easier to cover up small sections within your level.



Wall_4m

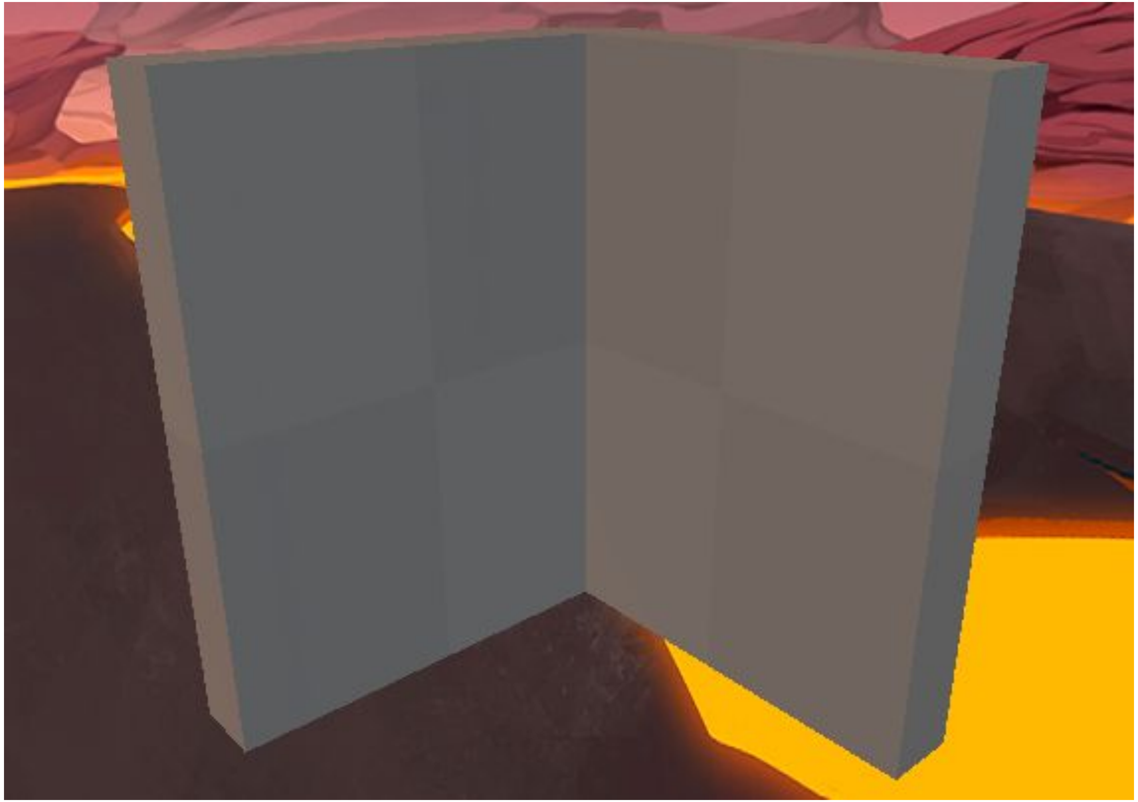
The *Wall_4m* prefab is the generic wall object that is used throughout almost every prefab created.

Much like the *Basic_Floor* prefab, it uses four cubes, whose dimensions are adjusted into a rectangle.



Wall_Corner

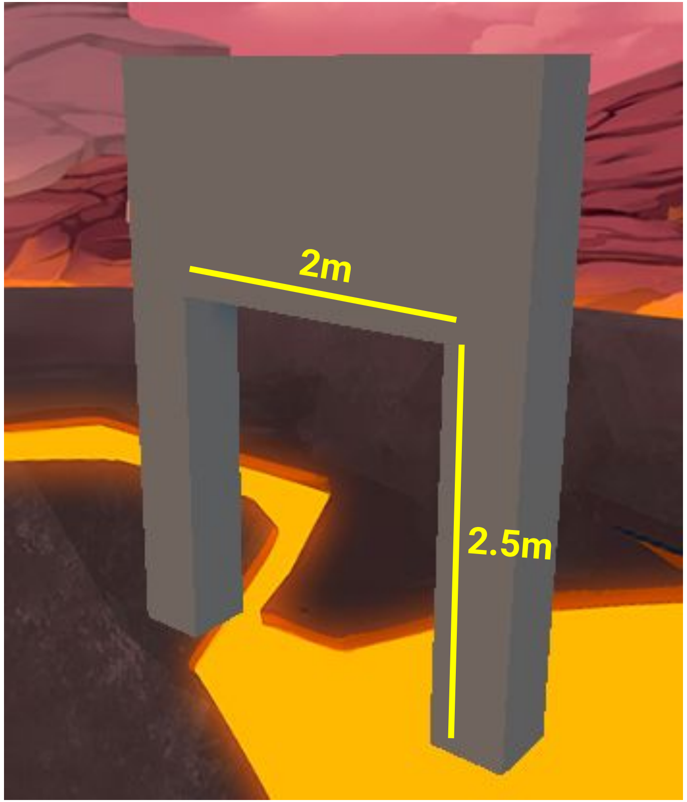
The *Wall_Corner* prefab uses two *Wall_4m* prefabs. This makes it simpler and modular when creating rooms or sections within a level that require corners.



Wall_Door

The *Wall_Door* prefab uses the same dimensions of a *Wall_4m* prefab.

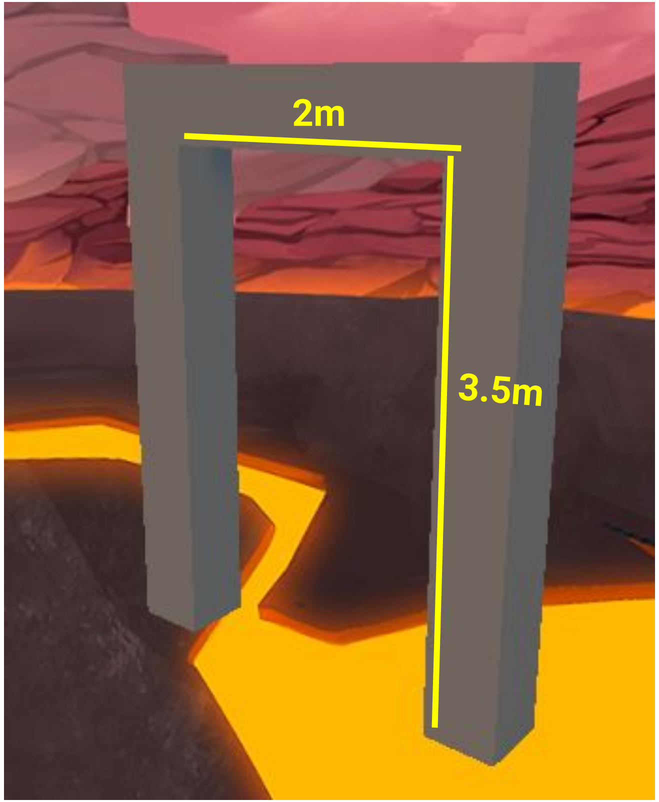
This prefab can be placed beside any wall prefab or with any floor prefab to allow an opening or exit within your level.



Wall_DoorLarge

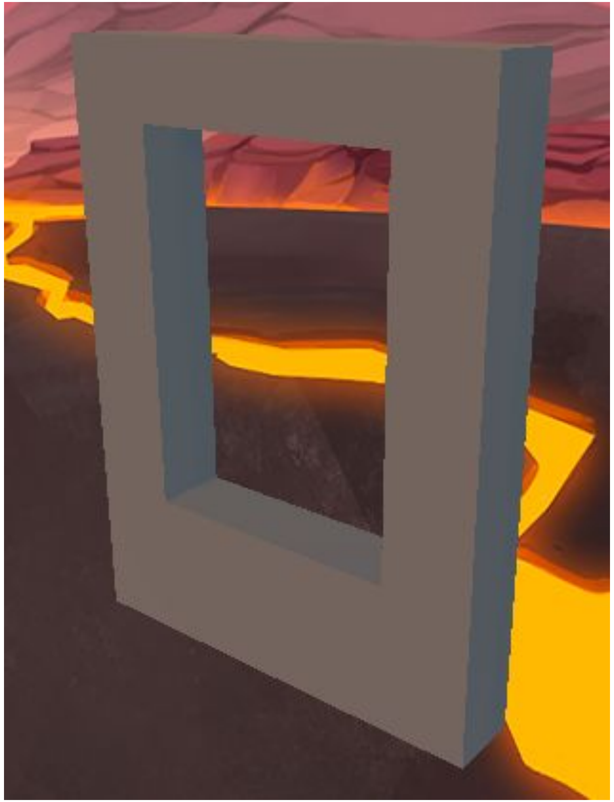
The *Wall_DoorLarge* prefab is exactly like the regular *Wall_Door* prefab.

The main difference is the height of the opening within the wall.



Wall_Window

The *Wall_Window* prefab was created to allow you to place an opening or showcase other sections of your level to players.

