Lumberyard

Getting Started Guide

Version 1.11

Lumberyard: Getting Started Guide

Table of Contents

Introduction to Lumberyard Editor

The *Getting Started Guide for Amazon Lumberyard Editor* is a tutorial series that guides you through the basics of building a game in Lumberyard. You'll create a lush mountainous landscape, build a maze within it, place a player character, position enemy characters, install a simple UI, and finally, export and play your game. By working through this series, you will gain an understanding of the fundamental workflows and tools within Lumberyard. As you proceed, you can refer back to the orientation level, which represents the completed level that you are building.

Before you get started, install Lumberyard.

Each of the tutorials in this 12-part tutorial series teaches or demonstrates a Lumberyard concept. Each tutorial takes 10 to 30 minutes, depending on your level of experience.

- *1: Understanding the Lumberyard Interface* (p. 3) – Open the orientation level to explore and learn Lumberyard's basic manipulation and navigation tools. This level also represents the end product of the game you'll build during this series. Refer to this level whenever you want to compare your level to the completed level.
- *2: Building the Environment* (p. 32) – Create your own new level and then import a heightmap and texture map to give your landscape a mountainous look. Place a player character to play and explore your level.
- *3: Gathering Your Building Blocks* (p. 46) – Add the doorway pieces of the maze structure. Learn how entities and components make objects behave like their real-world versions.
- *4: Using Slices to Build the Maze* (p. 60) – Use prebuilt slices to add the exterior and interior maze walls and lamp posts. Learn how the slices feature makes it simple to change one lamp post and then save the change to modify all the lamp posts at once.
- *5: Adding Enemy AI Characters* (p. 72) – Place enemy AI sentries inside the maze and set their behavior to patrol the area.
- *6: Scripting Gameplay* (p. 96) – Use Lua scripts to make doors open and close, require an item to be collected, and time the player's passage through the maze.
- *7: Applying Physics to Stacked Crates* (p. 147) – Stack crates to construct a wall. Apply physics properties to the crates so that the player can knock the crate wall down.
- *8: Sculpting the Terrain* (p. 155) – Create a lush vegetative environment by adding trees, grass, dirt, and rocks. Build a well-worn footpath that leads to and away from the maze.
- *9: Enhancing Your Level with Details* (p. 192) – Enhance the look of your environment by adding decals such as a door number and scorch marks. Create a more interesting environment by adding broken pipes, erupting steam, ambient particles in the air, and industrial junk.
- *10: Lighting the Environment* (p. 196) – Illuminate your level by creating environmental probes, setting the time of day, and adding lights to your lamps.
- *11: Setting Up the UI* (p. 209) – Incorporate a simple UI for your game.
- *12: Exporting Your Game* (p. 213) – Export your game to an executable file and play it.

You can skip to any tutorial in this series without first completing all preceding tutorials. To do this, open a prebuilt level file that has all the steps from that level and all preceding tutorials already completed. For example, if you want to explore the AI tutorial (number 5) without completing levels 1 through 4, open **GSG_04_Slices**.

To open a prebuilt level file

1. Install and open Lumberyard.
2. In the **Welcome to Lumberyard** screen, click **Open Level**.

3. Navigate to `Levels\GettingStartedGuide`. Select the level that you want to open and then click **Open**.

Start: 1: Understanding the Lumberyard Interface (p. 3)

1: Understanding the Lumberyard Interface

In this section, you learn the basics of Lumberyard's interface. You also explore and play the completed level for this tutorial series, giving you a solid understanding of what you'll be building.

This tutorial shows you how to do the following:

- Open Lumberyard's orientation level (p. 3) – Open the orientation level, explore the welcome screen, and preview the completed level that you'll be building.
- Identify the basic tools (p. 6) – Become familiar with the primary areas of Lumberyard's interface.
- Navigate in the viewport (p. 14) – Learn how to move efficiently around the level.
- Manipulate objects (p. 15) – Select (p. 17), Move (p. 19), Rotate (p. 23), and Scale (p. 27) objects.
- Configure auto backup (p. 30) – Configure incremental backups so that you don't lose your work in the event of a crash or freeze.

Downloading and Installing Lumberyard

Before you get started, download and install Lumberyard. Use the installation tutorial if you need instructions. After you have set up Lumberyard, launch the Lumberyard Editor and proceed to the next section.

Opening the Orientation Level

To familiarize you with Lumberyard's opening screen and basic controls, open the first tutorial level called **GSG_01_Orientation**.

To open the orientation level

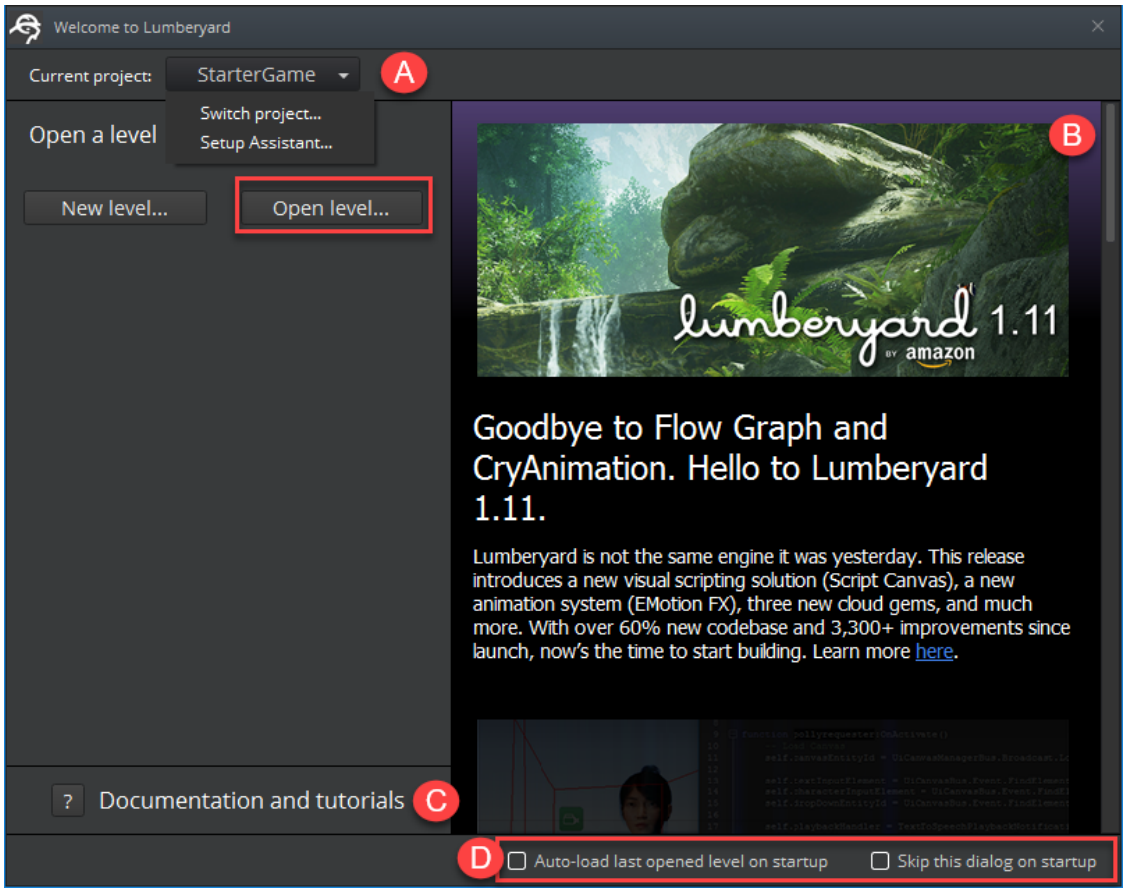
1. Launch Lumberyard Editor.

   The **Welcome** screen appears by default each time you open Lumberyard.

   On the welcome screen, you see the following:

   - **A** – **Current Project** – Displays the project that you are currently working on. If you click the project name, the following options appear:
     - **Switch project** – Closes Lumberyard Editor and opens **Project Configurator**, which you can use to switch to another project.
     - **Setup Assistant** – Closes Lumberyard Editor and opens Lumberyard Setup Assistant, where you can customize your build options.
   - **B** – **Welcome** messages – Displays the latest news about Lumberyard and its partners.

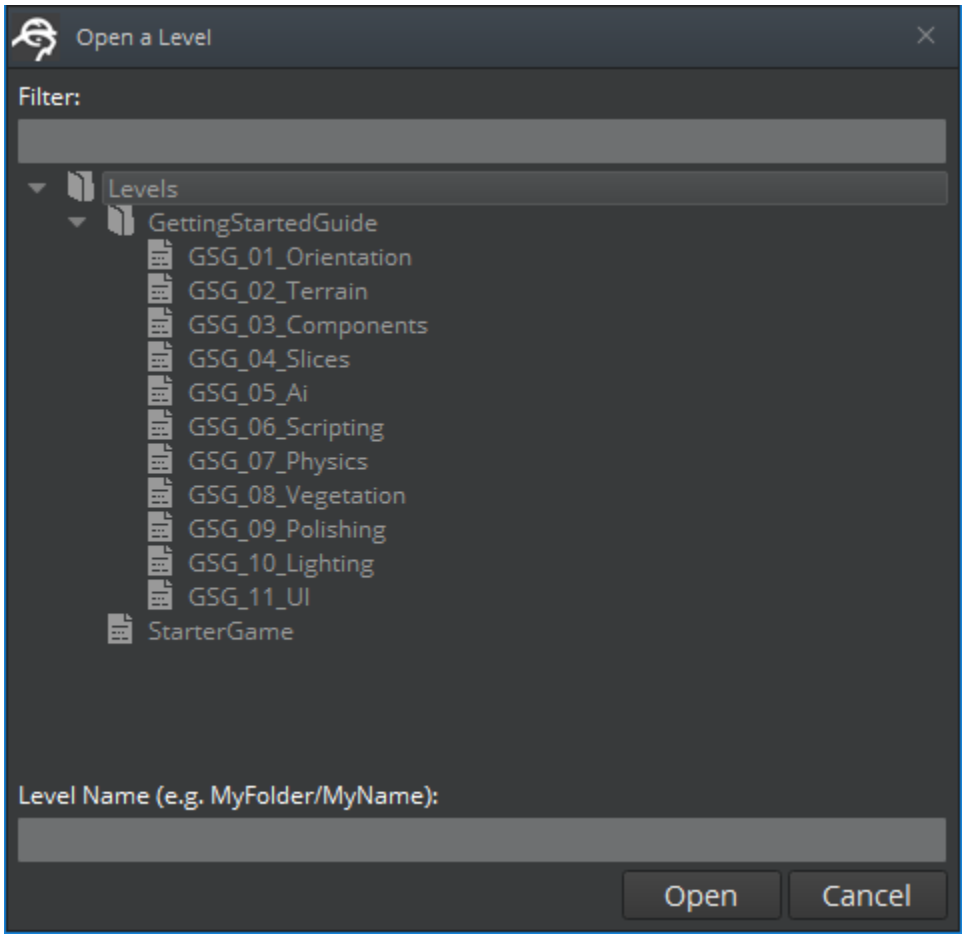
- **C** – **Documentation and Tutorials** – Opens the Lumberyard support page where you can find other tutorials and documentation about Lumberyard.
- **D** – **Welcome** screen options – Select check boxes to **Auto-load last opened level on startup** or **Skip this dialog on startup**.



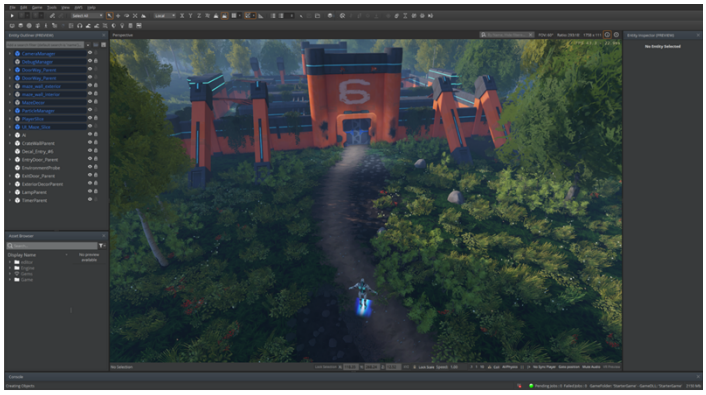
2. Click **Open level**.

   Navigate to `Levels/GettingStartedGuide`.

   Choose `GSG_01_Orientation` and then click **Open**.



The orientation level opens. It looks similar to the following image.



3. (Optional) To explore the level and play the game, press **Ctrl+G**.

   Use standard PC game controls – W, A, S, D, space to jump, left click to shoot.

   Try to complete all of the following:

   - Approach the entry door to trigger it to open. The timer starts when you enter the maze.
   - Shoot to kill enemy AI sentries.
   - Find and collect the pickup item, a glowing orb. This triggers the opening of the exit door.

- Shoot at and knock over the wall built from crates.
- Exit the maze. The timer stops, and a "Mission Complete" screen appears.
- Explore the level. Have fun!

Press **Esc** to quit.

Next: Identifying the Basic Tools (p. 6)

Identifying the Basic Tools

Lumberyard's interface is easy to use and is set up to match a common workflow. Become familiar with the interface names marked in the following image, as these are Lumberyard's basic tools.

As you proceed through the tutorial series, you'll use most of these tools and become familiar with them.

The next few sections describe these tools in greater detail and provide links to the *Amazon Lumberyard User Guide* for more details and advanced usage.

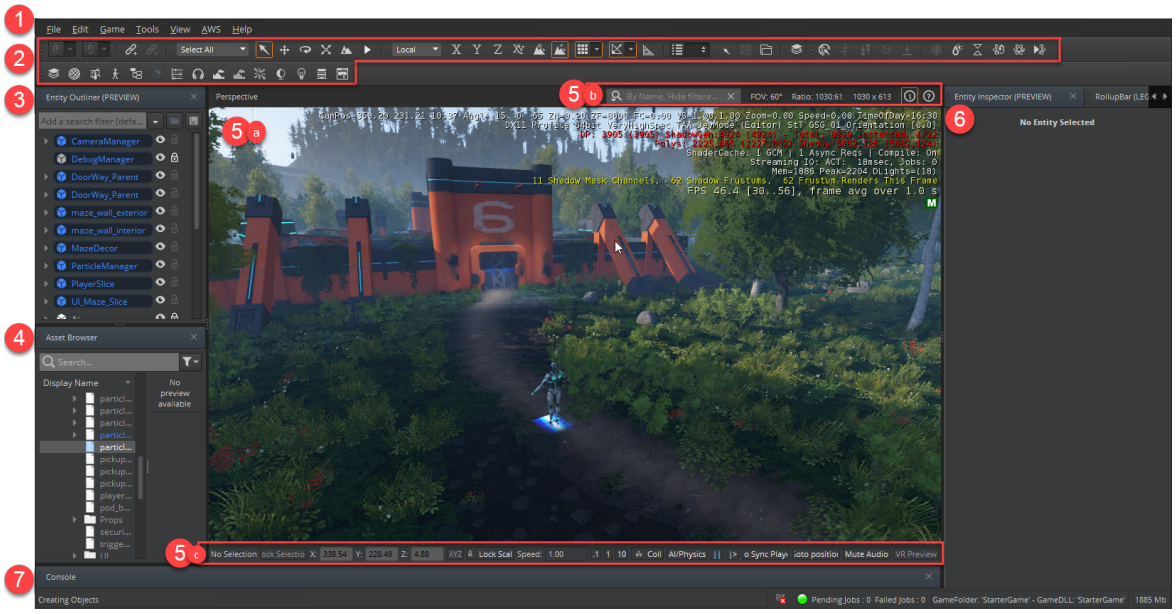
Interface Names

1. **Main menu** – Access all tools, editors, and commands.
2. **Top toolbar** – Open the most commonly used tools, editors, and commands.
3. **Entity Outliner** – View all the entities and slices in your level and perform actions on them.
4. **Asset Browser** – Drag assets that are available in your project into your level.
5. **Perspective**:

   5a. **Viewport** – View and navigate in your 3D game environment.

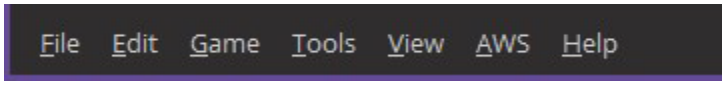
   5b. **Header** – Find objects quickly and adjust display options.

   5c. **Bottom toolbar** – Change the navigation speed and view the position of your selected object.
6. **Entity Inspector** – View, add, remove, and modify components on the currently selected entity.
7. **Console** – View a running list of all editor commands, processes, and output.

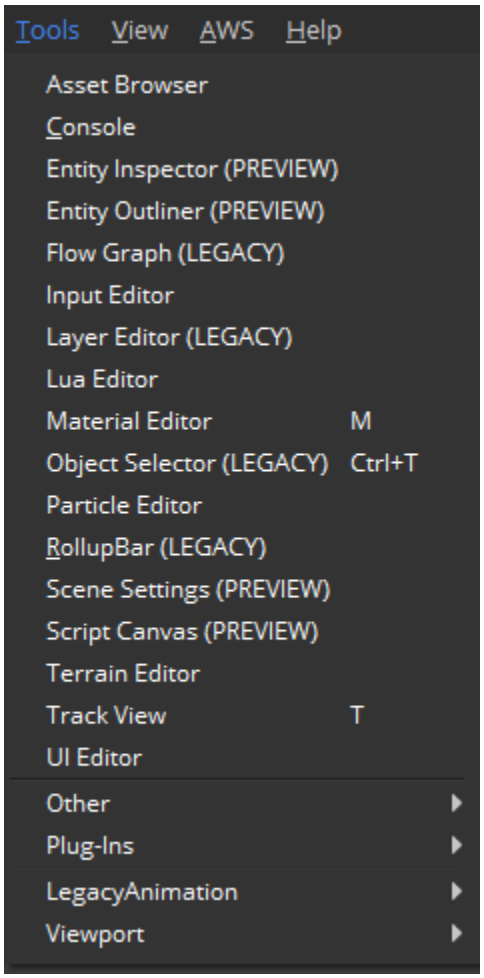


Next: 1: Main Menu (p. 7)

1: Main Menu

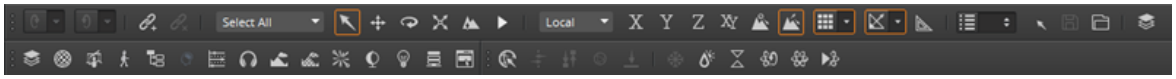


The **Main menu**, located at the top of the editor, contains all of Lumberyard's functions and settings. You can open tools that are also accessible from other parts of the interface. For example, every editor or tool that can be opened from the Lumberyard Editor toolbar can also be opened from the main menu. Click the **Tools** menu to see all available editors.



Next: 2: Top Toolbar (p. 8)

2: Top Toolbar

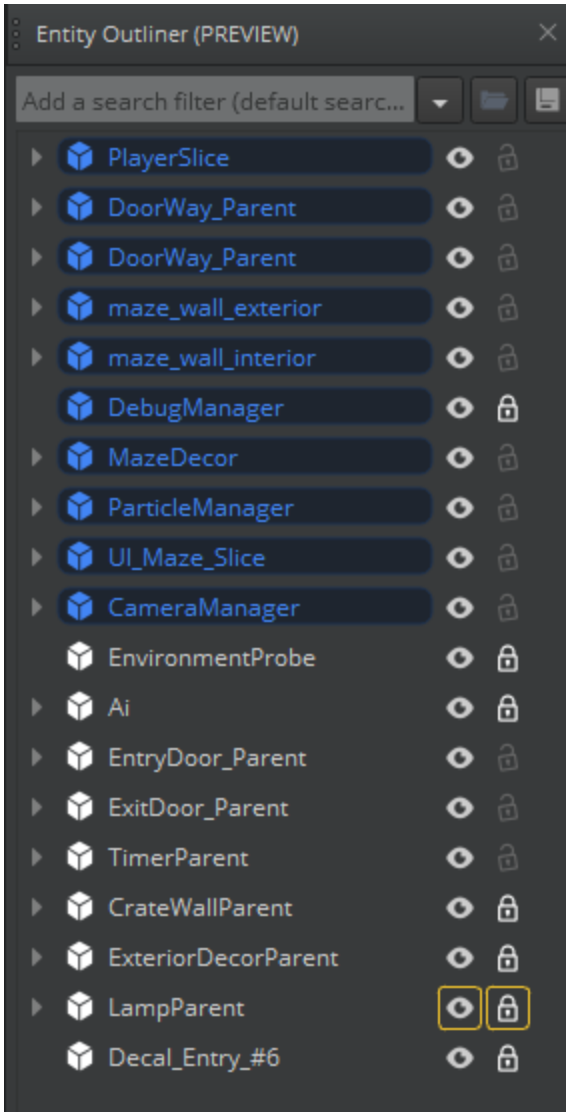


The top toolbar features quick access to Lumberyard's most commonly used tools and editors, such as the **Select**, **Move**, **Rotate**, and **Scale** tools. You'll learn more about these tools and editors as you proceed through this tutorial series.

For more information about the top toolbar, see Using the Top Toolbar in the *Amazon Lumberyard User Guide*.

Next: 3: Entity Outliner (p. 9)

3: Entity Outliner

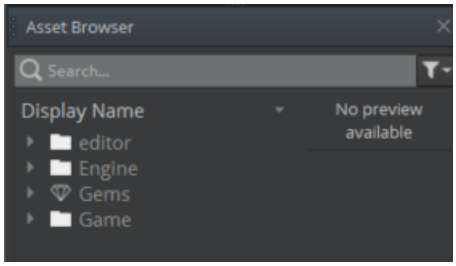


The **Entity Outliner** lists every entity and slice that is in your level. From here, you can find, select, lock, hide, and disable any entity or slice in your level. You can also create new entities, create child entities, create or instantiate slices, and more.

For more information about the **Entity Outliner**, see Entity Outliner in the *Amazon Lumberyard User Guide*.

Next: 4: Asset Browser (p. 10)

4: Asset Browser



The **Asset Browser** displays the file system for your project. You can find all assets that are in your project file system, such as 3D meshes, textures, and sound files. You can drag assets from the **Asset Browser** into the **Entity Outliner** or into the viewport.

For more information about the **Asset Browser**, see Asset Browser (Preview) in the *Amazon Lumberyard User Guide*.

Next: 5a: Perspective Viewport (p. 10)

5a: Perspective Viewport

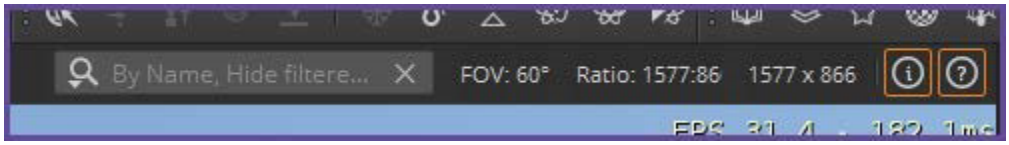
The **Perspective** viewport displays the 3D environment of your game. Here, you can navigate around your level and select objects and entities to manipulate.

See the section called "Navigating in the Viewport" (p. 14) later in this section to practice navigating in the viewport.

For more information about the **Perspective** viewport, see Using the Viewport in the *Amazon Lumberyard User Guide*.

Next: 5b: Perspective Header (p. 10)

5b: Perspective Header

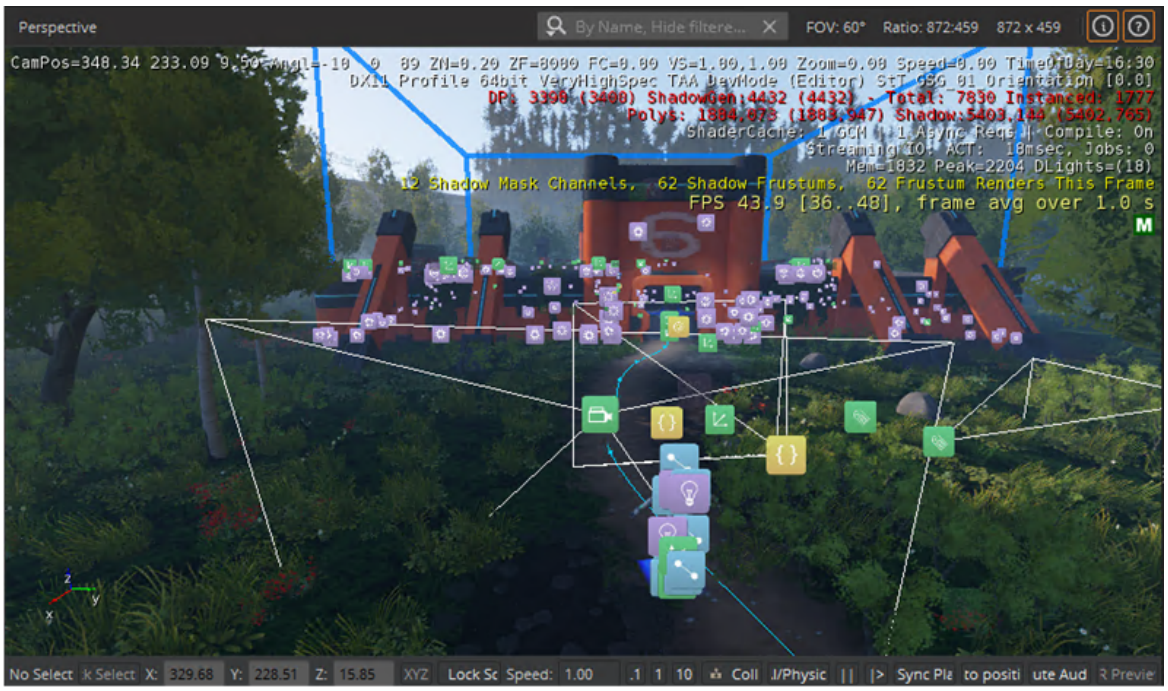


The **Perspective** header in the upper right corner of the **Perspective** viewport features displays options and a search bar to filter objects within the viewport.

Click the "i" icon  to change the level of information displayed in the viewport. Click it multiple times to cycle through the different levels of information.

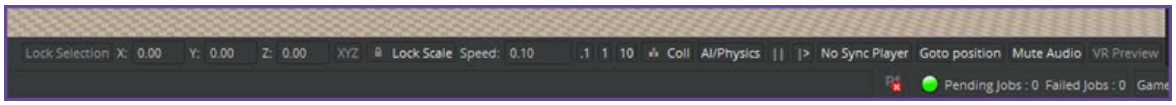
Click the "?" icon  to show or hide helpers. Helpers are the bounding boxes, icons, and lines that help you recognize different types of objects in your level.

Turn helpers off to get a clearer picture of what your game looks like. For example, the picture on the top shows the orientation level with helpers turned on. The picture on the bottom shows the same level with helpers turned off.



Next: 5c: Bottom Toolbar (p. 12)

5c: Bottom Toolbar

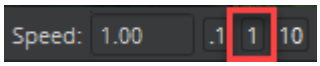


The toolbar at the bottom of the **Perspective** viewport displays information for objects selected in your viewport, such as its XYZ position in the level. You can also speed up or slow down navigation speed, mute audio, go to a specific position, and more.

The default navigation speed is `0.10`. Try speeding up and slowing down your navigation.

To change the viewport navigation speed

1. First, become familiar with the default navigation speed by pressing **W**, **S**, **A**, and **D** in turn to strafe forward, backward, left, and right.
2. In the bottom toolbar, click the **Speed** setting `1`.

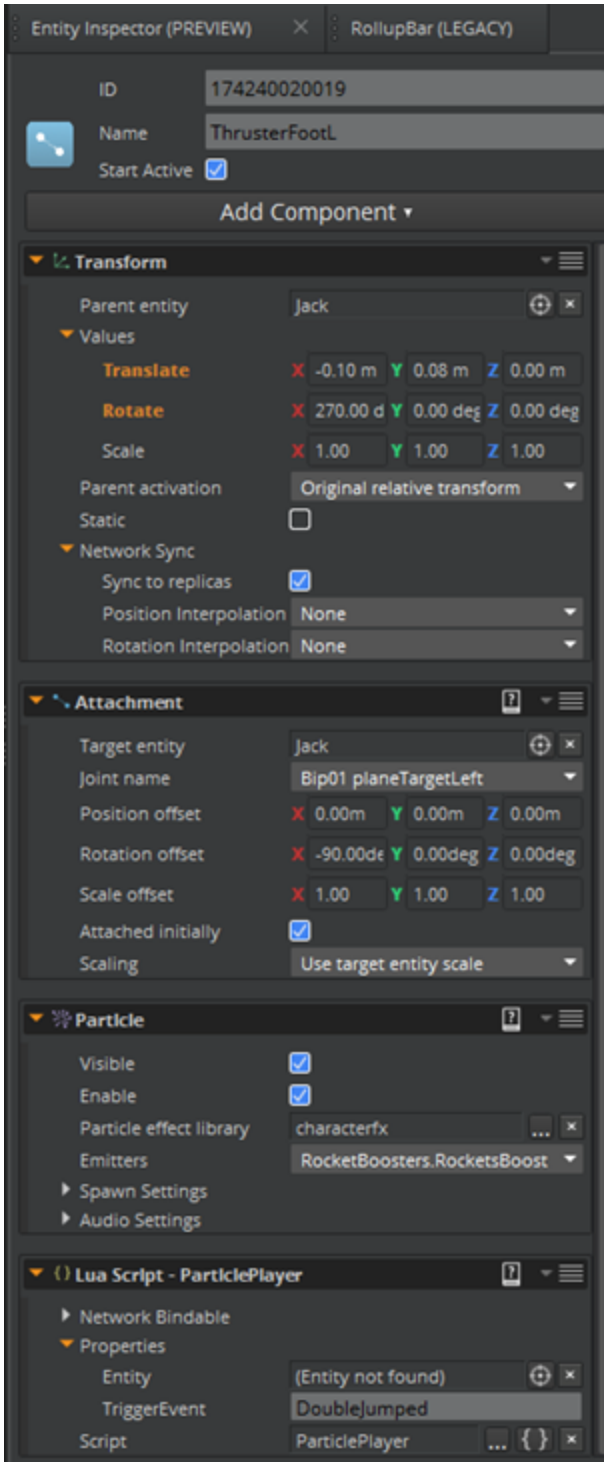
   Click in your viewport and press **W**, **S**, **A**, and **D** in turn to test the new, faster setting.
3. In the bottom toolbar, click the **Speed** setting `10`.

   Click in your viewport and press **W**, **S**, **A**, and **D** in turn to test this very fast setting, which you can use to navigate quickly over long distances.
4. You can also input custom speeds. Try a speed of `0.5`, for example. Try a speed of `5`. Experiment with the navigation speeds.

For more information about the other settings in the bottom toolbar, see Using the Bottom Toolbar in the *Amazon Lumberyard User Guide*.

Next: 6: Entity Inspector (p. 13)

6: Entity Inspector

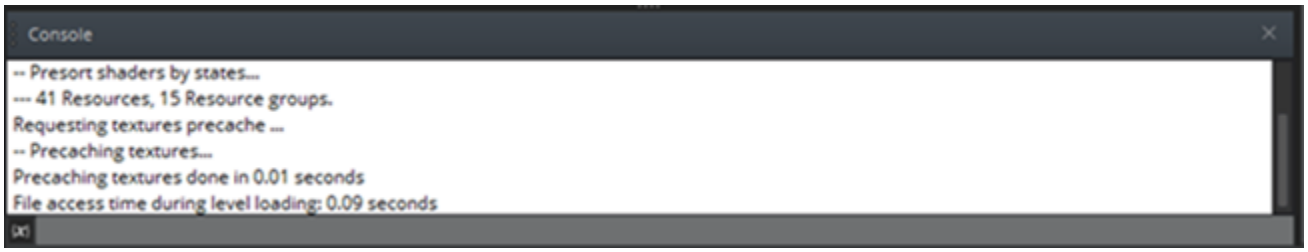


The **Entity Inspector** (on the right side of the viewport) displays all the components for the currently selected entity in the **Entity Outliner** (on the left side of the viewport). You use the **Entity Inspector** to add and remove components and to modify settings on components. This gives the entity customized abilities, such as triggering a particular action, making sounds, colliding with other objects, and so on.

For more information about the **Entity Inspector**, see **Entity Inspector** in the *Amazon Lumberyard User Guide*.

Next: 7: Console (p. 14)

7: Console



By default, only the **Console**'s title bar is visible. To view the console, which is below the **Perspective** viewport, drag the upper edge of the **Console** title bar up to expand it. The console displays input and output data for Lumberyard Editor and any custom code that is written for your level. Advanced users can type console commands into the gray bar on the bottom of the **Console** window.

For more information about the console, see **Using the Console Window** in the *Amazon Lumberyard User Guide*.

Next: Navigating in the Viewport (p. 14)

Navigating in the Viewport

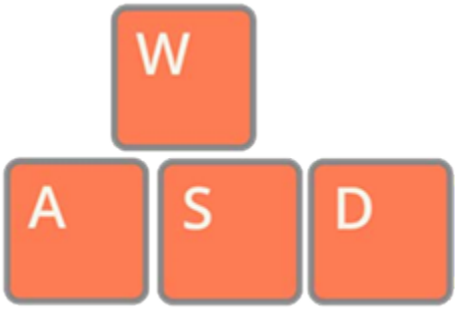
Lumberyard uses mouse and keyboard navigation controls that are familiar to most gamers.

Like most first-person shooter (FPS) games on a PC, a series of keyboard strokes controls forward, backward, left, and right strafing (movement without changing viewing direction).

A combination of mouse actions can zoom, change viewing direction, and pan.

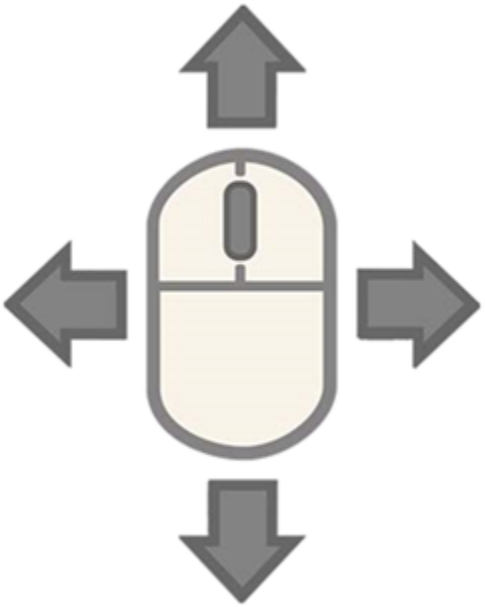
To navigate in the viewport

Click in the viewport to bring the focus to the viewport, and then experiment with the following keyboard and mouse navigation controls.



- **W** – Strafe forward
- **S** – Strafe backward
- **A** – Strafe left
- **D** – Strafe right

Tip
To increase strafe speed, hold **Shift** while pressing **W**, **S**, **A**, or **D**.



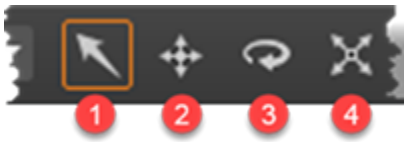
- Hold right mouse button (RMB) + direction – Look direction
- Hold middle mouse button (MMB) + direction – Pan direction
- Click left mouse button (LMB) – Select
- Drag left mouse button (LMB) – Marquee select
- **Alt** + right mouse button (RMB) + drag up or down – Rapid strafe forward and backward
- **Alt** + middle mouse button (MMB) + direction – Look direction

For a full list of keyboard shortcuts, see Using Keyboard Shortcuts in the *Amazon Lumberyard User Guide*.

Next: Manipulating Objects (p. 15)

Manipulating Objects

After you place an object in your viewport, you can perform the following actions on it:



1. Select it so that you can make adjustments to it.
2. Move it to another location.
3. Rotate it.
4. Scale it (make it larger or smaller).

You can quickly select any of these tools by using the following hotkeys:

- 1 – **Select**

- 2 – **Move**
- 3 – **Rotate**
- 4 – **Scale**

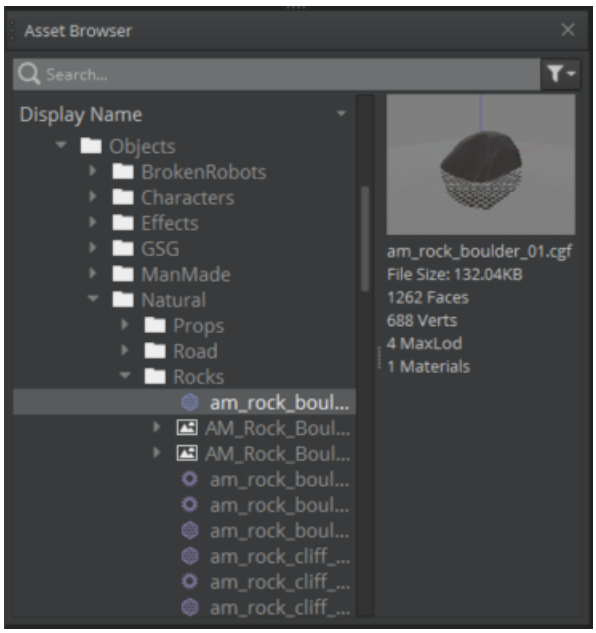
Use the following procedure to place a new object in your viewport.

Although there are a few boulders already in the orientation level, you develop your skills by placing a new one.

To place a new boulder in your viewport

1. In the **Asset Browser**, navigate to `Game\Objects\Natural\Rocks`.

   Select `am_rock_boulder_01.cgf`.



2. Drag from the **Asset Browser** directly into your viewport.

In the next sections, you practice manipulating your object using these tools.

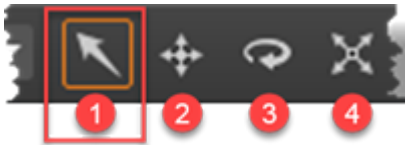
Next: Selecting an Object (p. 17)

Selecting an Object

You can select any object in your viewport that is not currently locked or hidden.

To select the boulder object

1. If your boulder is currently selected from the previous tutorial, click elsewhere in your viewport to unselect it.
2. Press **1** on your keyboard to activate the **Select** tool. Or you can click the **Select** icon on the top toolbar.



3. In the viewport, move your pointer over the boulder object.

   The boulder object becomes highlighted and the cursor changes to a **+** sign. Click to select it.

   Tip
   If you do not see a highlight on your boulder when you move your mouse pointer over it, you might have helpers turned off. To turn helpers on, click  in the upper right corner of the viewport.

When your object is selected and the **Select** tool is active, a gizmo appears on your object. The gizmo for **Select** is a set of three lines—one for each of the three axes (X, Y, Z).

Next, you learn how to move your object.

Next: Moving an Object (p. 19)

Moving an Object

You can move the currently selected object in your viewport. When you move an object with the **Move** tool, you can choose to move the object along the following:

- **An axis (X, Y, or Z)** – X and Y can be thought of as latitude and longitude, and Z as elevation.

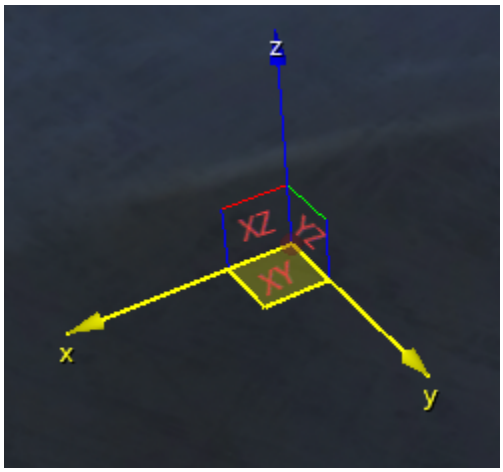
  For example, if you move along X, you move your object forward and backward only.

  If you move along Y, you move your object left and right only.

  And if you move along Z, you move your object up and down only.
- **A plane (XY, YZ, XZ)** – Move your object along two axes, or in two different directions, without moving it in the third.

  For example, if you choose the XY plane, you can move the object in the X and Y directions but not the Z direction. You can change its position anywhere on a flat landscape.

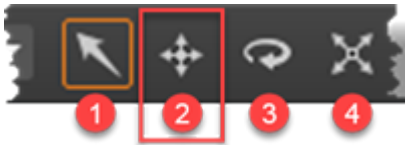
  If you choose the YZ or XZ plane, you can move the object as if it were against a wall. YZ and XZ can be thought of as walls that are perpendicular to one another.



To choose the axis or plane along which you want to move the object, move your mouse pointer over the gizmo until the plane or axis that you want is highlighted. Click to select it and then drag to move it.

To move a selected object

1. If your boulder is not still selected from the previous tutorial, select it.
2. Press **2** on your keyboard to activate the **Move** tool. Or you can click the **Move** icon on the top toolbar.

   When your boulder is selected with the **Move** tool, it looks like the following image.

3. Move your mouse pointer over the gizmo until the Z axis is selected. Click to select and then drag upward. Your boulder appears to float.

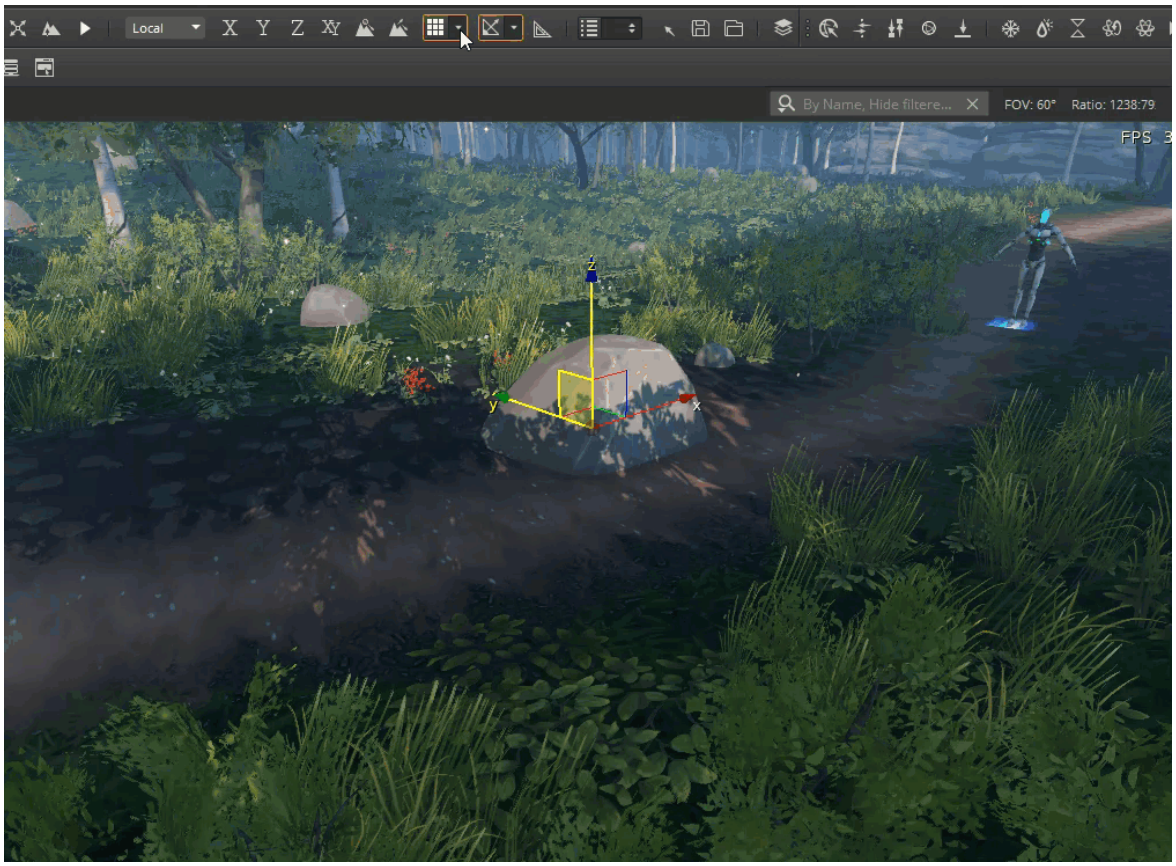
   Move it back down to sit partially beneath the ground.