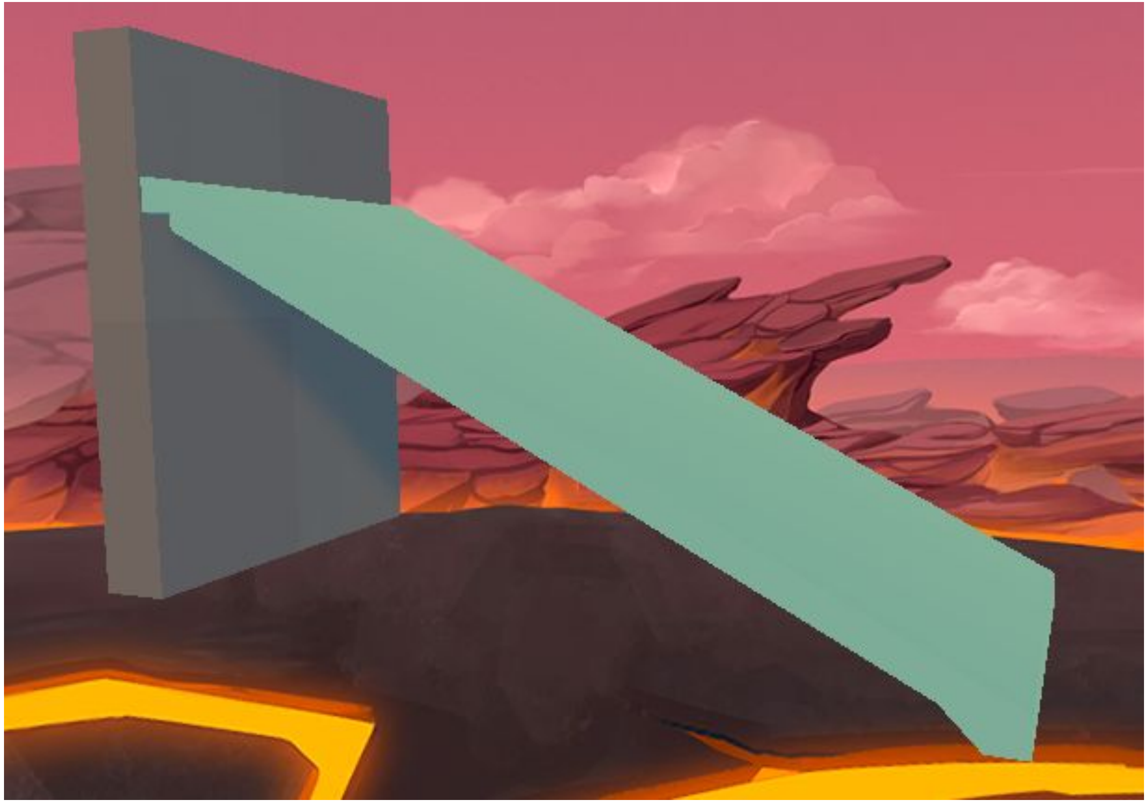


Ramps

Ramp_3m

The *Ramp_3m* prefab is a simple Ramp that is 3 meters high.

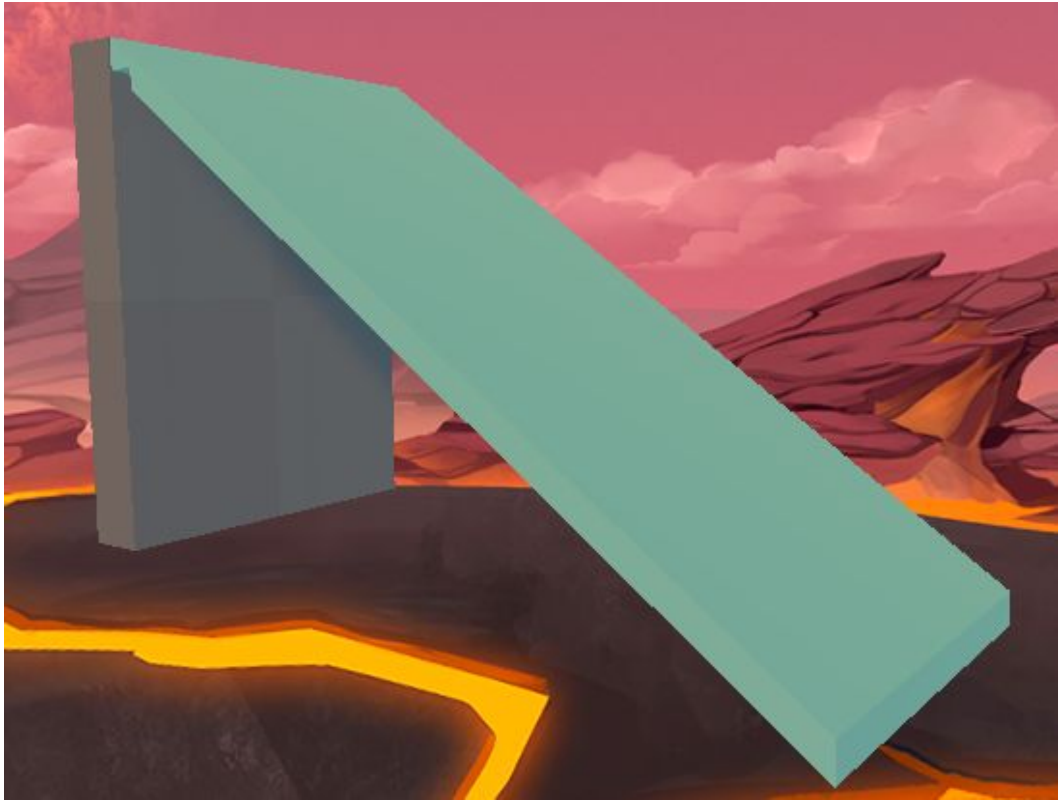
The *Wall_4m* seen within this screenshot is used to clearly indicate the height of the ramp.



Ramp_4m

The *Ramp_4m* prefab is a simple Ramp that is 4 meters high.

The *Wall_4m* seen within this screenshot is used to clearly indicate the height of the ramp.



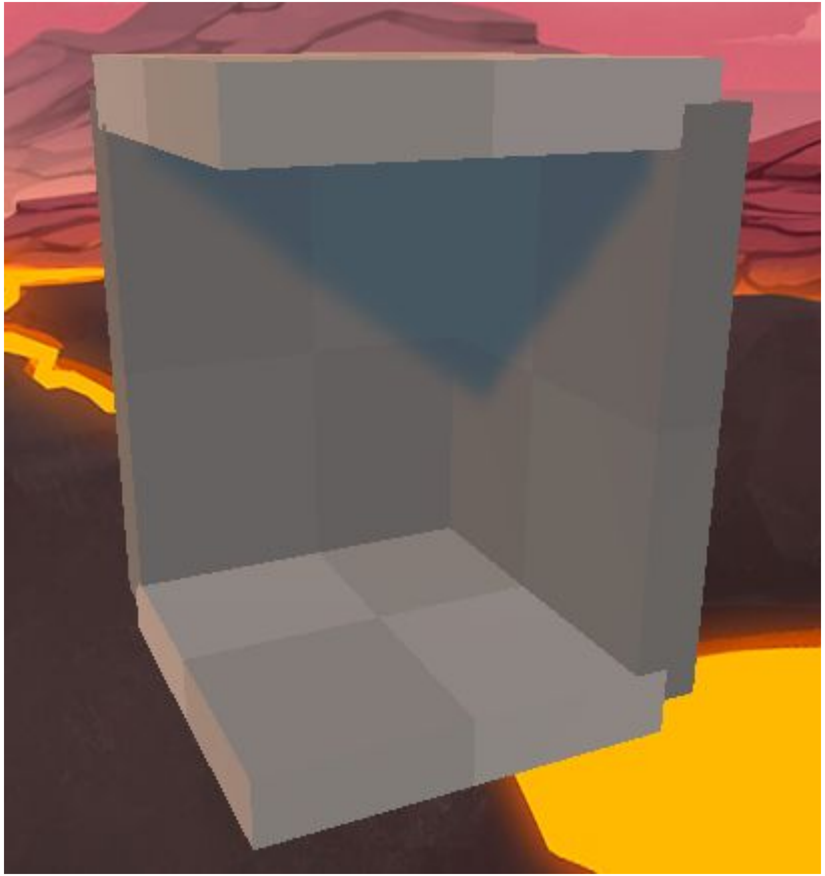
Catwalk Set

This Prefab set allows you to have small and tight sections in your level.

Cat_Corner

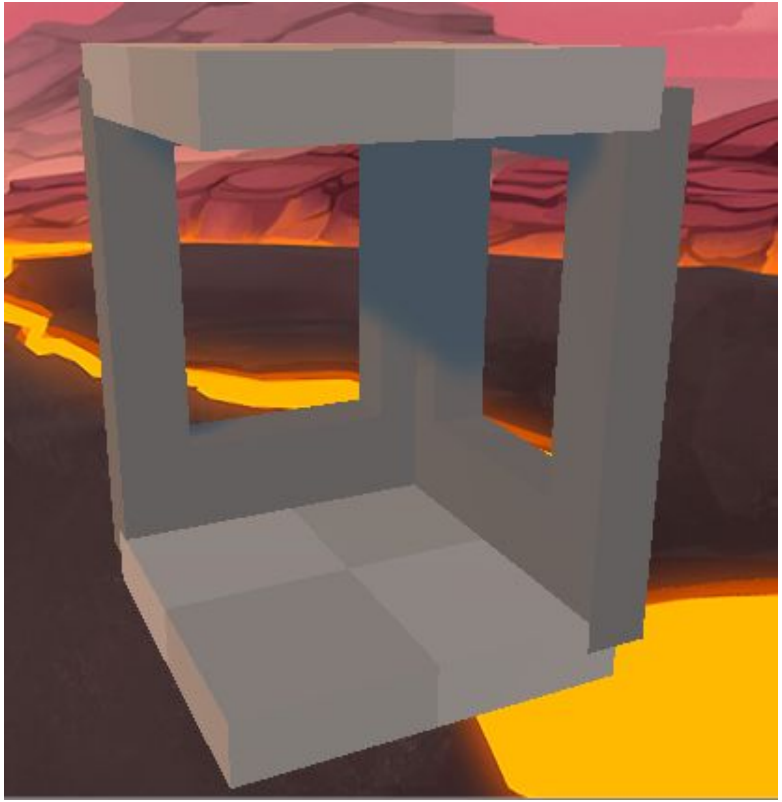
The *Cat_Corner* prefab, similar to the *Wall_Corner*, was created so that Level Design creation can be quick and modular.

This prefab uses a combination of *Basic_Floor* and *Wall_4m* prefabs.



Cat_Corner_Window

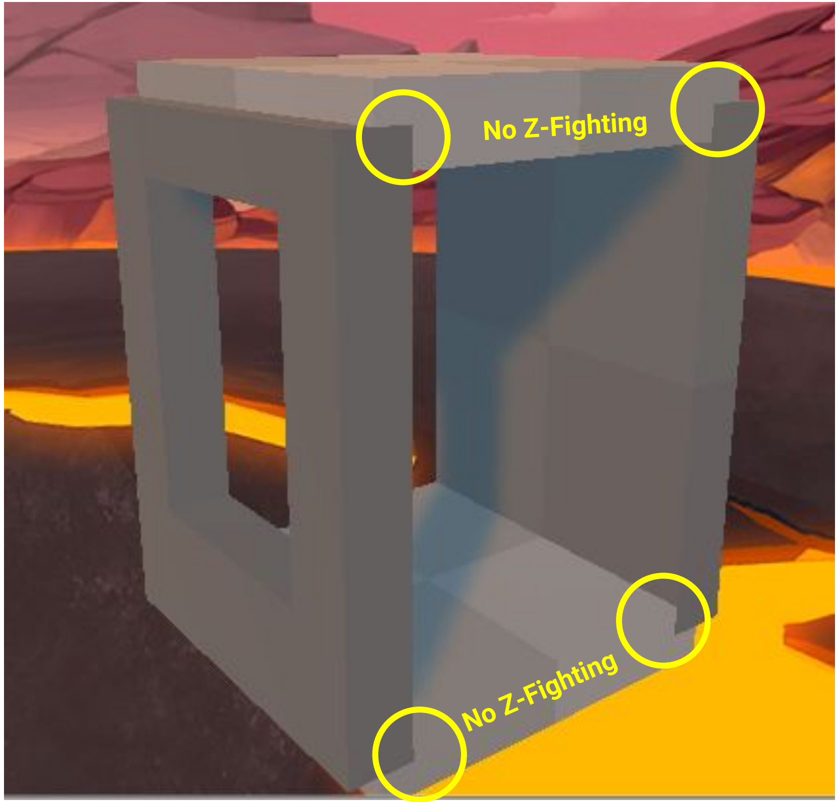
The *Cat_Corner_Window* prefab is similar to the *Wall_Corner*, but it uses the *Wall_Window* prefab instead of the *Wall_4m.*



Cat_Hall_Mix

The *Cat_Hall_Mix* prefab uses a combination of both *Wall_Window* and *Wall_4m* prefabs for its walls and a *Floor_HalfIndented* for both the ceiling and floor.

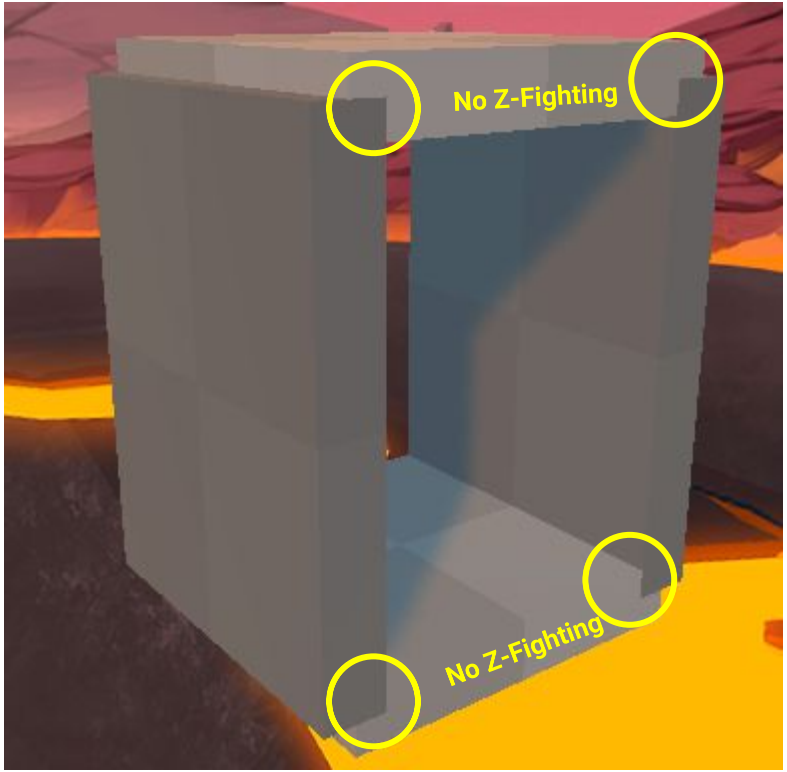
This is an example of how the notches on the floors are used to avoid Z-Fighting.



Cat_Hall_Wall

The *Cat_Hall_Wall* prefab uses the *Wall_4m* prefabs along with the *Floor_HalfIndented* prefab for both the ceiling and floor.

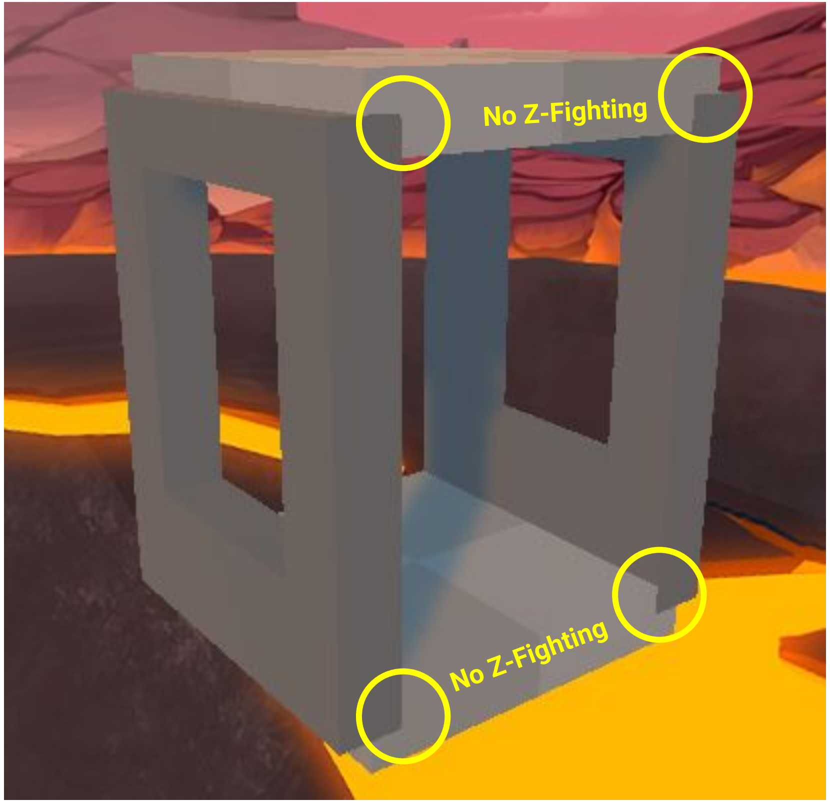
This is an example of how the notches on the floors are used to avoid Z-Fighting.



Cat_Hall_Window

The *Cat_Hall_Window* prefab uses the *Wall_Window* prefabs along with the *Floor_HalfIndented* prefab for both the ceiling and floor.

This is an example of how the notches on the floors are used to avoid Z-Fighting.

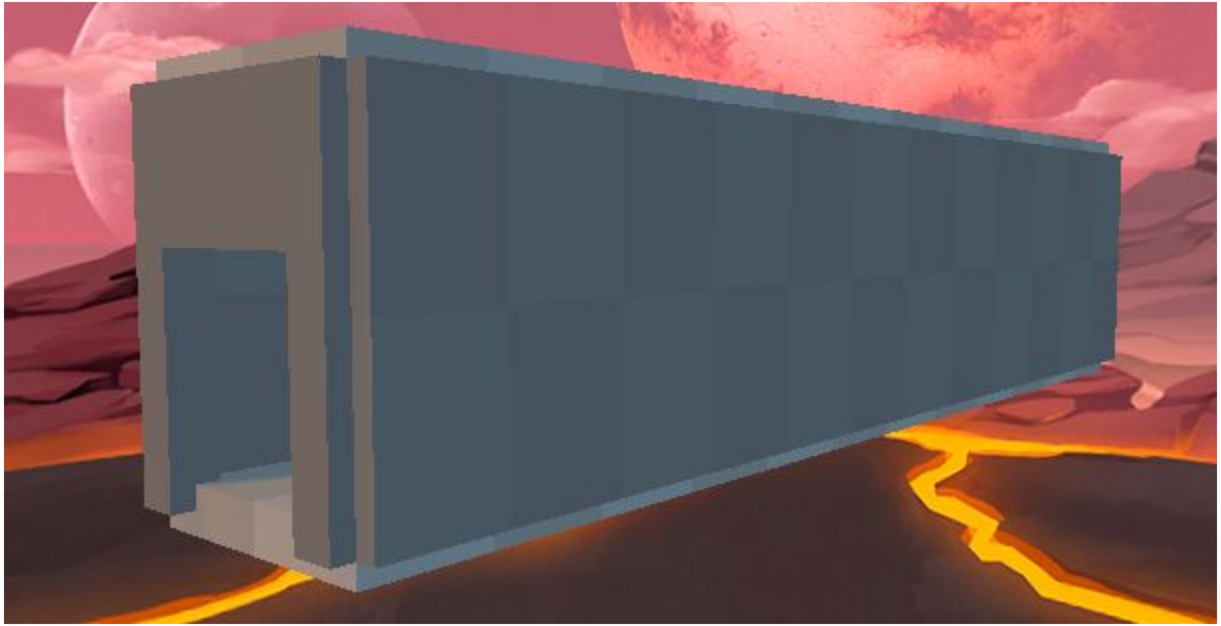


Crawl Set

This Prefab set is inspired by the Catwalk Set.

Crawl_Hall

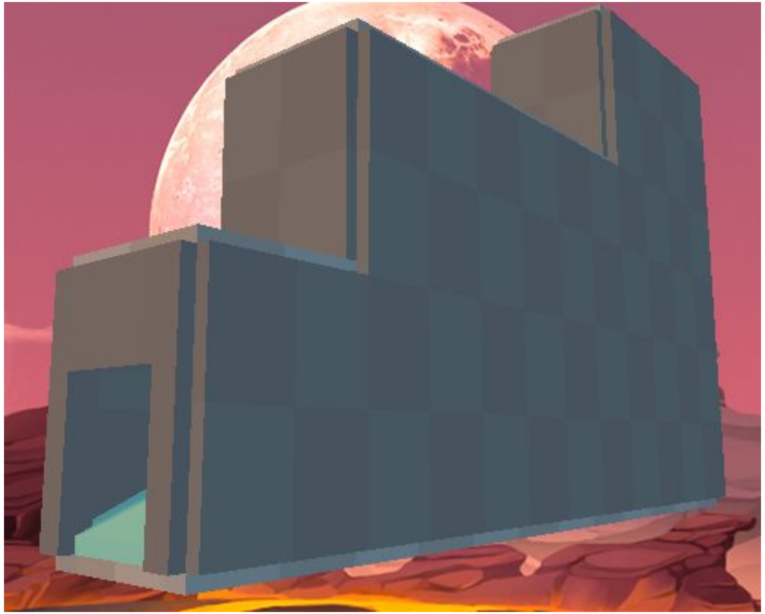
The *Crawl_Hall* prefab uses multiple *Wall_4m* prefabs along with the *Floor_HalfIndented* prefab for both the ceiling and floor.