4. Move your cursor over the gizmo until the small square indicating the XY plane is selected. Drag it to a different location along the ground.

Next: Snapping to Grid (p. 21)

Snapping to Grid

The **Snap to Grid** feature is on by default. With this feature on, you can move an object by aligning it to points along a grid. You can make the grid smaller or larger depending on your needs. The smaller your grid, the finer control you have when moving your objects. The following image shows the boulder being moved along a large grid.



To turn on snap grid and adjust grid size

1. If your boulder is not still selected from the previous tutorial, select it.
2. Select the **Move** tool.
3. In your top toolbar, locate the **Snap to Grid** icon. If it has an orange border around it, it is on. If it is off, click to turn it on.

   For example, in the following image, **Snap to Grid** is on.

4. Click the small arrow on the icon. Choose **4**.

5. Move the boulder in your viewport. Notice how it seems to jump in increments rather than move smoothly.
6. Turn off **Snap to Grid** and move your boulder again. Notice how it now moves smoothly for refined placement.

Next, you learn how to rotate your object.

Next: Rotating an Object (p. 23)

Rotating an Object

You can rotate the currently selected object in your viewport. When you rotate an object with the **Rotate** tool, you can rotate around the X, Y, or Z axis. Think of the rotational axes as a pole around which the object spins or rotates. For example, rotating around the Z axis would be like a car driving in circles around a pole in the ground. A large outer circle also surrounds the entire gizmo. Select and drag this circle to rotate the object in relation to the screen display.

You can distinguish which circles belong to which axis by their color marking (red circle = red X axis, green circle = green Y axis).

To rotate a selected object

1. If your boulder is not still selected from the previous tutorial, select it.
2. Press **3** on your keyboard to activate the **Rotate** tool. Or you can click the **Rotate** icon on the top toolbar.

When your boulder is selected with the **Rotate** tool, it looks like the following image.

3. Move your mouse pointer over the gizmo until the Z axis is selected. Click to select and then drag left and right. Your boulder spins like a top.

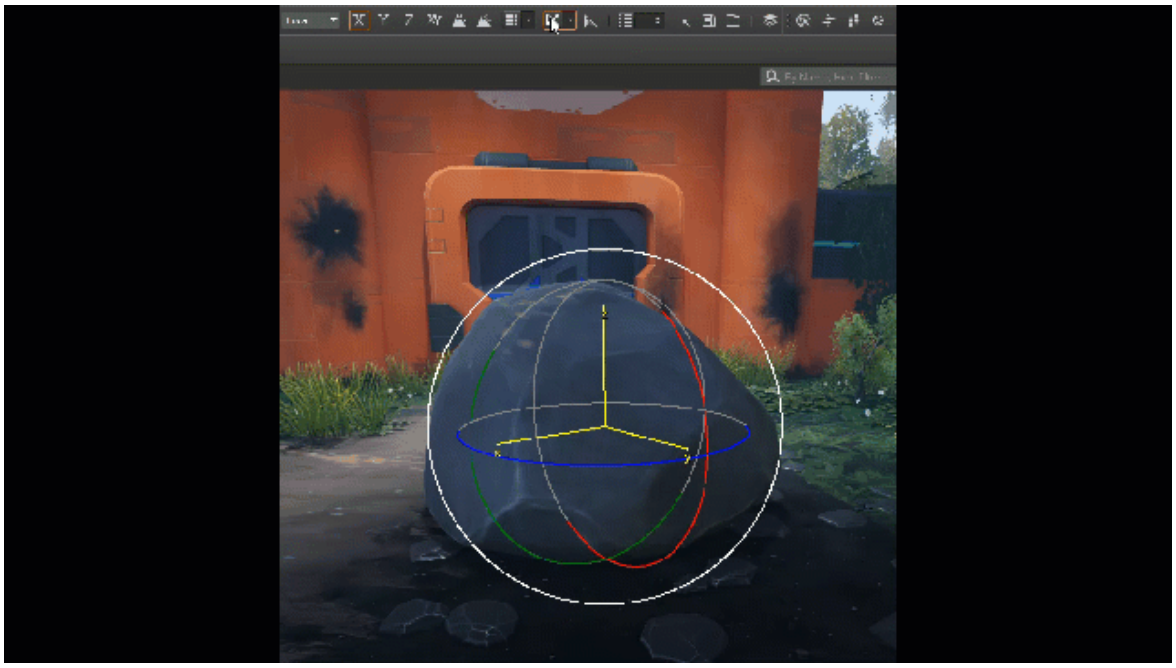
Experiment with rotating along the other axes.

Next: Snapping to Angle (p. 26)

Snapping to Angle

The **Snap Angle** feature is on by default. With this feature on, you can rotate an object by incremental degrees. You can make the snap angles smaller or larger depending on your needs. The smaller your snap angle, the finer control you have when rotating your objects. The following image demonstrates **Snap Angle** both on and off.

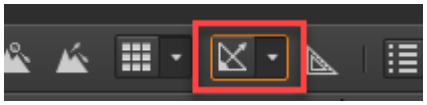
Snap Angle (on by default) sets the incremental degrees of rotation that an object snaps to when you rotate it. To turn **Snap Angle** off or on, click the icon. To customize the **Snap Angle** degrees of rotation, click the down arrow to the right of the icon. Adjust this value to increase or decrease the degree of rotation with each snap.



To turn on Snap Angle and adjust the grid size

1. If your boulder is not still selected from the previous tutorial, select it.
2. Select the **Move** tool.
3. In your editor toolbar, find the **Snap Angle** icon. If it has an orange border around it, it is on. If it is off, click to turn it on.

   In the following image, **Snap Angle** is on.

4. Click the small arrow on the icon. Choose **45**.

5. Rotate the boulder in your viewport. Notice how it seems to rotate in increments rather than smoothly.
6. Turn off **Snap Angle** and rotate your boulder again. Notice how it now rotates smoothly for refined placement.

Next, you learn how to scale your boulder to make it smaller or larger.

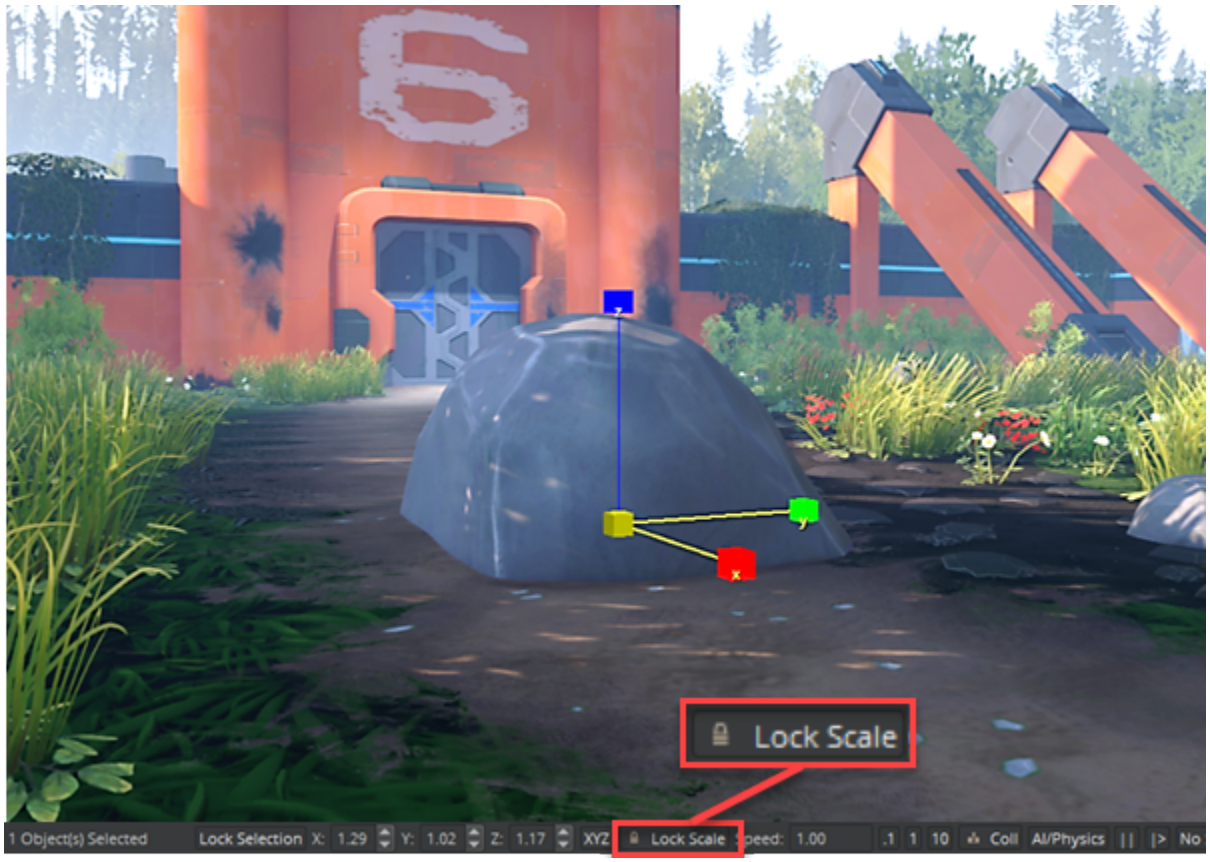
Next: Scaling an Object (p. 27)

Scaling an Object

To scale an object means to make it larger or smaller.

By default, the proportions of your object are locked. This means that whichever axis you choose to scale, your object grows or shrinks on all axes at the same time. When you scale it up, it simply looks like a bigger version of itself. When you scale it down, it looks like a smaller version of itself.

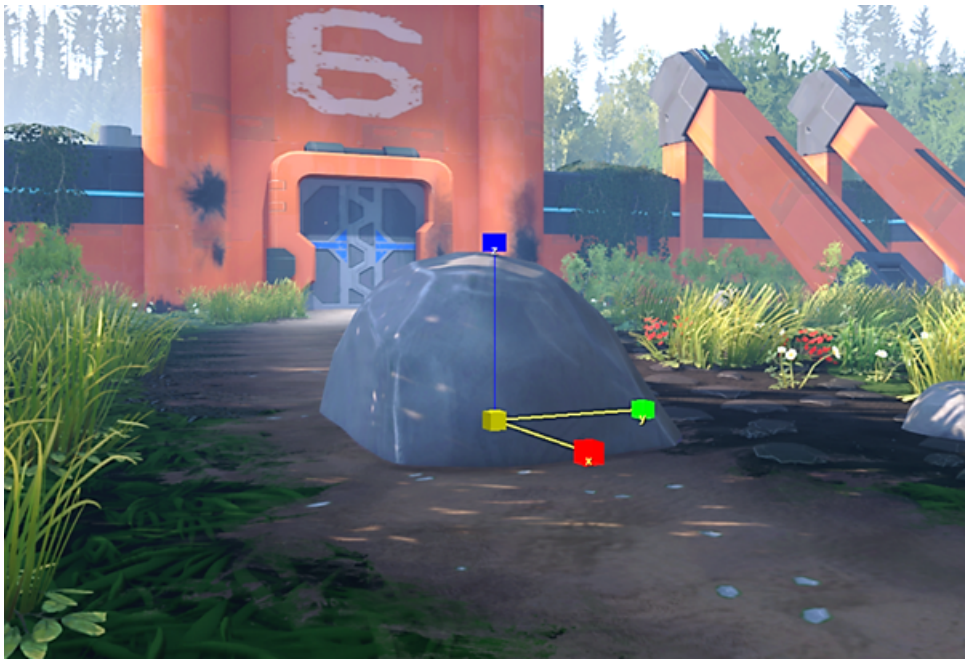
You can, however, make an object taller without affecting its width, or wider without affecting its height. To do this, click the **Lock Scale** toggle at the bottom of the perspective viewport. This unlocks the proportions so that you can scale just one axis.



To scale a selected object

1. If your boulder is not still selected from the previous tutorial, select it.
2. Press **4** on your keyboard to activate the **Scale** tool. Or you can click the **Scale** icon on the editor toolbar.

3. Move your cursor over the gizmo until it is highlighted. Drag up or down to make the boulder larger or smaller.

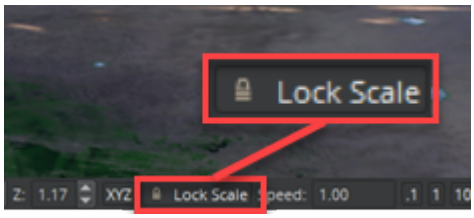


4. To scale your boulder along one axis only, such as making it taller without making it wider:

   a. At the bottom of the viewport, in the toolbar, click the **Lock Scale** toggle.

      This unlocks the proportions so that you can adjust one axis without affecting the others.

      If **Lock Scale** has an orange border, that means it is on. If it is on, click it once to turn it off.



   b. Move your pointer over the scale gizmo until the Z axis is highlighted. Drag up to make the boulder taller.

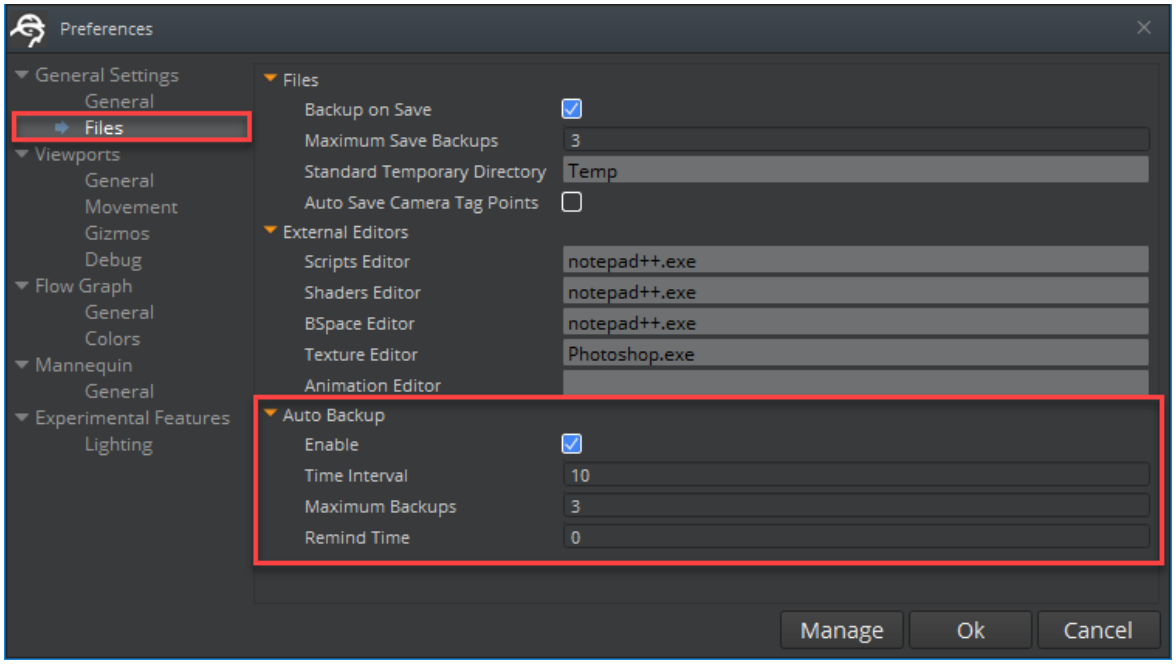
      Experiment with the X and the Y axes as well.

Next: Configuring Auto Backup (p. 30)

Configuring Auto Backup

Lumberyard's **Auto Backup** feature is off by default. **Auto Backup** saves your level file automatically at specified time intervals. This prevents the loss of your progress if an unexpected event causes a crash or freeze while you are working in your level.

To enable this feature, in the main menu, choose **Edit**, **Editor Settings**, **Global Preferences**. Under **General Settings**, choose **Files**. In the right panel, you can customize your **Auto Backup** settings.



Next: 2: Building the Environment (p. 32)

2: Building the Environment

In this tutorial shows you how to do the following:

Next: Creating a Level (p. 32)

Creating a Level

Lumberyard currently works with level files as its primary system for organizing game play content. Each level file contains the level file along with support files that hold game data, scripts, terrain, and any additional references required for the level file to run in-game.

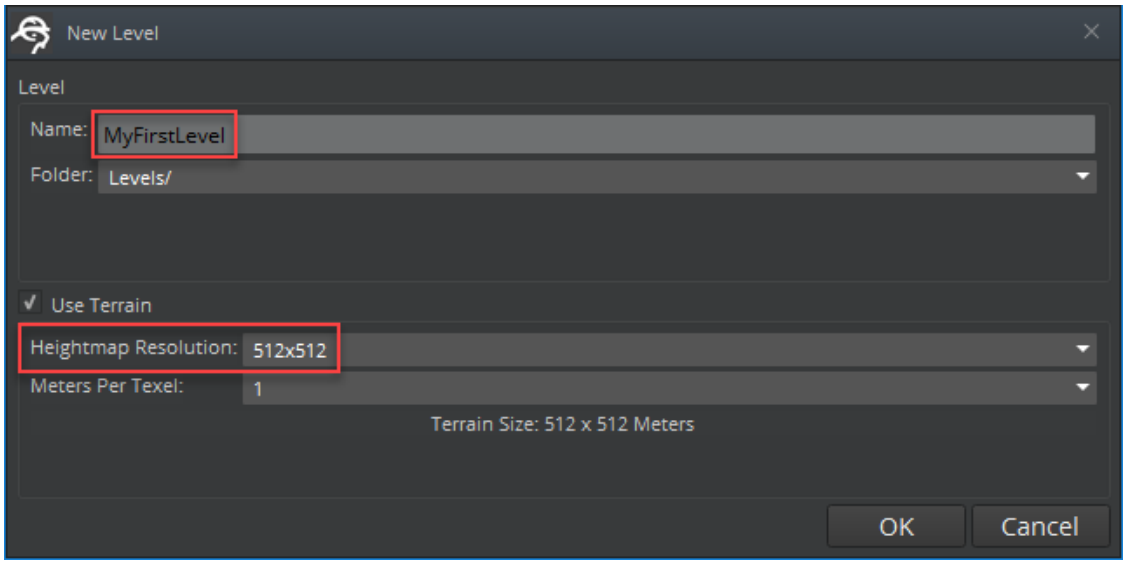
You start by creating a new level within the Starter Game project, which you are already in if you have followed the tutorial up to this point.

When you create a level, Lumberyard creates a level file within the project directory. In this case, that is `dev\StarterGame\Levels\<levelname>`. Staying in the Starter Game project directory means that you have easy access to all the assets that are associated with the project. You'll use many of these assets throughout this tutorial series. Such assets include prebuilt 3D objects, scripts, textures, and much more.

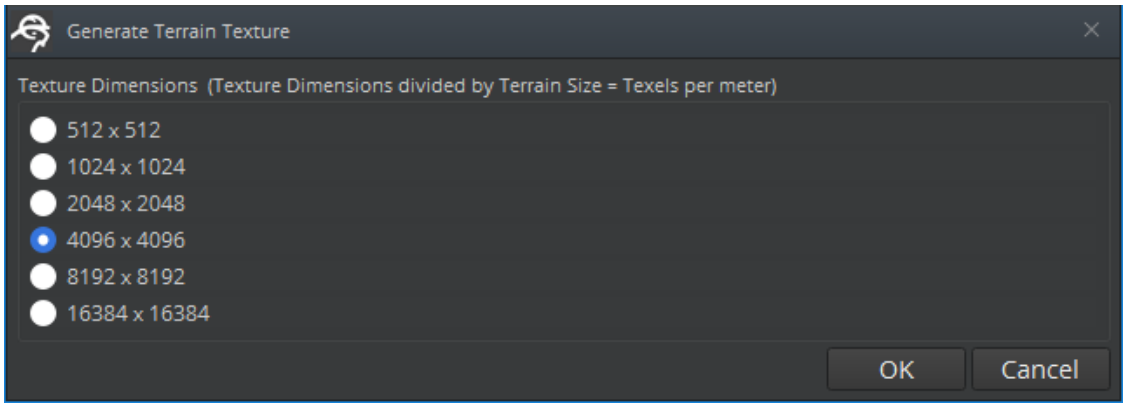
To create a new level

1. You can create a new level in three ways. Use one of the following methods to create a new level:
   - From the main menu: Choose **File**, **New**.
   - With hot keys: Press **Ctrl+N**.
   - From the **Welcome** window: Click **New Level**.
2. For **Name**, type **`MyFirstLevel`**.

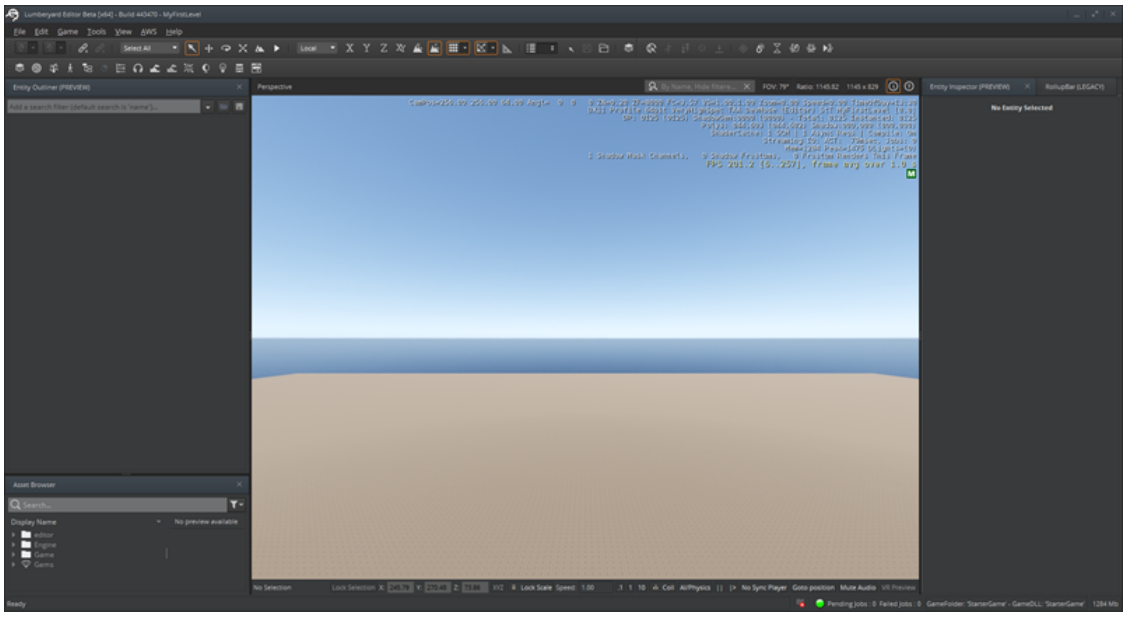
   Level names can consist of alphanumeric characters. You cannot use spaces or special characters except hyphen [-] and underscore [_].
3. Set **Heightmap Resolution** to **`512x512`**. Click **OK**.



4. Select the **4096x4096** texture dimension. This value is in pixels. Click **OK**.



You have an empty level with a flat terrain mesh in your level file. You'll start with this empty file to build your game sample.



Next: Importing a Heightmap (p. 34)

Importing a Heightmap

You can import a prebuilt heightmap to quickly place mountains and valleys in your level. Lumberyard supports using heightmaps generated by third party tools such as World Machine. A heightmap is a raster image that stores values, such as surface elevation data, for display in your Lumberyard game levels.

Lumberyard uses heightmaps in the following ways:

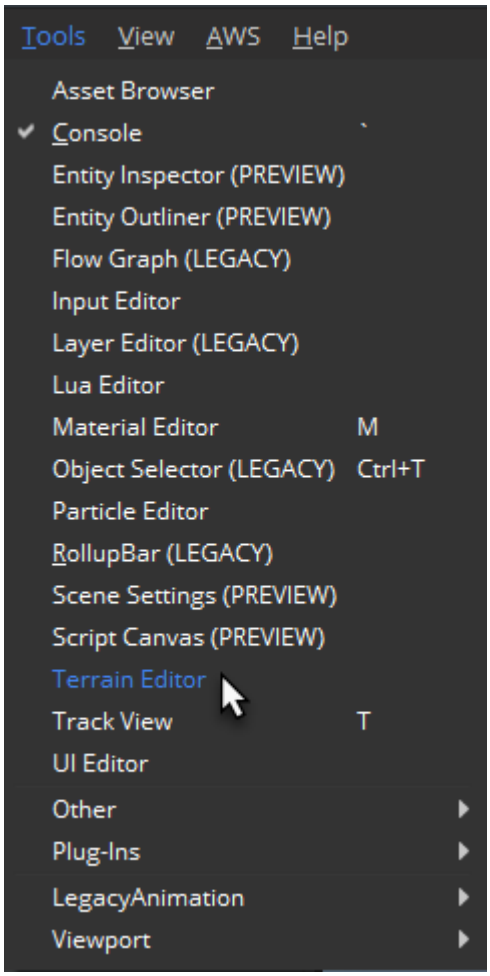
- Bump mapping – Uses 3D data to create shadows in materials
- Displacement mapping – Determines the geometric position of points over the textured surface
- Terrain mapping – Converts heightmap into a terrain mesh.

Lumberyard also has a set of terrain generation and manipulation tools that you can use to create terrain from scratch.

For this tutorial, you'll create terrain by importing a heightmap.

To import a heightmap

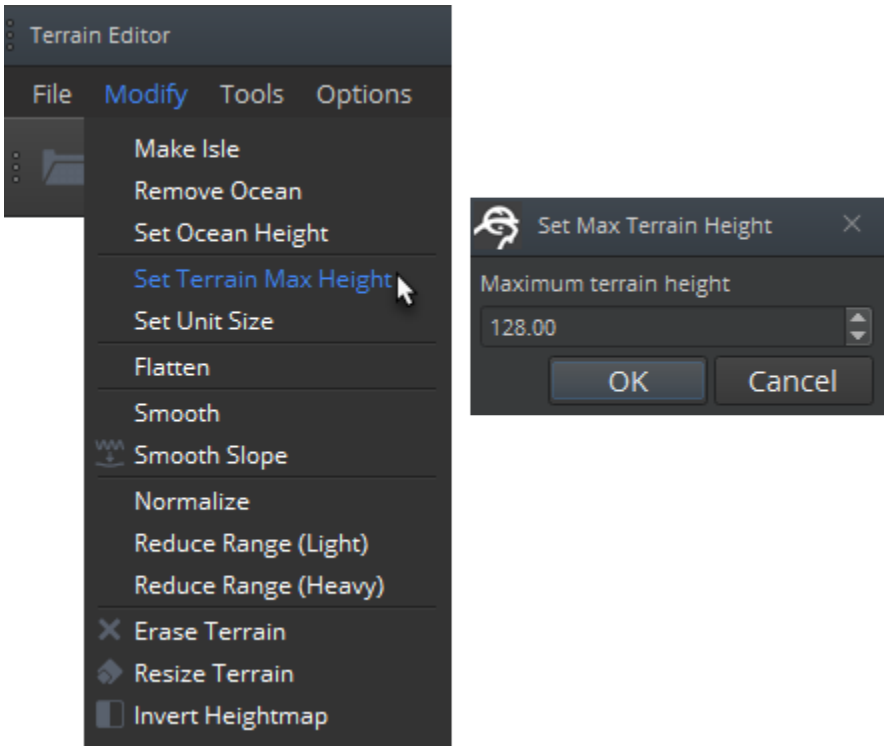
1. Open the **Terrain Editor** by choosing **Tools**, **Terrain Editor**.



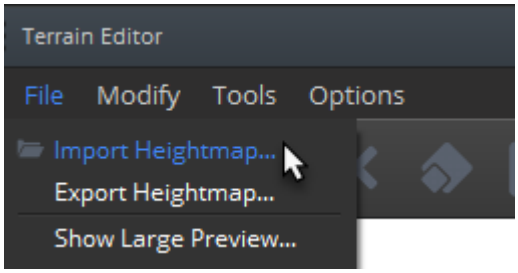
2. Set the maximum height of your terrain by choosing **Modify**, **Set Terrain Max Height**.

   Enter **128**.

   This means that the tallest point in your terrain will be no more than 128 meters.



3. Choose **File**, **Import Heightmap**.



Navigate to `\dev\StarterGame\Textures\Heightmaps` and select `FTUE_heightmap_test.tif`.

In your viewport, you see the initial creation of valleys and mountains as determined by the heightmap.

4. Press **Ctrl+S** to save your level.

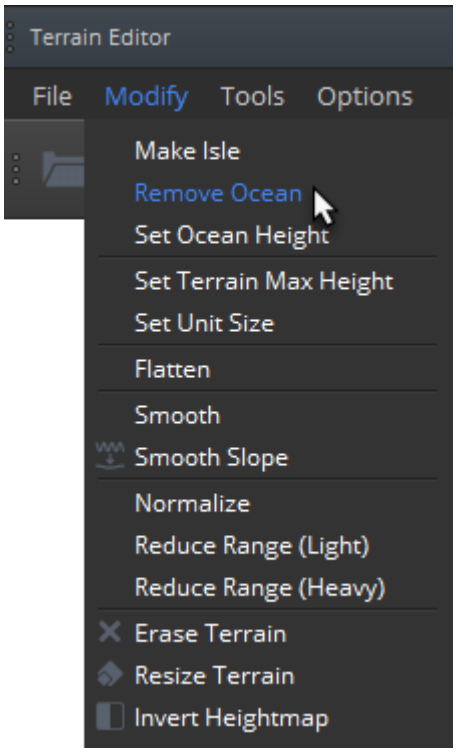
Next: Removing the Ocean (p. 37)

Removing the Ocean

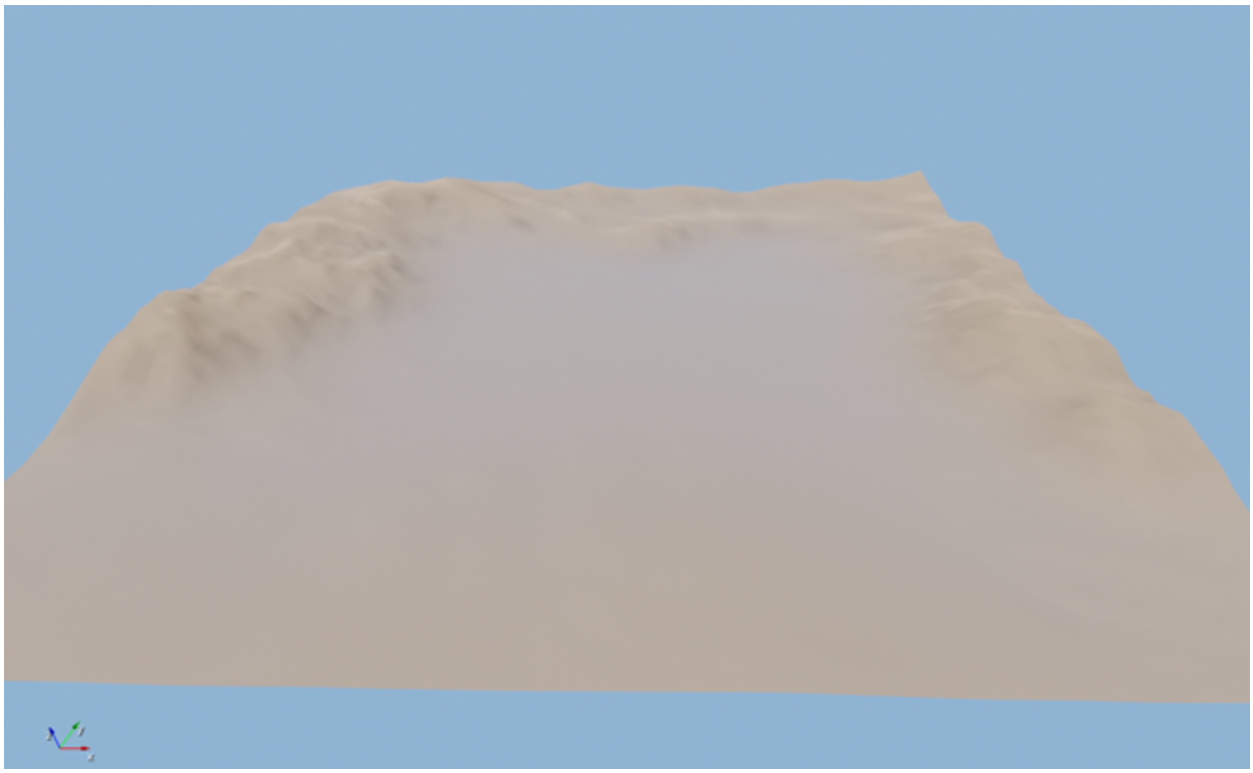
At this point in the tutorial, most of the land mass is under the ocean. To remedy this, you could lower or hide the ocean. In this tutorial, you learn how to completely remove the ocean from your level.

To remove the ocean

1. If **Terrain Editor** is not open, open it by choosing **Tools**, **Terrain Editor**.
2. Choose **Modify**, **Remove Ocean**.



After your ocean is removed, your level should look similar to the following image.



3. Press **Ctrl+S** to save your level.

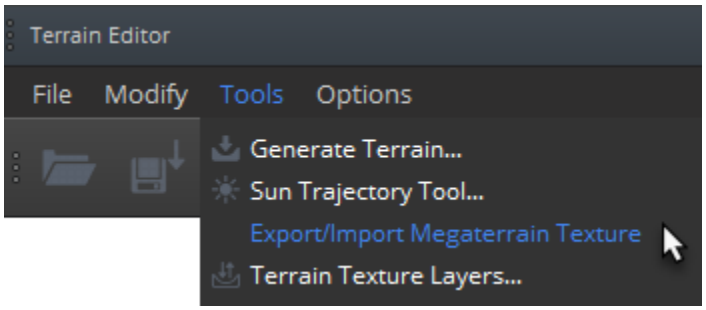
Next: Importing a Megaterrain Texture (p. 39)

Importing a Megaterrain Texture

In this tutorial, you import a megaterrain texture. A megaterrain texture is a diffuse texture that covers the entire terrain. This texture is visible from a distance and presents sweeping vistas. As the camera moves closer to the terrain, the megaterrain texture is replaced by more detailed terrain textures.

To import a megaterrain texture

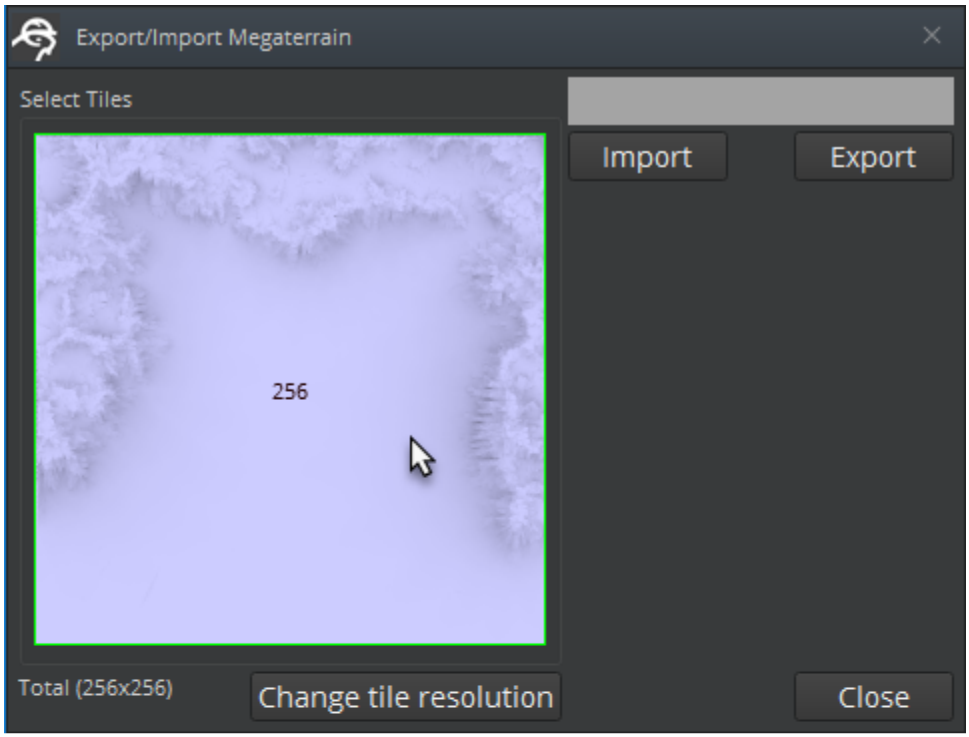
1. If **Terrain Editor** is not open, open it by choosing **Tools**, **Terrain Editor**.
2. Choose **Tools**, **Export/Import Megaterrain Texture**.

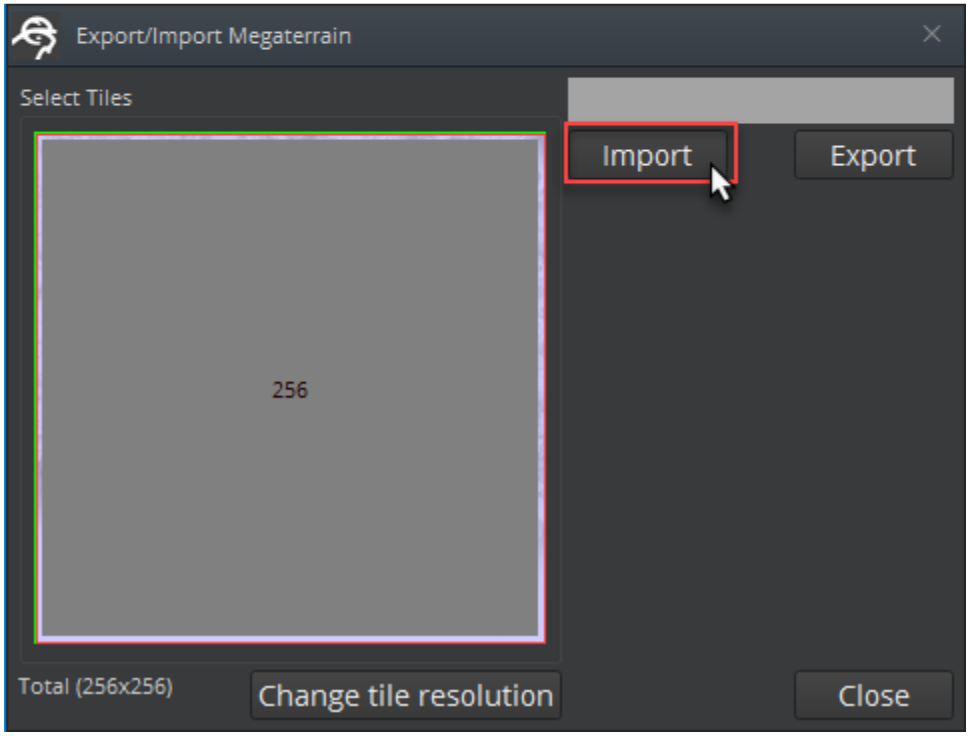


3. Beneath **Select Tiles**, click the heightmap image. The box turns gray.

   Click **Import**.

   Note
   Only one tile exists. This is a result of setting the initial tesselation, or **Meters per Texel** setting, to **1** when you created the level.





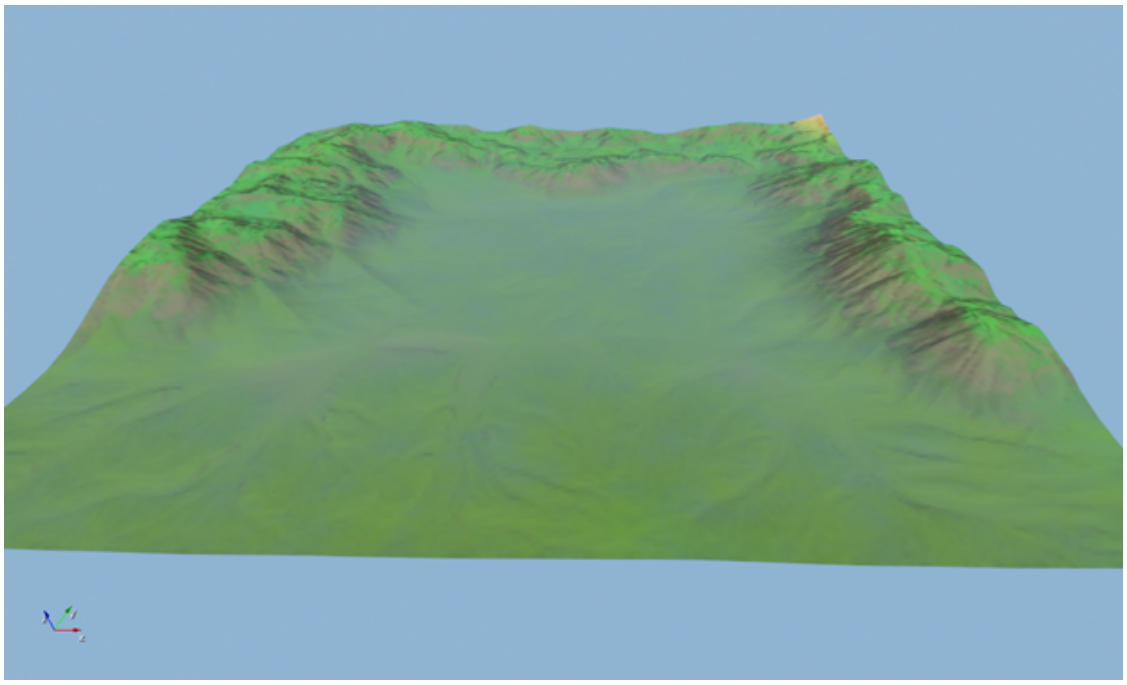
4. Navigate to `\dev\StarterGame\Textures\Terrain` and select `FTUE_MegaTexture_02.bmp`. Click **Open**.

   Lumberyard takes a few moments to generate the texture on the terrain.

5. Click **Close** in the **Export/Import Megaterrain** window.

Close the **Terrain Editor**.

Observe that your terrain now has shades of green and brown to represent grass and soil, as well as shadows to give the landscape realism.