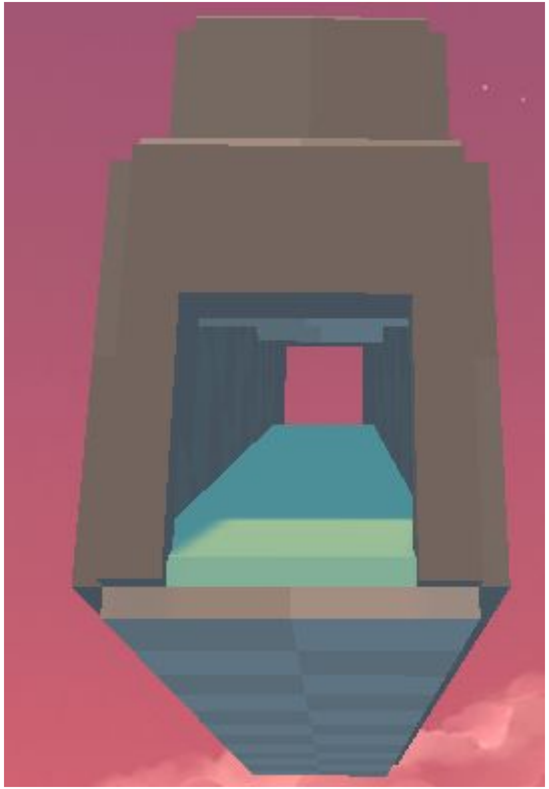


Crawl_Ramp

The *Crawl_Ramp* prefab uses an 8m high ramp along with multiple *Wall_4m* prefabs .

Floor_HalfIndented prefabs are also used for the ceiling and floor.





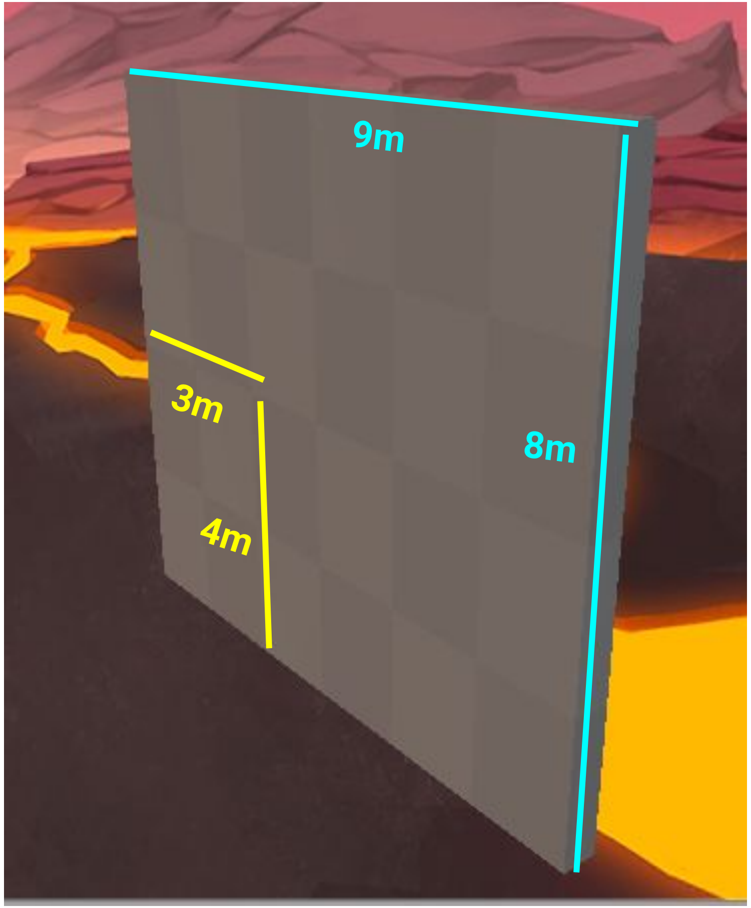
Dungeon Set

This Prefab set permits the rapid creation of labyrinthian levels.

Dun_Wall

The *Dun_Wall* prefab uses 6 *Wall_4m* prefabs.

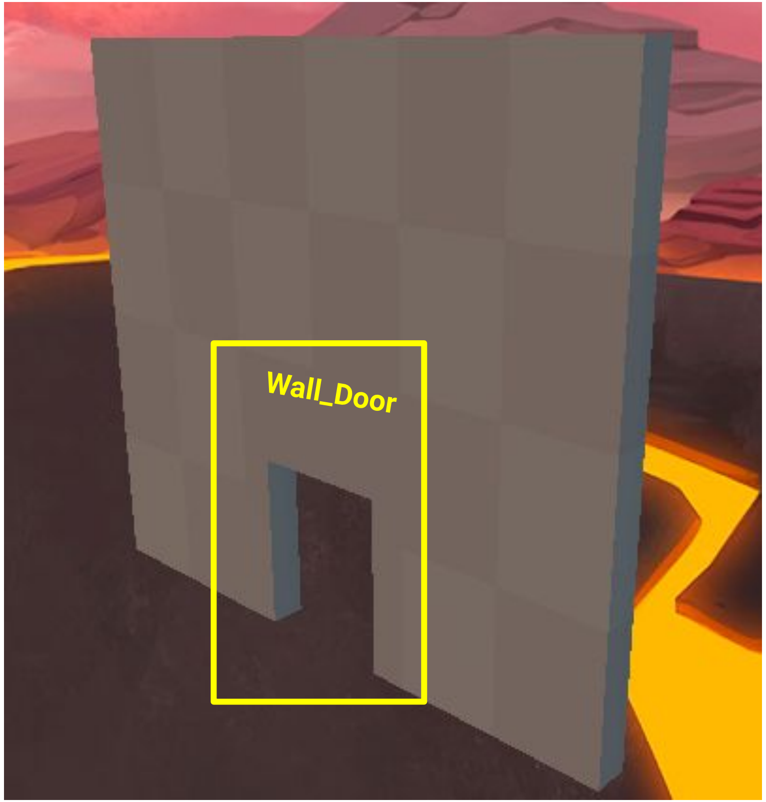
The standard wall size used in the Dungeon Set uses a 9 x 8 dimension.



Dun_Wall_Open

The *Dun_Wall_Open* prefab replaces the center bottom *Wall_4m* prefab with a *Wall_Door* prefab.

The standard wall size used in the Dungeon Set uses a 9 x 8 dimension.

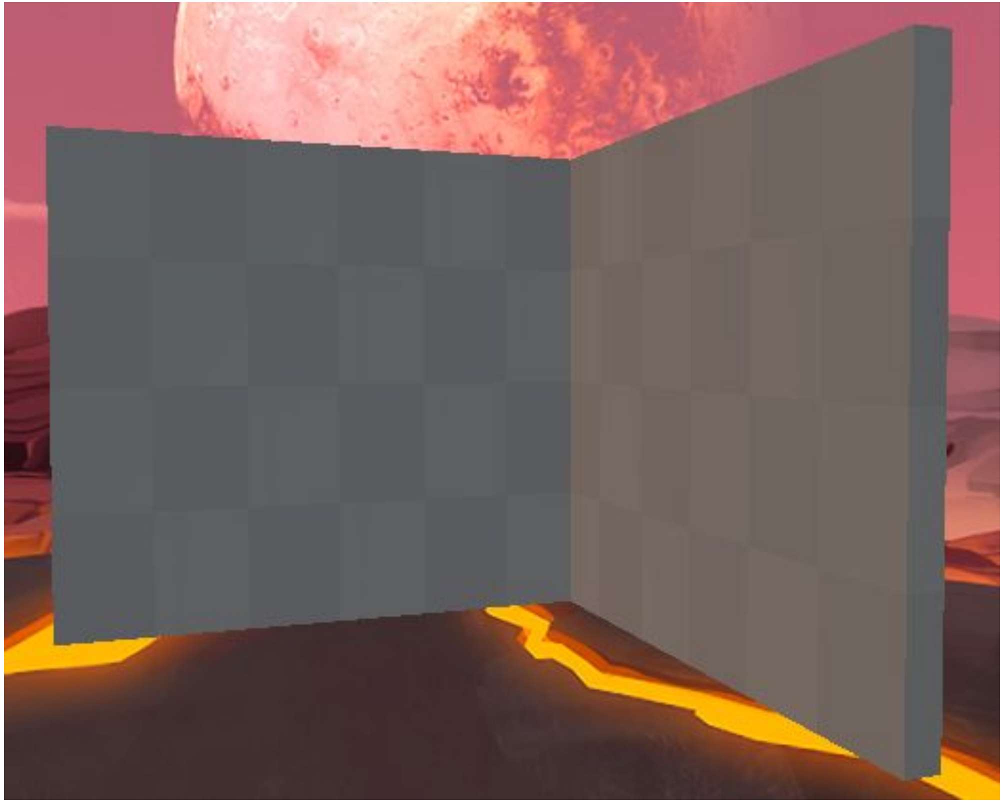


Dun_Corner

The *Dun_Corner* prefab uses 2 *Dun_Wall* prefabs.