6. Press **Ctrl+S** to save your level.

Next: Placing the Player Character (p. 41)

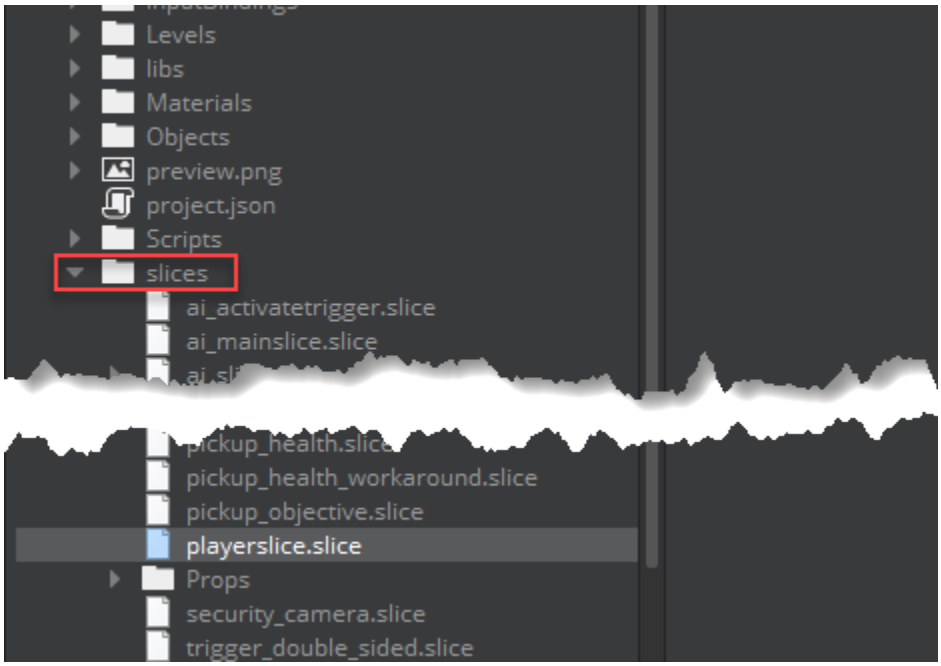
Placing the Player Character

In this tutorial, you place a player character so that you can move around and explore the level from a player's perspective. To do this, you drag the character controller slice from the **Asset Browser** into your viewport. Slices are described in a later tutorial.

The character controller is Jack the robot, which you played in the previous tutorial in the orientation level. You also place the camera manager slice, which improves player input controls from the keyboard. The camera manager slice also helps to manage the switching of additional cameras, which you place later in this tutorial series.

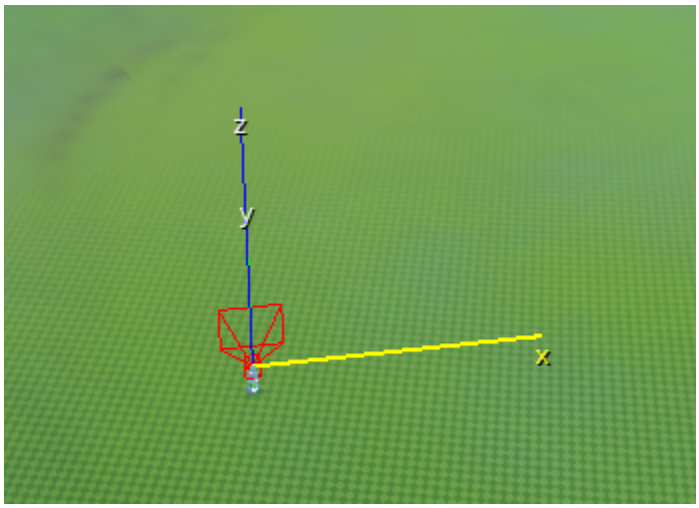
To place the character controller and camera manager

1. The **Asset Browser** automatically opens in Lumberyard's default layout, in the lower left corner. If you don't see it, choose **Tools**, **Asset Browser** to open it.
2. In the **Asset Browser**, navigate to `Game/slices`.

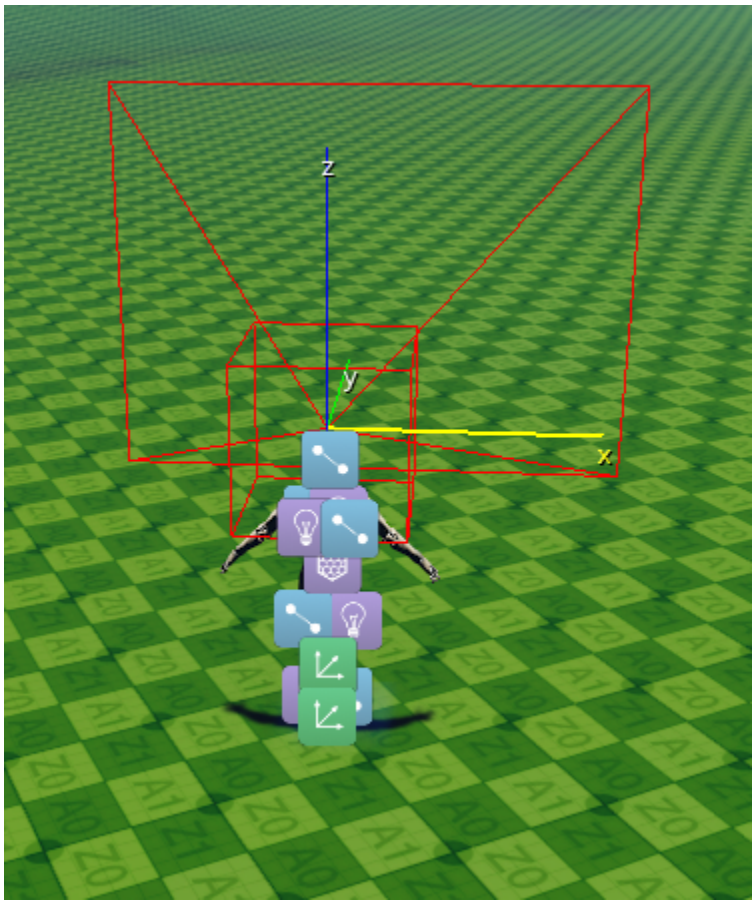


3. Drag `playerslice.slice` into the viewport.

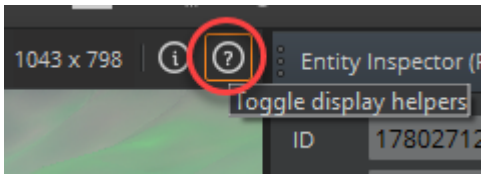
   This places the robot character into your level, which you can control.



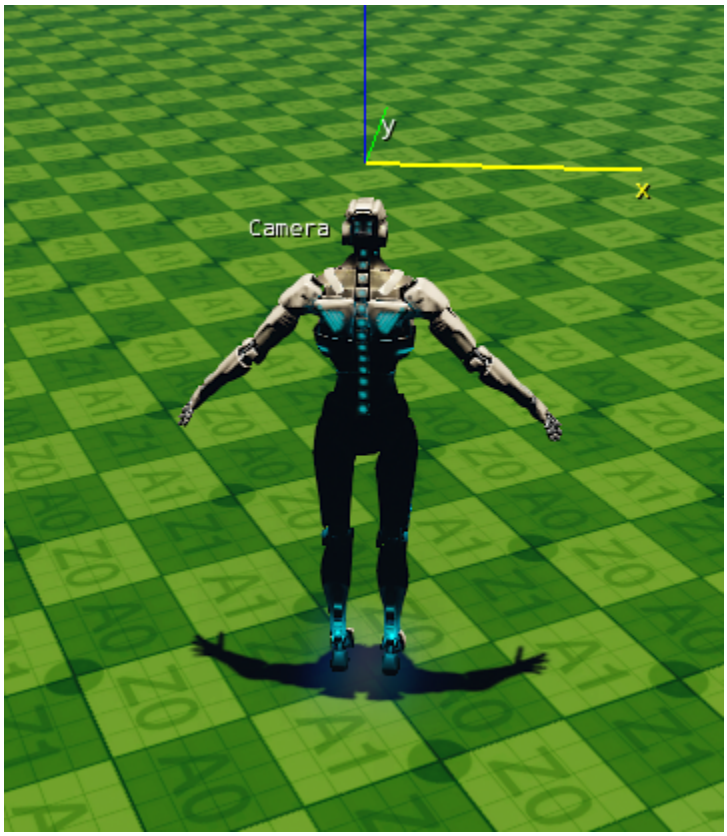
   Zoom in to view the character in detail.



4. Because the character controller is composed of a number of entities, all the helper icons are visible. To hide these, click the hide helpers toggle in the top right of the viewport.

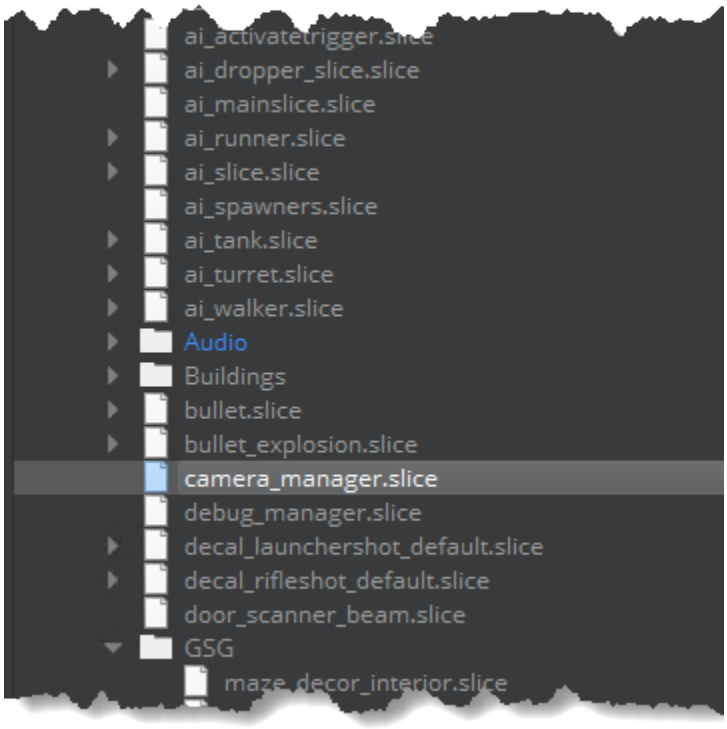


You can now see the character's details clearly.



5. To place the camera manager, drag `camera_manager.slice` from the same directory into your viewport.

   The camera manager is not required to be situated in any particular place in your level. However, for organizational purposes, you can place it near the player slice for future reference.



6. Press **Ctrl+S** to save your level.
7. Click **Ctrl+G** to play the game and explore the terrain. You can control the character using **W**, **A**, **S**, **D** keyboard controls, and click to shoot. You can also use a PC game controller if preferred.

Next: 3: Gathering your Building Blocks (p. 46)

Additional Information

- Component Entity System
- Component Reference
- Entity Outliner
- Entity Inspector
- Asset Browser

3: Gathering Your Building Blocks

In this tutorial, you use component entities and slices to begin building the maze. In a later tutorial, you use Lumberyard's component entity system to place game elements such as meshes, lights, sounds, trigger areas, cameras, and more. This is a key workflow in Lumberyard. The slices system is Lumberyard's version of what other game engines call *prefabs*. You can use slices, which are easily updated, to quickly develop your project.

When placing entities, you typically begin with an empty entity. You then add components to it to define the appearance, behavior, and properties of that entity.

You will learn how to do the following:

- Create an empty entity and name it.
- Add a mesh component to the entity.

  You use the mesh component to customize the entity's appearance by selecting a premade mesh, which is a 3D model file.
- Add collider and physics components to the entity.

  Characters can interact with an entity only if collider and physics components are present. Otherwise, characters and other objects can pass through the entity without colliding.
- Use the **Asset Browser** to add an entity to your level.
- Duplicate an entity.
- Group existing entities beneath a new parent entity.

The component entity system is Lumberyard's approach to composing complex entities out of simpler entities.

You can add components to an entity to build the behavior you want. You can then edit component settings in the editor and create scripts to change and extend the behavior of an entity.

When you group or nest entities, you can save the group as a slice. This creates a `.slice` file within your game project directory. You can use Lumberyard's **Asset Browser** to drag that slice into your level. This is called instantiating a slice. You can do this as many times as you want. You can also nest slices within other slices to create more complex slices.

When you make changes to a slice instance, you can save those changes to all instances of that slice in your project. However, you can also customize any part of any slice instance (except the source slice) and keep the customization in just that instance.

Lumberyard uses components to allow system-level features of the engine to be extended by writing new core components. For example, the character manager component gives animated characters a tick for each frame and acts as a bridge between the physics and animation systems. Components are a critical feature in Lumberyard's modular game engine architecture.

Topics

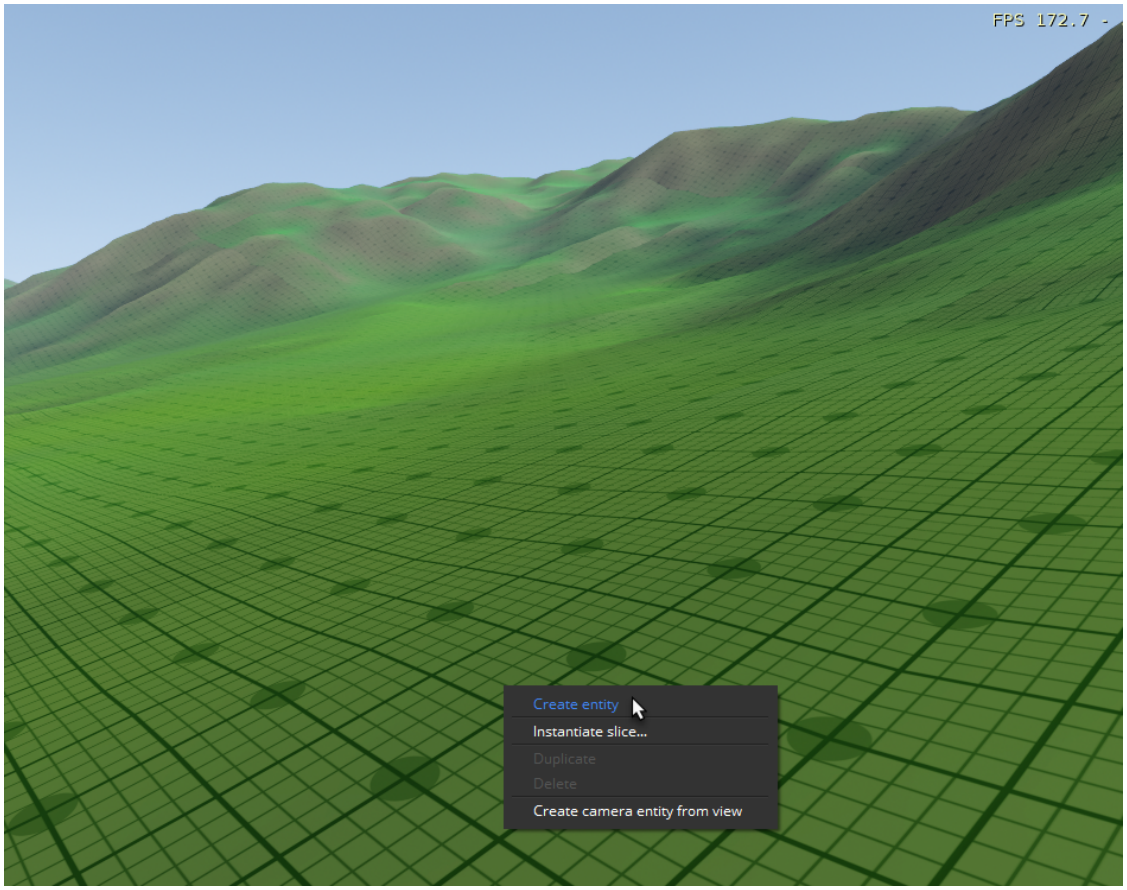
Next: Creating an Entity (p. 47)

Creating an Entity

In this tutorial, you create and name a new entity. You'll name this entity 'doorway' and, in the next tutorials, add components to it that make it look like a doorway and give it collision data.

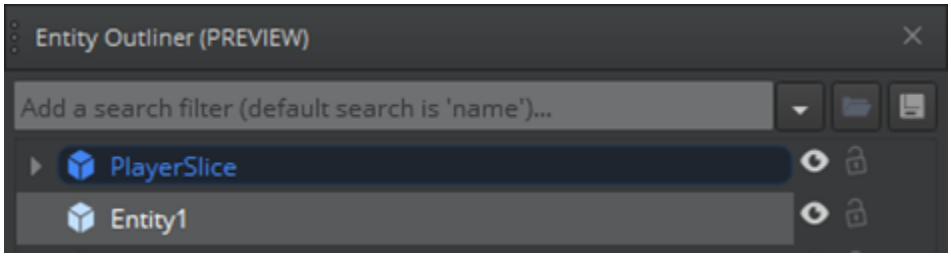
To create a doorway entity

1. First, you create a new, empty entity. To do this, in Lumberyard's perspective viewport, right-click and choose **Create entity**.



This action creates an empty entity, which means it currently has no components.

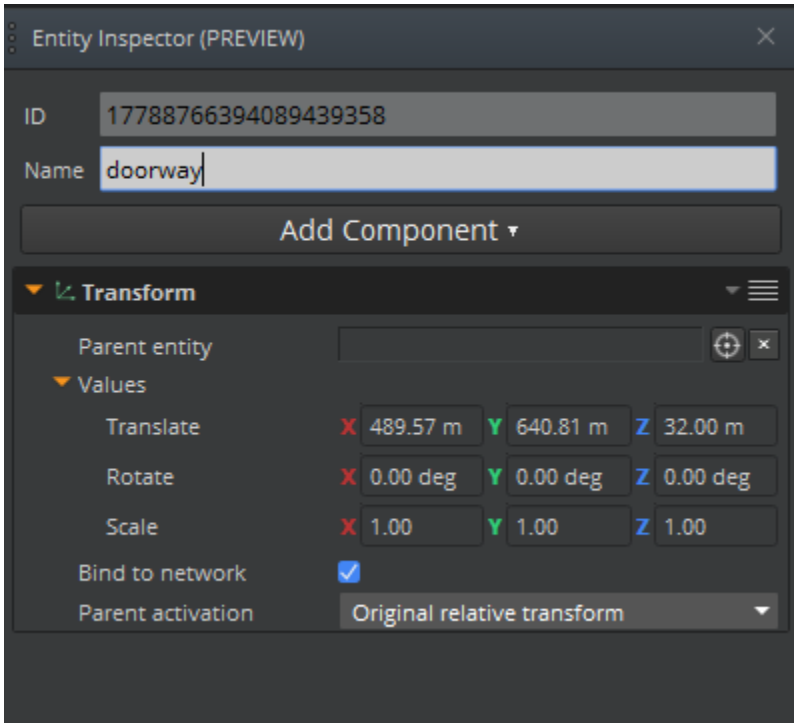
You can view your new empty entity in the **Entity Outliner**, which lists all entities in the level. Its default placement is to the left of the viewport. In the **Entity Outliner**, you should see the **PlayerSlice** that you placed in an earlier tutorial.



2. Click **Entity1** to select your entity.
3. Next, you name the entity. To do this, you modify entity properties in the **Entity Inspector**, which is located by default to the right of the viewport.

   For **Name**, `replace the **`Entity1`** text with **`doorway`**.

   The **Entity Inspector** lists the components on the currently selected entity. Every entity by default has a transform component, which defines the entity's position, rotation, and size.



4. Press **Ctrl+S** to save your level.

Next: Adding a Mesh Component (p. 48)

Adding a Mesh Component

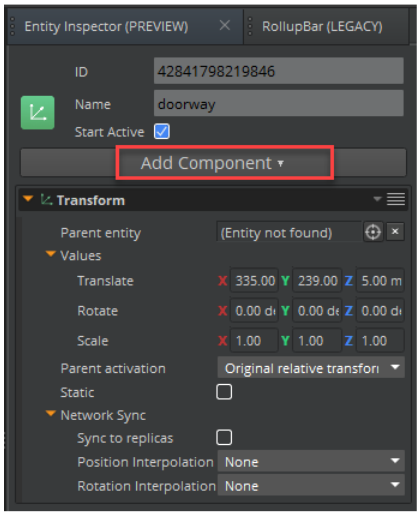
In this tutorial, you add a mesh component to the doorway entity and reference a prebuilt `.cgf` file. The `.cgf` file provides the 3D mesh data that defines what the entity looks like.

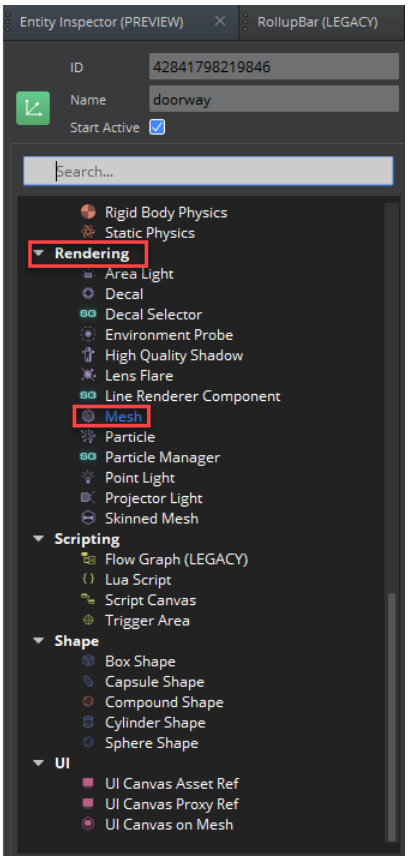
To add a mesh component

1. In the **Entity Inspector**, click **Add Component**.

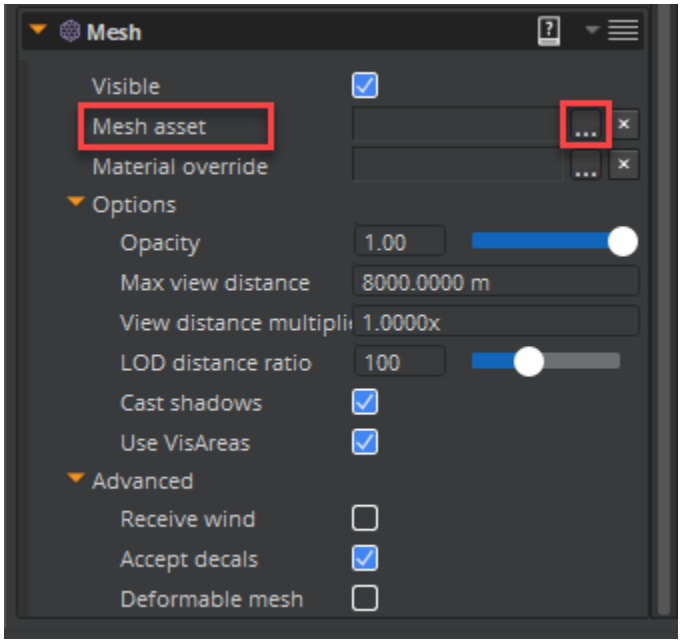
   Under **Rendering**, select **Mesh**.

   Tip
   To quickly find a component, type its name into the **Search** bar.



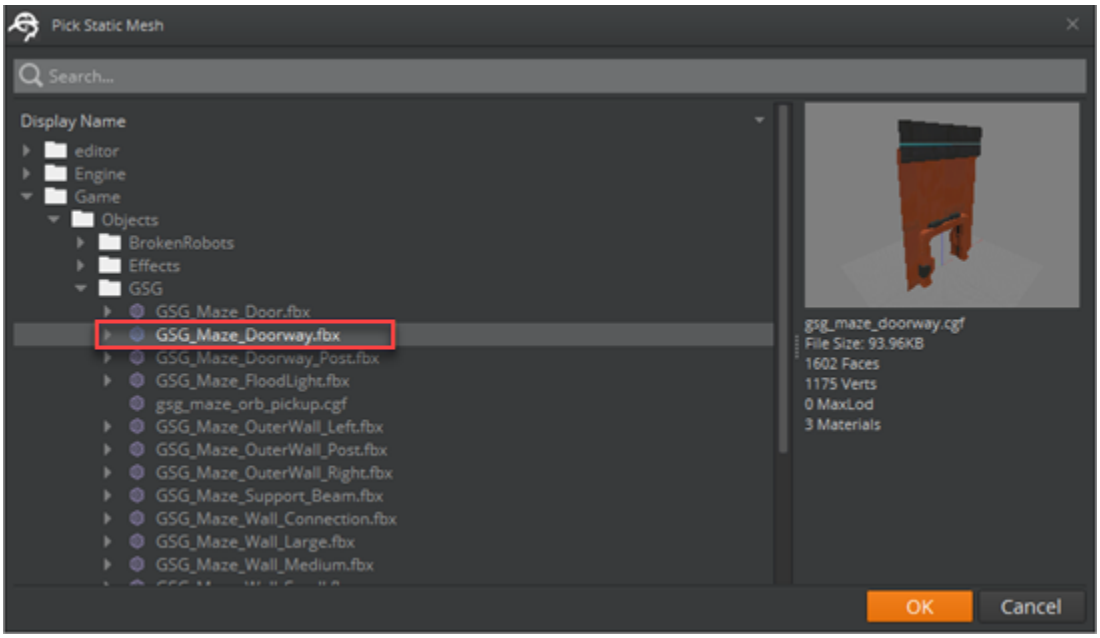


2. Under **Mesh**, next to **Mesh asset**, click the browse (**...**) button.

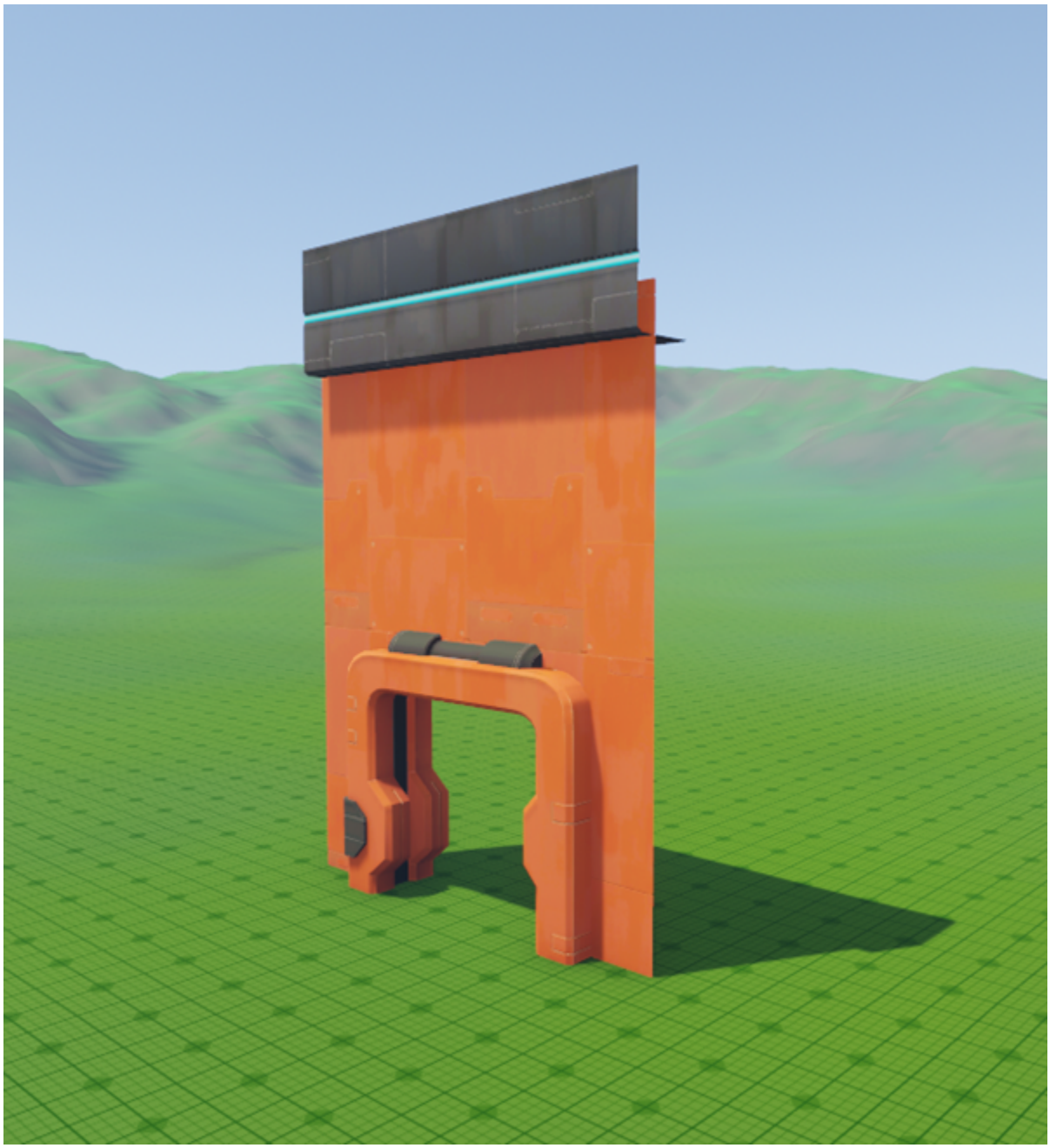


In the browser window, navigate to `Game/Objects/GSG`. Select `GSG_Maze_Doorway.cgf`, and then click **OK**.

This file is the 3D mesh that defines the appearance of the entity.



Your entity now appears as a doorway.



3. Press **Ctrl+S** to save your level.
4. Press **Ctrl+G** to play the game.

   Observe that your character cannot interact with the doorway entity but instead can pass right through it. You will add collider and physics components in the next section so that your character can interact with the entity.

Next: Adding Collider and Physics Components (p. 52)

Adding Collider and Physics Components

Characters can interact with entities only when entities have a collider component and a physics component. Collider components add an outline to your entity in a shape that characters can collide with. Physics components engage Lumberyard's physics system. Together, these components give entities their interactive qualities.

Adding a Mesh Collider

The collider components add an outline to your entity in a shape that characters can collide with. Lumberyard includes two types of collider components:

- **Mesh Collider** – Collision outline comes from the mesh asset defined in the **Mesh** component.
- **Primitive Collider** – Collision outline comes from a shape component (cube, sphere, cylinder, and so on).

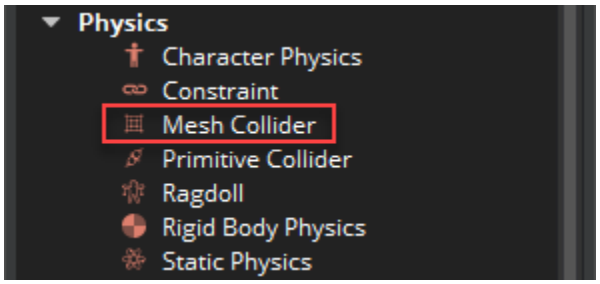
In this tutorial, you add the **Mesh Collider** component to your doorway.

To add a mesh collider component

- In the **Entity Inspector**, click **Add Component**.

  Under **Physics**, select **Mesh Collider**.

  Tip
  To quickly find a component, type its name into the **Search** bar.



Adding Static Physics

The physics component engages Lumberyard's physics system. Lumberyard includes two types of physics components:

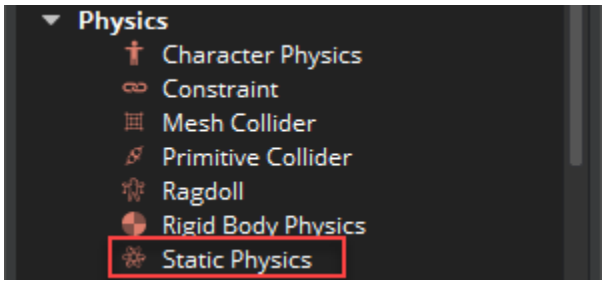
- **Static Physics** – Properties for objects that require collision information but are not designed to move, such as a building.
- **Rigid Body Physics** – Properties for objects that can be moved by physics-implemented motion events, such as a ball inteded to roll or a wall intended to fall over.

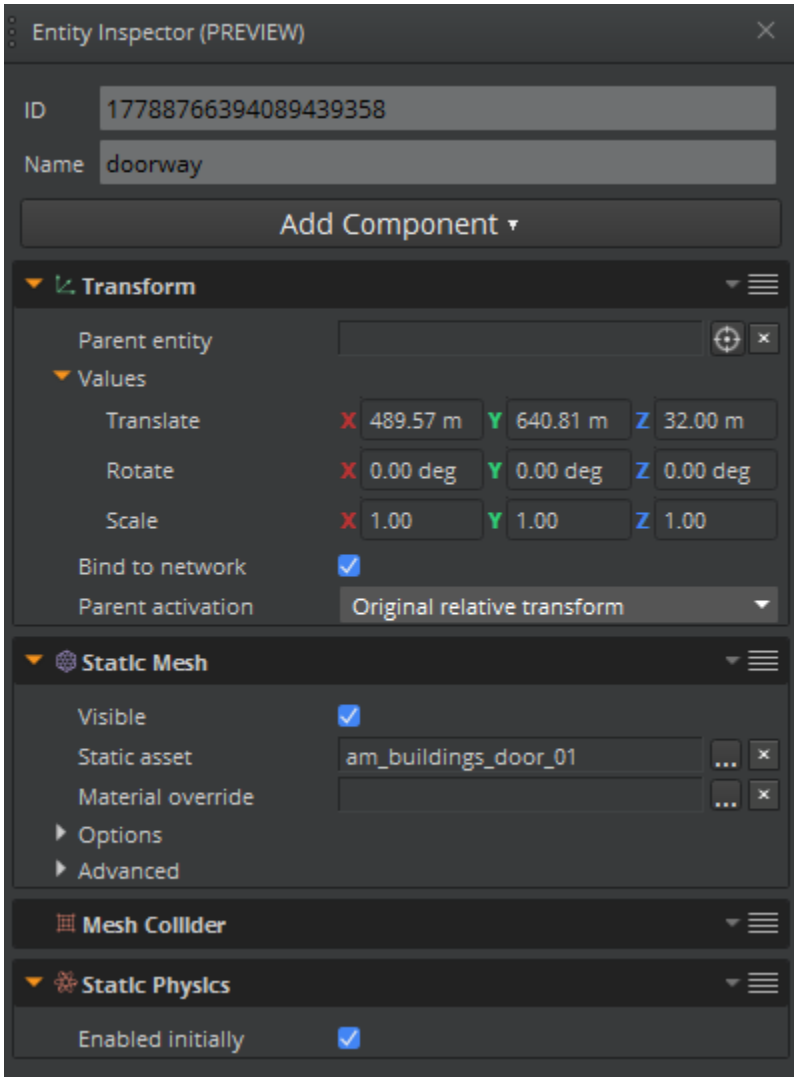
In this tutorial, you add the **Static Physics** component to your doorway.

To add a physics component

- In the **Entity Inspector**, click **Add Component**.

  Under **Physics**, select **Static Physics**.





Press **Ctrl+G** to play the game.

Observe that your character can now collide with the doorway.

Next: Using Asset Browser to Create an Entity (p. 54)

Using the Asset Browser to Create an Entity

You can also use Lumberyard's **Asset Browser** to create an entity in your level.

The **Asset Browser** appears by default in Lumberyard's bottom left corner and displays a tree view of the files in your project directory. You can drag an asset file directly from the asset browser into your viewport. When you do, Lumberyard automatically adds the **Mesh** component and sets the **Mesh asset** as the file you dragged.

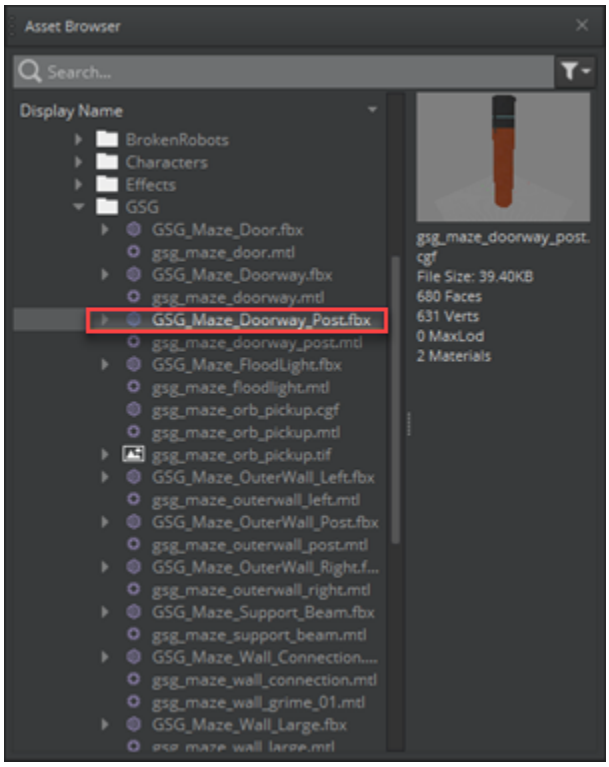
This method is slightly faster than adding a mesh component and then specifying the mesh asset. You must still add the collider and static physics components.

To add a post entity using the Asset Browser

1. The **Asset Browser** is open in Lumberyard's default layout and is located in the lower left corner. If it is not open, choose **Tools**, **Asset Browser**.

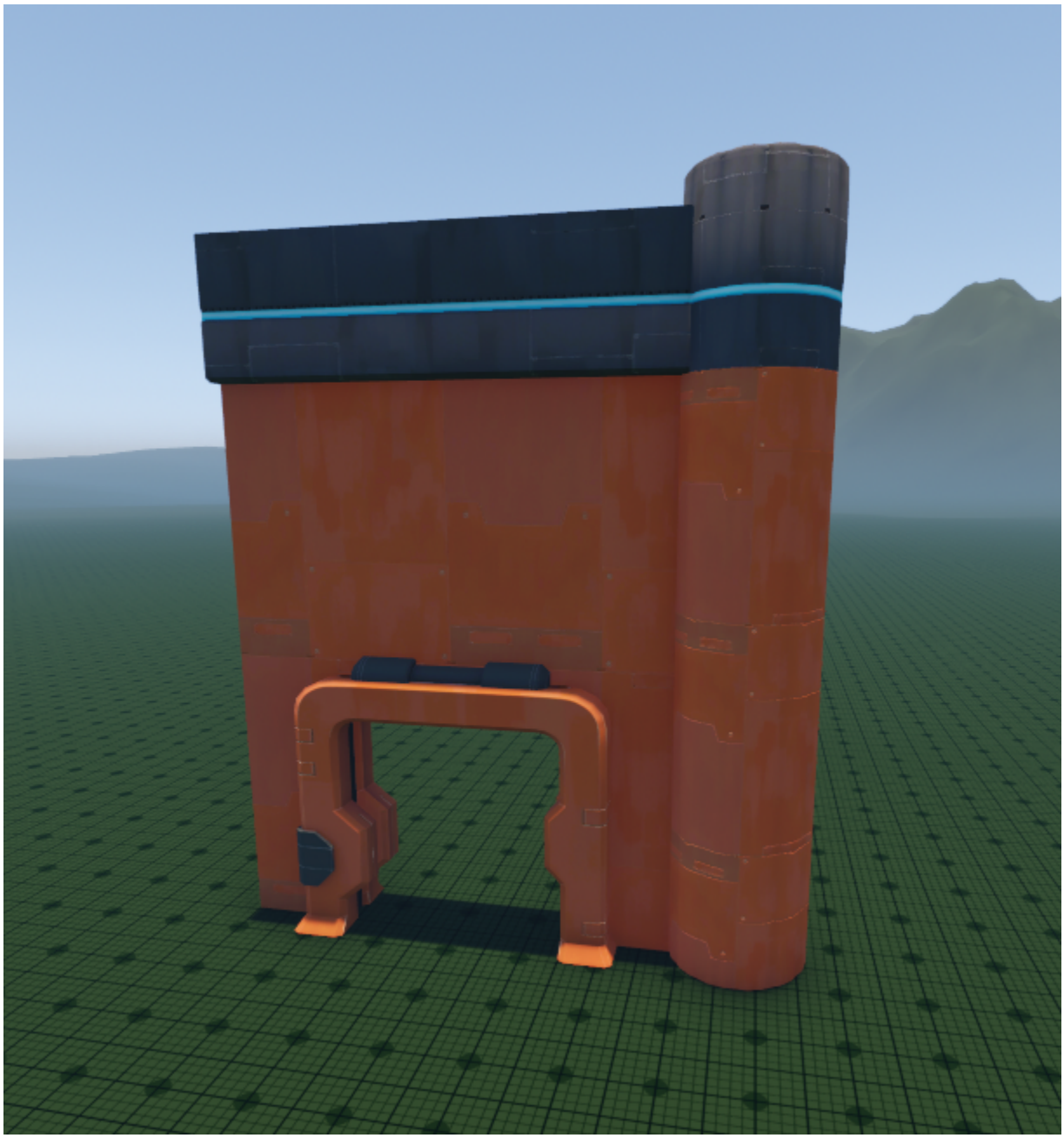
   In the **Asset Browser**, navigate to `Game/Objects/GSG`.

   Select `GSG_Maze_Doorway_Post.fbx`, and then drag it into your viewport.



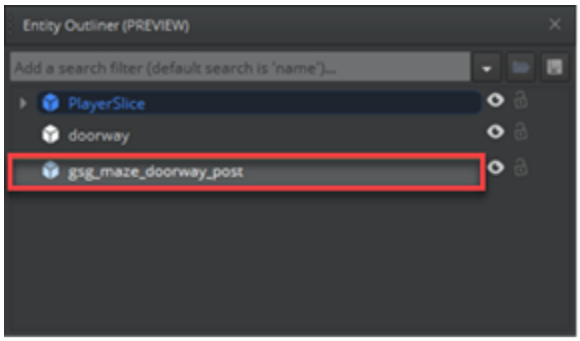
2. Use the positioning tools to move the post so that it is next to the gate as shown in the following image.

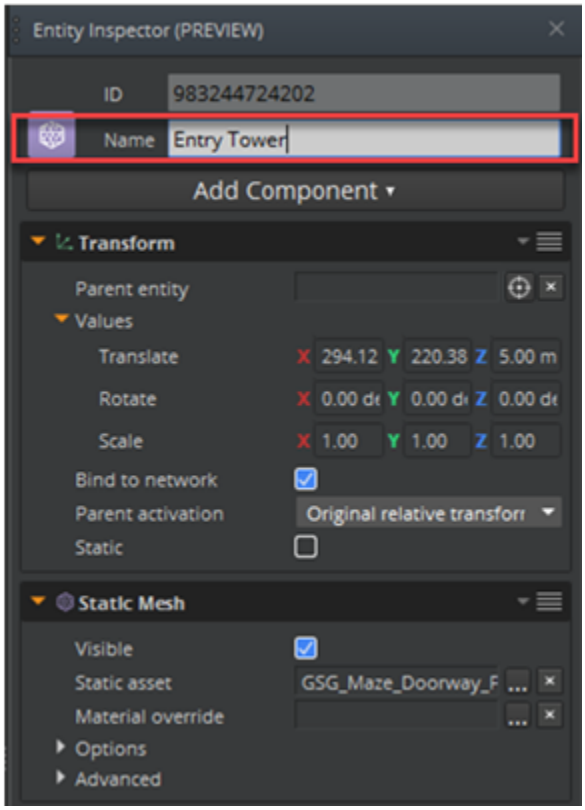
   Tip
   Snap to Grid is on by default. This makes objects snap to points on a grid. To more precisely place objects, turn off **Snap to Grid**.



3. Ensure that the post entity is selected in the **Entity Outliner**.

   In the **Entity Inspector**, for **Name**, type **`Entry Tower`**. This renames your entity.