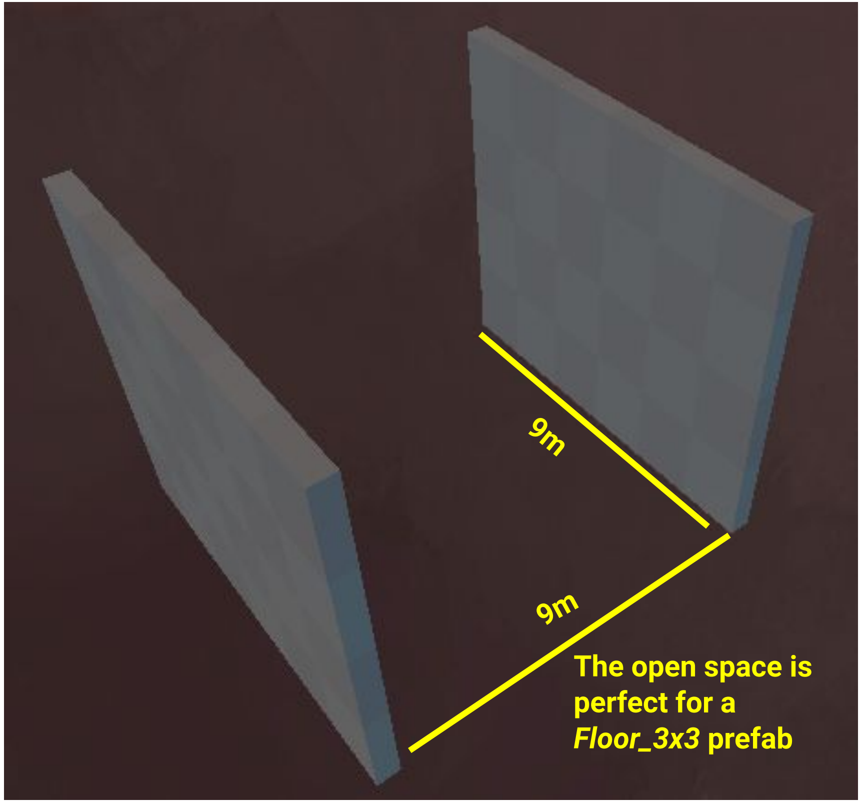
Similar to all the other Corner prefabs, this was constructed speed up level creation.

The standard wall size used in the Dungeon Set uses a 9 x 8 dimension.



Dun_Hall

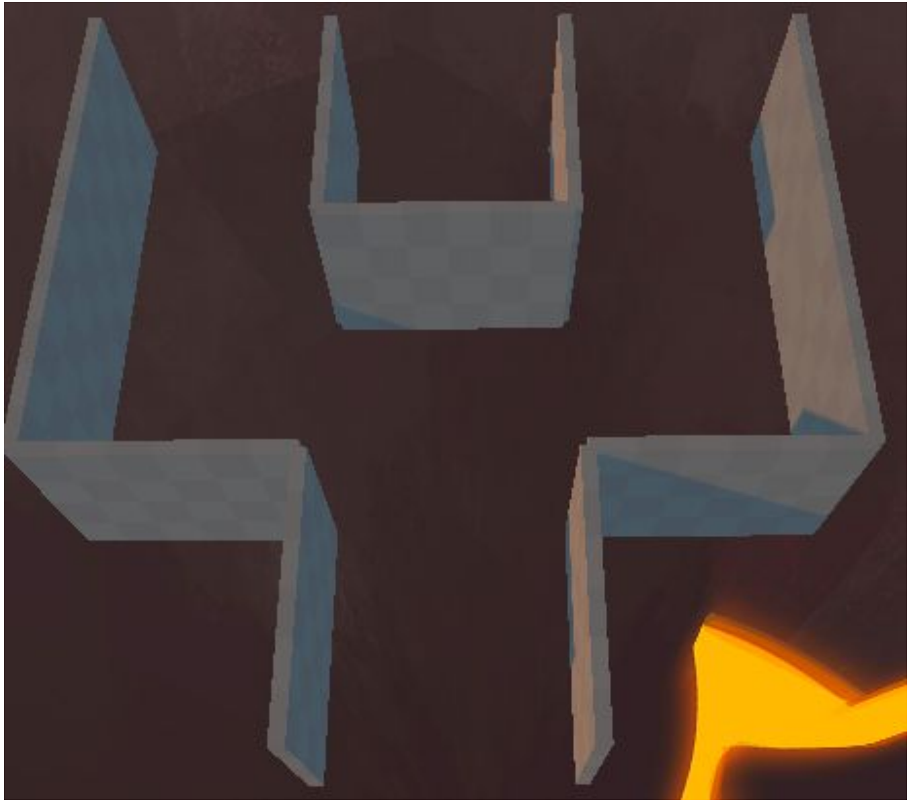
The *Dun_Hall* prefab uses 2 *Dun_Wall* prefabs and are separated by a length of 9m.



The open space is perfect for a *Floor_3x3* prefab

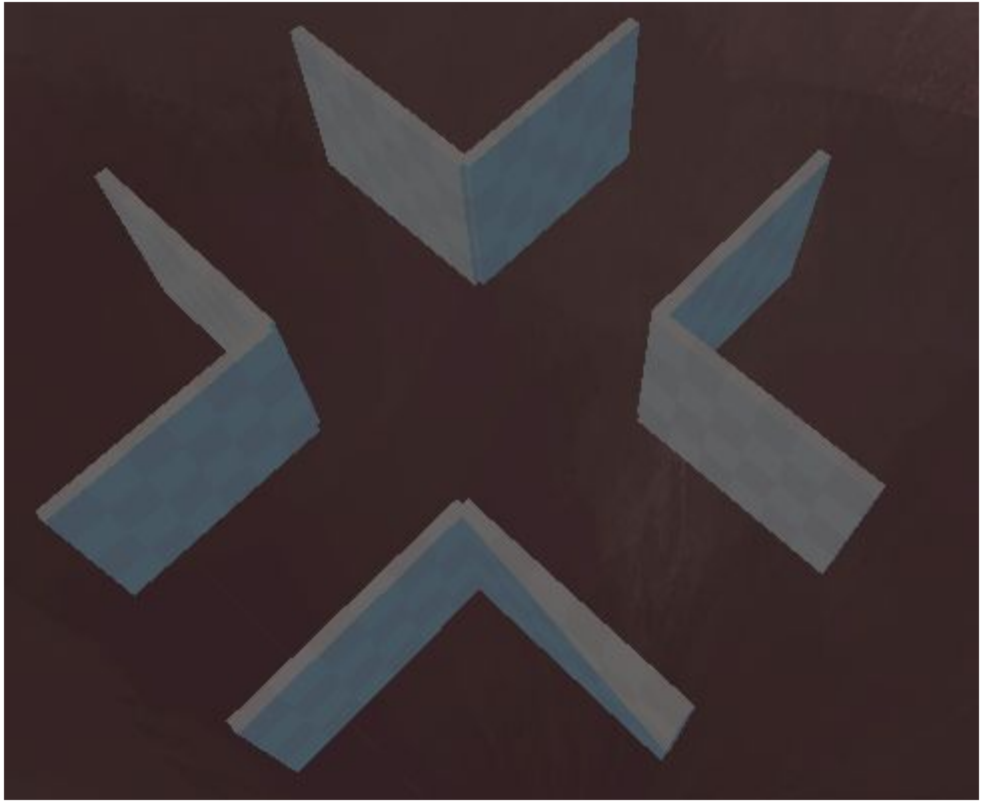
Dun_Fork

The *Dun_Fork* prefab permits you to open your paths and offer players multiple routes to take.



Dun_Cross

The *Dun_Cross* prefab allows you quickly expand your level's routes.

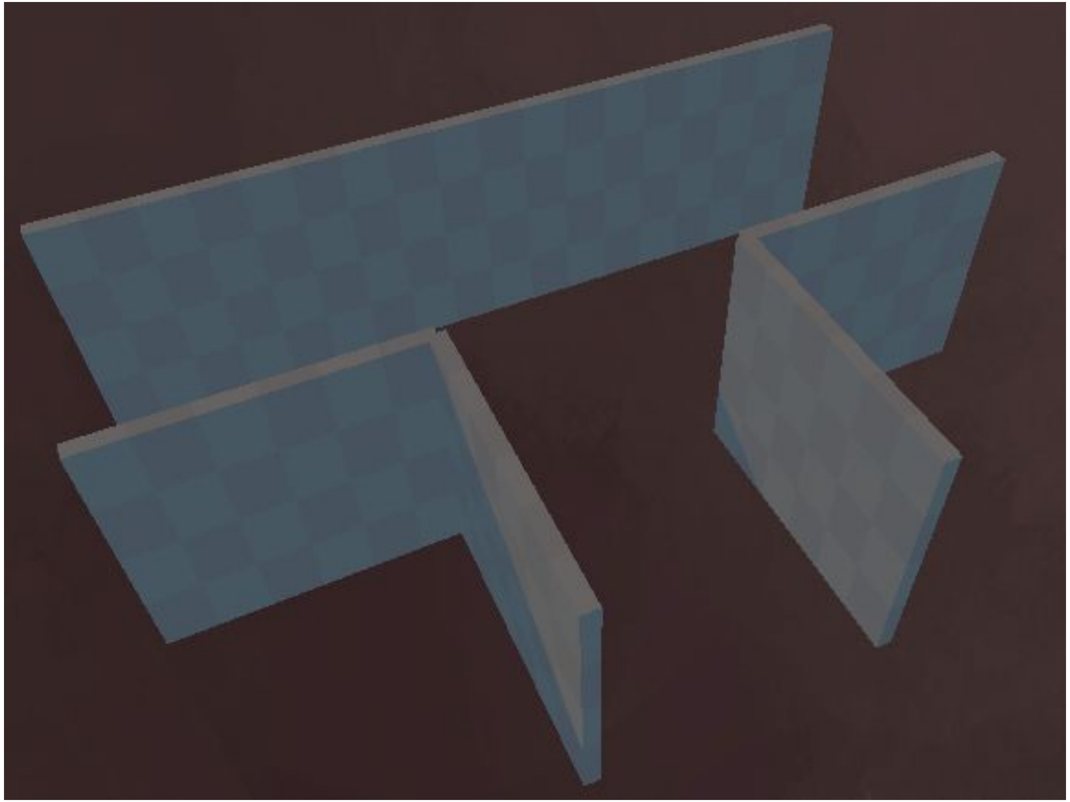


Dun_Hall_Corner

The *Dun_Hall_Corner* prefab, like other corner prefabs, is present to allow for a rapid creation of levels.

Dun_Hall_T

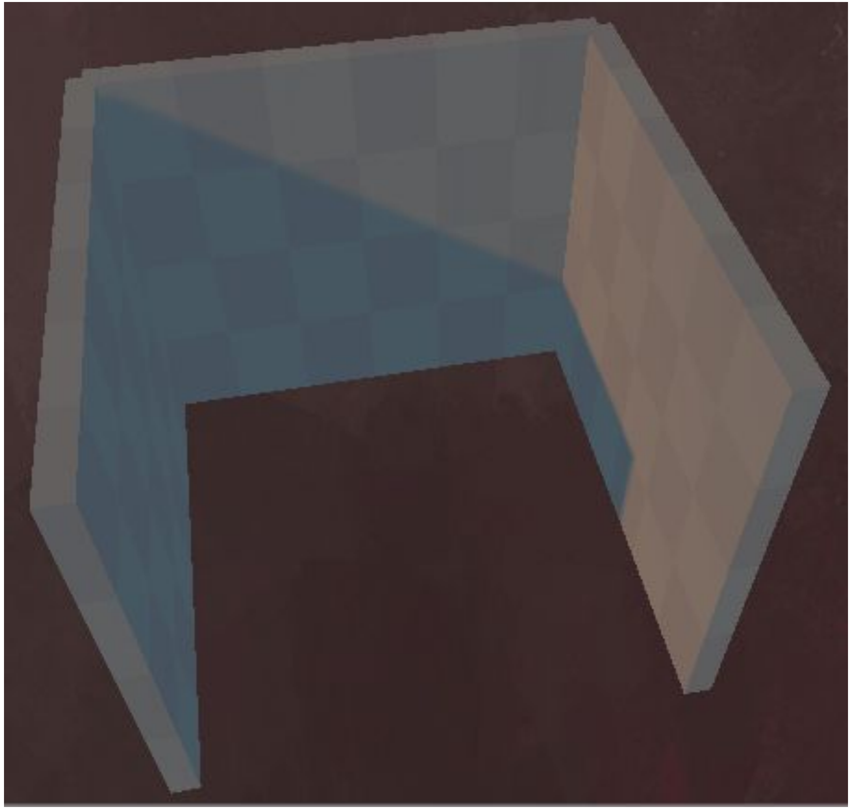
The *Dun_Hall_T* prefab permits you to open your paths and offer players multiple routes to take.



Dun_DeadEnd

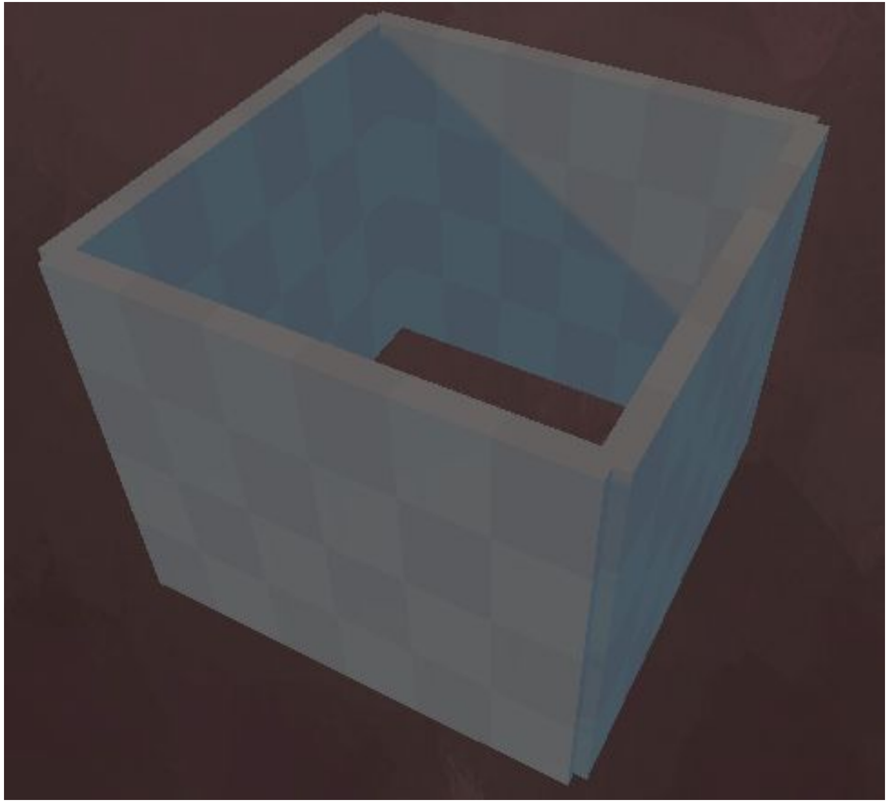
The *Dun_DeadEnd* prefab can be used to close off some of your paths in your level.

You can use this prefab to either reward the player with some pick ups or have them backtrack and find the correct route.



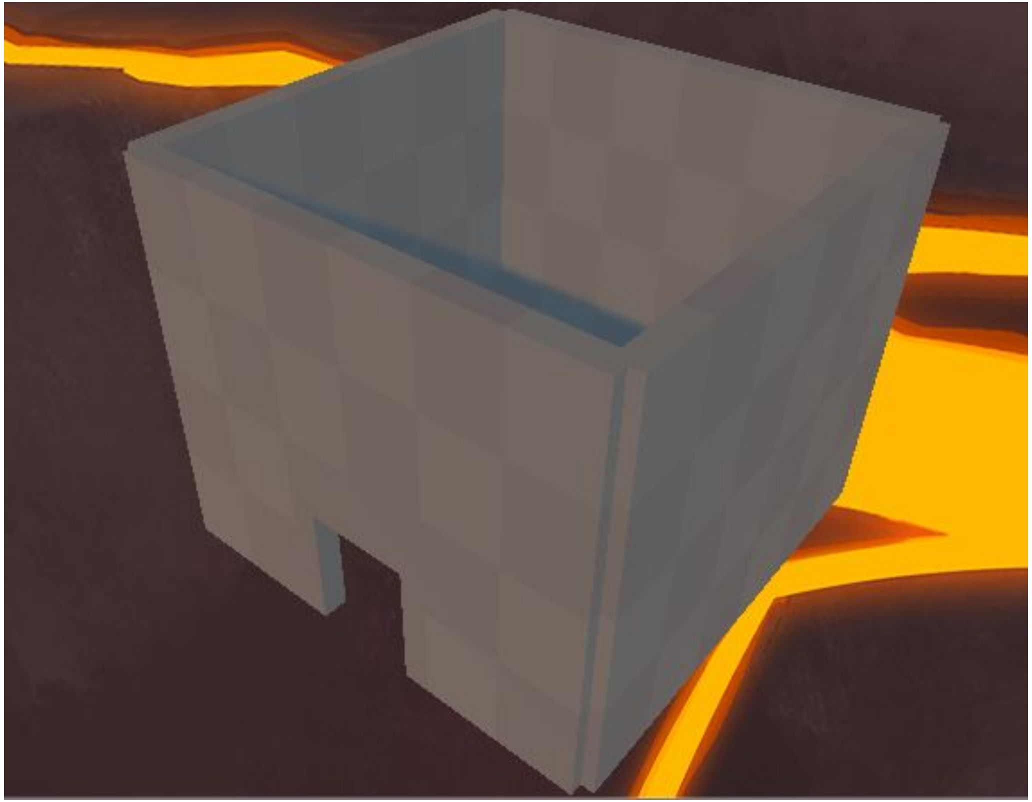
Dun_Tower

The *Dun_Tower* prefab is designed to be stacked. It permits you to create vertical sections in your level.



Dun_Tower_Open

The *Dun_Tower_Open* prefab, adjusts the previous *Dun_Tower* prefab by including a *Wall_Door*.



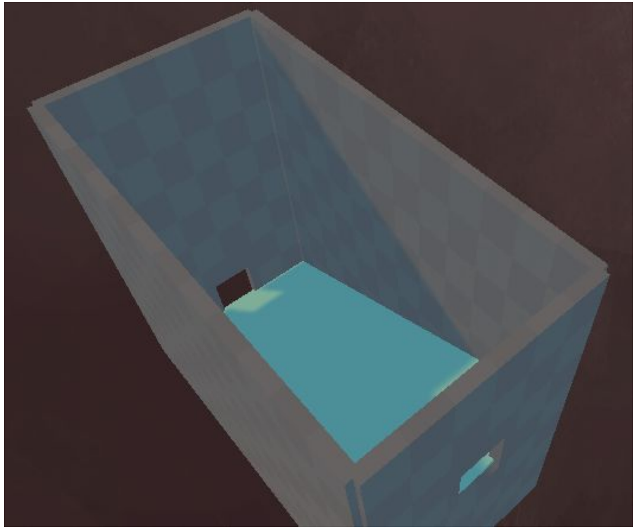
Dun_BigDoor

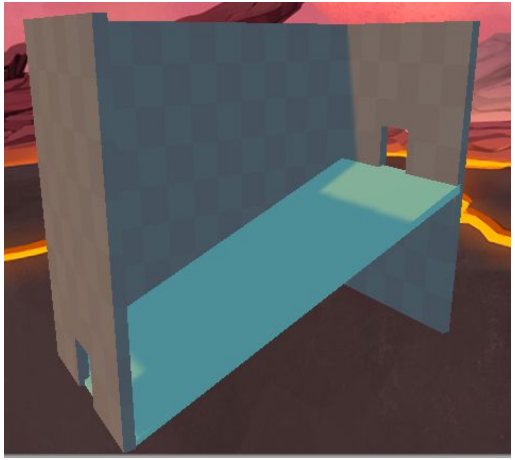
Because the Dungeon Set uses a new standard of metrics, the *Dun_BigDoor* prefab was created to better serve a level creation where walls are 9m tall.

Dun_Descent

The *Dun_Descent* prefab allows your Dungeon levels to either ascend or descend.

Similar to the *Crawl_Ramp*, the *Dun_Descent* ramp is 8m high.