4. Add the mesh collider (p. 52) and static physics (p. 52) components, as you learned in a previous procedure.

   In the next tutorial, you learn how to duplicate an entity.

5. Press **Ctrl+S** to save your level.

Next: Duplicating an Entity (p. 56)

Duplicating an Entity

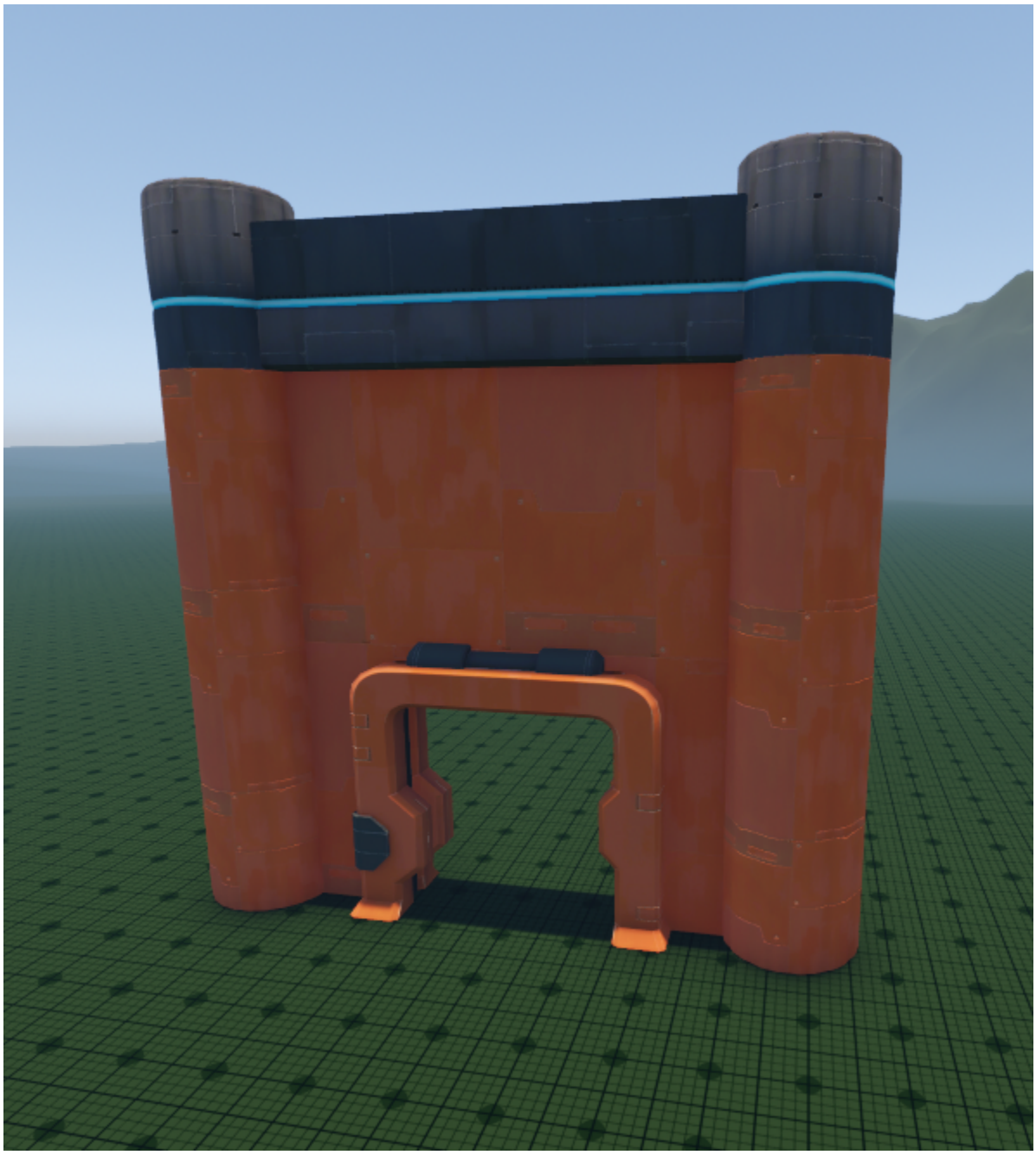
You can easily duplicate a selected entity. The duplicated entity is identical to the original entity, and even has the same initial transform (position, scale, rotation). This means that it is duplicated on top of the original entity and is not visible until you move your mouse pointer and click to place it.

1. If the tower is not still selected, then select it.

   Duplicate the tower by pressing **Ctrl+D**. Move your mouse pointer to adjust the second tower's position, and then click to place it.

   > **Tip**
   > **Snap to Grid** (p. 21) is on by default. This makes objects snap to points on a grid when you move them. To more precisely place objects, turn off **Snap to Grid**.

2. Using the essential tools (p. 15) you learned about in an earlier tutorial, adjust the position of your entities to match the following picture.



3. Press **Ctrl+S** to save your level.

Next: Parenting Entities (p. 57)

Parenting Entities

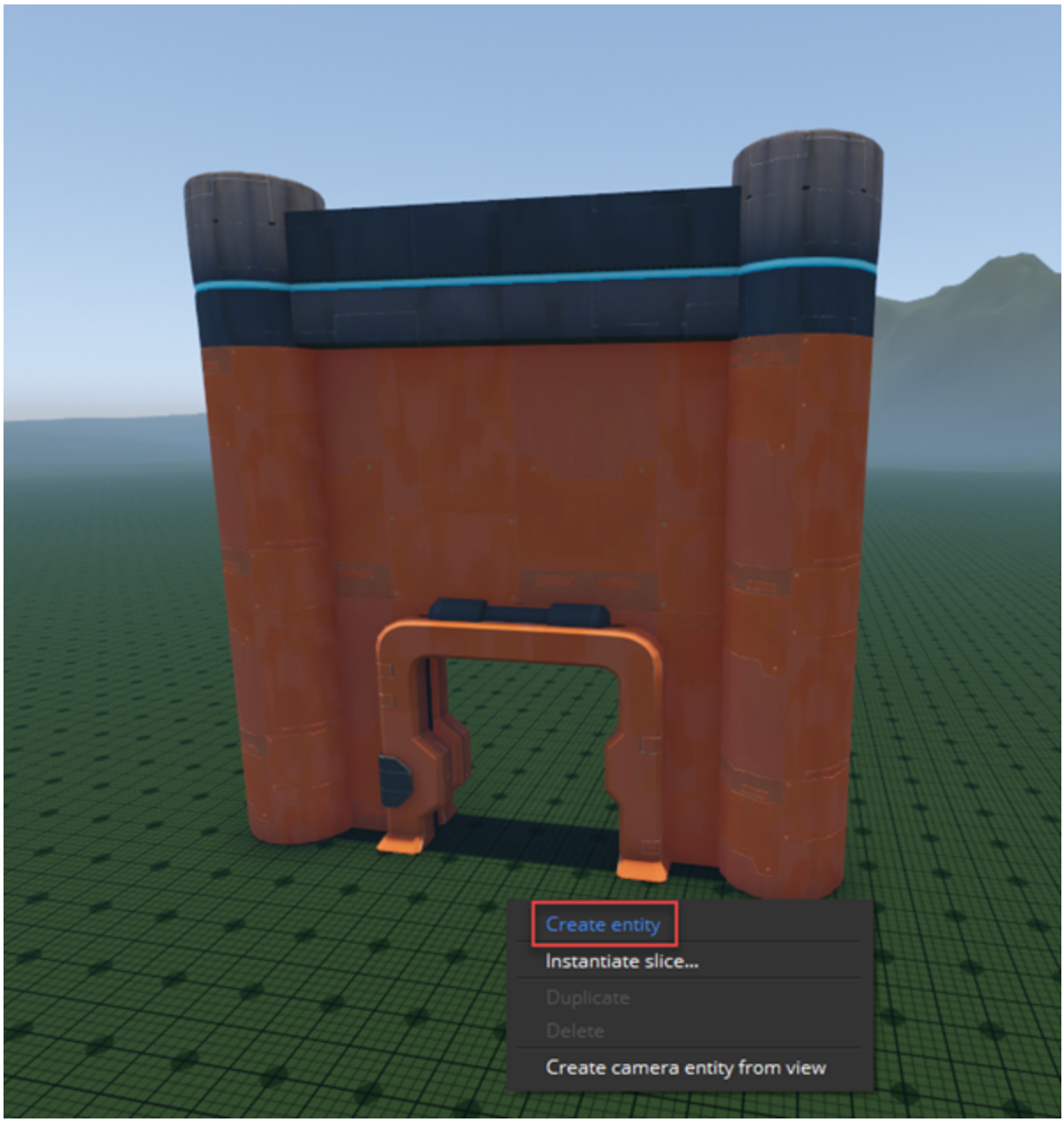
You can organize your entities by grouping them together beneath another entity. This is called parenting. The top entity is called the parent entity, and any entities grouped beneath it are called child

entities. You can view the parent-child relationship in the **Entity Outliner**. Grouping or parenting your entities helps to keep your level organized.

In this tutorial, you use the **Entity Outliner** to group together the three entities that you created in the previous tutorial.

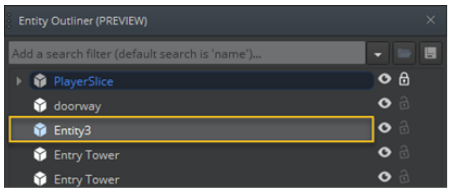
To group entities

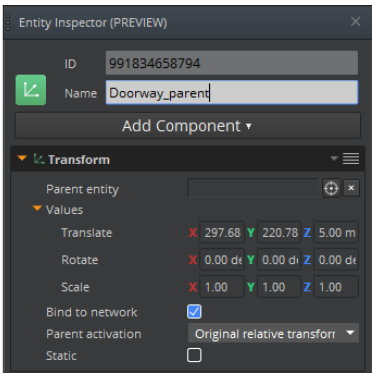
1. Create a new, empty component: In your viewport, near the bottom center of the doorway entity, right-click an empty area and then choose **Create entity**.



2. Select the new entity in the **Entity Outliner**.

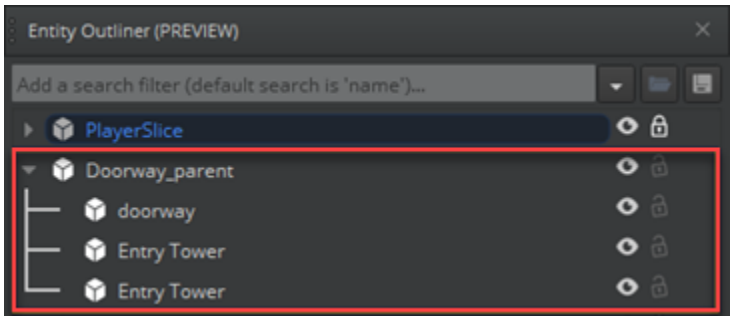
   In the **Entity Inspector**, rename this new entity to **`Doorway_parent`**.





3. In the **Entity Outliner**, select the doorway and two tower entities. Then drag the selection onto the **Doorway_parent** entity.

   The **Doorway_parent** entity is now the parent of the doorway and two tower entities, which are now child entities.



4. Press **Ctrl+S** to save your level.

You have parented the three entities. When you move, scale, and rotate the parent entity, the gate pieces are moved, scaled, and rotated as well. You can, however, still select the individual entities and move them independently.

Next: 4: Using Slices to Build the Maze (p.)

4: Using Slices to Build the Maze

A slice is a special kind of asset (a `.slice` file) that is composed of a group of entities and their components. You can place, or instantiate, slices throughout your level quickly and easily. If you modify one slice instance, such as adding a component or changing a property, you can save your changes to update all the other slice instances.

You can also independently edit an individual slice instance and then not save the change. This results in a unique slice instance.

To begin building your maze, you do the following:

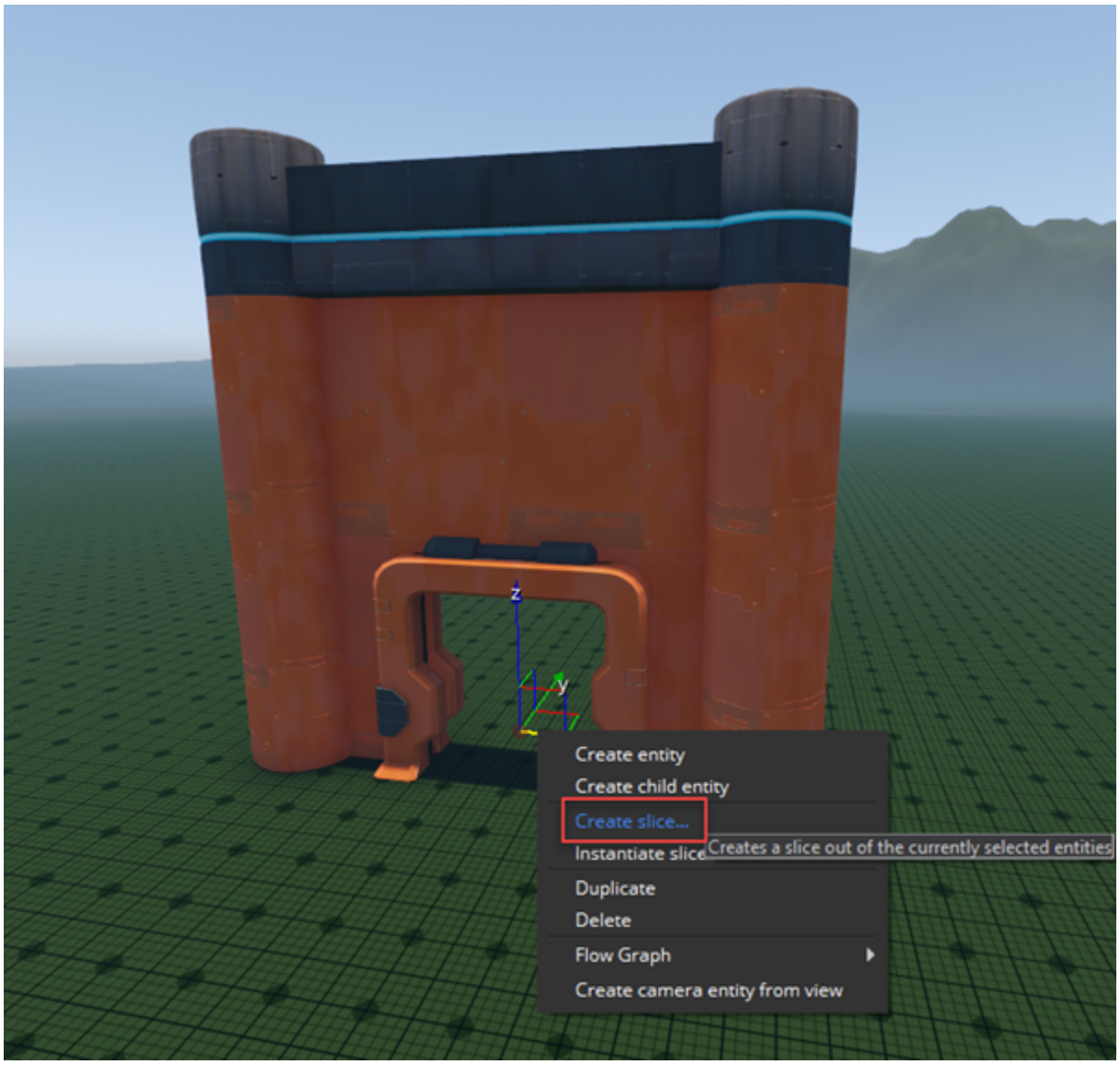
1. Create a doorway slice.
2. Instantiate (place) prebuilt slices for exterior and interior maze walls.
3. Align the positions of your wall slices.
4. Lock wall slices from being selected.
5. Instantiate (place) multiple instances of the lamp slice.
6. Modify the properties of one lamp slice instance.
7. Save the changes to all lamp slice instances.

Next: Creating a Doorway Slice (p. 60)

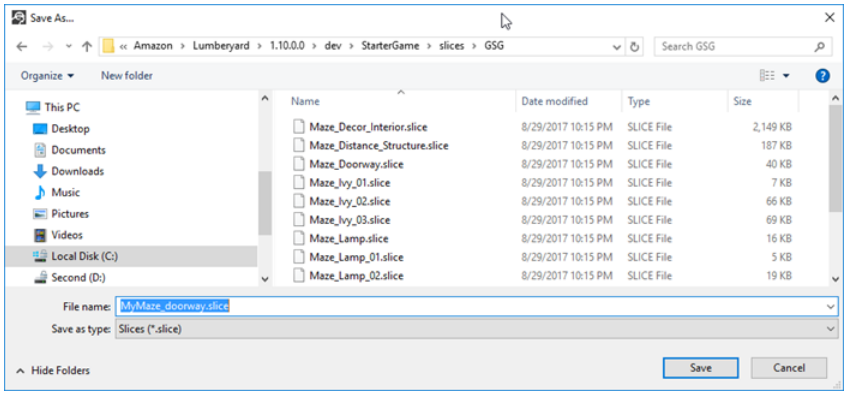
Creating a Doorway Slice

In previous tutorials, you grouped entities to make a doorway group. In this tutorial, you learn how to save that group as a new slice.

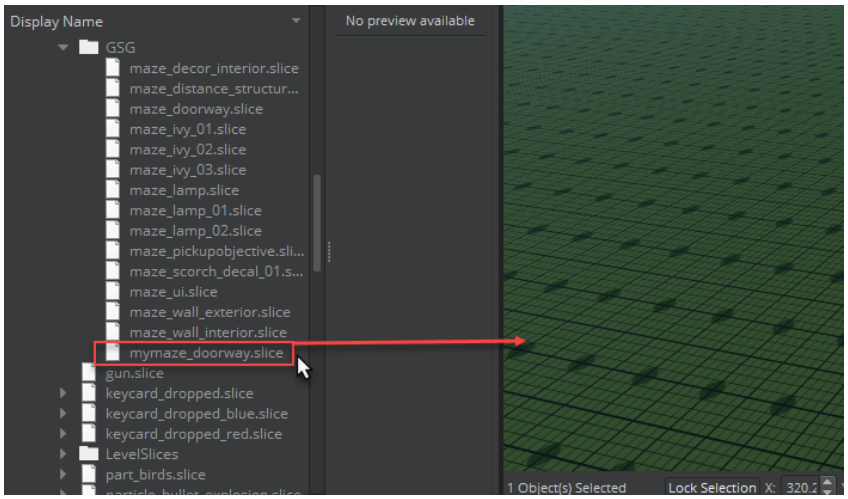
1. Save the doorway group as a doorway slice: In the perspective viewport or the **Entity Outliner**, right-click the **Doorway_parent** entity. Choose **Create slice**.



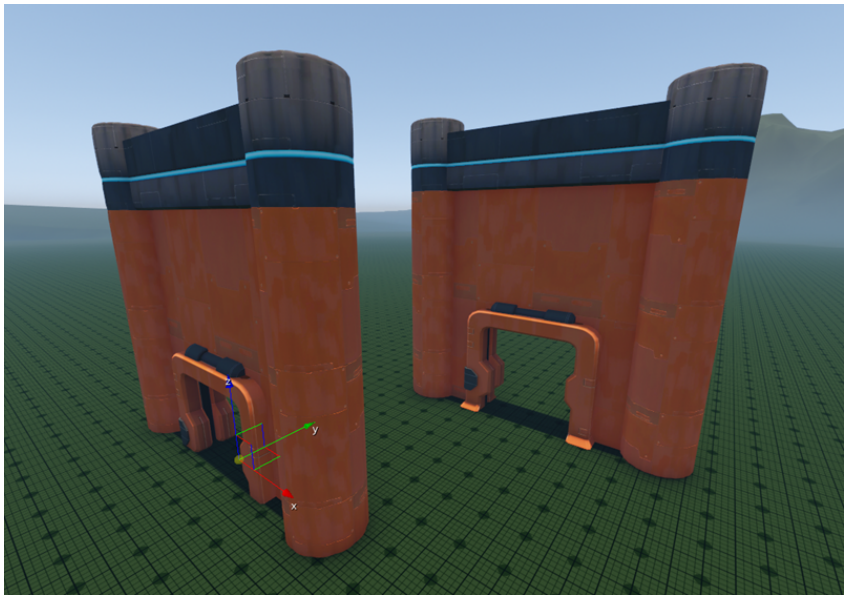
2. In the **Save As** dialog box, navigate to `Game\slices\GSG` and save your slice as `MyMaze_doorway.slice`. Click **Save**.



3. Instantiate (place) the doorway slice: In the **Asset Browser**, navigate to `Game\Slices\GSG`. Drag the newly created `MyMaze_doorway.slice` into the perspective viewport.



You should now have two doorways in your level, similar to the following image.



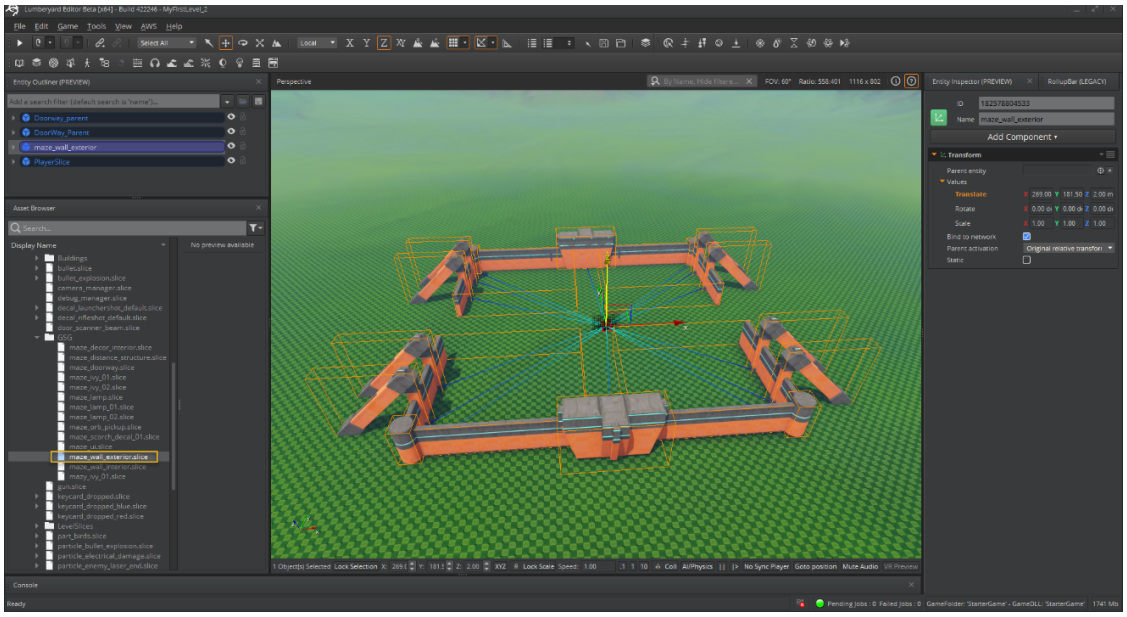
4. Press **Ctrl+S** to save your level.

Next: Instantiating Prebuilt Slices (p. 62)

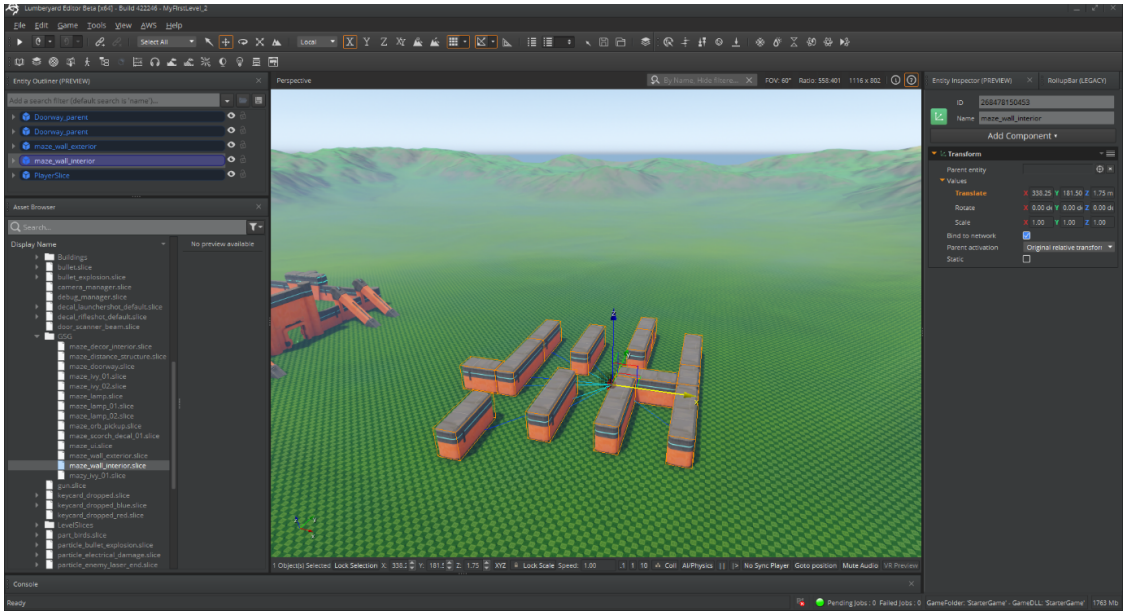
Instantiating Prebuilt Slices

In this tutorial, you instantiate, or place, two previously created slices into the perspective viewport: the exterior maze walls and the interior maze walls.

1. In the **Asset Browser**, navigate to `Slices\GSG`. Drag `maze_wall_exterior.slice` into the perspective viewport.



2. In the **Asset Browser**, from within `Slices\GSG`, drag `maze_wall_interior.slice` into the perspective viewport.



3. Press **Ctrl+S** to save your level.

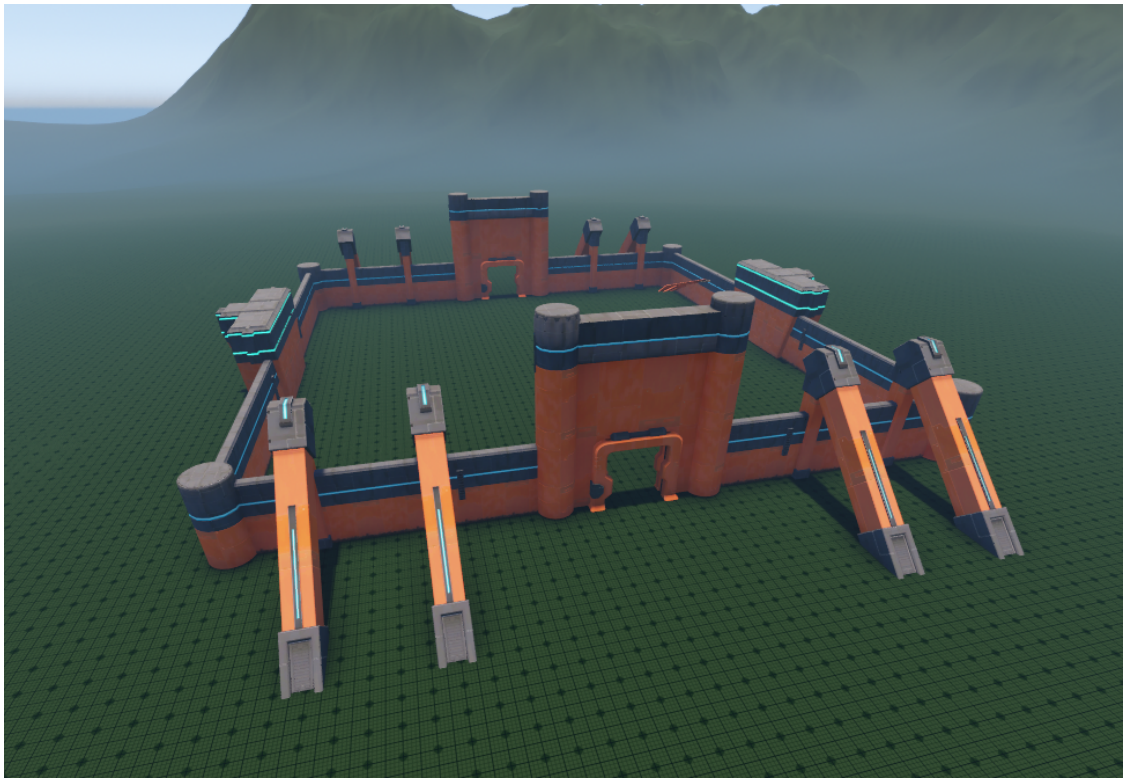
Next: Aligning the Slices (p. 63)

Aligning the Slices

In this tutorial, you use the **Move** tool to align your doorway parent slices with the exterior wall's open spaces. Then you use the **Align to Object** tool to arrange the interior maze walls within the exterior maze walls. The **Align to Object** tool sets the parent components for the interior and exterior walls to the same relative position.

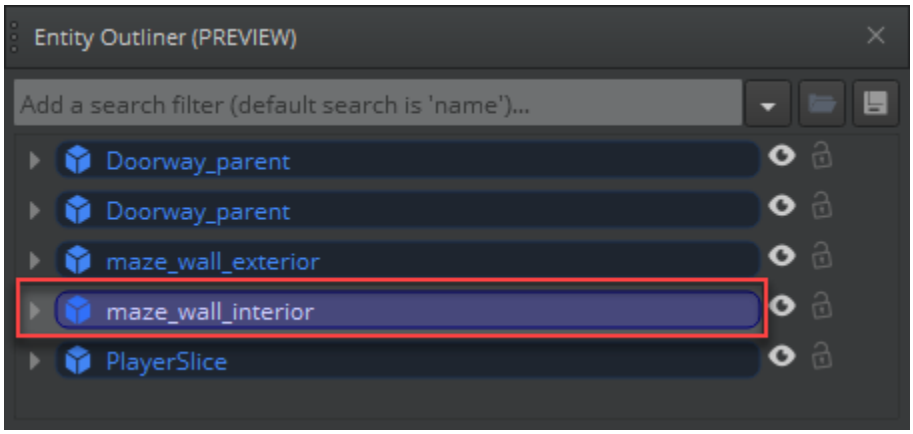
To align exterior walls with doorways

- Use the **Move** tool to adjust the exterior wall and both doorways so that they are aligned as in the following picture.



To align exterior and interior walls

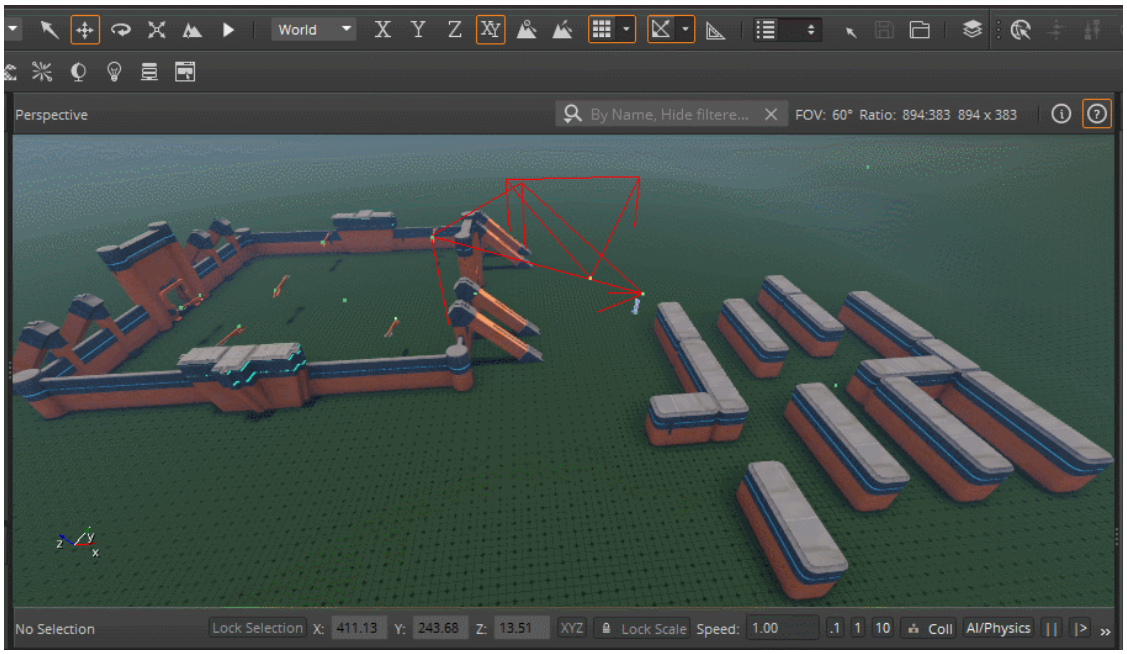
1. In the **Entity Outliner** or in the viewport, select **maze_wall_interior**.



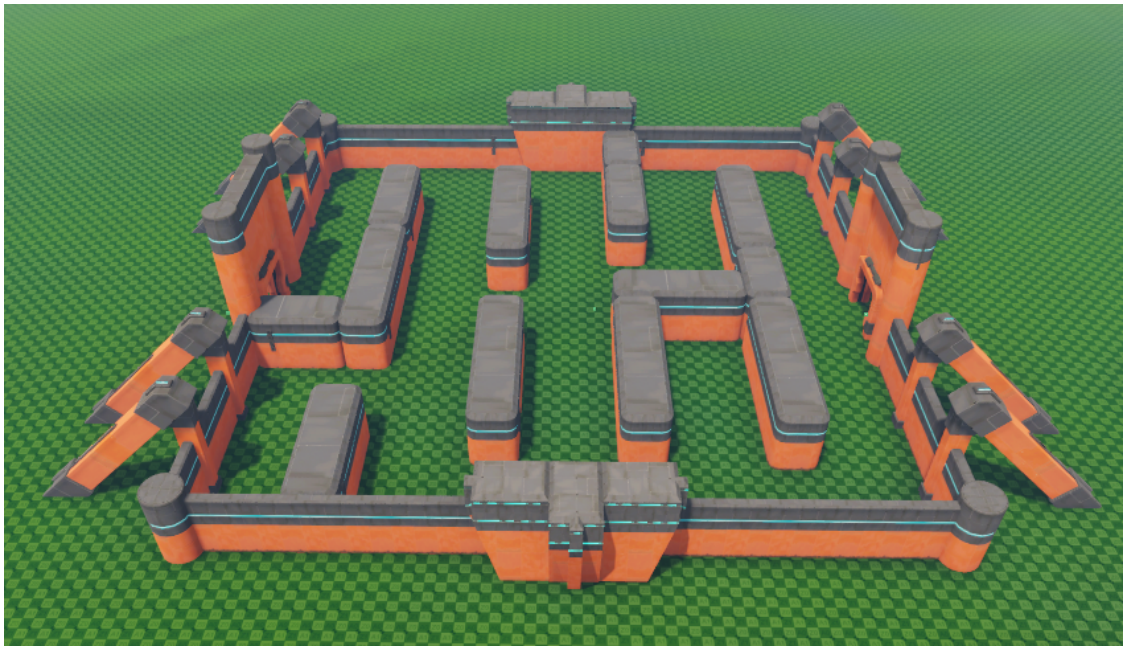
2. On the toolbar, click the **Align to object** icon .

The alignment takes place once you select another entity in the viewport to align it with.

3. In the viewport, select the **maze_wall_exterior** slice. The green dot in the center of the exterior walls indicates the exterior maze wall slice, as shown in the following animated image.



The interior and exterior walls are now aligned and look like the following image.



4. Press **Ctrl+S** to save your level.

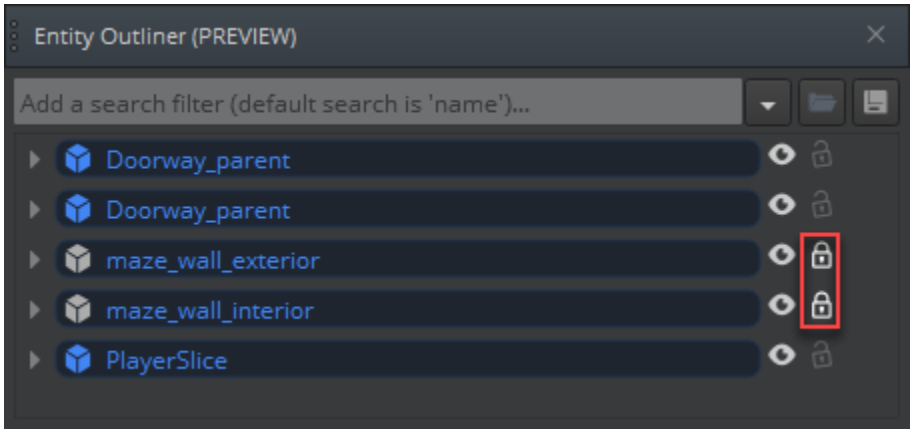
Next: Locking Wall and Doorway Slices (p. 66)

Locking Wall and Doorway Slices

For now, you are done adjusting the maze walls and doorways. You can prevent these slices from interfering with your future work by locking them. This prevents you from selecting them accidentally.

To lock the wall and doorway slices

1. In the **Entity Outliner**, locate **maze_wall_exterior** and **maze_wall_interior**.
2. To the right of **maze_wall_exterior** and **maze_wall_interior**, click the lock icon.



3. Repeat the preceding step for the **Doorway_parent** slices.
4. Press **Ctrl+S** to save your level.

The maze and doorway slices are now locked and cannot be moved by accident. To modify them at any time, click the lock icons again to unlock the slices.

Next: Placing Lamp Post Slices (p. 66)

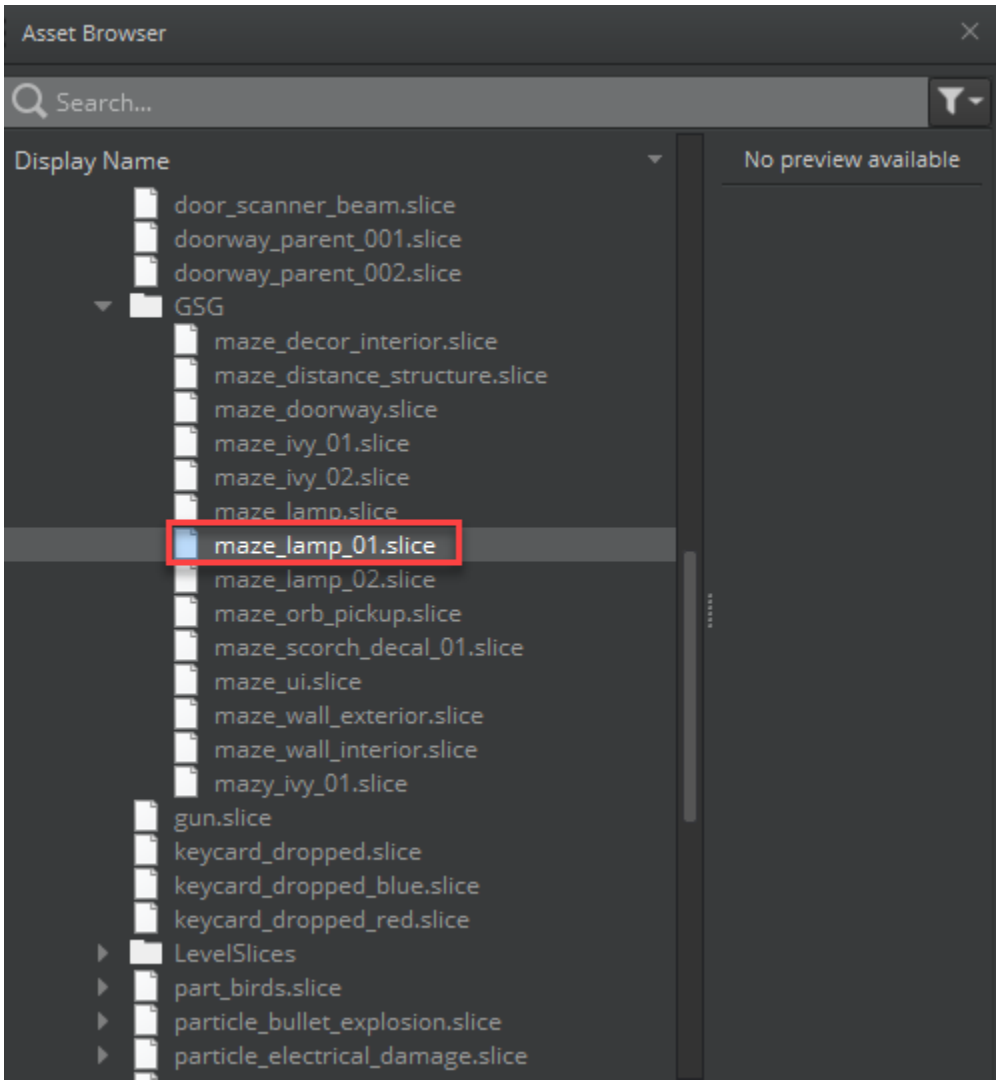
Placing Lamp Post Slices

Instantiate, or place, multiple instances of the lamp post slice to add some overhead lamps to the maze.

To place lamp post slices

1. In the **Asset Browser**, navigate to `/slices/GSG`.

   Drag `maze_lamp_01.slice` into the viewport.

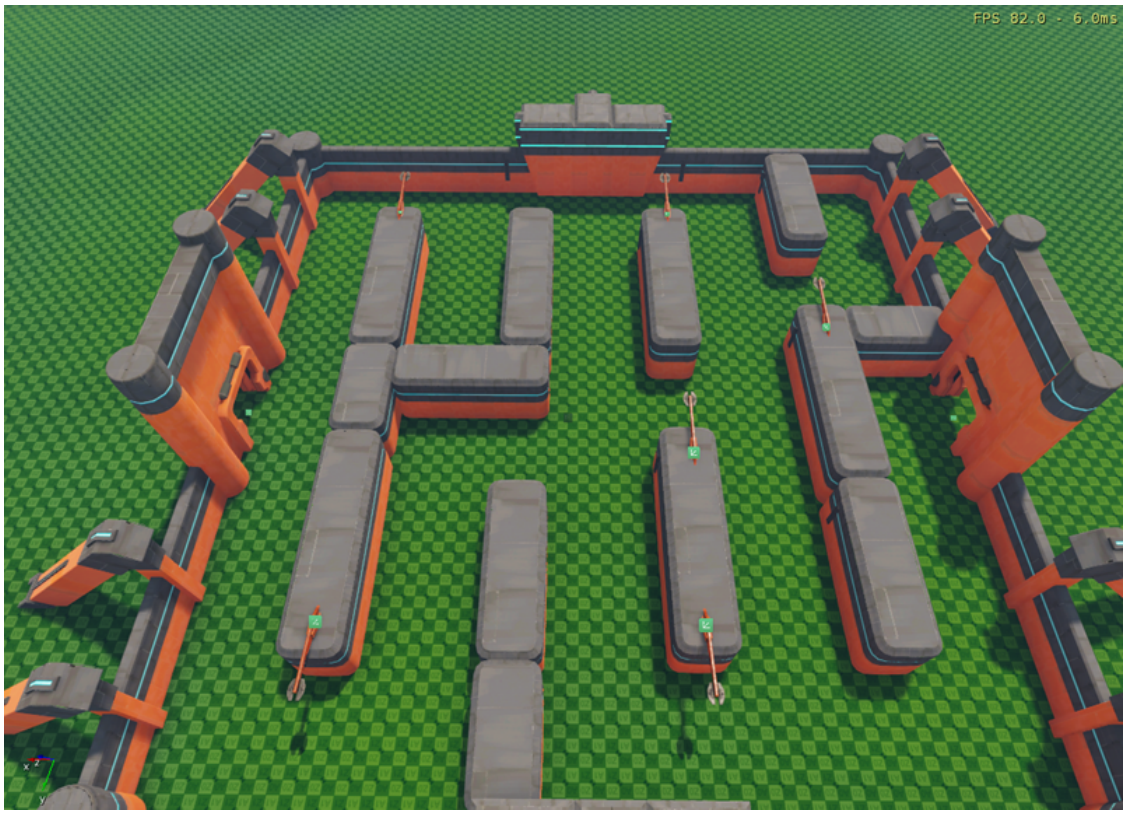


2. Position the lamp post on top of one of the interior maze walls, similar to the following image.

3. In the **Entity Outliner**, right-click **Lamp** and then choose **duplicate**. This creates a copy of the lamp. Move your mouse pointer to the position you want it, and then click to place it.