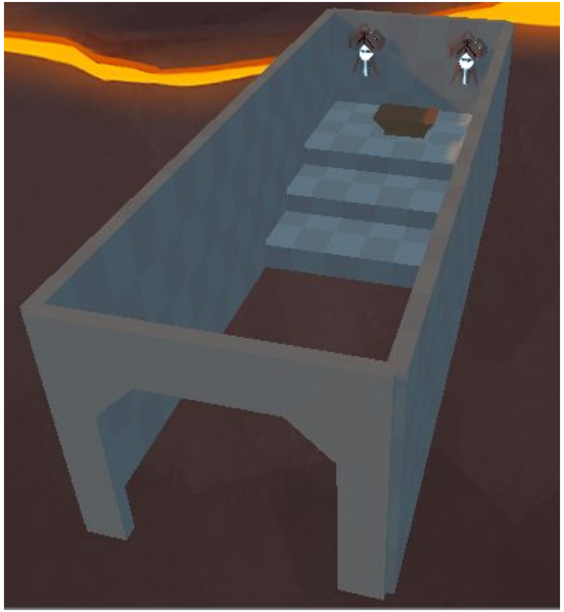


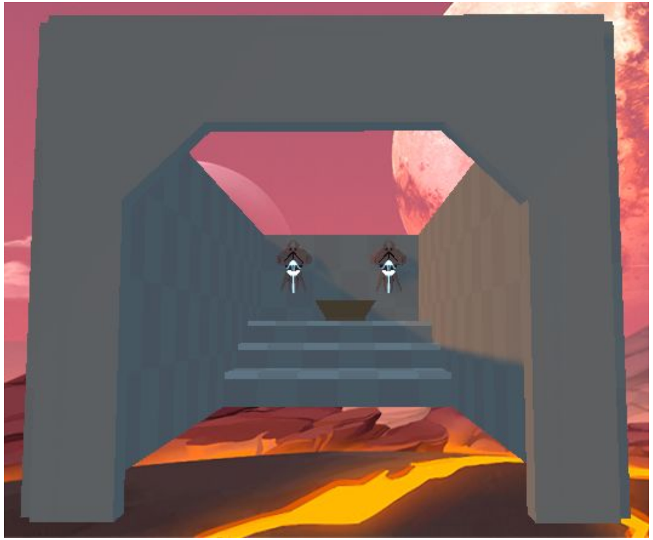


Rooms Set

Room_Altar

The *Room_Altar* prefab uses a combination of prefabs and unique props: *Statue_Wall* and *Altar*.





Room_Medium

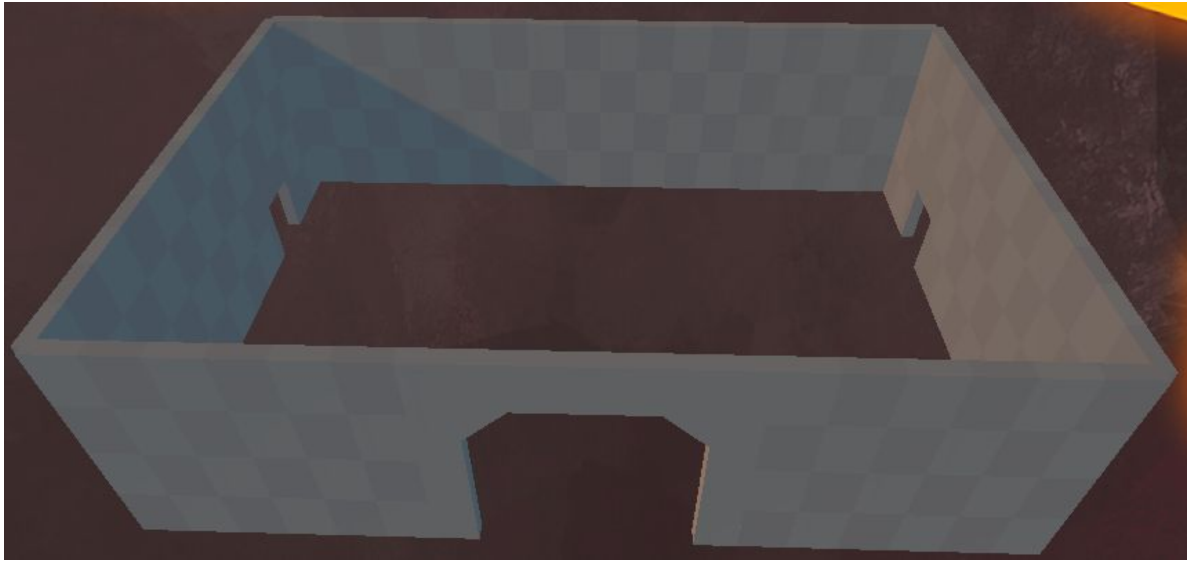
The *Room_Medium* prefab uses a combination of the Dungeon Set prefabs.

This prefab features 4 openings, with a floor size of 15x15 *(45m x 45m)*.

Room_Small_T

The *Room_Small_T* prefab uses a combination of the Dungeon Set prefabs.

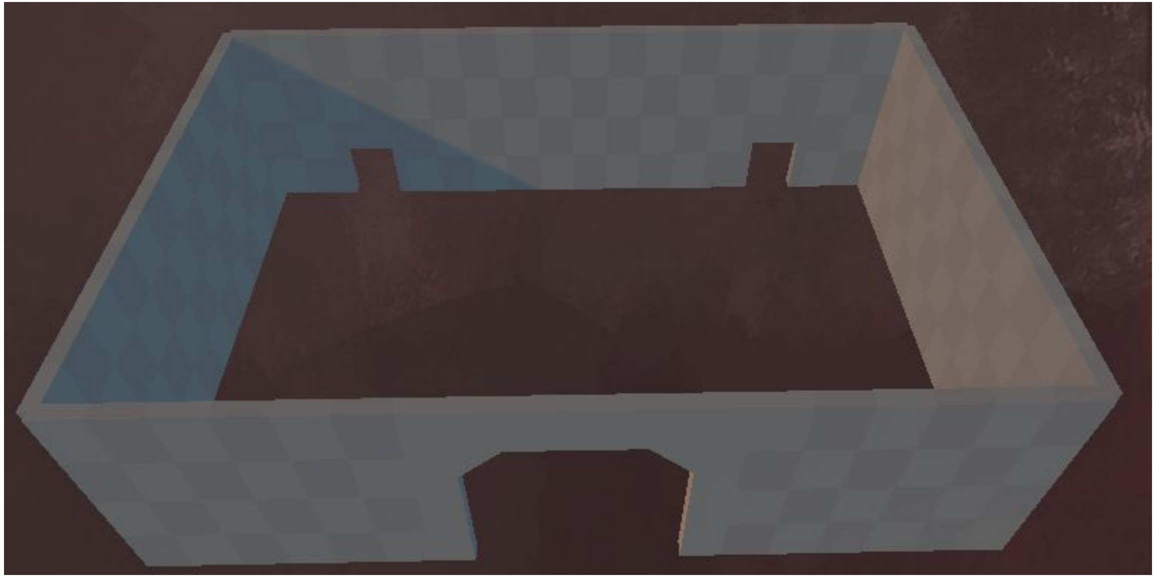
This prefab features 3 openings in a “T” shape with a floor size of 9x6 *(27m x 18m)*.



Room_Small_Y

The *Room_Small_Y* prefab uses a combination of the Dungeon Set prefabs.

This prefab features 3 openings in a “Y” shape with a floor size of 9x6 *(27m x 18m)*.



Stairwell Set

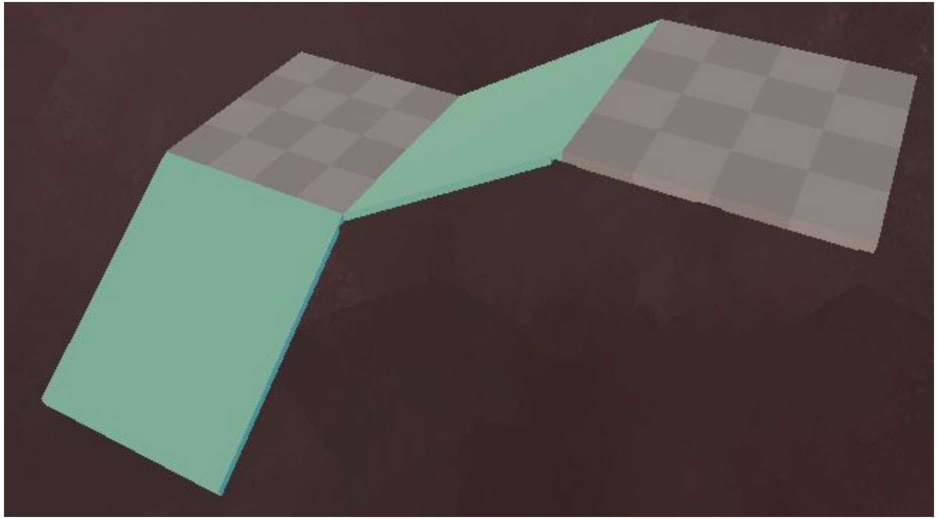
This Prefab set uses a combination of ramps and floors to make quick ascending/descending sections of your level.

Stairs_Large

The *Stairs_Large* prefab uses a combination of the *Ramp_4m* and *Floor_Indented* prefabs.

Both the Ramp sections and the floor sections are doubled in size so that traversing through these stairs is safer.

This prefab can easily be configured within your Dungeon Set levels.

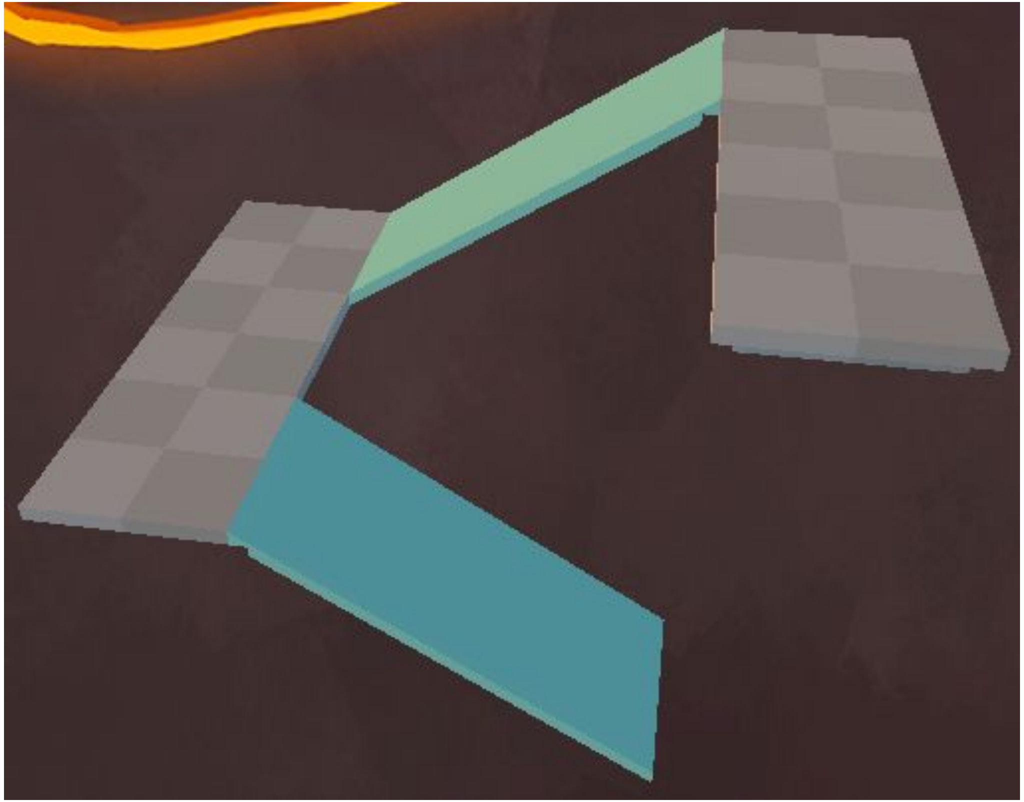


Stairs_Small

The *Stairs_Small* prefab uses a combination of the *Ramp_4m* and *Floor_Indented* prefabs.