   Tip
   You can duplicate an entity by selecting it in the viewport and pressing **Ctrl+D**.

4. Repeat this process to place four or five more lamps into the scene, anywhere you want.



5. Press **Ctrl+S** to save your level.

Next: Saving Slice Changes (p. 69)

Saving Slice Changes

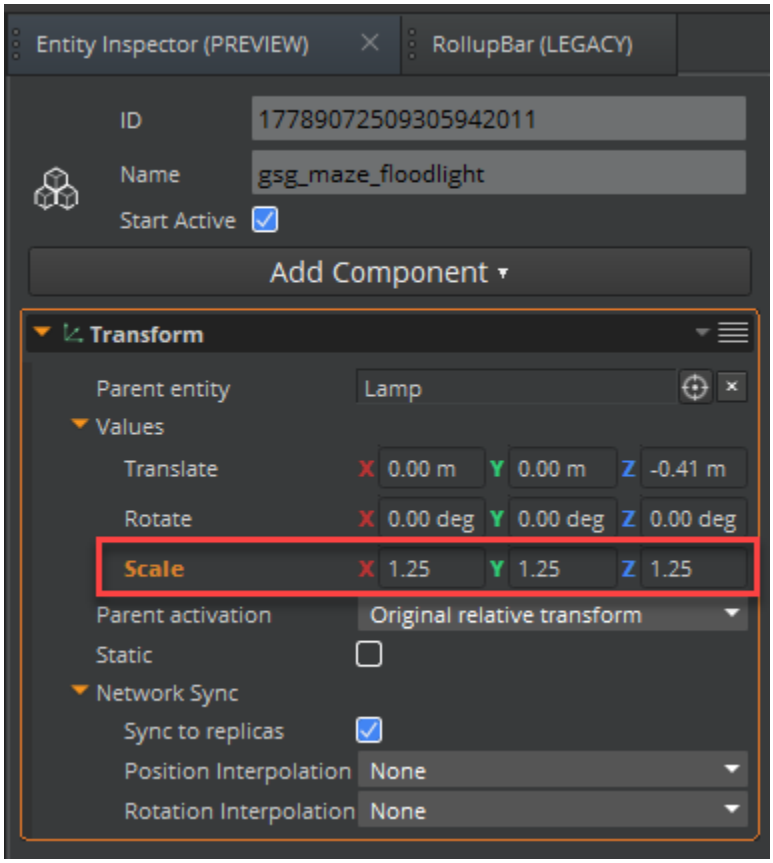
If you modify one of the lamps you just placed, you can save the changes to all of the lamp slice instances. This means, for example, that you can make one of the lamps larger; when you save that change, all the other placed lamps become larger as well.

To change the lamp's size and save the changes

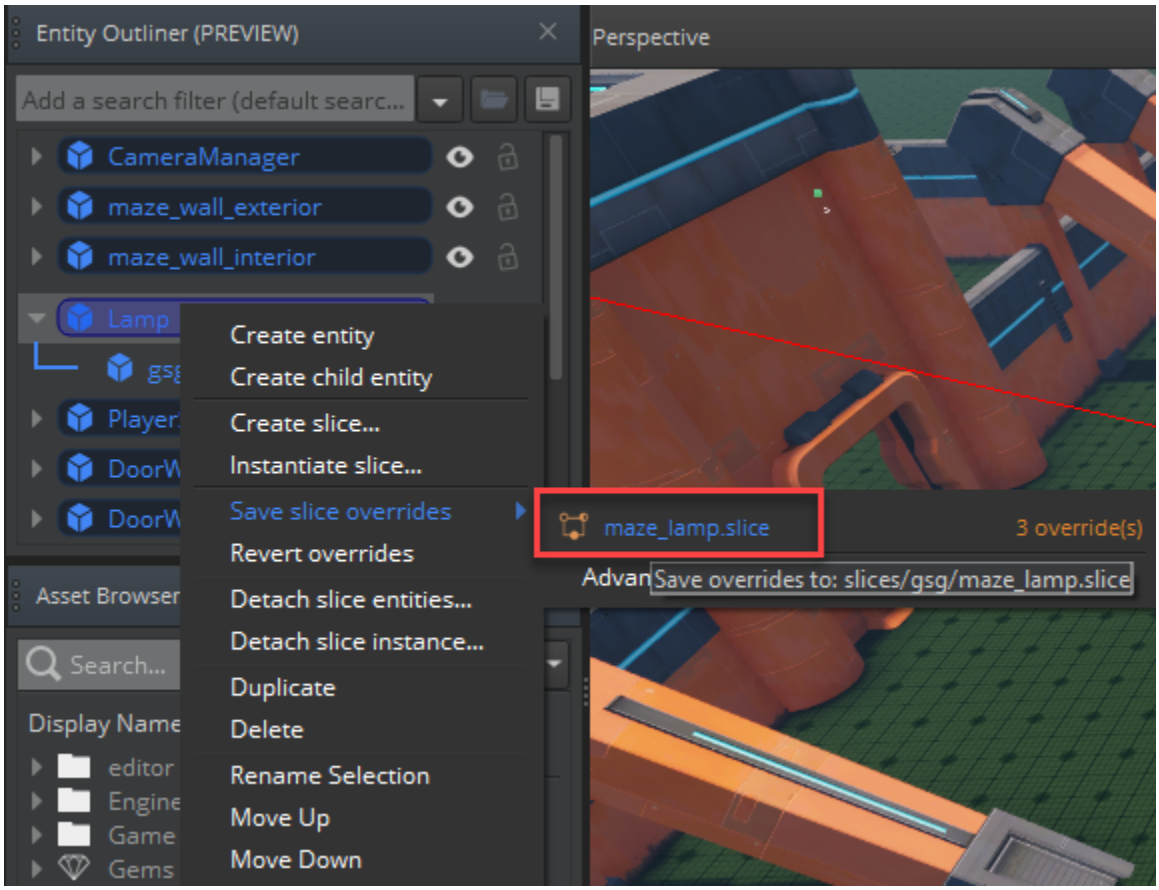
1. In the **Entity Outliner**, select one of the lamp slices. Click the arrow icon to expand the parent entity. This displays the entities in the slice. In this lamp slice, there is only one entity. Select the entity **gsg_maze_floodlight**.

2. In the **Entity Inspector**, change the **Scale** values:

   - **x**: `1.25`
   - **y**: `1.25`
   - **z**: `1.25`



3. In the **Entity Outliner**, select the parent node **Lamp**. Right-click, point to **Save slice overrides** and then choose **maze_lamp.slice**.



Observe how all of the lamps are updated to the new scale. The editor has saved this update to the slice, and all slice instances are automatically updated to reflect those changes.

4. Press **Ctrl+S** to save your level.
5. Press **Ctrl+G** to play your level.

   Press **Esc** to quit.

Next: 5: Adding Enemy AI Characters (p. 72)

5: Adding Enemy AI Characters

Lumberyard uses artificial intelligence, or AI, to create responsive behavior in nonplayer characters (NPCs). NPCs are typically placed as enemy characters. Adding these AI opponents to your level adds conflict and tension to the player's experience.

In the following tutorials you learn how to set up and place enemy AI characters within the maze area. You use component entities and a Lua script to create a simple AI experience.

The processes outlined in this tutorial are based on the AI design that is used for Starter Game.

Topics

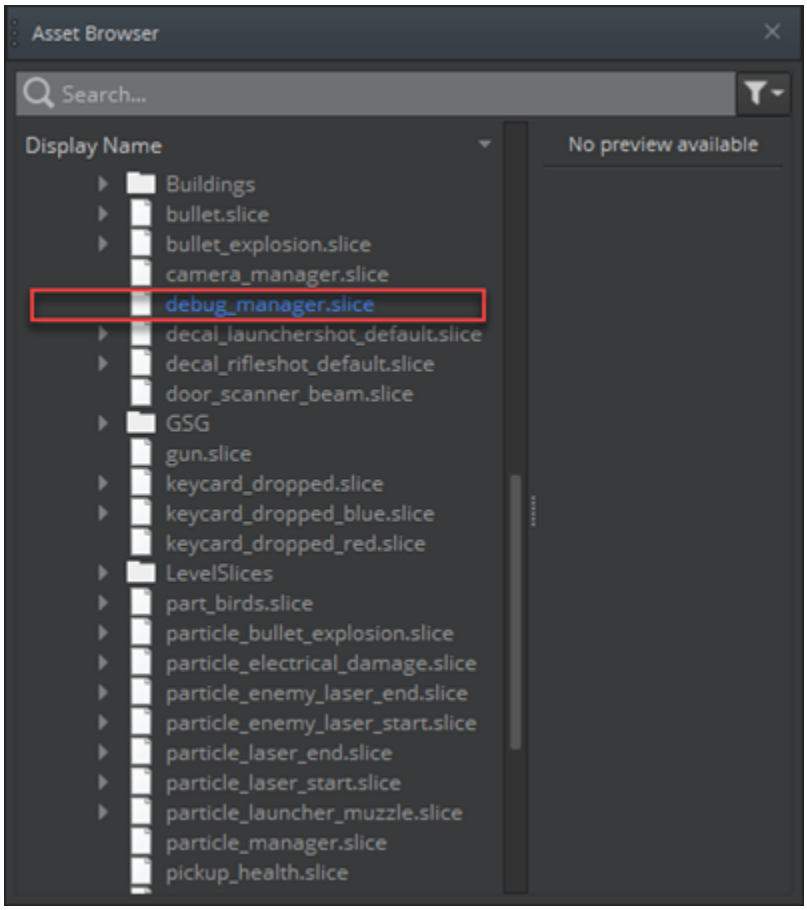
Next: Setting Up the AI Trigger Area (p. 72)

Setting up the AI Trigger Area

To set up the AI trigger area

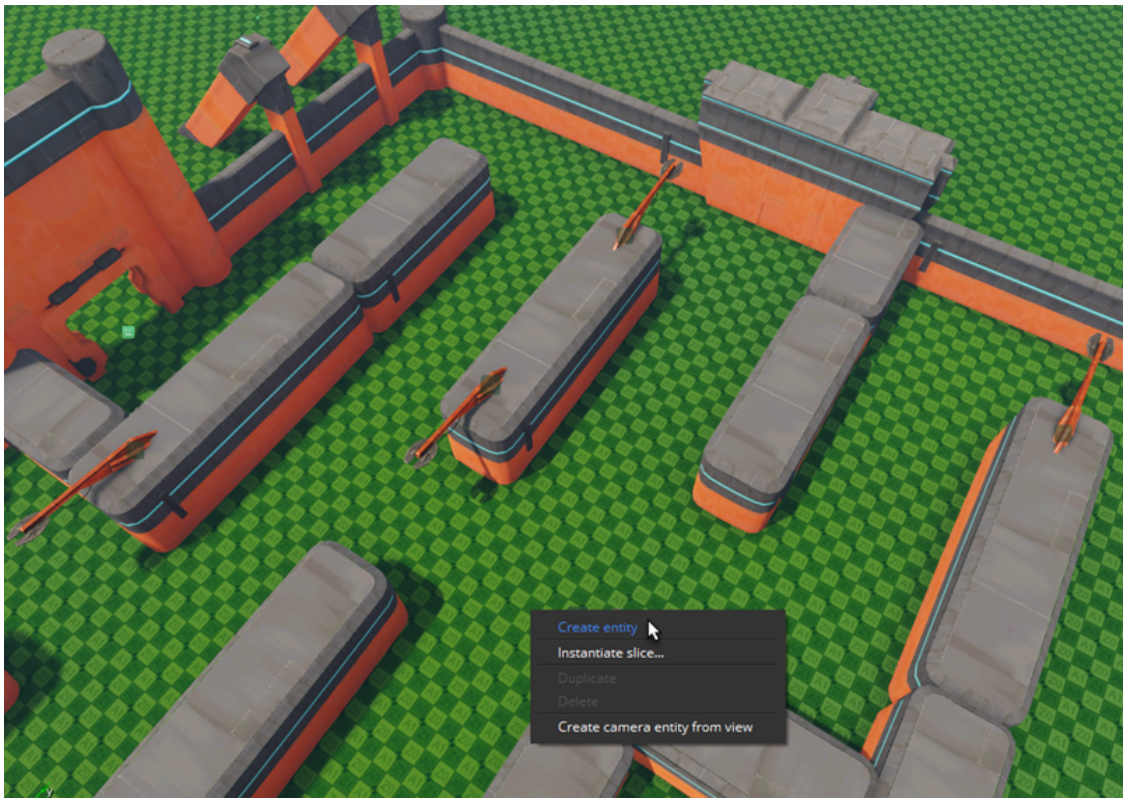
1. In the **Asset Browser**, navigate to `\Game\Slices`. Drag `Debug_manager.slice` into the viewport.

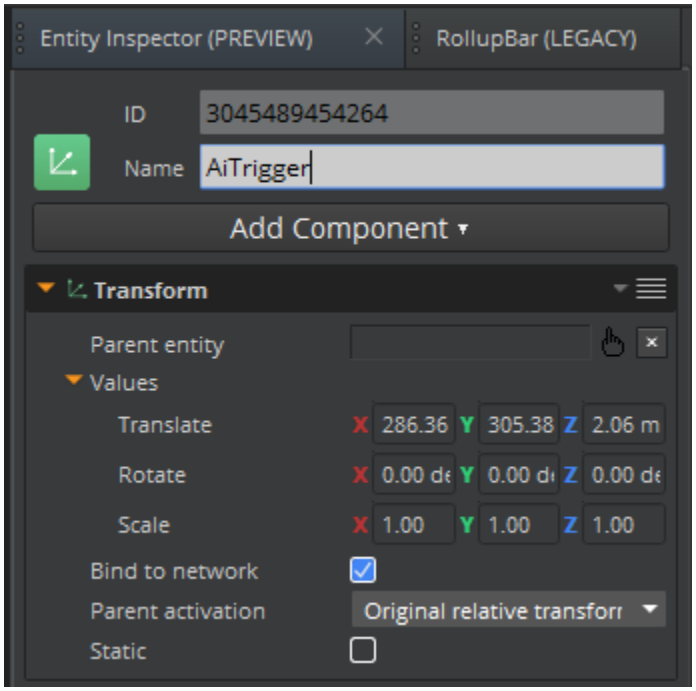
   This slice manages some of the AI script's functional behavior.



2. In the viewport, right-click the center of the maze and choose **Create entity**.

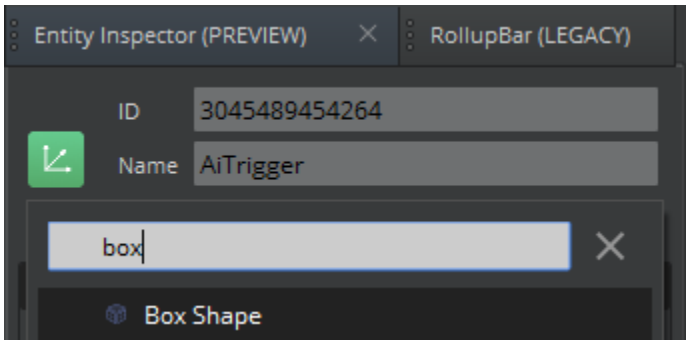
   This action creates an empty entity. In the **Entity Inspector**, name this entity **`AiTrigger`**.





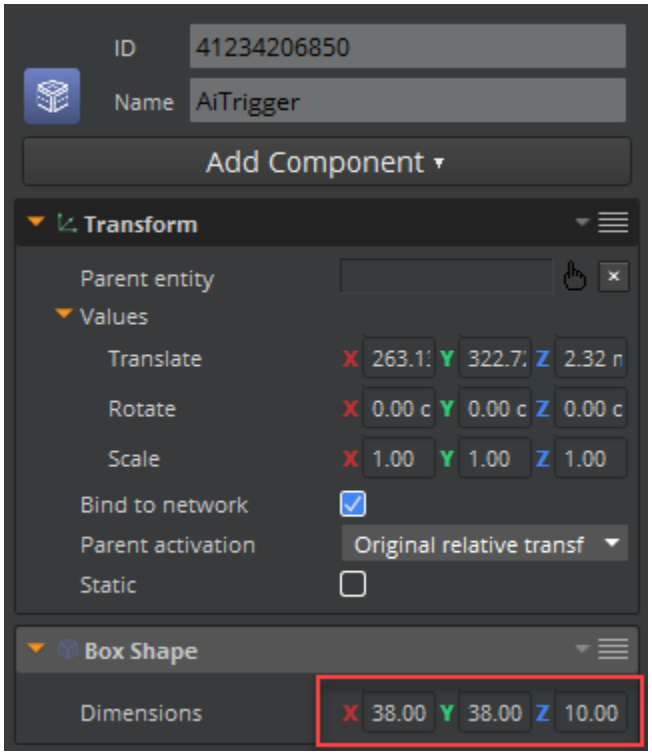
3. In the **Entity Inspector**, click **Add Component**. Under the **Shape** category, choose the **Box Shape** component.

   Tip
   To find the **Box Shape** component quickly, type `box` in the **Search** bar.



4. In the **Entity Inspector**, in the **Box Shape** component, set the box's **Dimensions** to **X**: **38**, **Y**: **38**, **Z**: **10**.

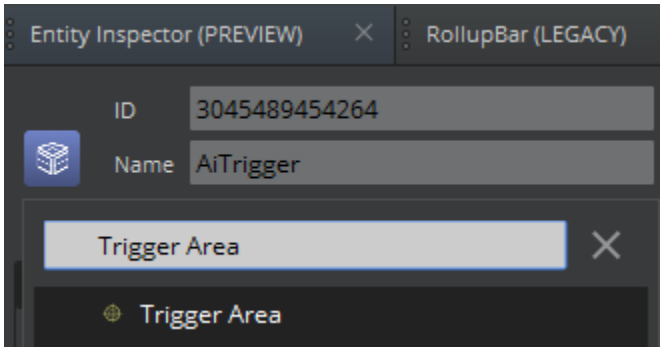
   The AI is restricted to these dimensions, which approximate the size of the maze.



5. Use the Move tool to center the box so that it fits around the maze wall.

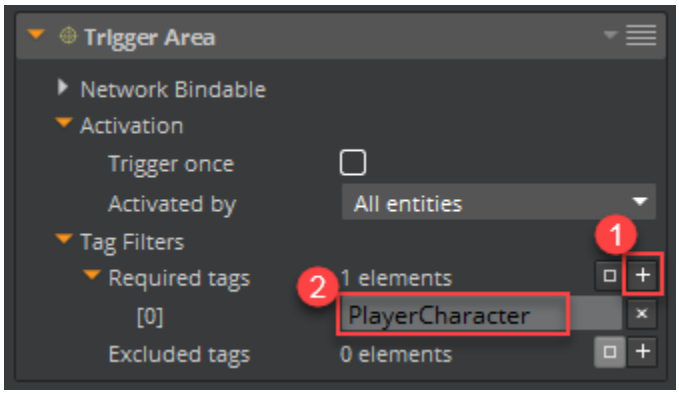
   Tip
   If the box does not appear in the viewport, click the **?** icon in the upper-right corner of your viewport.

6. With the **AiTrigger** entity still selected, in the **Entity Inspector**, click **Add Component**. Under **Scripting**, choose **Trigger Area**.

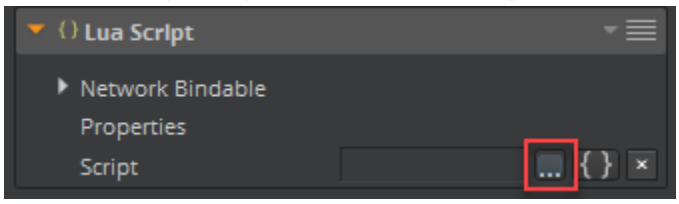


7. In the **Trigger Area** component, under **Tag Filters**, click **+** next to **Required tags**.

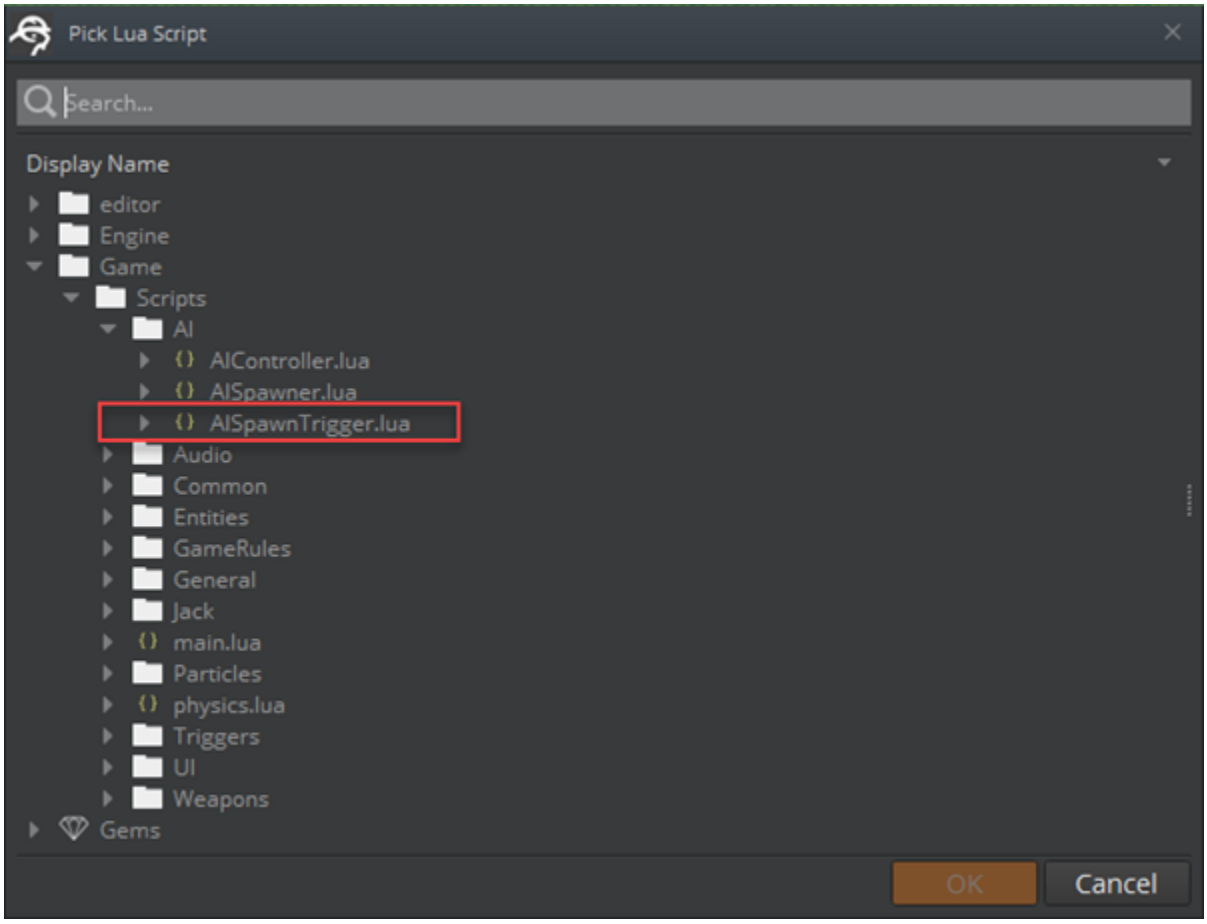
   In the box labeled **[0]**, type **`PlayerCharacter`**.



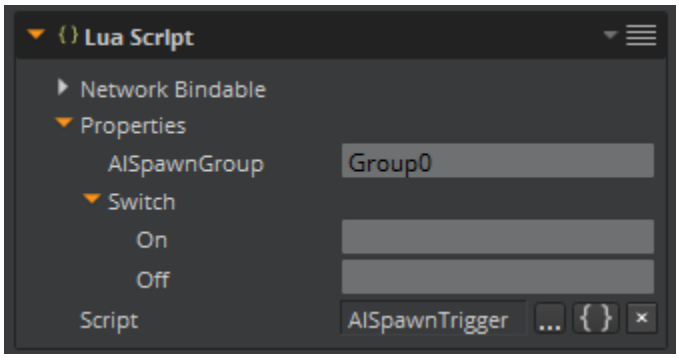
8. In the **Entity Inspector**, click **Add Component**. Under **Scripting**, click **Lua Script**.
9. In the **Lua Script** component, next to the **Script** box, click (**...**) to browse for a script file.



Navigate to `\Game\Scipts\AI`. Select **AISpawnTrigger.lua**. Click **OK**.

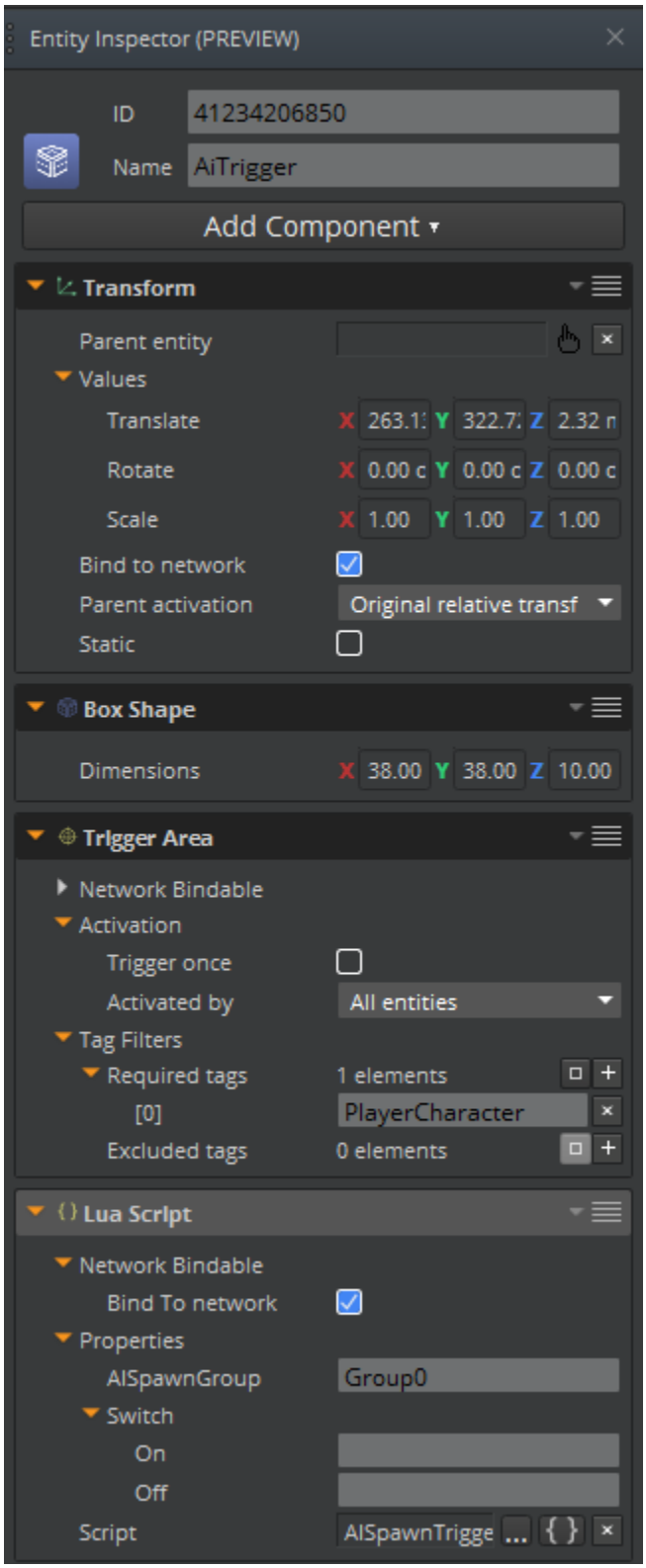


10. In the **Lua Script** component, under **Properties**, in the **AiSpawnGroup** box, type **`Group0`**.



When the player enters the trigger area defined by this entity's shape, the AI scripts in the group are activated. When the player leaves the trigger area, the AI scripts are deactivated.

Your **Entity Inspector** should look like the following image.



11. Press **Ctrl+S** to save your level.

Note
During gameplay, the player controller must start outside the maze trigger area and then enter it in order to activate the AI characters. If the game starts inside the trigger area, the player controller must leave and then reenter to trigger the AI.

Next: Defining the AI Navigation Area (p. 80)

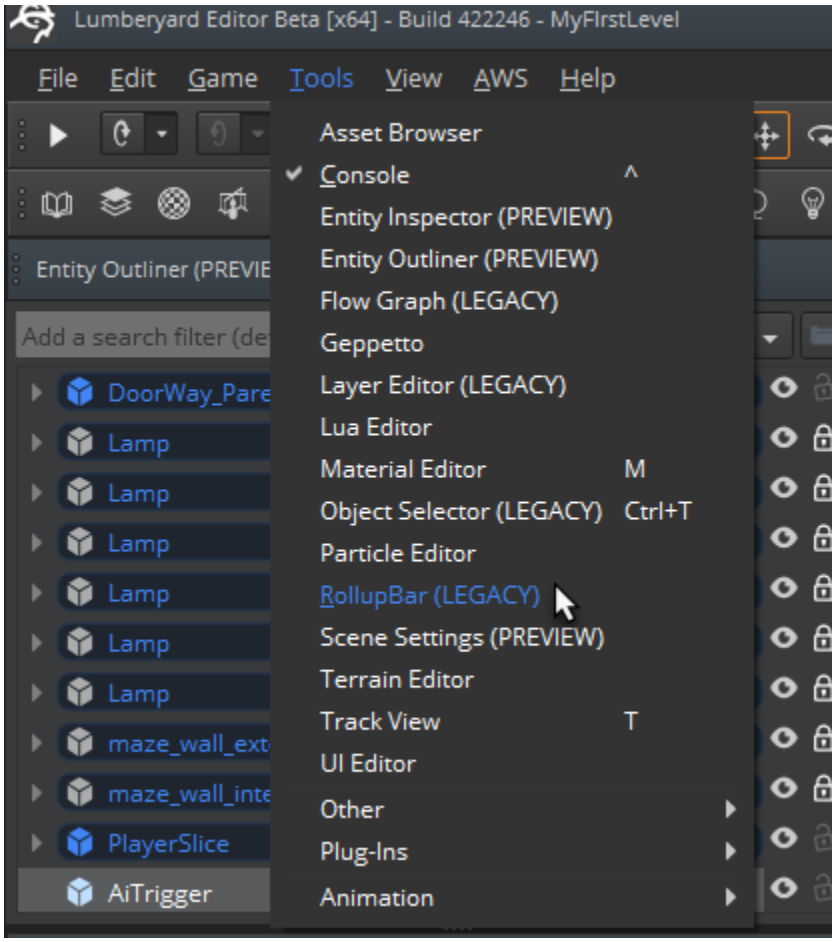
Defining the AI Navigation Area

The AI navigation area is the traversable area for the AI characters.

In this tutorial you use a legacy feature called **Rollup Bar** to define the AI navigation area.

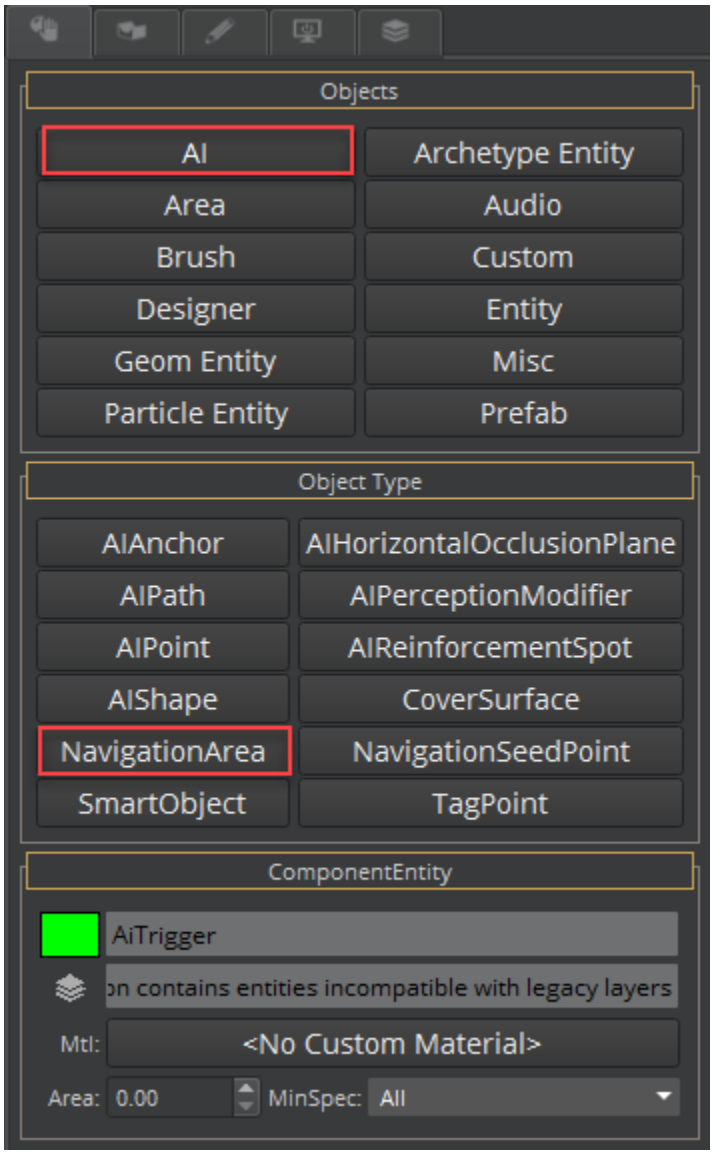
To define the AI navigation area

1. Open the **Rollup Bar** by clicking **Tools**, **RollupBar (LEGACY)**.



2. Under **Objects**, click **AI**.

   Under **Object Type**, click **NavigationArea**.



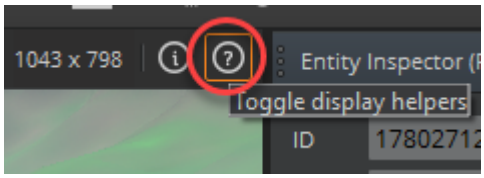
3. The **Navigation Area** tool is a point-and-click shape builder. You'll create a box that is within the **AiTrigger** volume that you created earlier in this tutorial.

   To create the box, click one corner inside the maze walls and then work your way around to the other corners. On the fourth corner, double-click to exit the tool and close the area that you defined.

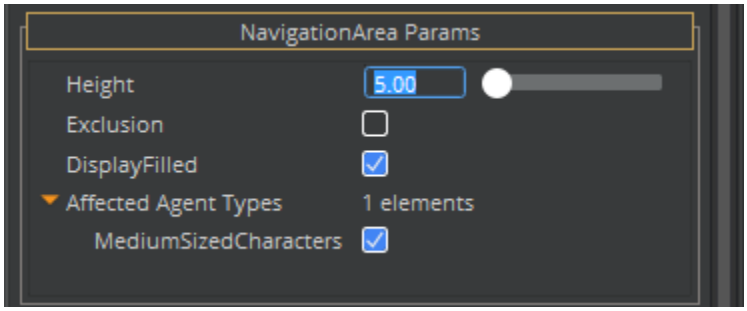
   Tip
   If the tool restricts you from moving in certain directions, ensure that no directional restrictions are selected in your toolbar.

   Tip
   If you cannot see the outline of the navmesh as you click, you likely have helpers turned off.

   Turn on helpers  in the top toolbar to view the navmesh outline.



4. Adjust the height of the AI navigation area. To do this, in the **Rollup Bar**, under **NavigationArea Params**, set **Height** to `5.00`.



5. Press **Ctrl+S** to save your level.

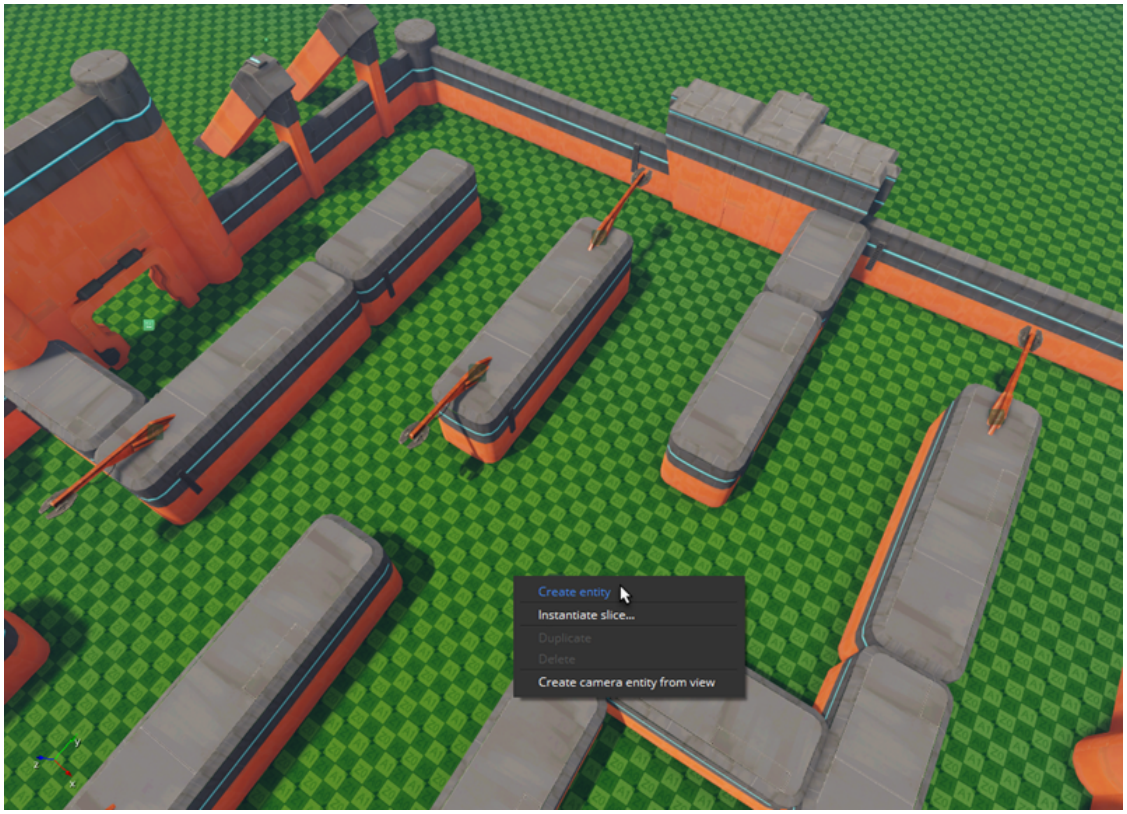
Next: Creating AI Spawn Points (p. 83)

Creating AI Spawn Points

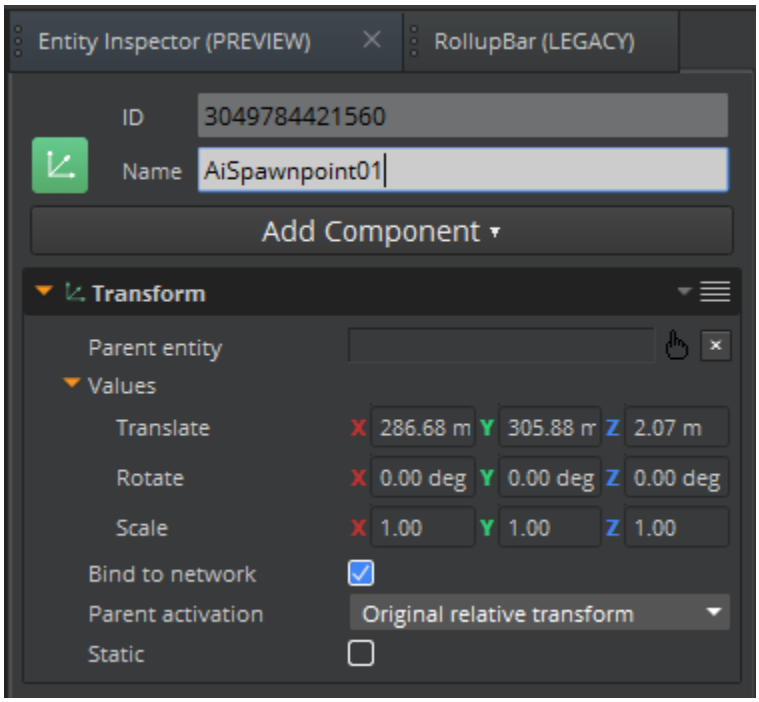
You can use AI spawn points to define the exact location that an enemy AI appears in the level when activated.

To create AI spawn points

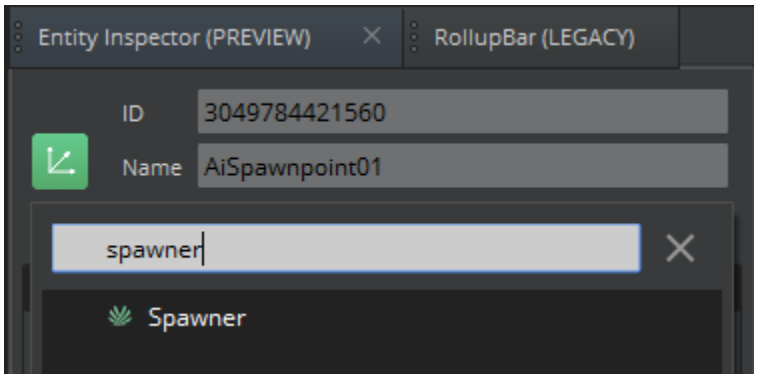
1. In the viewport, right-click in the center of your maze and choose **Create entity**.



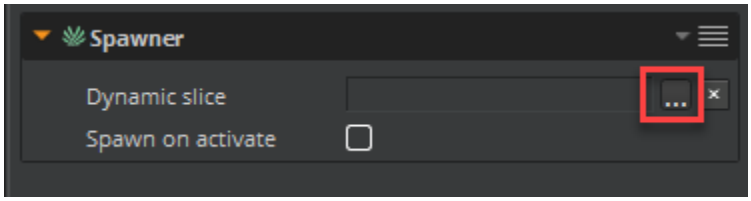
2. In the **Entity Inspector**, name the new entity **`AiSpawnpoint01`**.



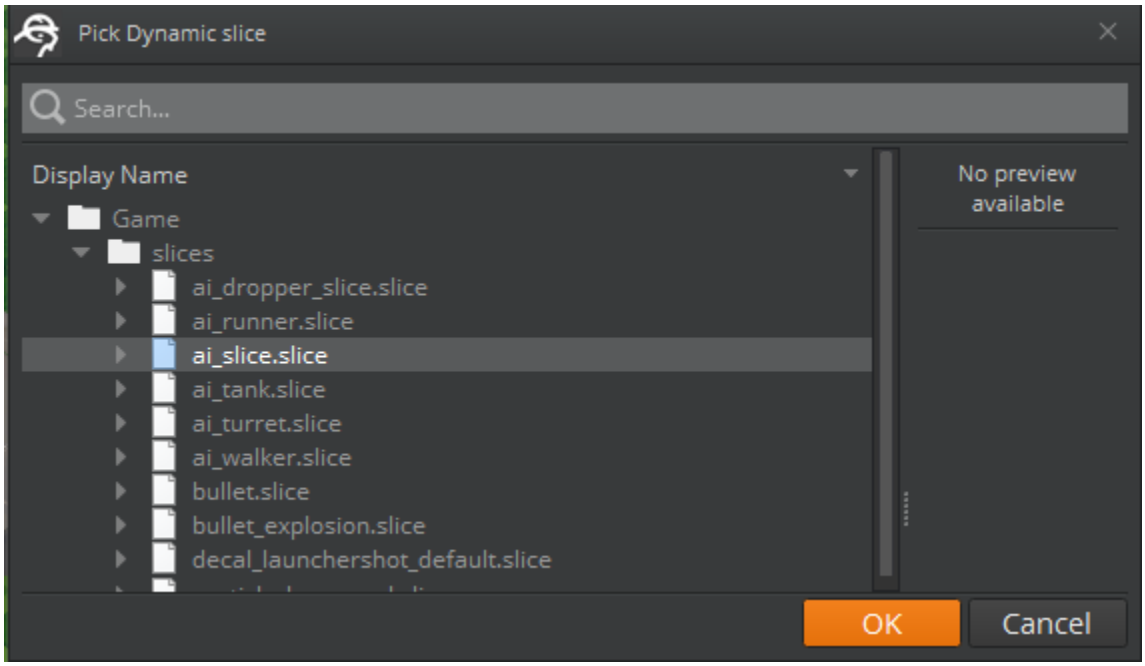
3. Click **Add Component**. Under the **Gameplay** category, click the **Spawner** component.



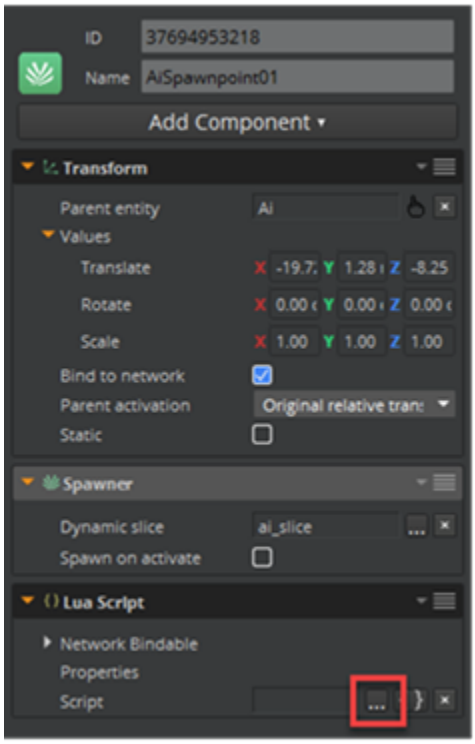
4. In the **Spawner** component, next to **Dynamic slice**, click **(...)**.

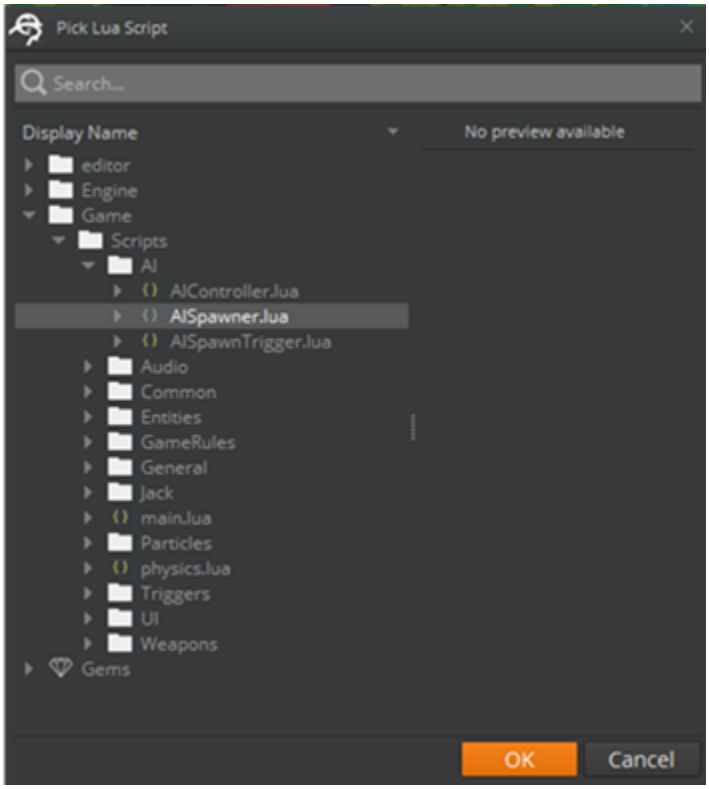


5. In the **Pick Dynamic slice** window, navigate to `Game\slices`. Select `ai_slice.slice` and click **OK**.



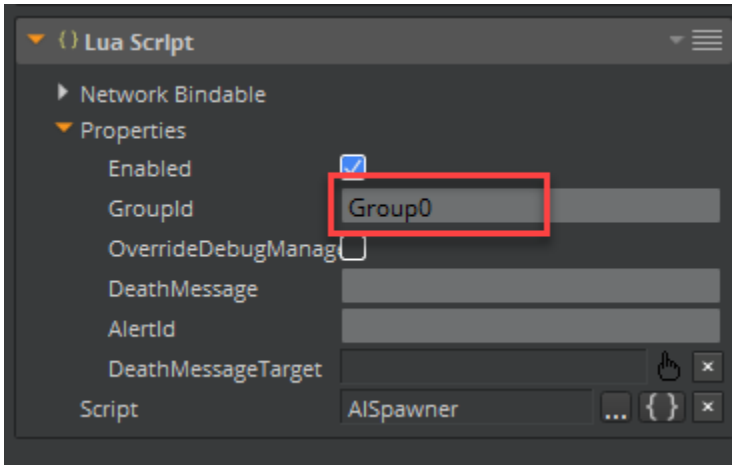
6. Add a Lua Script component to the **AiSpawnpoint01** entity. To do this, make sure the **AiSpawnpoint01** entity is still selected in the **Entity Outliner**. And then in the **Entity Inspector**, click **Add Component** and then click **Lua Script**.
7. To add the script to the **Lua Script** component, next to **Script**, click **(...)**. Navigate to `Game\Scripts\AI`. Select `AISpawner.lua` and then click **OK**.





8. In the **Lua Script** component's **GroupId** box, type **`Group0`**.

This links the **Spawner** component on this entity to the **AiTrigger** entity's **Lua Script** component, telling the trigger what group ID to spawn.



9. Press **Ctrl+S** to save your level.

Next: Adding Patrol Waypoints (p. 86)

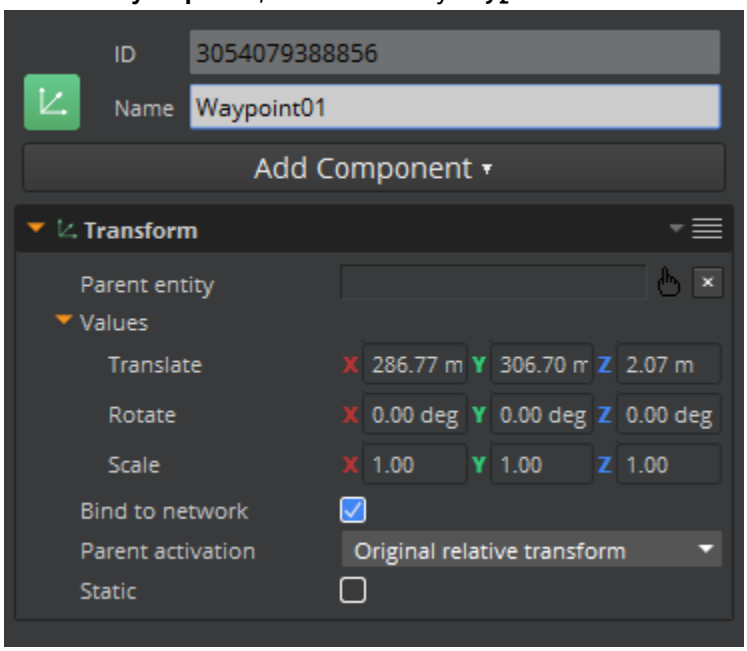
Adding Patrol Waypoints

A spawned AI character needs a defined path to walk along as it patrols the maze. To define the path, you place way points.

To add AI waypoints

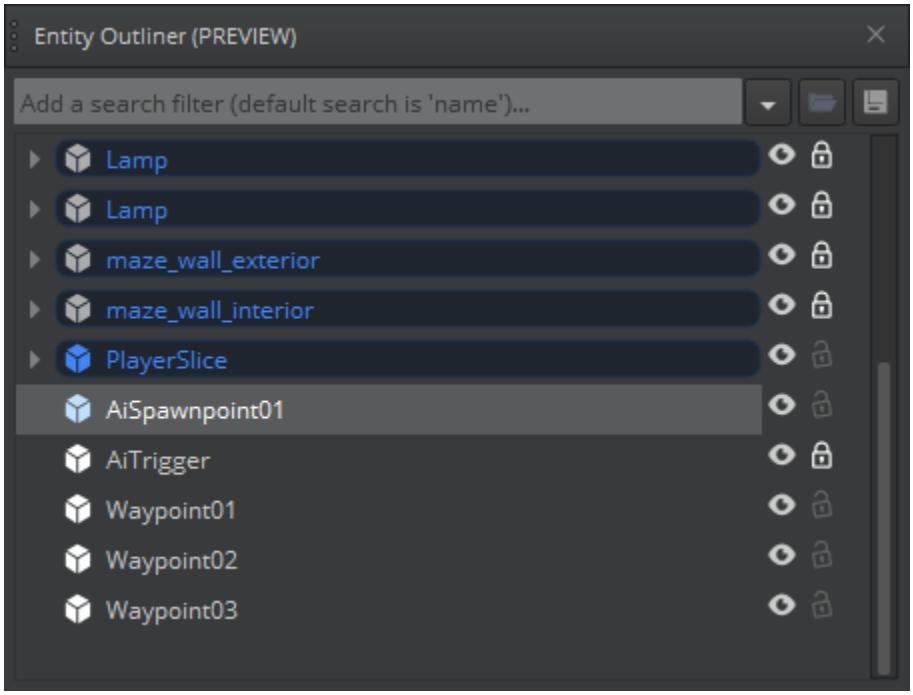
1. In the viewport, in the center of the maze near the spawn point entity that you created earlier, right-click and choose **Create Entity**.

2. In the **Entity Inspector**, name the entity **`Waypoint01`**.

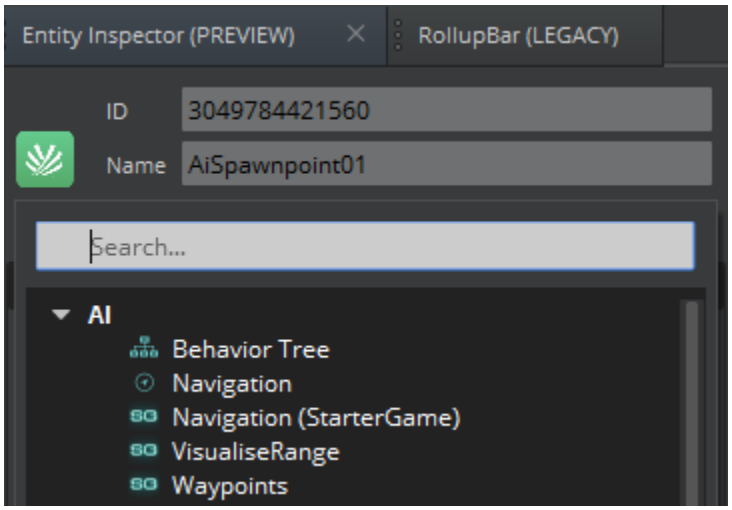


3. A short distance down the walkway from where you created your first waypoint, create another entity. Name it **`Waypoint02`**.
4. Create a third waypoint a short distance down a path perpendicular to the first two waypoints, so that the path between the three points forms an L shape. Name the waypoint **`Waypoint03`**.

5. In the **Entity Outliner**, select the **AiSpawnpoint01** entity you created earlier.



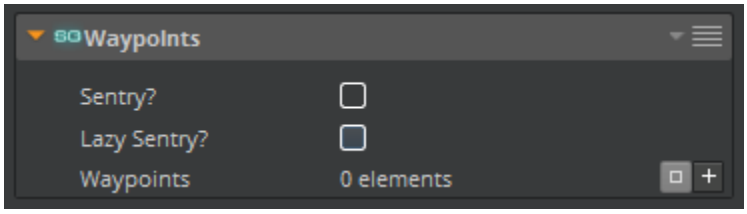
6. In the **Entity Inspector**, click **Add Component**. Under **AI**, click **Waypoints**.



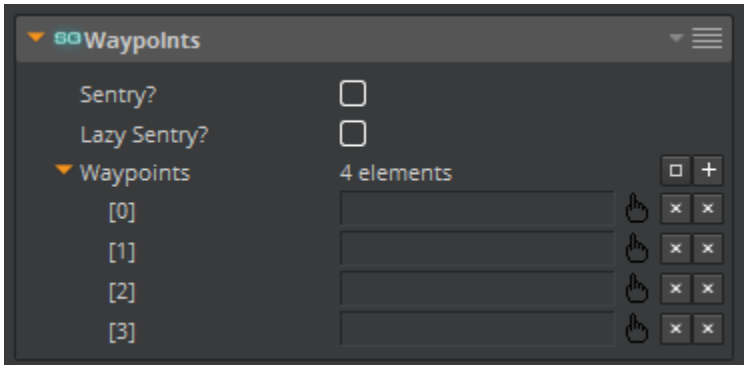
7. In the **Waypoints** properties, clear the **Sentry?** and **Lazy Sentry?** check boxes. This allows the AI to patrol between waypoints.

   The **Sentry?** property, when selected, causes the AI to stand in a fixed position, periodically turning to face different directions.

   The **Lazy Sentry?** property, when selected, causes the AI to look only in the direction it faced when it was spawned.



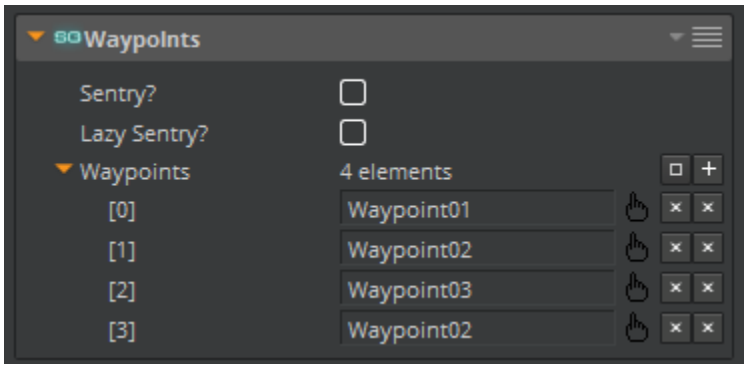
8. In the **Waypoints** properties, add four **Waypoints** elements by clicking **+** four times.



9. Use the pick icon  to select a waypoint for each of your waypoint elements.

   To do this, click the pick icon and then select the waypoint element in the viewport.

   Select the waypoints for the elements in the following order: **Waypoint01**, **Waypoint02**, **Waypoint03**, **Waypoint02**.



This sequence causes the AI to walk from **Waypoint01** to **Waypoint02** to **Waypoint03**, and then back to **Waypoint02**. In the next sequence, it begins at **Waypoint01**.

> **Note**
> Not setting waypoints causes the enemy to operate in sentry mode. This means that he stands at his spawn point and turn occasionally.
> When an AI spawns, it faces the in **X** direction (as opposed to the Y or Z direction).

10. Press **Ctrl+S** to save your level.

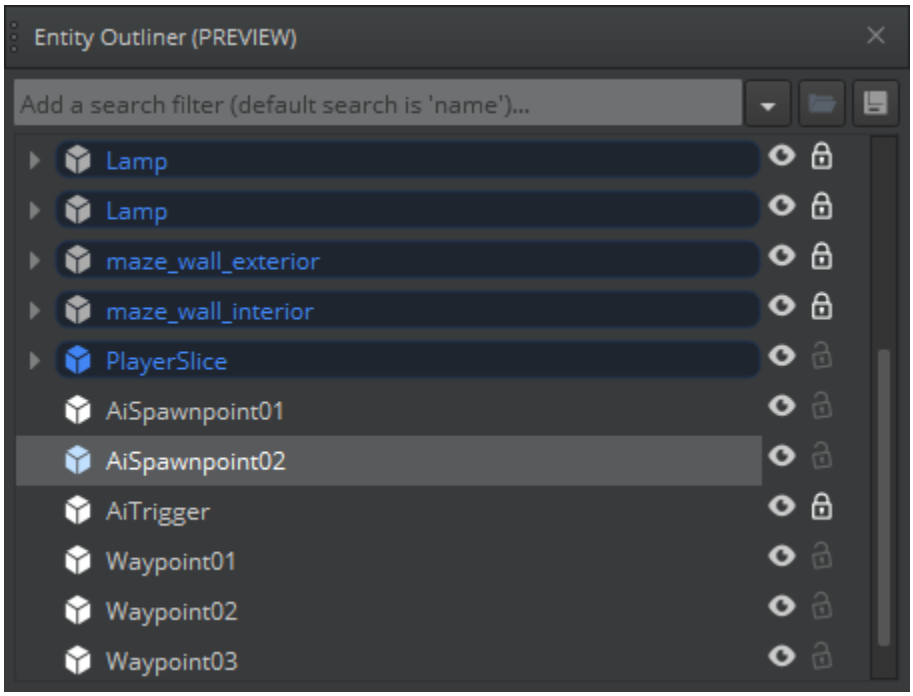
Next: Adding a Second AI Character (p. 90)

Adding a Second AI Character

With the same procedures you used to place the first AI character, you can place a second enemy AI that takes a slightly different path as it patrols the maze.

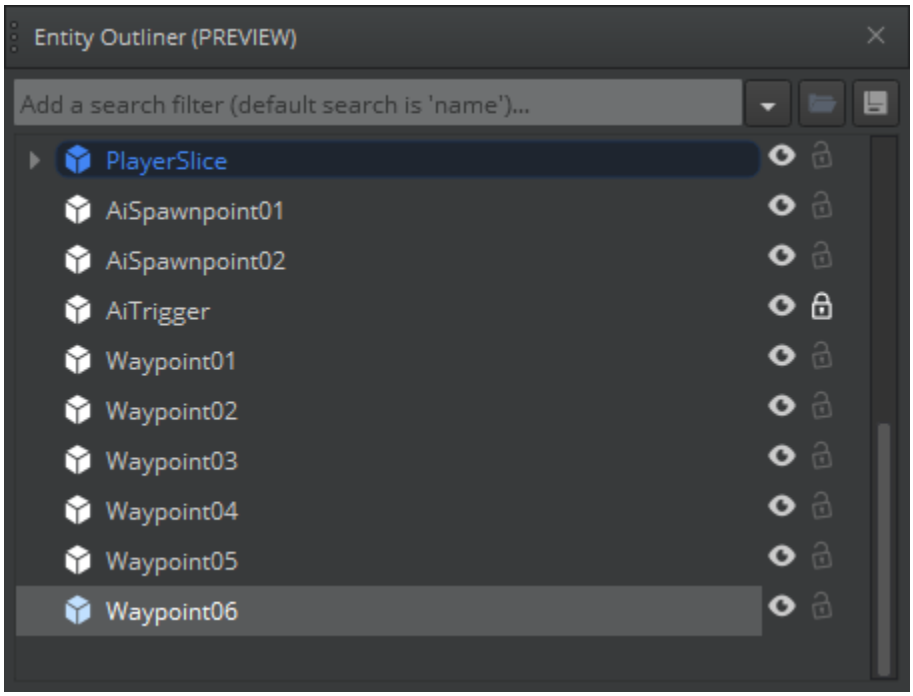
To add a second AI character

1. In the **Entity Outliner**, select and right-click **AiSpawnpoint01**. Choose **Duplicate**. Name the duplicated spawn point **Spawnpoint02**.



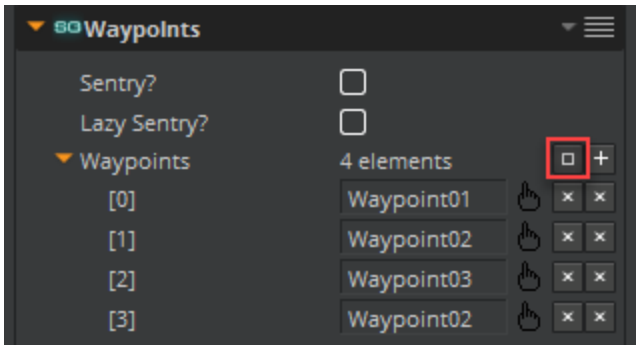
2. In the viewport, move the new spawn point to a location near the L shape of your waypoints, as shown in the following image.

3. Create three new entities to use as waypoints. Name them **Waypoint04**, **Waypoint05**, and **Waypoint06**.



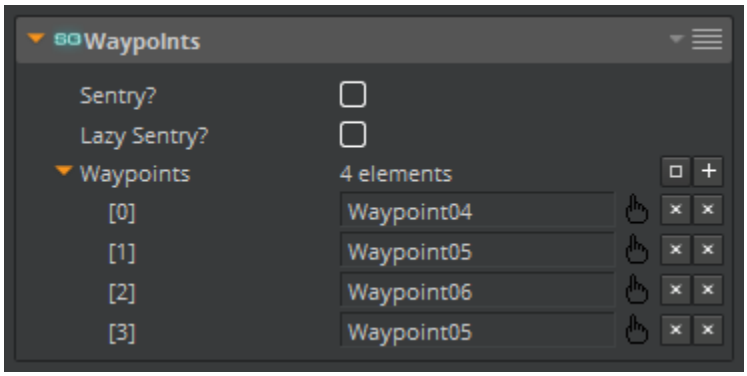
4. In the **Entity Outliner**, select **Spawnpoint02**.

   In the **Entity Inspector**, in the **Waypoints** component, clear all the existing waypoints by clicking the square icon next to the **Waypoints** setting.



5. Next to **Waypoints**, click the **+** icon. Add the following waypoints in this sequence:

   - **Waypoint04**
   - **Waypoint05**
   - **Waypoint06**
   - **Waypoint05**



6. Press **Ctrl+S** to save your level.

Next: Organizing Your AI Entities (p. 93)

Organizing Your AI Entities

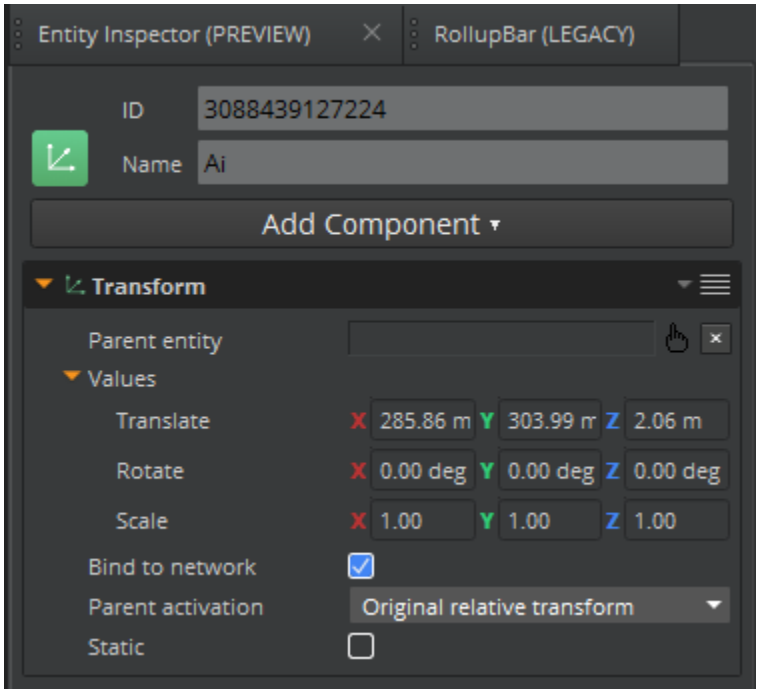
Keeping your entities organized helps you to manage your assets as you build the rest of the level.

In this tutorial you create a parent entity and then place all the AI entities into it.

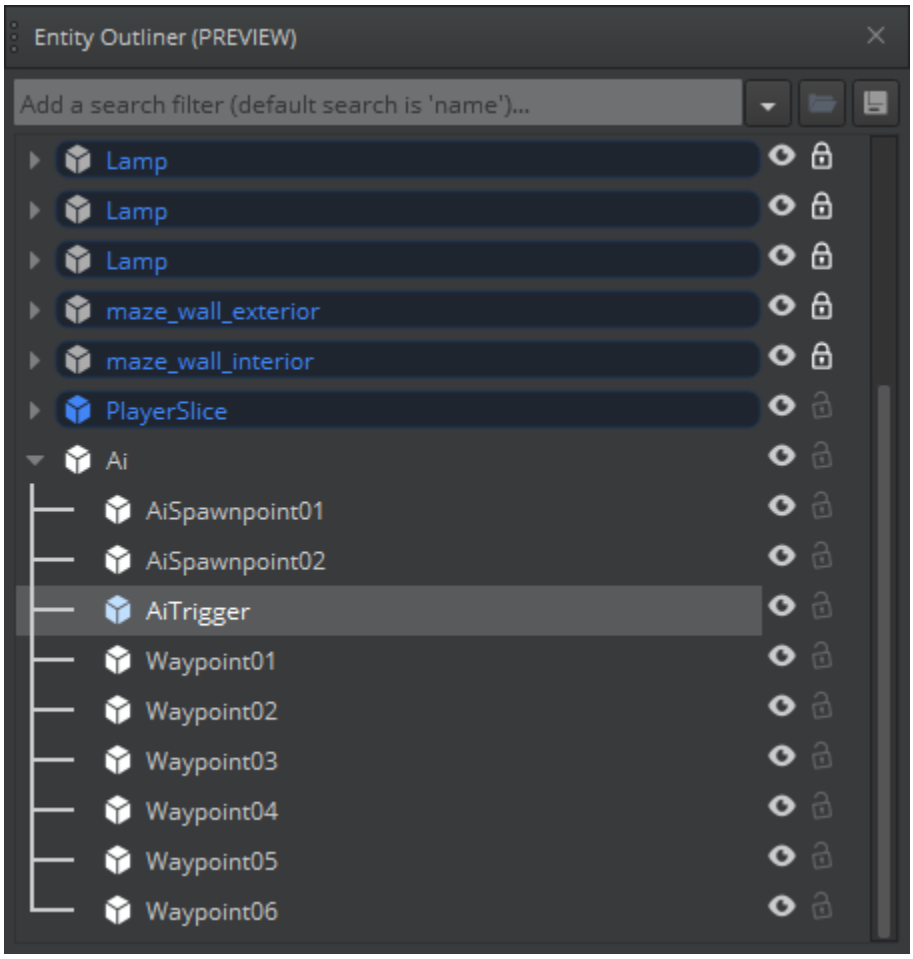
To organize your AI entities

1. Right-click in the viewport and choose **Create entity**. This new entity will be the parent entity for your AI entities.

   Name the new entity `Ai`.



2. In the **Entity Outliner**, drag the **AiSpawnpoint**, **AiTrigger**, and **Waypoint** entities onto the new **Ai** entity.



3. Press **Ctrl+S** to save your level.
4. Press **Ctrl+G** to play your level.

   Press **Esc** to quit.

This completes the AI tutorial set. In the next set of tutorials, you add simple scripts to craft gameplay.

Next: Adding the Particle Manager (p. 94)

Adding the Particle Manager

Next you'll add the particle manager slice to the level. The particle manager activates a range of visual fx in the game such as shoot and hit effects. It significantly enhances the combat experience with the AI that you have placed in the level.

To add the Particle Manager slice

1. In the **Asset Browser**, navigate to `Game\slices`.
2. Drag `particle_manager.slice` into the viewport.
3. Press **Ctrl+S** to save your level.

4. Press **Ctrl+G** to play your level.

   Shoot at the walls and at your sentries. Notice how weapon fire and hit FX now appear.

5. To quit, press **Esc**.

Next: 6: Scripting Gameplay (p. 96)

6: Scripting Gameplay

You use Lua script commands to script gameplay events such as triggering doors to open when the game character approaches and to close once he's through. You set up a timer to display in the UI as well as triggers that start and stop it. Finally, you add a pickup item that the character needs to exit the maze.

Scripting the simple gameplay involves the following tasks:

- Set up the entry door.
- Set up the exit door.
- Add a pickup item.
- Add a timer.

Topics

Next: Adding the Entry Door (p. 96)

Adding the Entry Door

In this tutorial, you'll create an entry door and add components that make it act solid and open upon approach.

To set up your entry door, you do the following:

1. Place the door entity (p. 96).
2. Add the physics and collider components to the door (p. 99).
3. Create a trigger volume (p. 102).
4. Add scripts to the entry door entity to do the following (p. 105):
   - Trigger an event upon exit and another event upon entry.
   - Change the position of an object and link it as the entry event.
   - Change the position of an object and link it as the exit event.
5. Define the door as the target object for the Lua scripts (p. 112).
6. Add realism by connecting sounds to the door's opening and closing events (p. 114).

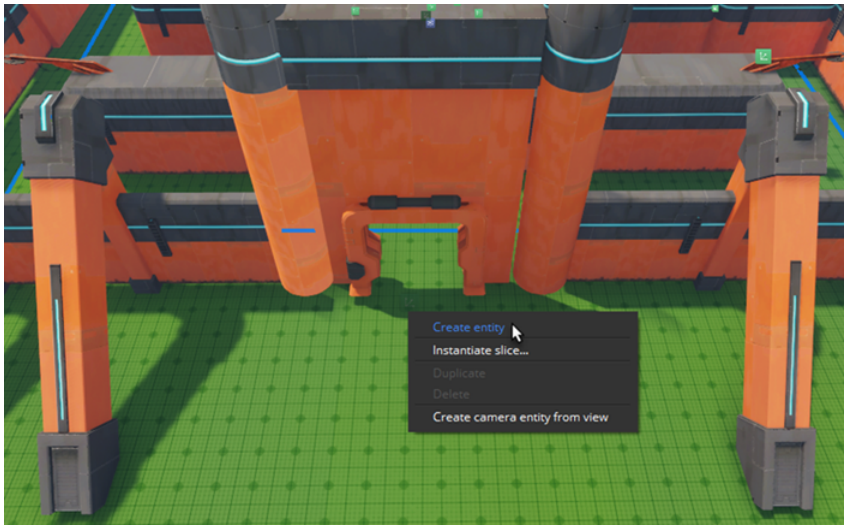
Next: the section called “Placing the Door Entity” (p. 96)

Placing the Door Entity

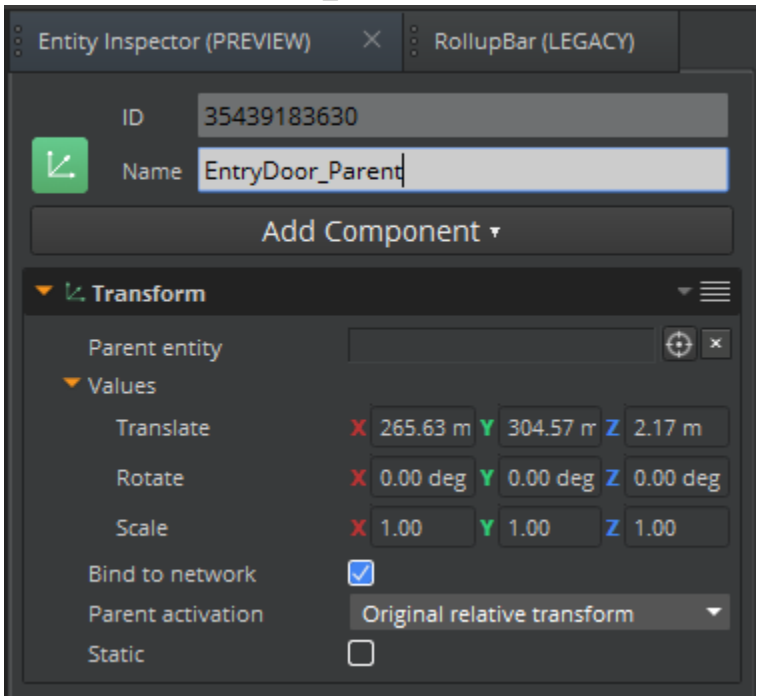
Earlier in this series of tutorials you placed a frame and posts, leaving an open doorway. Now it's time to place a door in that doorway.

To place a door

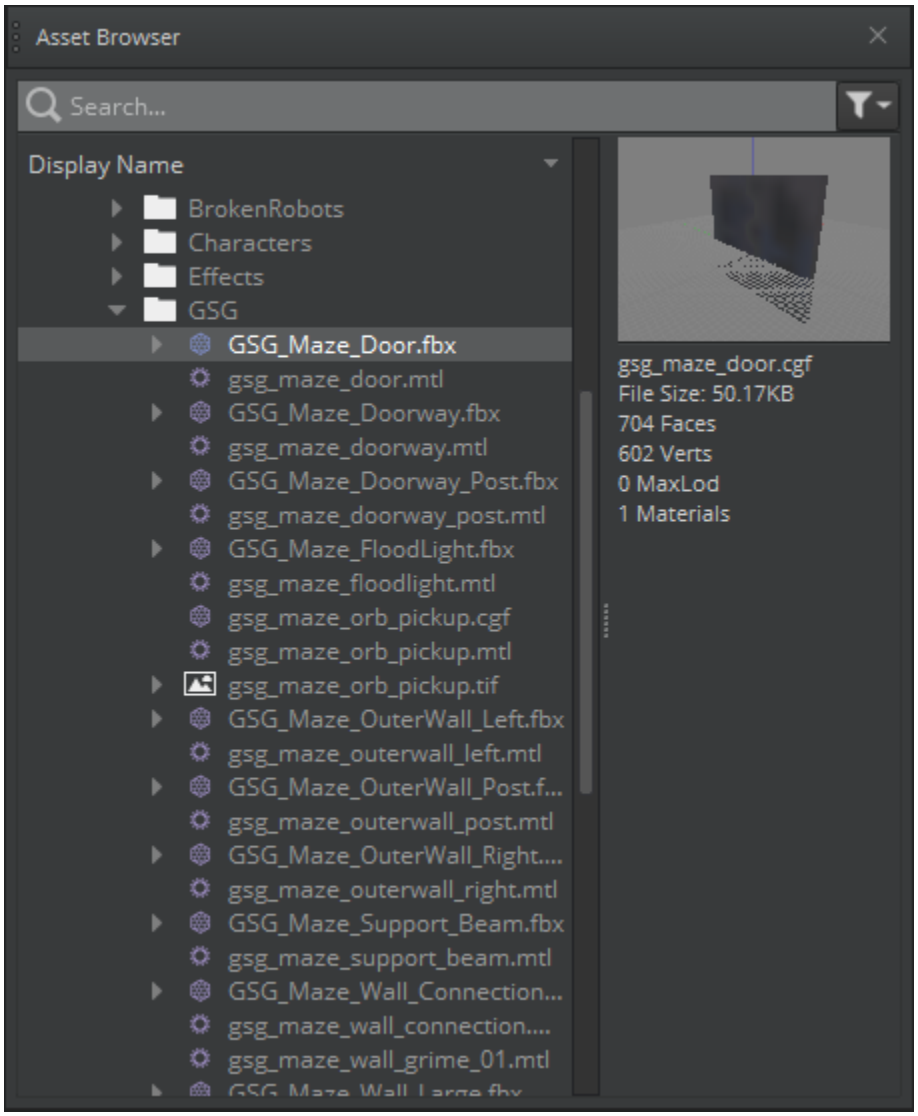
1. First, you create an empty entity to organize all your entry door entities. To do this, right-click in your viewport just outside one of your maze doors (but not inside of it).

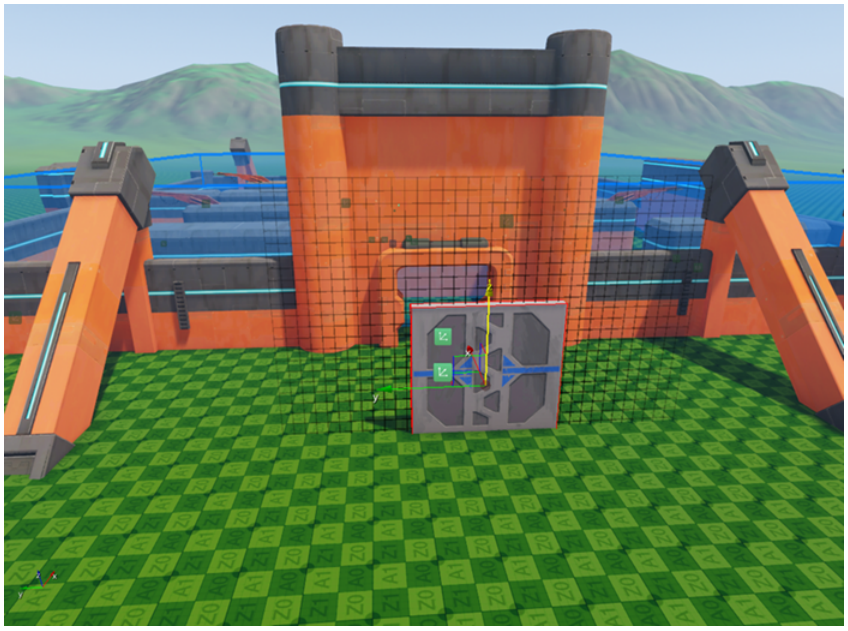


Name the entity **`EntryDoor_Parent`**.

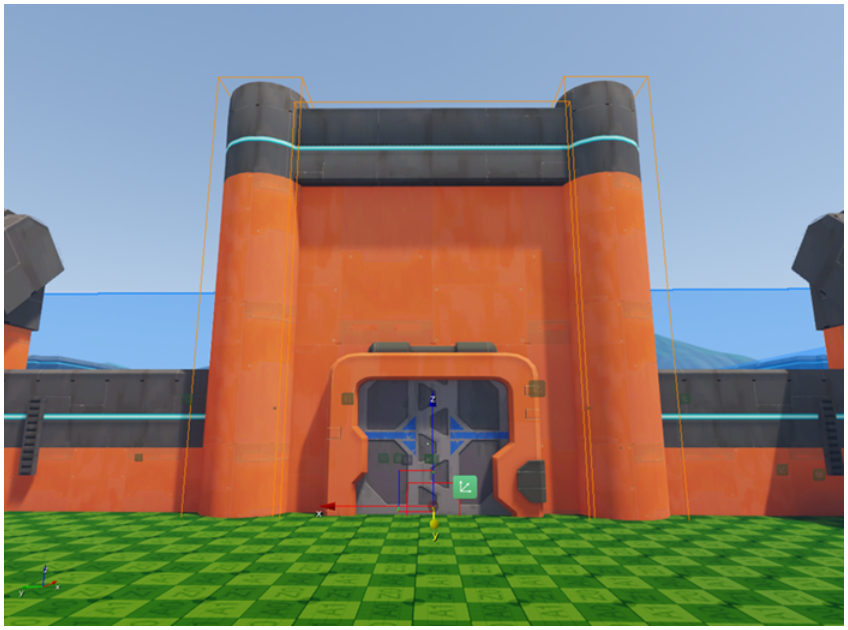


2. To add the door: In the **Asset Browser**, navigate to `Objects\GSG\` and drag `GSG_Maze_Door.fbx` into the viewport.


Asset Browser
Search...
Display Name
BrokenRobots
Characters
Effects
GSG
GSG_Maze_Door.fbx
gsg_maze_door.mtl
GSG_Maze_Doorway.fbx
gsg_maze_doorway.mtl
GSG_Maze_Doorway_Post.fbx
gsg_maze_doorway_post.mtl
GSG_Maze_FloodLight.fbx
gsg_maze_floodlight.mtl
gsg_maze_orb_pickup.cgf
gsg_maze_orb_pickup.mtl
gsg_maze_orb_pickup.tif
GSG_Maze_OuterWall_Left.fbx
gsg_maze_outerwall_left.mtl
GSG_Maze_OuterWall_Post.f...
gsg_maze_outerwall_post.mtl
GSG_Maze_OuterWall_Right....
gsg_maze_outerwall_right.mtl
GSG_Maze_Support_Beam.fbx
gsg_maze_support_beam.mtl
GSG_Maze_Wall_Connection...
gsg_maze_wall_connection....
gsg_maze_wall_grime_01.mtl
gsg_maze_door.cgf
File Size: 50.17KB
704 Faces
602 Verts
0 MaxLod
1 Materials



3. Use the **Move** tool to align the door in the open doorway so that it looks like the door is closed.



4. To keep your entities organized, drag **gsg_maze_door** onto **EntryDoor_Parent**.

   EntryDoor_Parent now has **gsg_maze_door** as a child.

Next: the section called "Adding Physics and Mesh Collider Components" (p. 99)

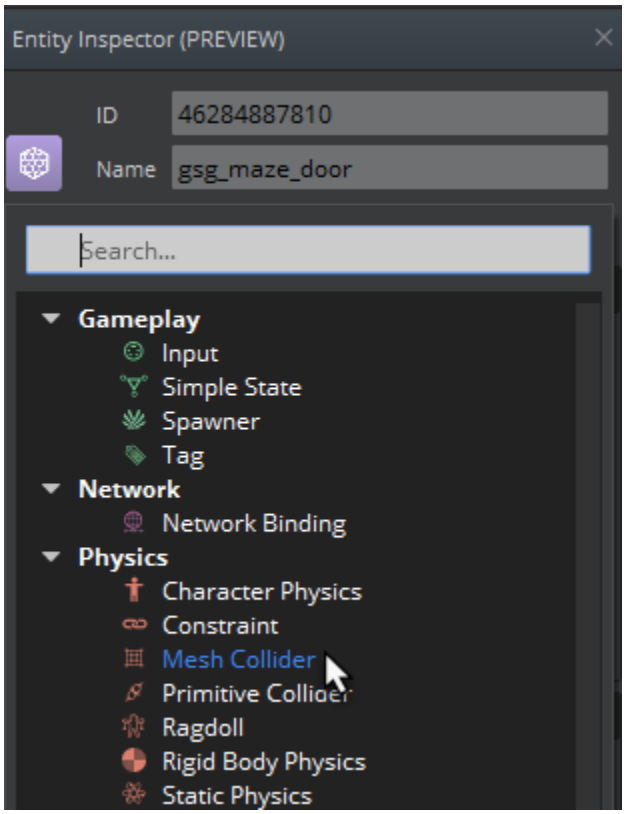
Adding Physics and Mesh Collider Components

As you learned in an earlier tutorial, a character can interact and collide with an object only if the object has collider and physics components. Without these components, a character passes right through the object.