Because it is narrower than the *Stairs_Large* prefab, navigating them is more challenging for the player.

This prefab can easily be configured within your Dungeon Set levels.

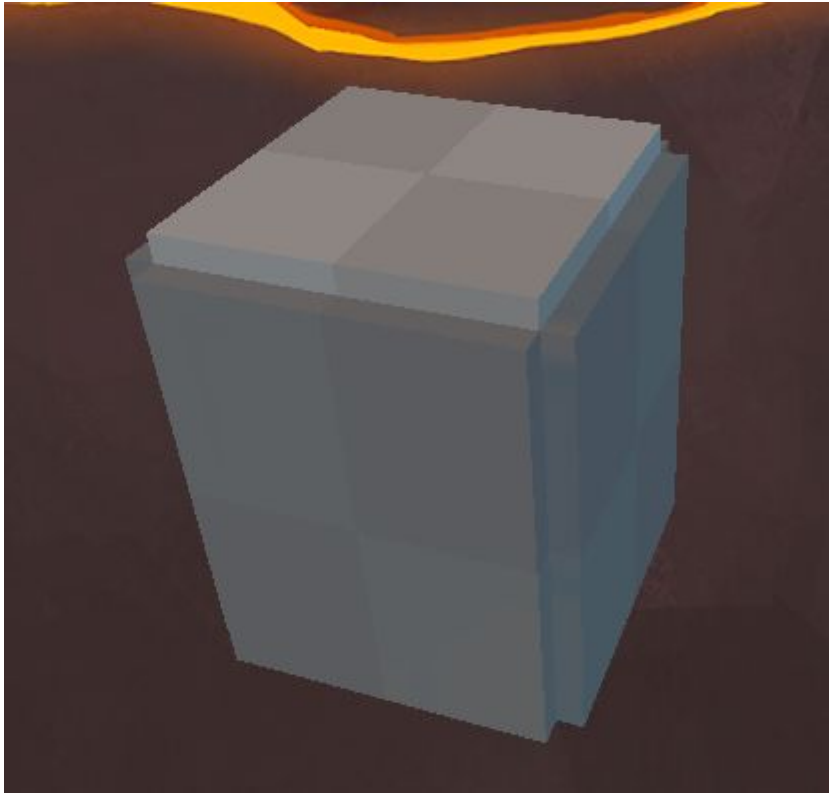


Steps

HighStep

The *HighStep* prefab uses a combination of the *Wall_4m* and *Basic_Floor* prefabs.

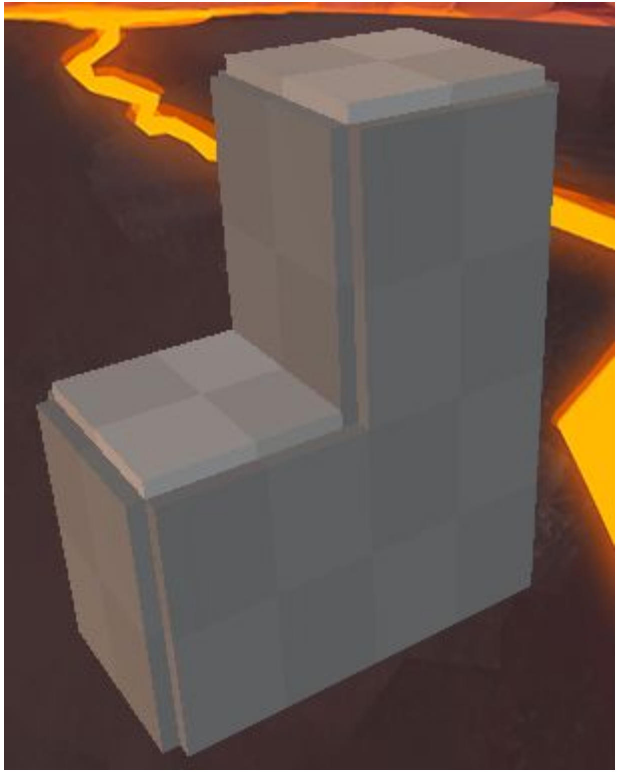
This prefab can be used for high jump sections, or to help block off certain areas while still allowing players to view above it.



HighStep_Double

The *HighStep_Double* prefab transforms the *HighStep* prefab into an 8m high object.

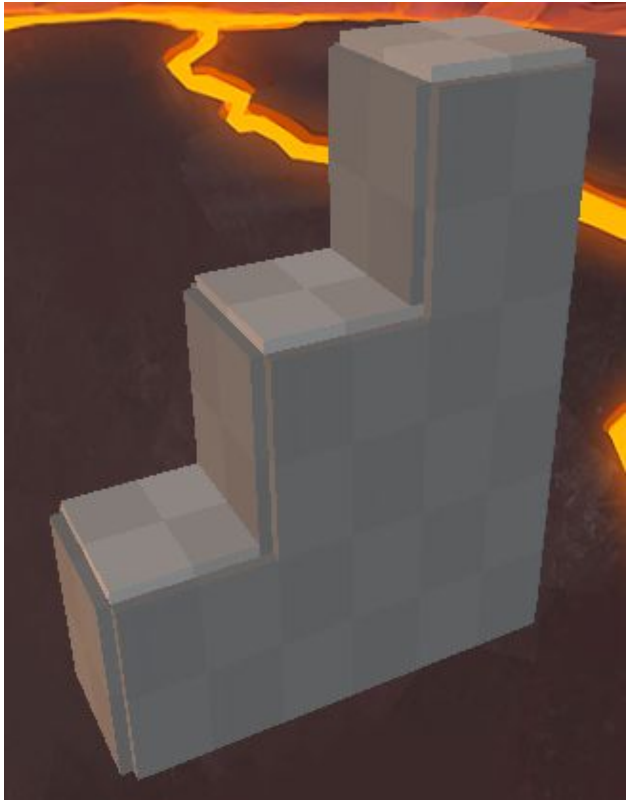
This prefab can be used for high jump sections, or to help block off certain areas while still allowing players to view above it.



HighStep_Triple

The *HighStep_Triple* prefab transforms the *HighStep* prefab into a 12m high object.

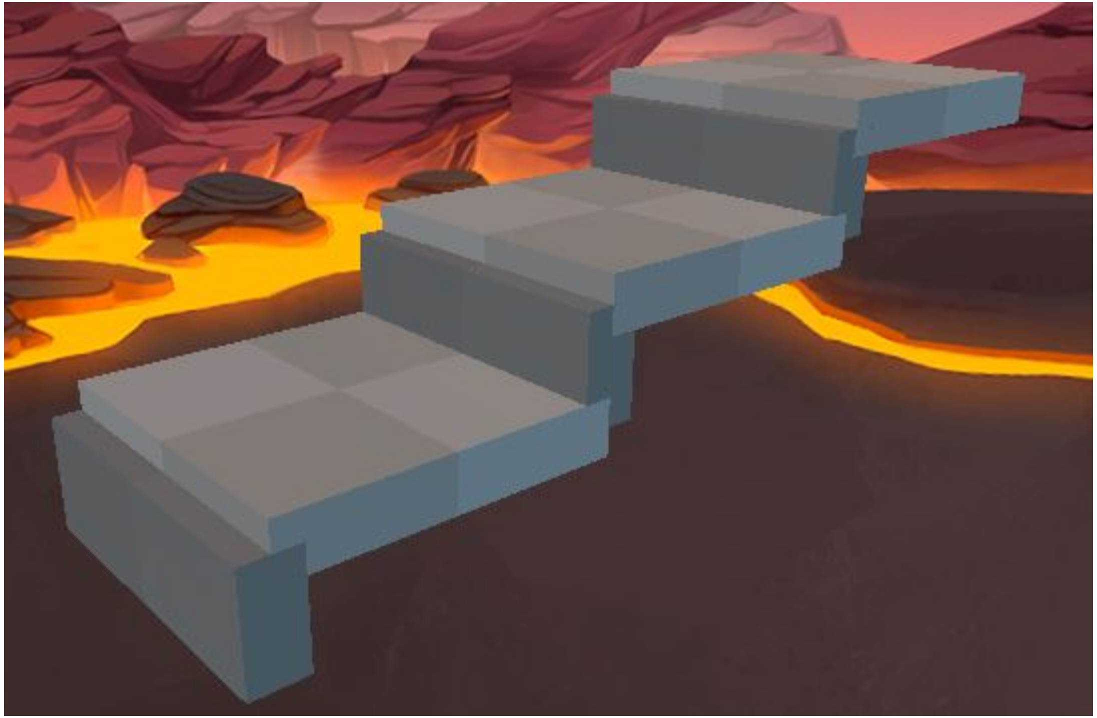
This prefab can be used for high jump sections, or to help block off certain areas while still allowing players to view above it.



ShortSteps

The *ShortSteps* prefab uses a combination of *Wall_1m* and *Basic_Floor* prefab.

Unlike the *HighStep* prefabs, these steps can be traversed by both Players and Enemies.



ShortSteps_Corner

The *ShortSteps_Corner* prefab can be used to close off any *ShortStep* section within your level.

This prefab uses a combination of *Wall_1m* and *Basic_Floor* prefabs.