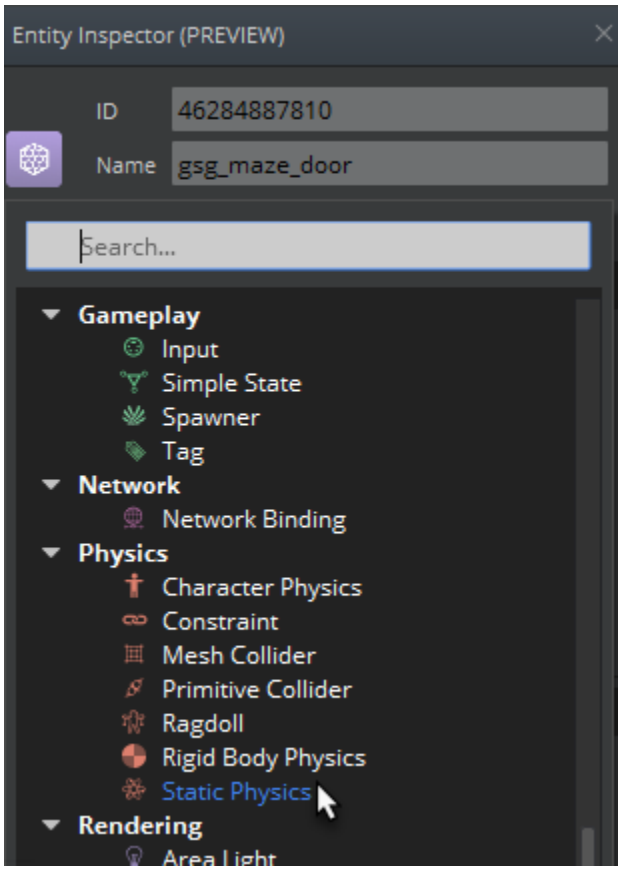
To add mesh collider and physics components to the door

1. In the **Entity Outliner**, select **gsg_maze_door**.

In the **Entity Inspector**, click **Add Component**. Under **Physics**, select **Mesh Collider**.



2. Click **Add Component** again. Under **Physics**, select the **Static Physics** component.



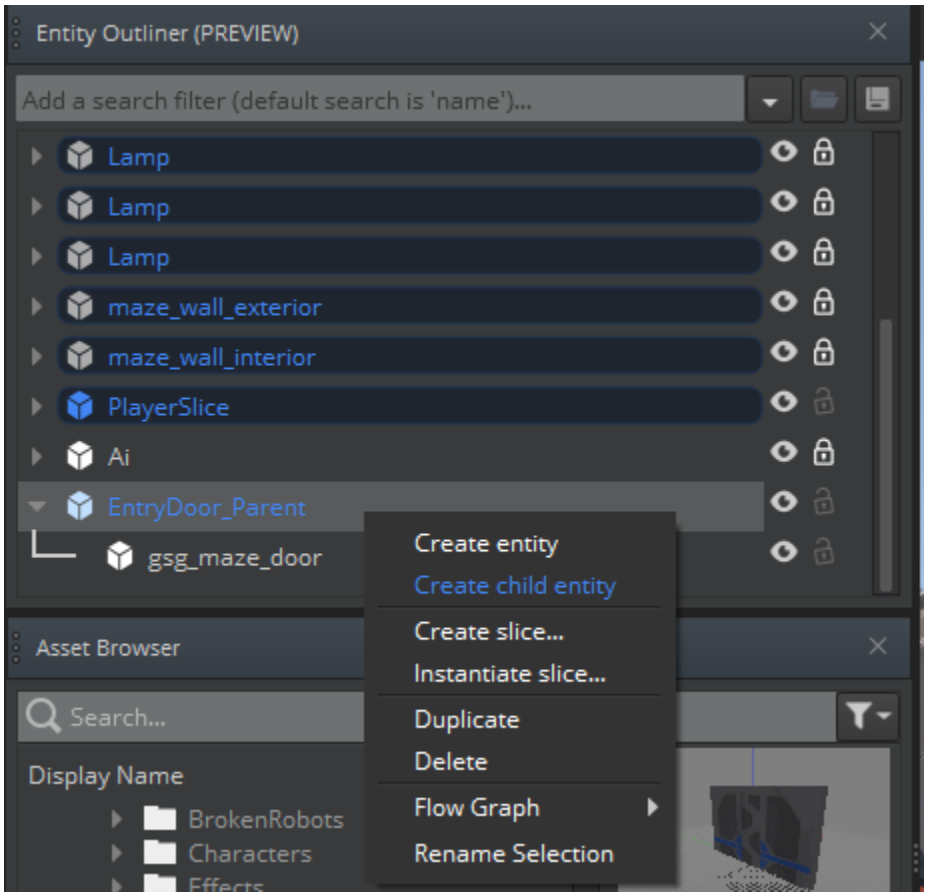
Next: the section called "Creating the Trigger Volume" (p. 102)

Creating the Trigger Volume

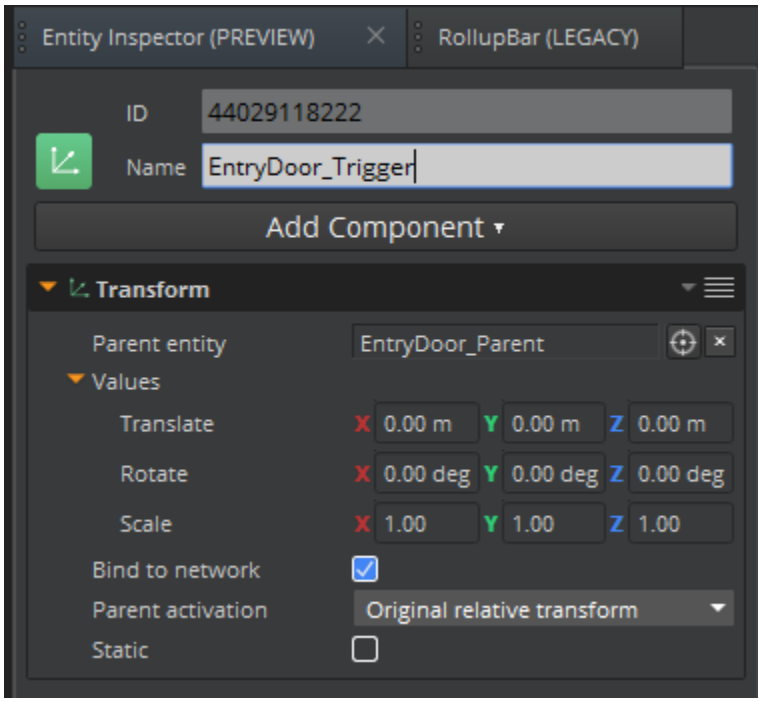
A trigger area is an invisible cube that you can set to trigger an action when a gameplay character enters and exits an area. You create a box-shaped trigger area and then move it to just outside of the maze door.

To create a trigger area

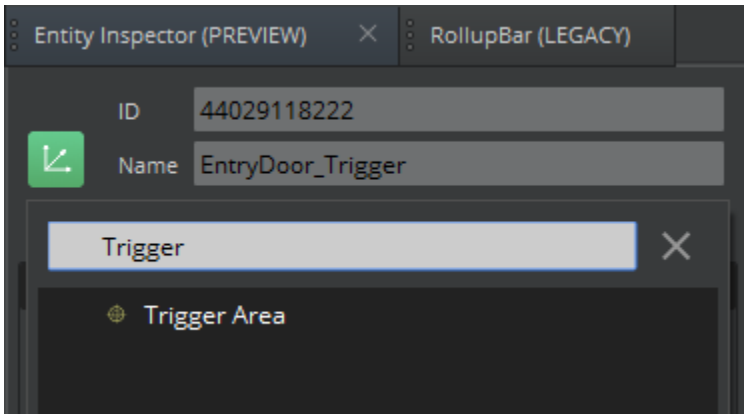
1. In the **Entity Outliner**, right-click the **EntryDoor_Parent** entity, and then select **Create child entity**.



2. In the **Entity Inspector**, rename this new entity **EntryDoor_Trigger**.

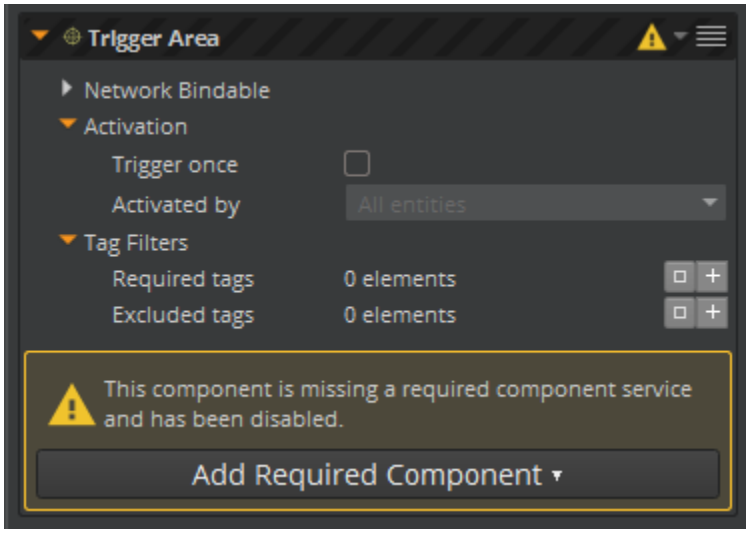


3. Ensure that the **EntryDoor_Trigger** is selected in the **Entity Outliner**. In the **Entity Inspector**, click **Add Component**. Under **Scripting**, select **Trigger Area**.

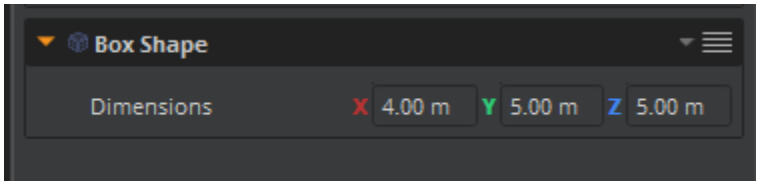


4. When adding any component that requires another component to function properly, the **Entity Inspector** displays a message: "This component is missing a required component service and has been disabled." To properly use the **Trigger Area** component, you must also add a shape component.

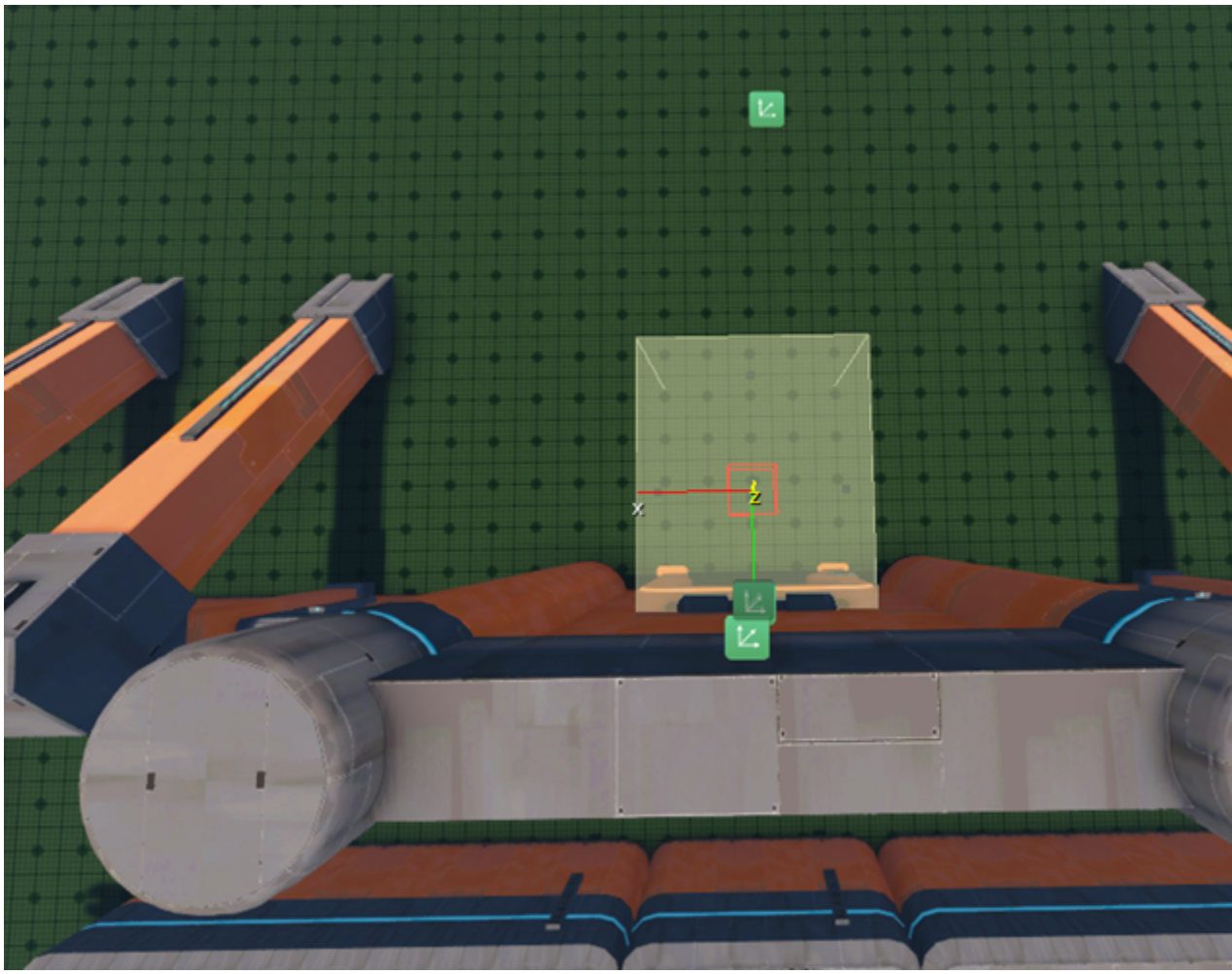
   Click **Add Required Component**, and then click **Box Shape**.



5. Set the box shape's **Dimensions** to **X**: `4.00` m, **Y**: `5.00` m, **Z**: `5.00` m.



6. Use the **Move** tool to adjust the position of the box-shaped trigger area so that it is just outside of the door. Align its edge with the inside edge of the door.



When the player walks toward the door and into the trigger area, the entry trigger action occurs —in this case, the door opens. The player then walks through the door, exiting the trigger volume, causing the exit trigger action—the door closes. This trigger area currently has no triggers set up; you do this in the next section.

Next: Adding Entry and Exit Scripts (p. 105)

Adding Open and Close Scripts

You need to add three Lua scripts to the trigger entity that you created. To do this, you add the **Lua Script** component three times and then select the script for each component.

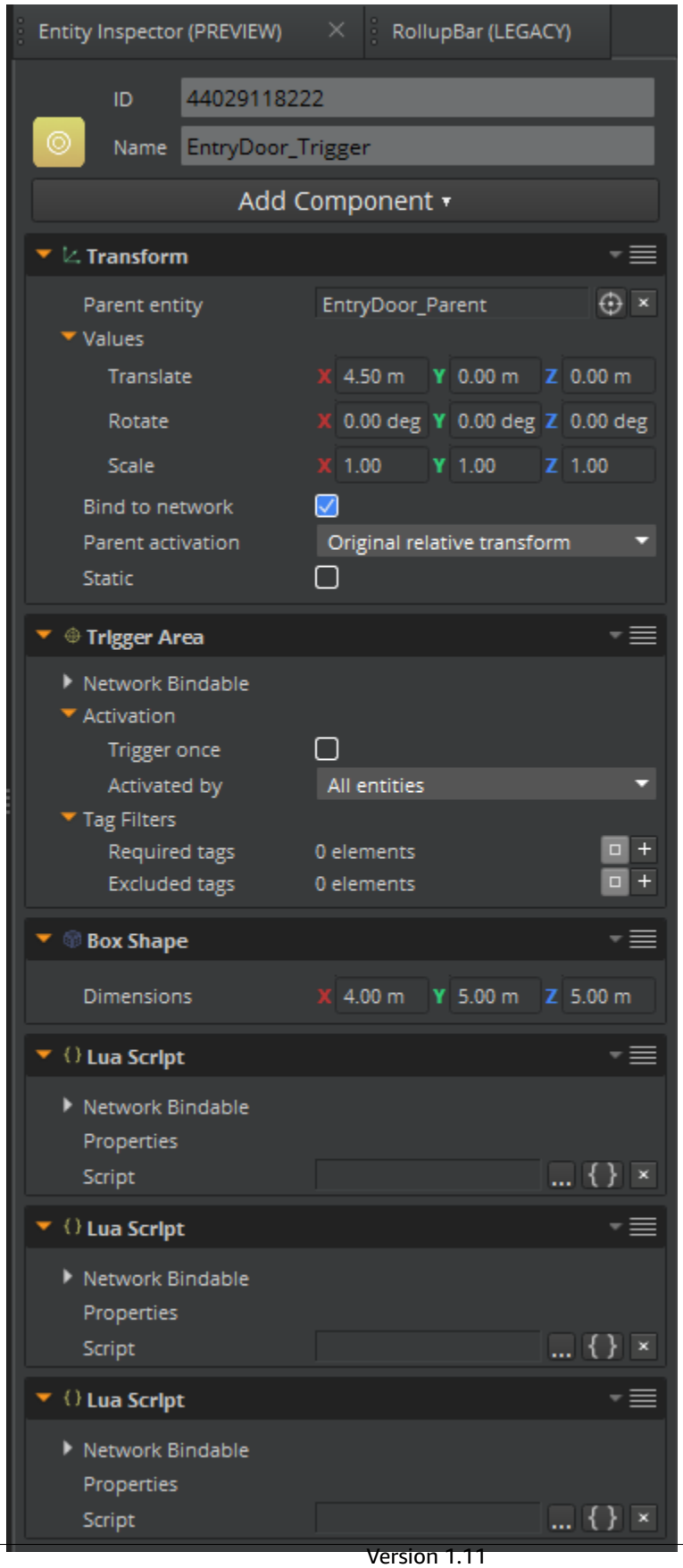
The first script that you add (`TriggerEvent.lua`) tells the trigger area what events to execute when a player enters and exits the area.

The second and third scripts that you add (`MoveObject.lua`) define those events specified in the first script, and link them to a script that moves the door. The `OpenExitDoor` script moves the Z position of the door so that it disappears up into the structure. The `CloseExitDoor` script moves the Z position back to its original, closed position.

To add Lua scripts to your trigger entity

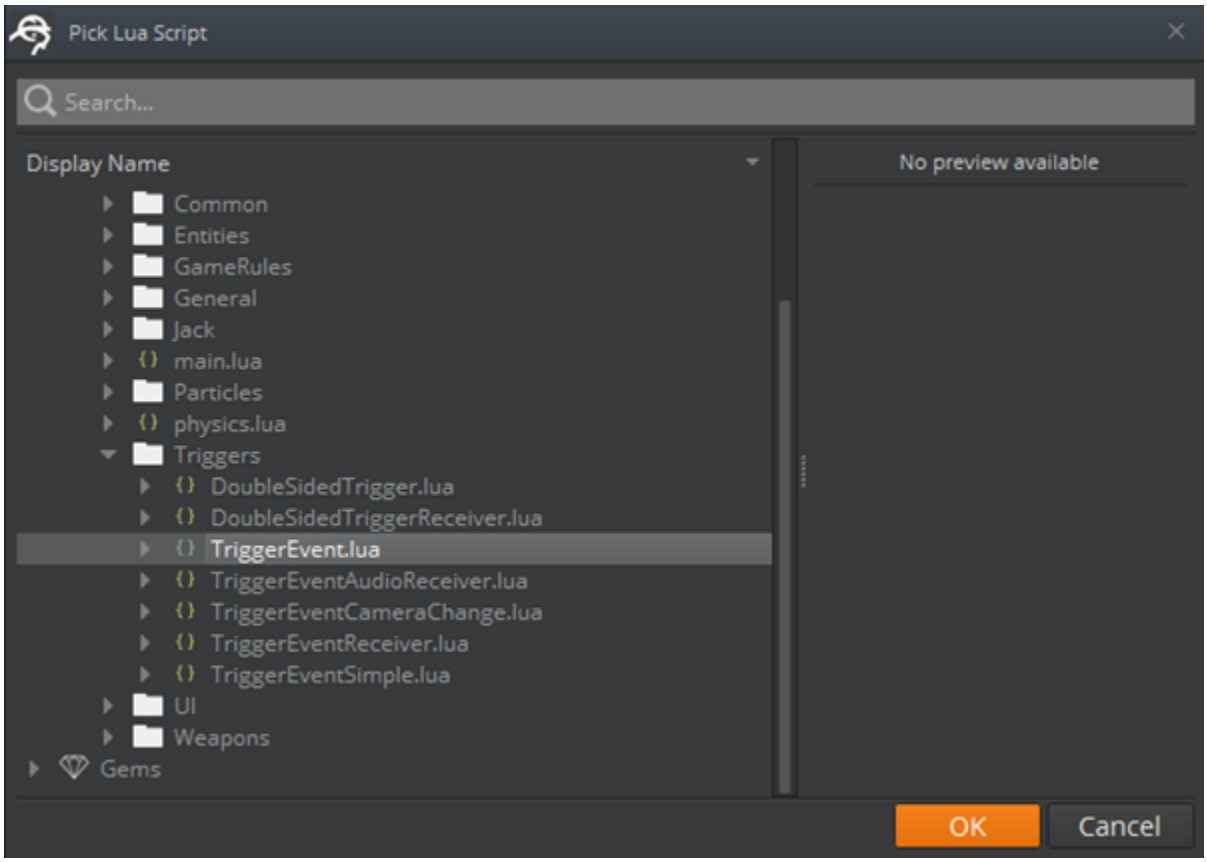
1. In the **Entity Outliner**, select the **EntryDoor_Trigger** entity.
2. In the **Entity Inspector**, click **Add Component**. Under **Scripting**, select **Lua Script**.

   Repeat this two more times so that the **EntryDoor_Trigger** has a total of three **Lua Script** components.


Entity Inspector (PREVIEW)
RollupBar (LEGACY)
ID
44029118222
Name
EntryDoor_Trigger
Add Component
Transform
Parent entity
EntryDoor_Parent
Values
Translate
X 4.50 m
Y 0.00 m
Z 0.00 m
Rotate
X 0.00 deg
Y 0.00 deg
Z 0.00 deg
Scale
X 1.00
Y 1.00
Z 1.00
Bind to network
Parent activation
Original relative transform
Static
Trigger Area
Network Bindable
Activation
Trigger once
Activated by
All entities
Tag Filters
Required tags
0 elements
Excluded tags
0 elements
Box Shape
Dimensions
X 4.00 m
Y 5.00 m
Z 5.00 m
Lua Script
Network Bindable
Properties
Script
Lua Script
Network Bindable
Properties
Script
Lua Script
Network Bindable
Properties
Script

3. In the first **Lua Script** component, next to the **Script** box, click **Browse** (**...**).

   Navigate to `Scripts\Triggers`. Select `TriggerEvent.lua`. Click **OK**.



   A list of options appears below the **Properties** heading in the **Lua Script** component.

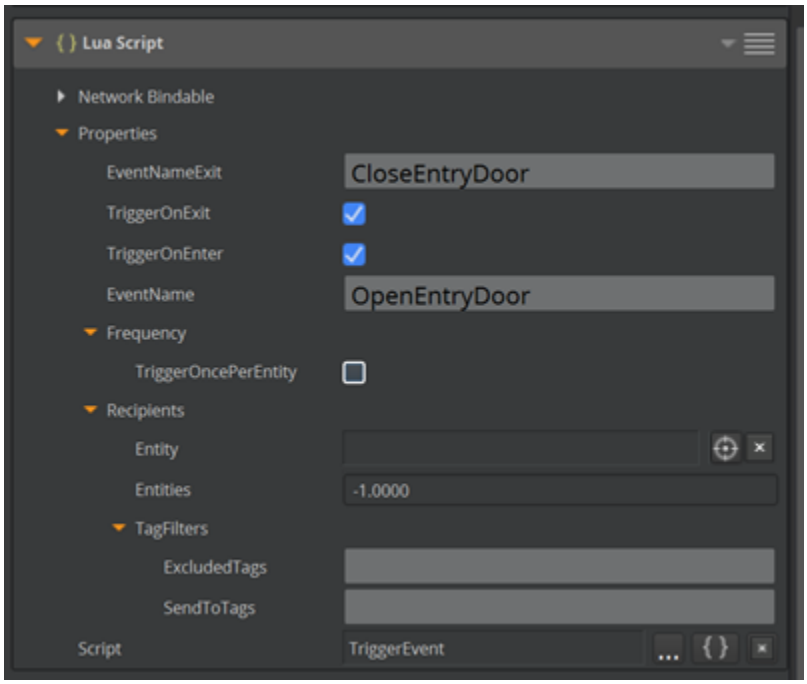
   For now, you'll configure the **EventNameExit** and **EventName** properties with names that you'll later define with the `MoveObject.lua` script. These can be any name you like; they just have to match the event names you configure in a later step.

4. Set the following options:

   - **EventNameExit** – Type **`CloseEntryDoor`**
   - **TriggerOnExit** – Selected
   - **TriggerOnEnter** – Selected
   - **EventName** – Type **`OpenEntryDoor`**
   - **TriggerOncePerEntity** – Unselected

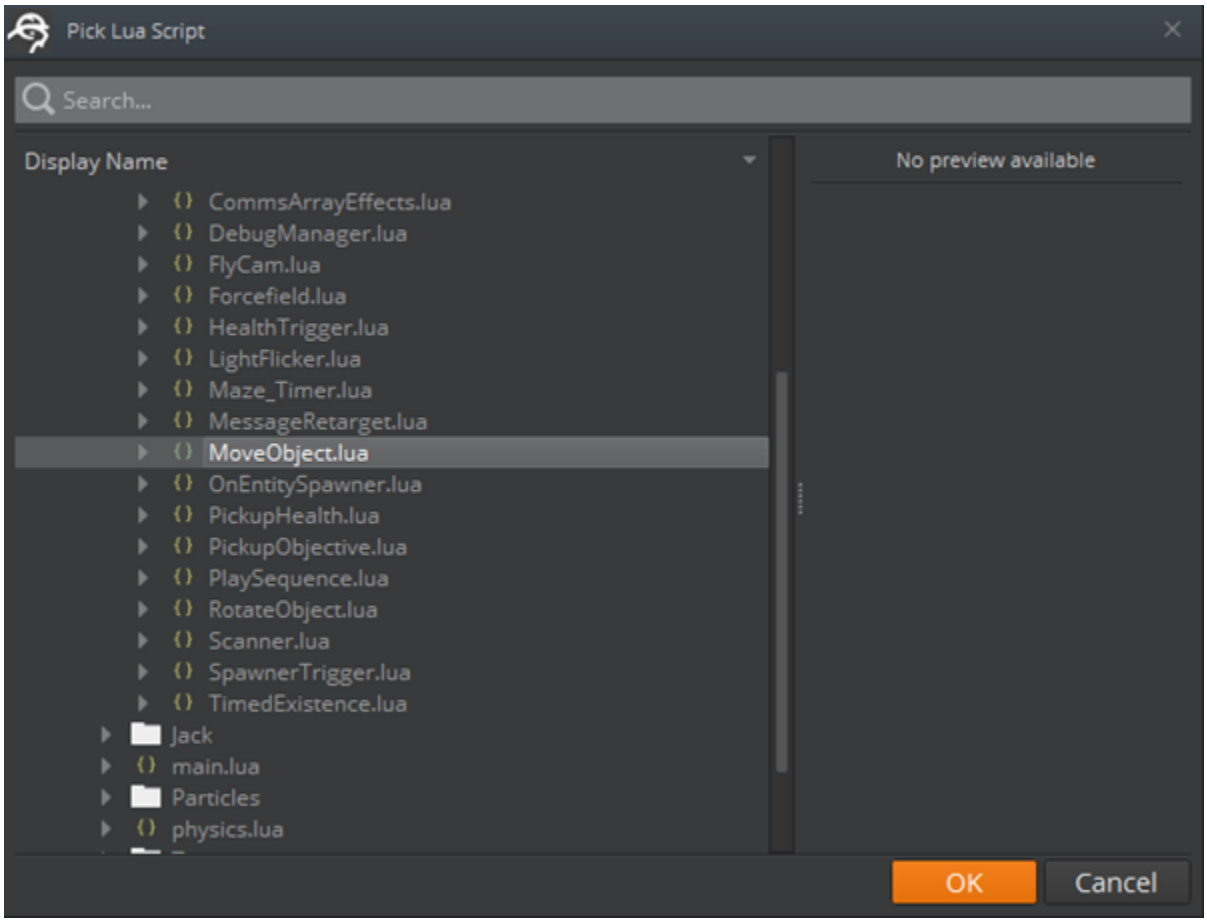
   These settings configure the named events to be triggered upon entry and exit.

   The **TriggerOncePerEntity**, when selected, sets that trigger volume to trigger just once by that entity, in this case the player character. Some game events may require this based on design.



5. Next, you'll add the script that moves the door into its open position.

   In the second **Lua Script** component, click **Browse (...)**. Navigate to `Game\Scripts\General`. Select `MoveObject.lua`. Click **OK**.

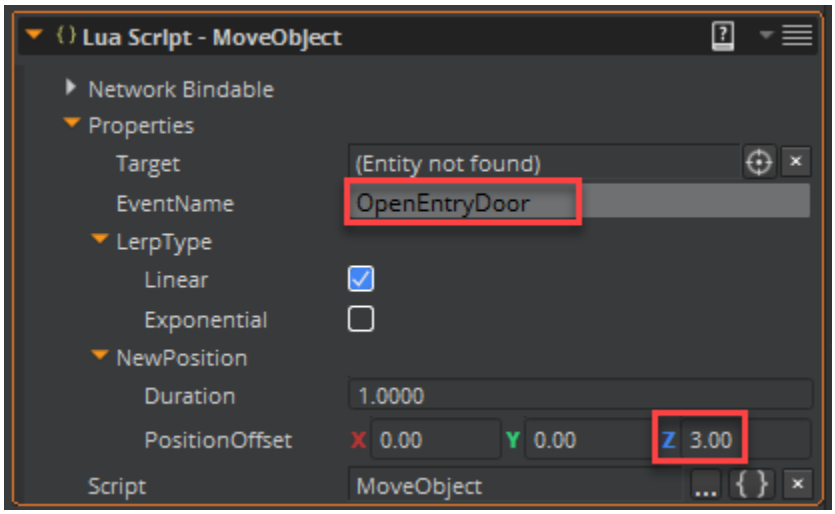


A list of options appears below the **Properties** heading in the **Lua Script** component.

6. As stated previously, the **EventName** can be any name you like, though it must match the **EventName** in the `TriggerEvent.lua` property.

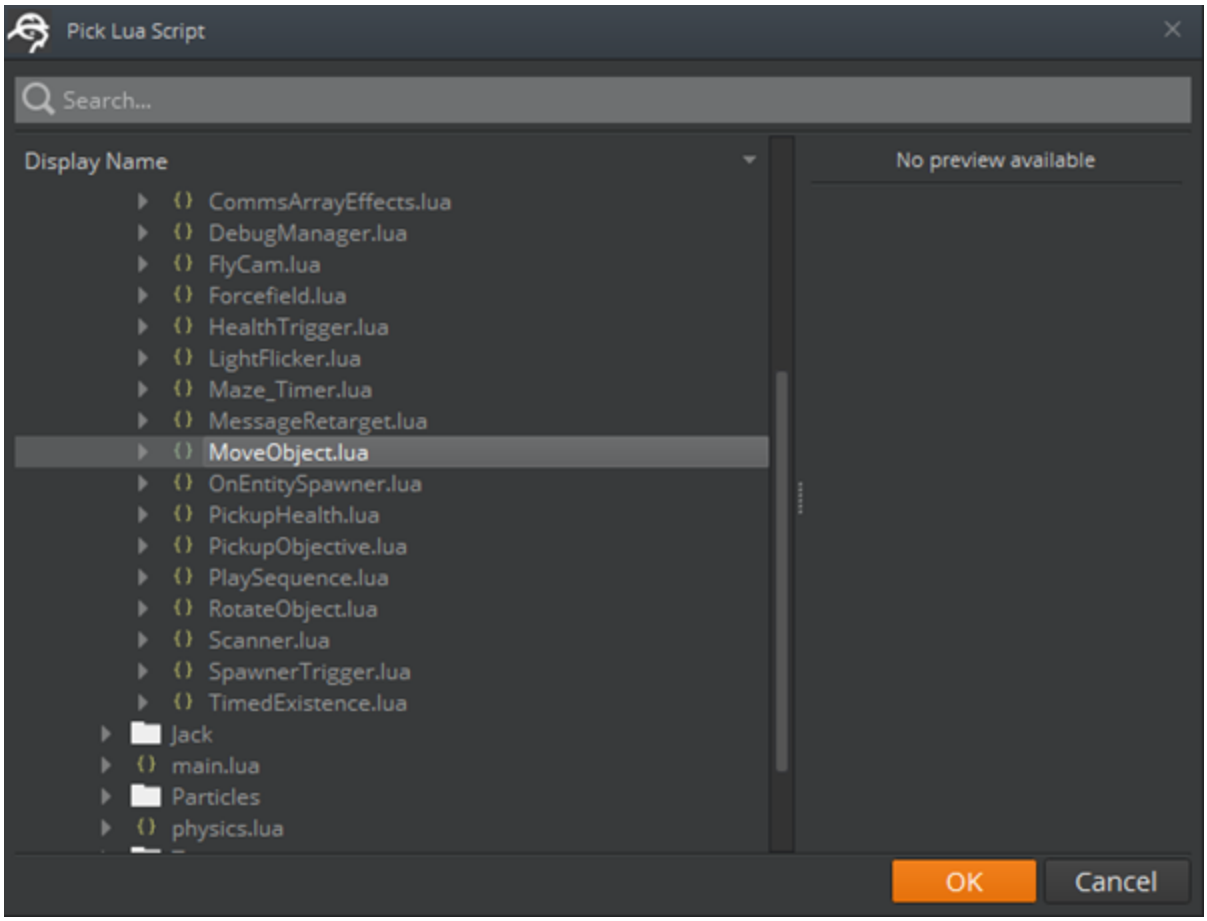
   Set the following options:

   - **EventName** – Type **`OpenEntryDoor`**
   - **PositionOffset** – Change **Z** to **`3.00`**



When the **OpenEntryDoor** event is triggered, this script moves the **Target**, which you define as the door in the next tutorial. The target is moved to an open position, which is determined by the relative offset that is specified by the **PositionOffset** values.

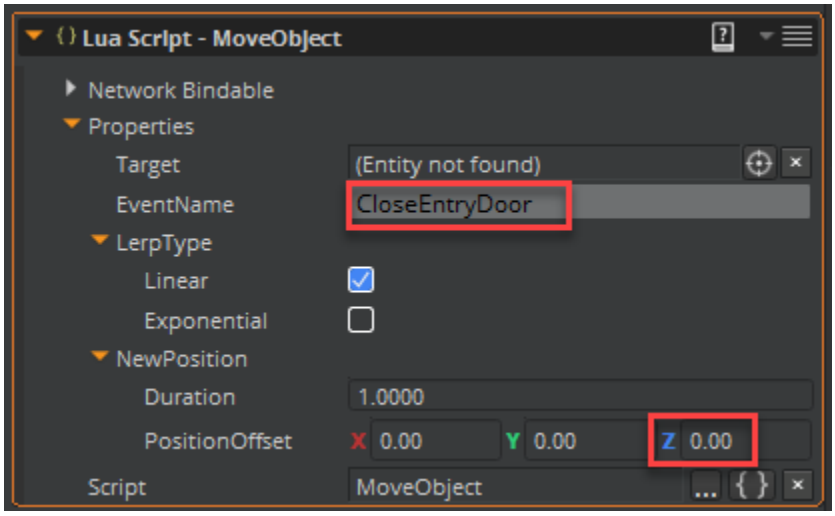
7. In the third **Lua Script** component, add the same `MoveObject.lua` script as you did for the previous **Lua Script** component.



A list of options appears below the **Properties** heading in the **Lua Script** component.

8. Set the following options:

   - **EventName** – Type `CloseEntryDoor`
   - **PositionOffset** – Set **X**, **Y**, and **Z** to `0.00`



   This script moves the **Target** back to its original, closed position when the **CloseEntryDoor** event is triggered.

Next: the section called "Defining the Door as the Target" (p. 112)

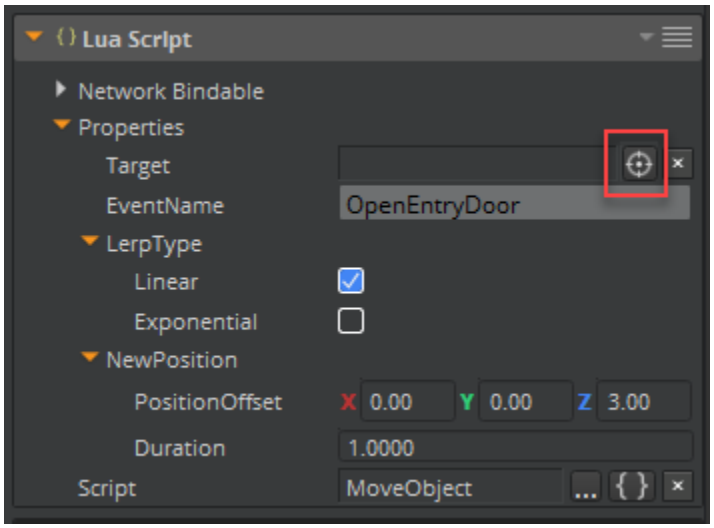
Defining the Door as the Target

The Lua scripts that you added in the previous section don't yet have a target to act upon. You need to specify the door as the target of the script's actions.

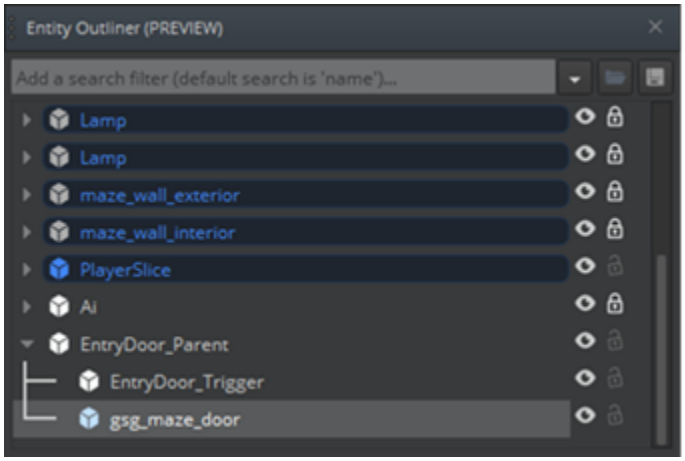
To define the door as the Lua script target

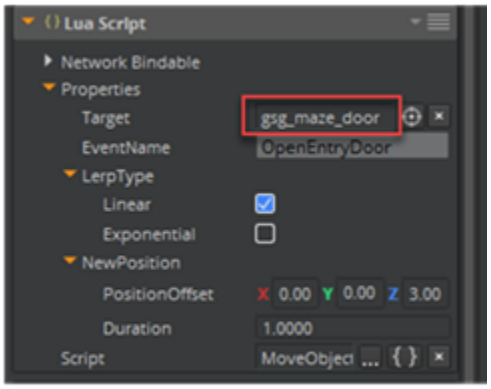
1. In the **Entity Outliner**,select **EntryDoor_Trigger**.

   In the **Entity Inspector**, in the second Lua Script component—the one with the **EventName** as **OpenEntryDoor**—click the picker icon  next to the **Target** box.



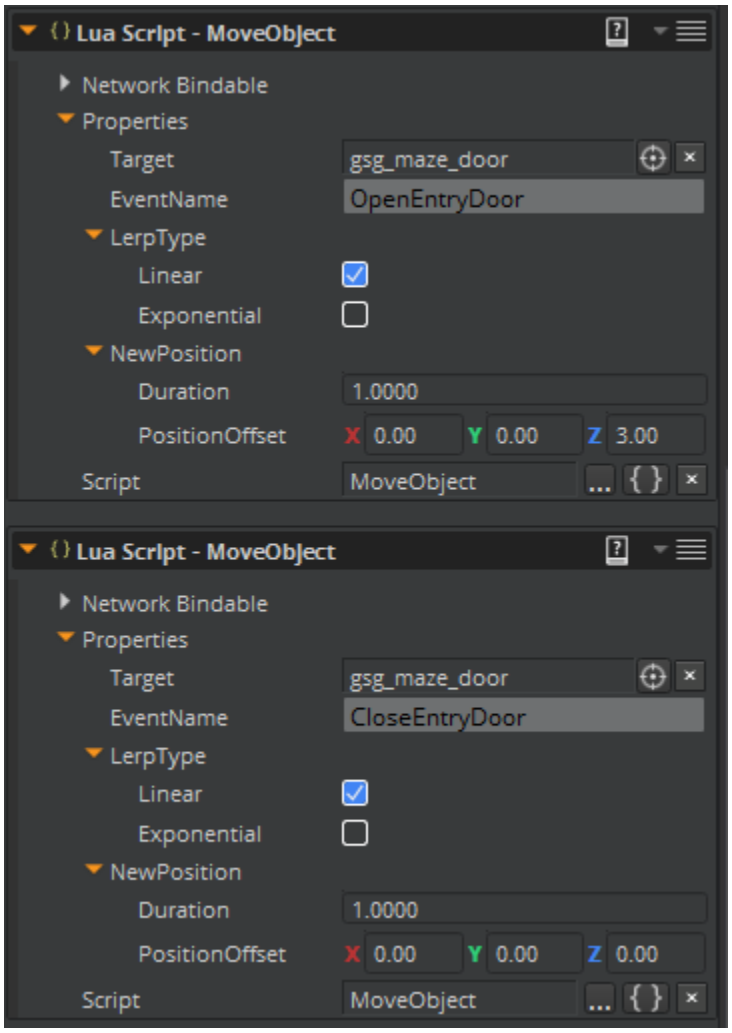
2. In the **Entity Outliner**, select **gsg_maze_door**. This fills the **Target** in the **Lua Script** component with the entity that you selected (**gsg_maze_door**).





3. Repeat the previous step for the third Lua script—the one with the **EventName** as **CloseEntryDoor**.

   Your two **Lua Script** components now look like the following image.



4. Press **Ctrl+S** to save your level.
5. Press **Ctrl+G** to play your level.

   Test your trigger area and triggers by approaching the door. The door should open when you enter the trigger area and close when you leave it.

   You may need to fine-tune the positions to ensure that the door is opening to the position you want.

   Press **Esc** to quit.

Next: the section called “Adding Door Sounds” (p. 114)

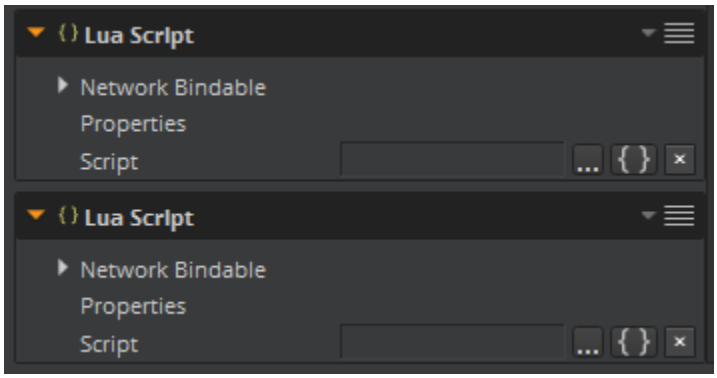
Adding Door Sounds

You add two **Lua Script** components and two audio components to the **EntryDoor_Trigger** entity to trigger sounds of a door opening and closing.

To add the Lua Script component for door sounds

1. In the **Entity Outliner**, select **EntryDoor_Trigger**.

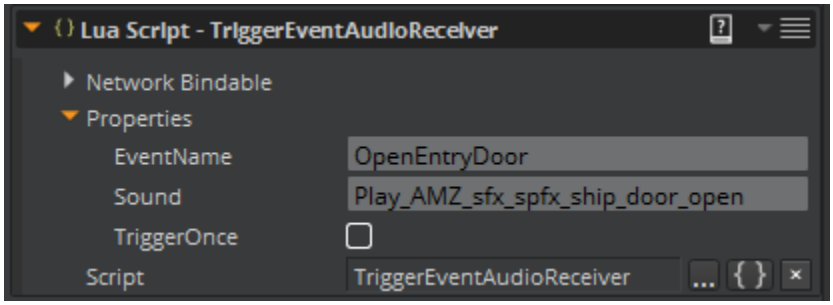
In the **Entity Inspector**, click **Add Component**, and then add the **Lua Script** component. Repeat this step to add a second **Lua Script** component.



2. In the first **Lua Script** component, click **Browse (...)** next to the **Script** box.

   Navigate to `Game\Scripts\Triggers` and select `TriggerEventAudioReceiver.lua`. Click **OK**.
3. In the properties that appear, set the following values:

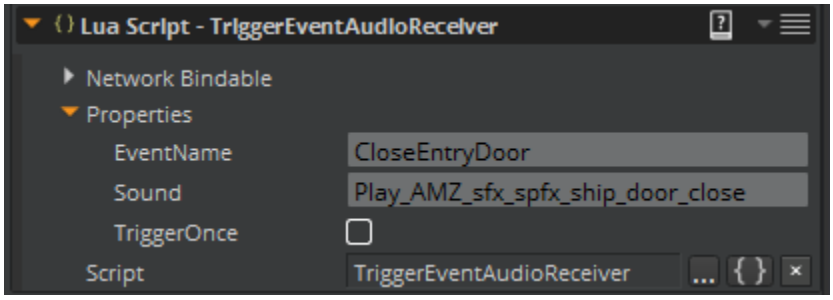
   - **EventName** – Type **`OpenEntryDoor`**
   - **Sounds** – Type **`Play_AMZ_sfx_spfx_ship_door_open`**
   - **TriggerOnce** – Clear the check box



4. In the second Lua Script component, click **Browse (...)** next to the **Script** box.

   Navigate to `Game\Scripts\Triggers` and select `TriggerEventAudioReceiver.lua`. Click **OK**.
5. In the properties that appear, set the following values:

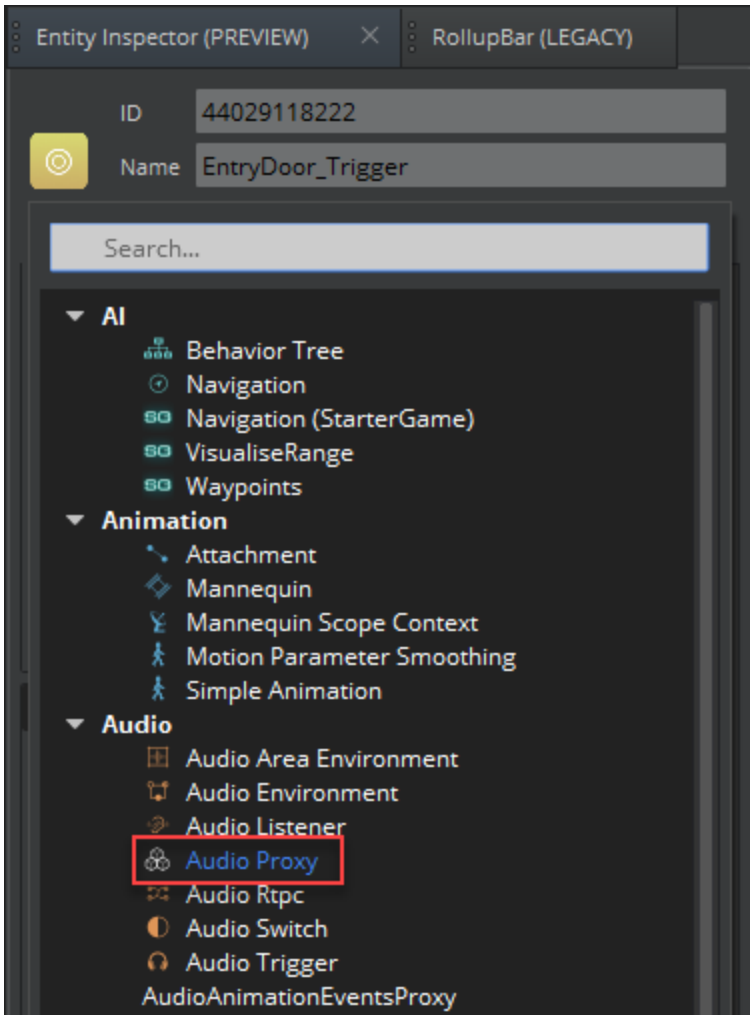
   - **EventName** – Type **`CloseEntryDoor`**
   - **Sounds** – Type **`Play_AMZ_sfx_spfx_ship_door_close`**
   - **TriggerOnce** – Clear the check box



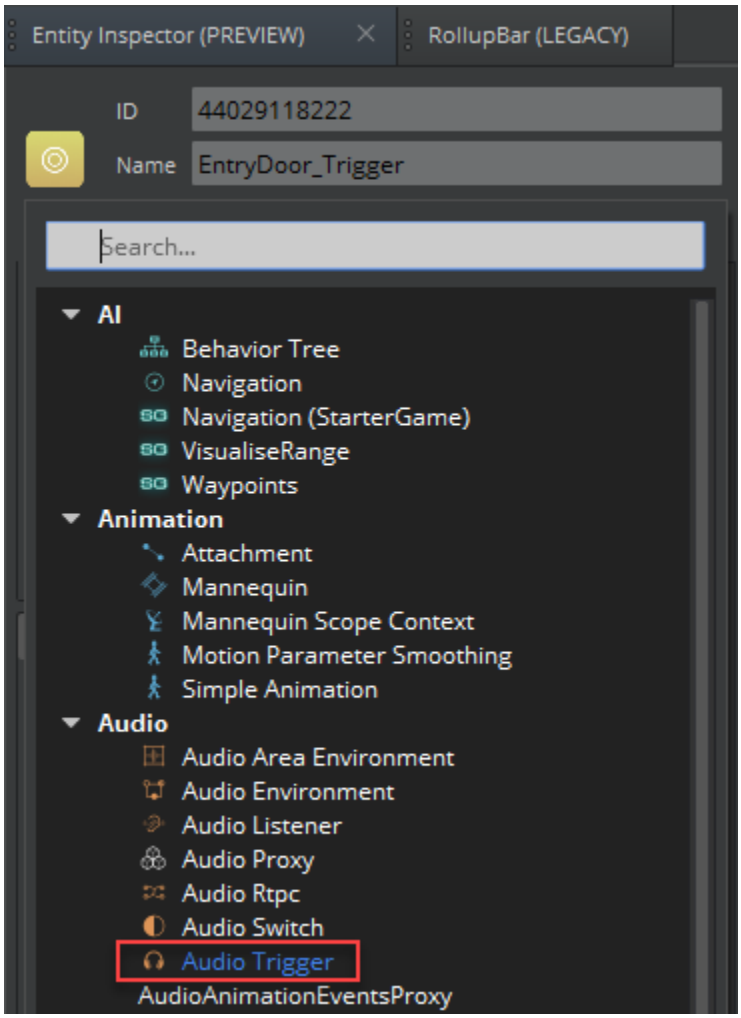
Now you add two audio components: **Audio Proxy** and **Audio Trigger**. The **Audio Proxy** component is required on any entity that uses audio. The **Audio Trigger** component triggers the audio.

To add the audio components

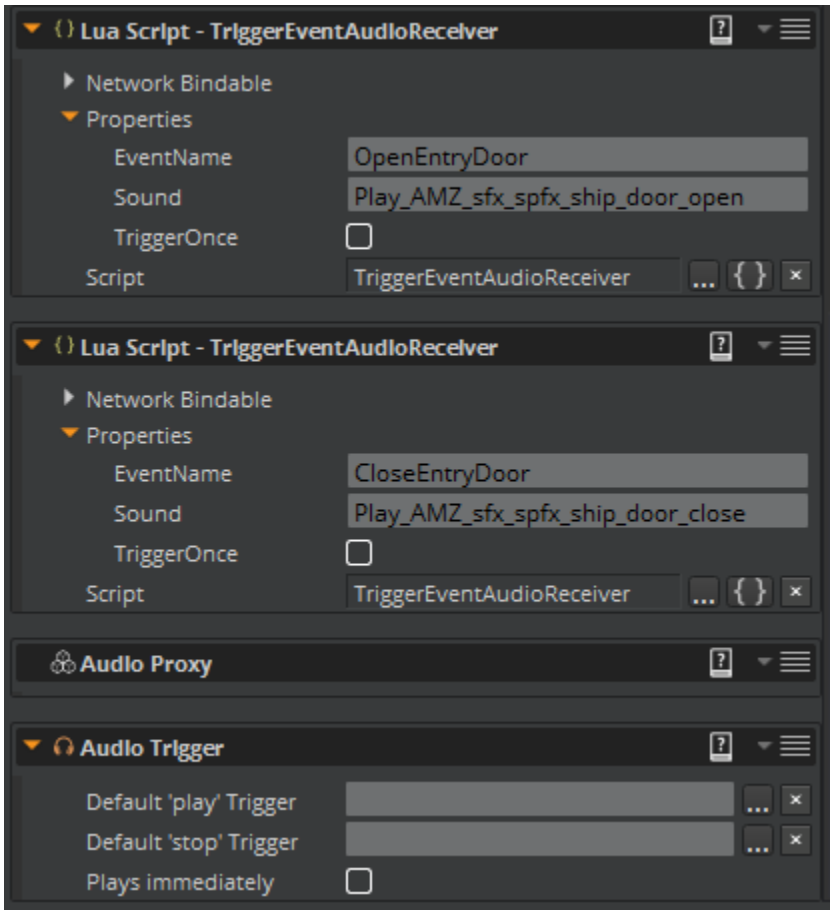
1. In the **Entity Outliner**, select the **EntryDoor_Trigger** entity.
2. In the **Entity Inspector**, click **Add Component**. Under **Audio**, select **Audio Proxy**.



3. Click **Add Component** again. Under **Audio**, select **Audio Trigger**.



Your two **Lua Script** components and your two audio components should look like the following.



4. Press **Ctrl+S** to save your level.
5. Press **Ctrl+G** to play your level. When you approach the door, you should hear the door opening and closing in concert with those actions.

   Press **Esc** to quit.

Next: Adding the Exit Door (p. 118)

Adding the Exit Door

Setting up the exit door uses a procedure that is similar to setting up the entry door. To add the exit door, you do the following:

- Duplicate and rename the door entity (p. 119) – Create a copy of the existing entry door and rename it.
- Modify the conditions that open the exit door (p. 120) – Remove the trigger area and prepare the door to open only after the player collects a hidden pickup item.
- Set up a new trigger area (p. 123) – Add a new trigger area that closes the door after the player collects the pickup item and exits the maze.

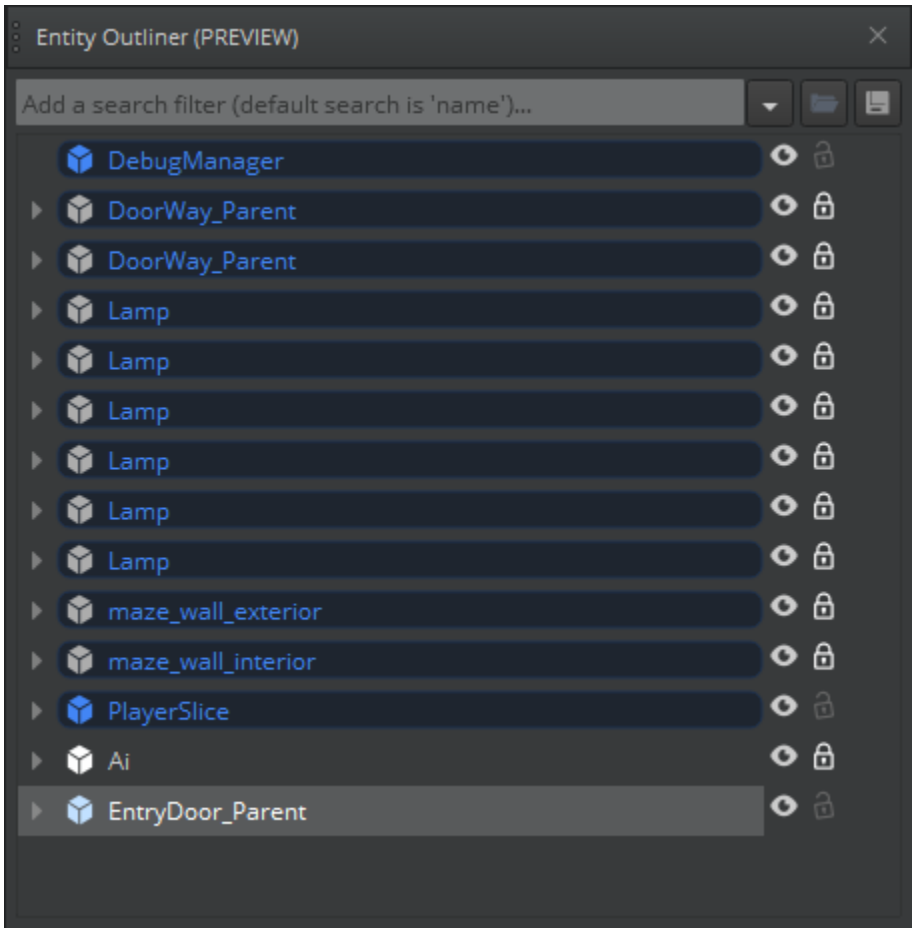
Next: Duplicating and Renaming the Door Entity (p. 119)

Duplicating and Renaming the Door Entity

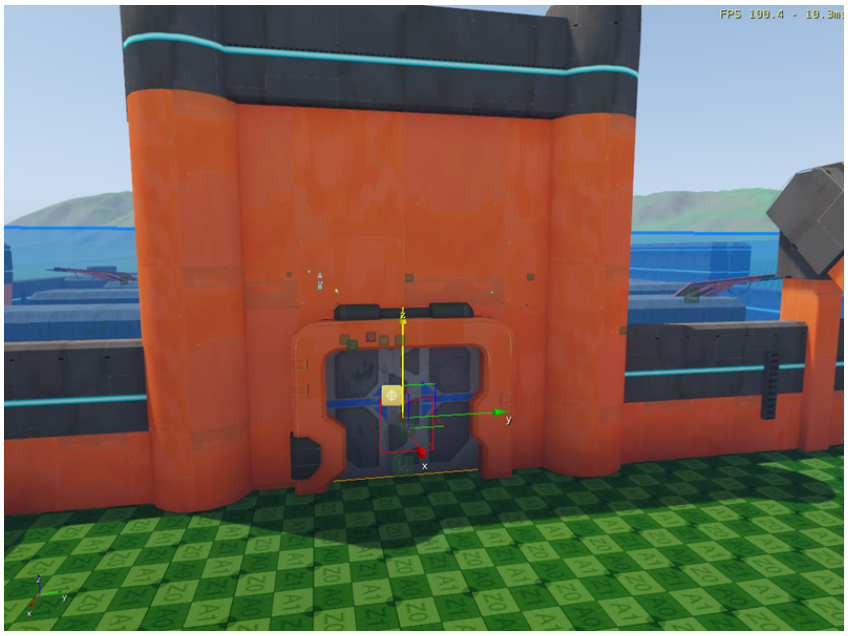
Because you have already performed the tasks of setting up the entry door, there is no need to do it all again from the beginning to create your exit door. Instead, you'll duplicate the existing entry door and then modify its name and properties.

To duplicate and rename the door entity

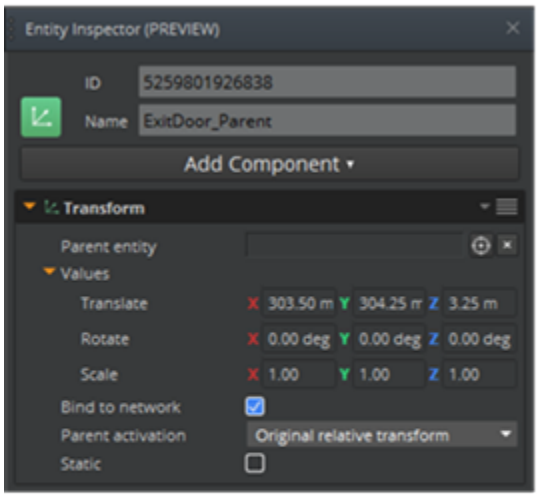
1. In the **Entity Outliner**, select **EntryDoor_Parent**. To duplicate it, press **Ctrl+D**.

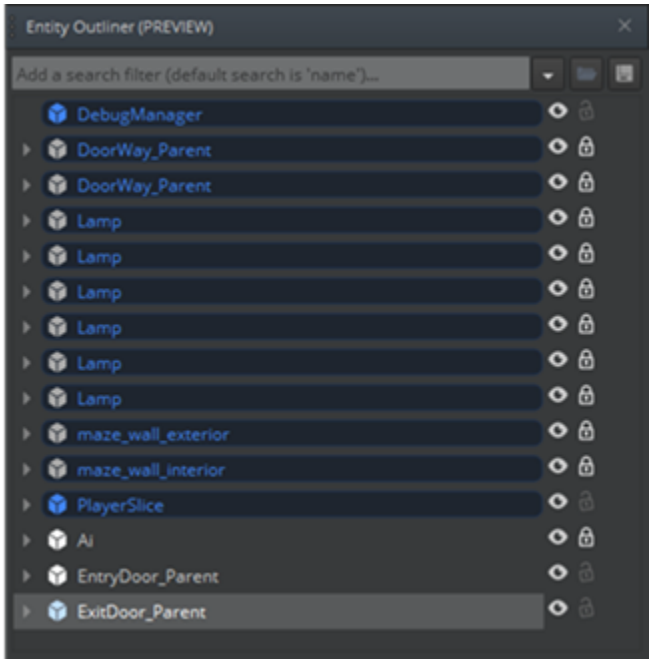


2. Use the **Move** tool to move the duplicated door entity to the other side of the maze. Align the door in the opening of the exit doorway.



3. In the **Entity Outliner**, select the duplicated entity. In the **Entity Inspector**, rename it **`ExitDoor_Parent`**.





4. Press **Ctrl+S** to save your level.

Next: Adding Lua Scripts to Exit Door (p. 120)

Modifying Conditions to Open Exit Door

To exit the maze, you'll set up a different trigger condition to open the exit door. Unlike with the entry door, the door should not open when the player approaches it. The door should open only after the player has collected the hidden pickup item.

In this tutorial, you'll remove the trigger volume that opens the door when the player approaches. You'll aslo prepare your exit door to open only after the player has collected a hidden pickup within the maze. You'll configure your pickup item and link it to this door in the section called "Adding a Pickup Item" (p. 127).

To modify the exit door

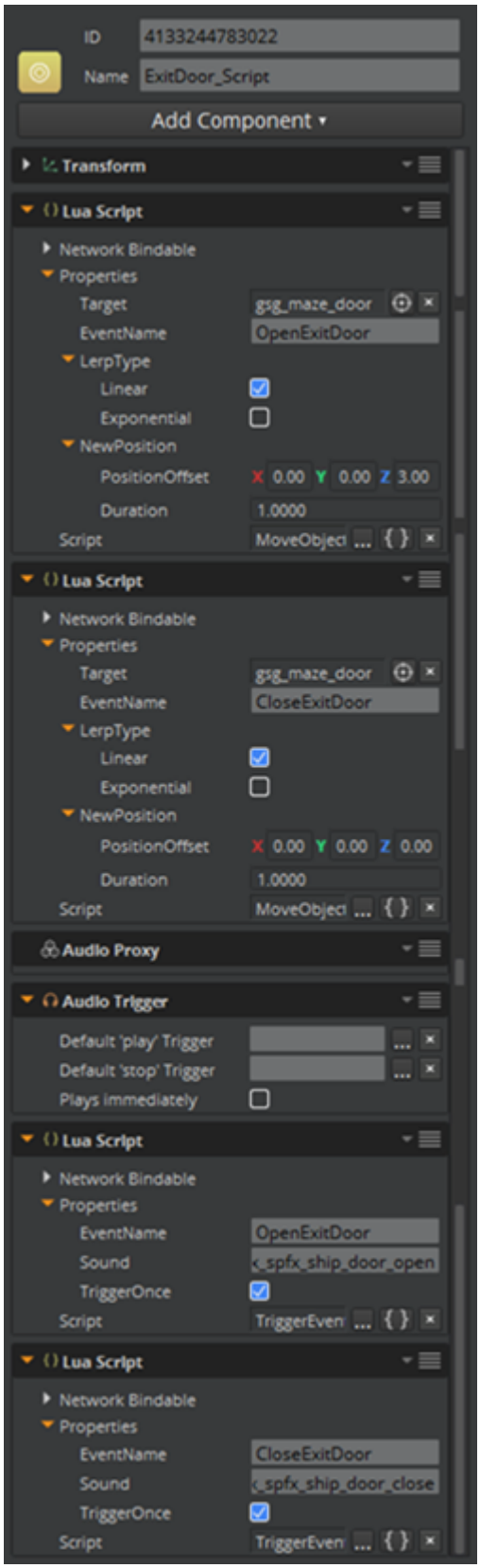
1. In the **Entity Outliner**, expand the **ExitDoor_Parent** group.

   Select the **EntryDoor_Trigger** entity and, in the **Entity Inspector**, rename it to **`ExitDoor_Script`**.
2. In the **Entity Outliner**, select **ExitDoor_Script**.

   Delete the following components. To do this, right-click the component header and click **Remove component**.

   - **Trigger Area**
   - **Box Shape**
   - **Lua Script** (the one with the **Script** set as `TriggerEvent`)
3. Because you copied the entry door, all the script names in the **Entity Inspector** refer to the entry door. You edit the script names to refer to the exit door.

   In the **Entity Inspector**, change all the **EventName** settings to reflect "exit" door rather than "entry" door. Your finished modifications should look similar to the following image.


ID
4133244783022
Name
ExitDoor_Script
Add Component
Transform
Lua Script
Network Bindable
Properties
Target
gsg_maze_door
EventName
OpenExitDoor
LerpType
Linear
Exponential
NewPosition
PositionOffset
X 0.00 Y 0.00 Z 3.00
Duration
1.0000
Script
MoveObject
Lua Script
Network Bindable
Properties
Target
gsg_maze_door
EventName
CloseExitDoor
LerpType
Linear
Exponential
NewPosition
PositionOffset
X 0.00 Y 0.00 Z 0.00
Duration
1.0000
Script
MoveObject
Audio Proxy
Audio Trigger
Default 'play' Trigger
Default 'stop' Trigger
Plays immediately
Lua Script
Network Bindable
Properties
EventName
OpenExitDoor
Sound
spfx_ship_door_open
TriggerOnce
Script
TriggerEven
Lua Script
Network Bindable
Properties
EventName
CloseExitDoor
Sound
spfx_ship_door_close
TriggerOnce
Script
TriggerEven

Next: Creating a New Trigger Area (p. 123)

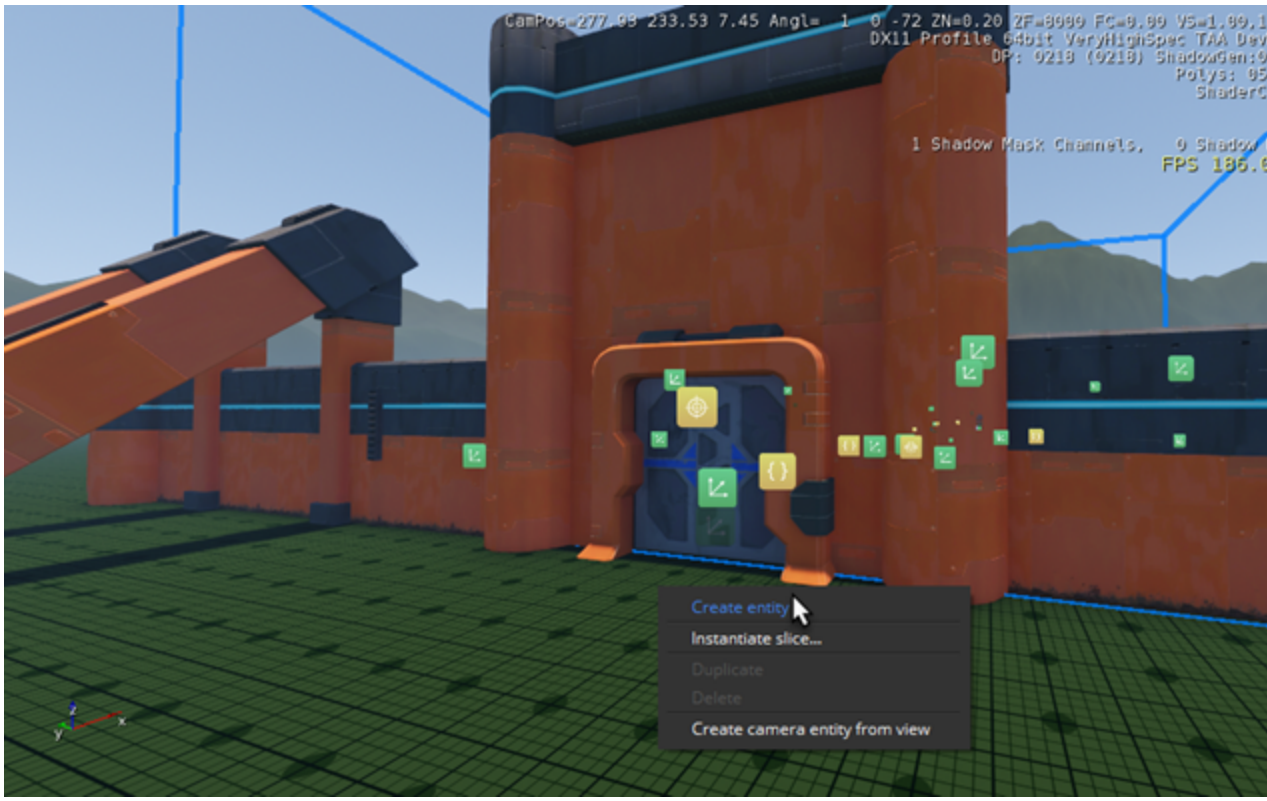
Creating a New Trigger Area

In the previous tutorial, you removed your exit door's ability to open upon approach. However, you still want the exit door to close behind the player after exiting.

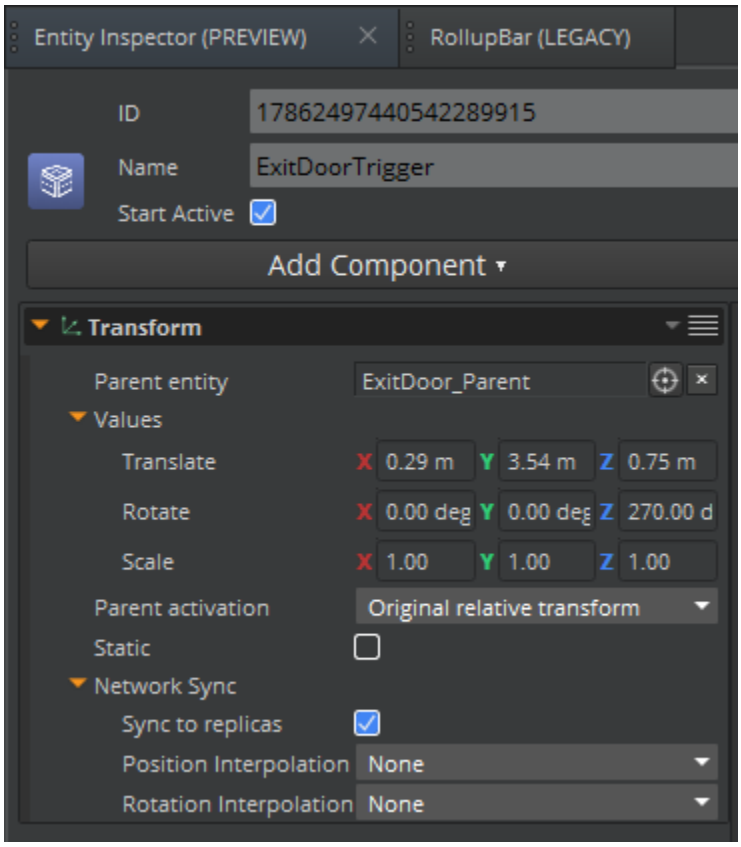
To do this, you'll create a new entity to act as the trigger area and place the trigger area on the outside edge of the maze exit door so that it triggers to close after the player exits.

To create a new trigger area for the exit door

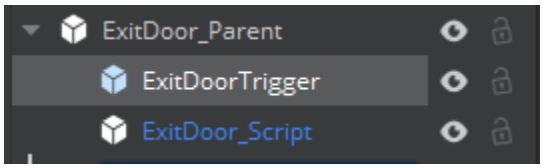
1. In the viewport, just outside of the exit door, create a new entity. To do this, right-click and choose **Create Entity**.



   In the **Entity Inspector**, name it `ExitDoorTrigger`.



2. In **Entity Outliner**, drag **ExitDoorTrigger** into **ExitDoor_Parent**.

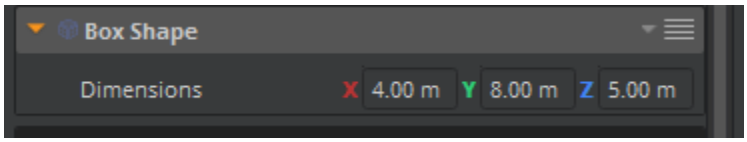


3. Make sure **ExitDoorTrigger** is still selected in **Entity Outliner**.

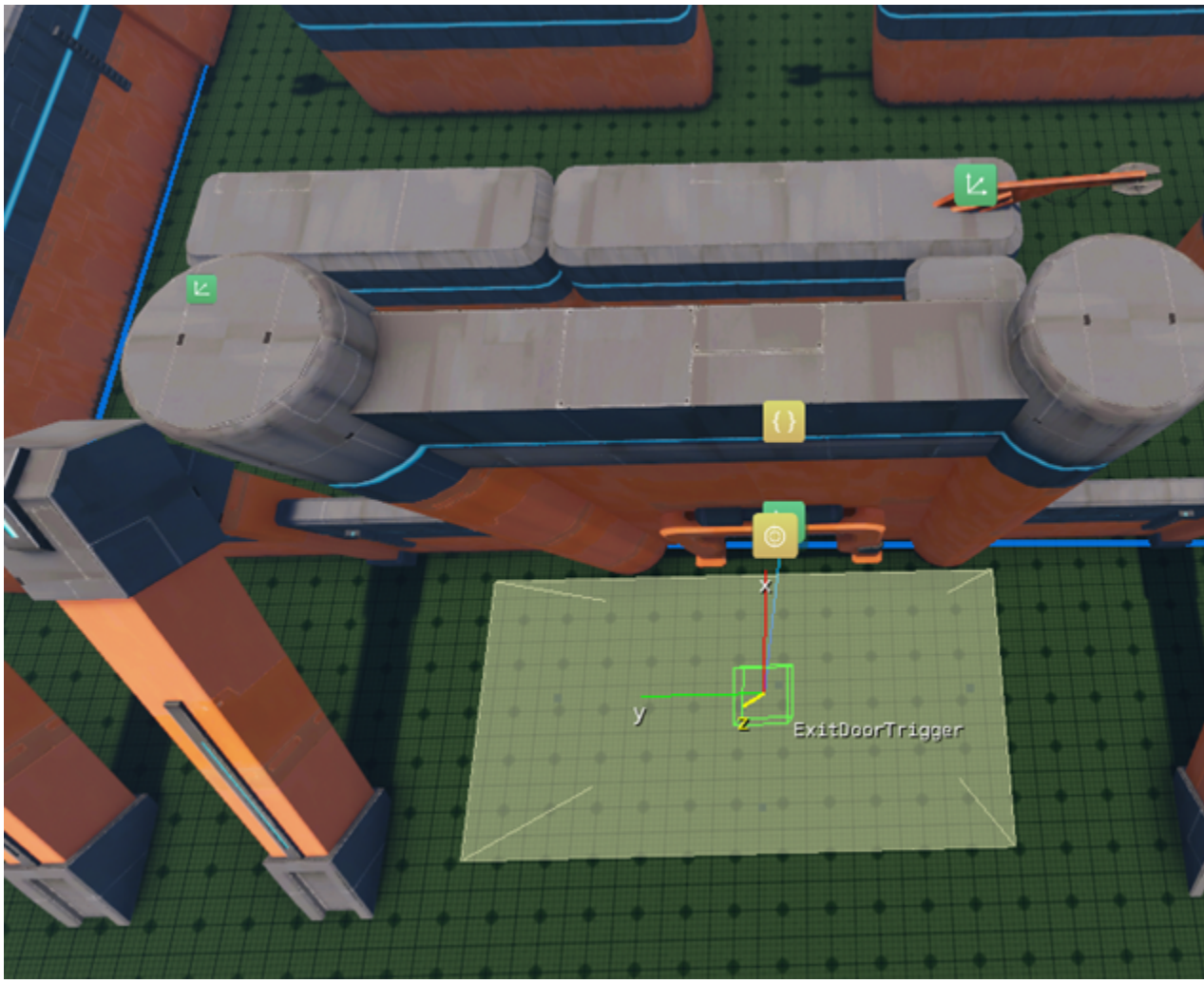
   In the **Entity Inspector**, add the following components:

   - **Box Shape**
   - **Trigger Area**
   - **Lua Script**

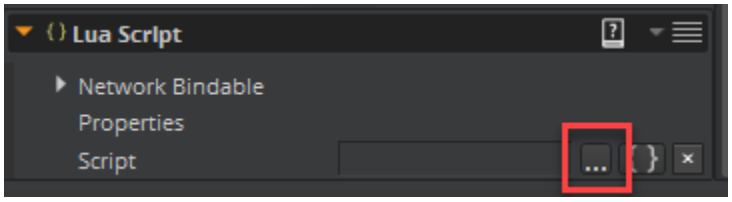
4. Modify the values of the **Box Shape** component's dimensions to **X**: `4.00`, **Y**: `8.00`, **Z**: `5.00`.



5. Use the **Move** tool to position the trigger area so that it is approximately two units outside of the door's exit side. Your trigger area should look similar to the following image.

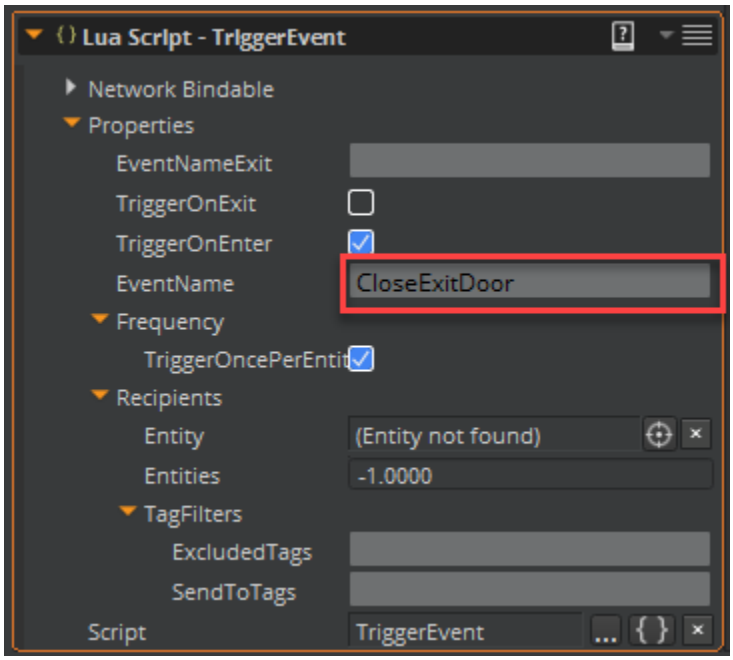


6. In the **Entity Inspector**, in the **Lua Script** component, next to **Script**, click the browse ('**...**') button.

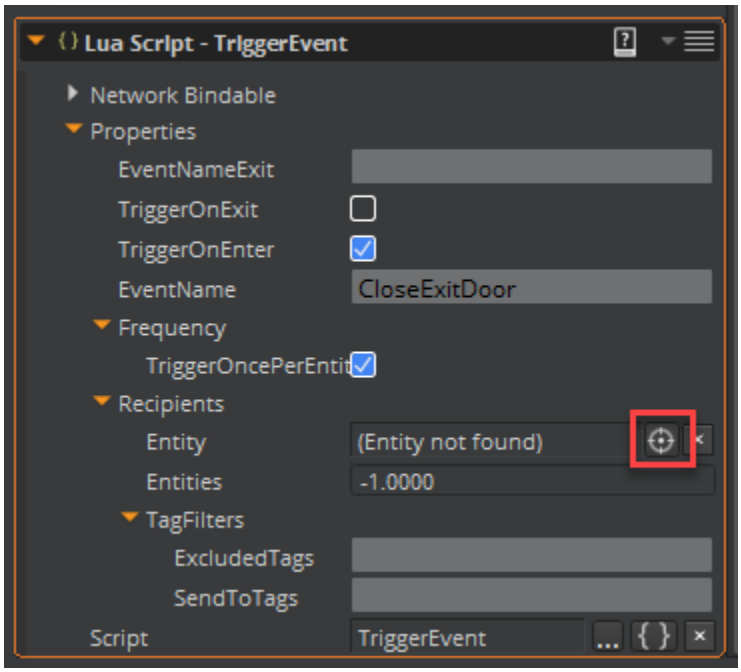


Navigate to `Scripts\Triggers`. Select `TriggerEvent.lua`. Click **OK**.

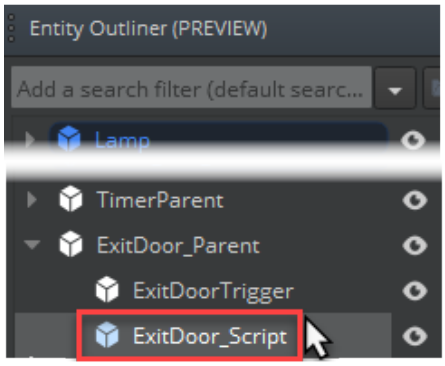
7. In the **EventName** box, type **`CloseExitDoor`**.

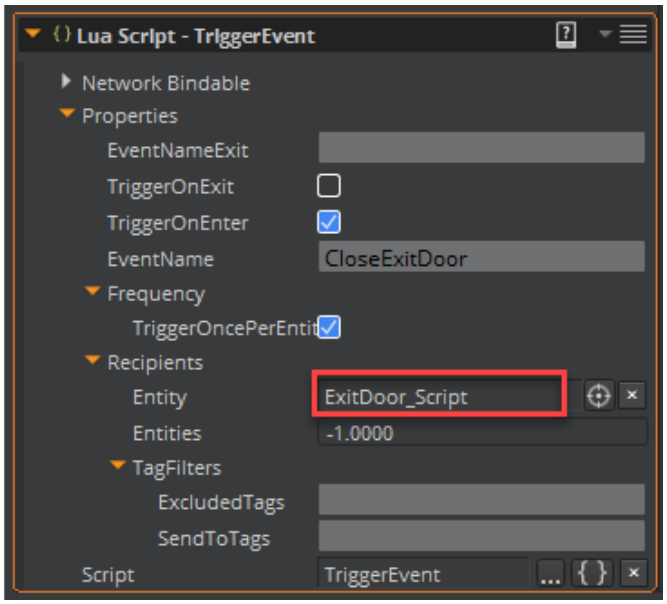


8. Under **Recipients**, next to **Entity**, click the picker icon .



In the **Entity Outliner**, select **ExitDoor_Script**. This places **ExitDoor_Script** in the **Entity** box.





This points the new trigger area that you created back to the script that opens or closes the door.

9. Press **Ctrl+S** to save your level.

Next: Adding a Pickup Item (p. 127)

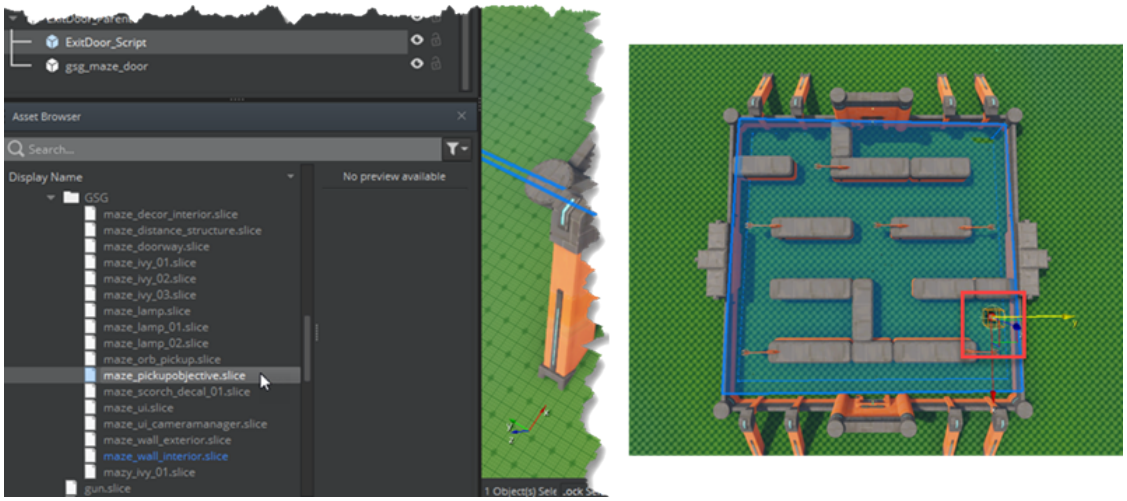
Adding a Pickup Item

Now you add an item in the maze that the player must pick up before exiting the maze. The pickup item is a prebuilt slice that you access through the **Asset Browser**. After you place the pickup item slice, you customize the **Lua Script** component.

To place a pickup item

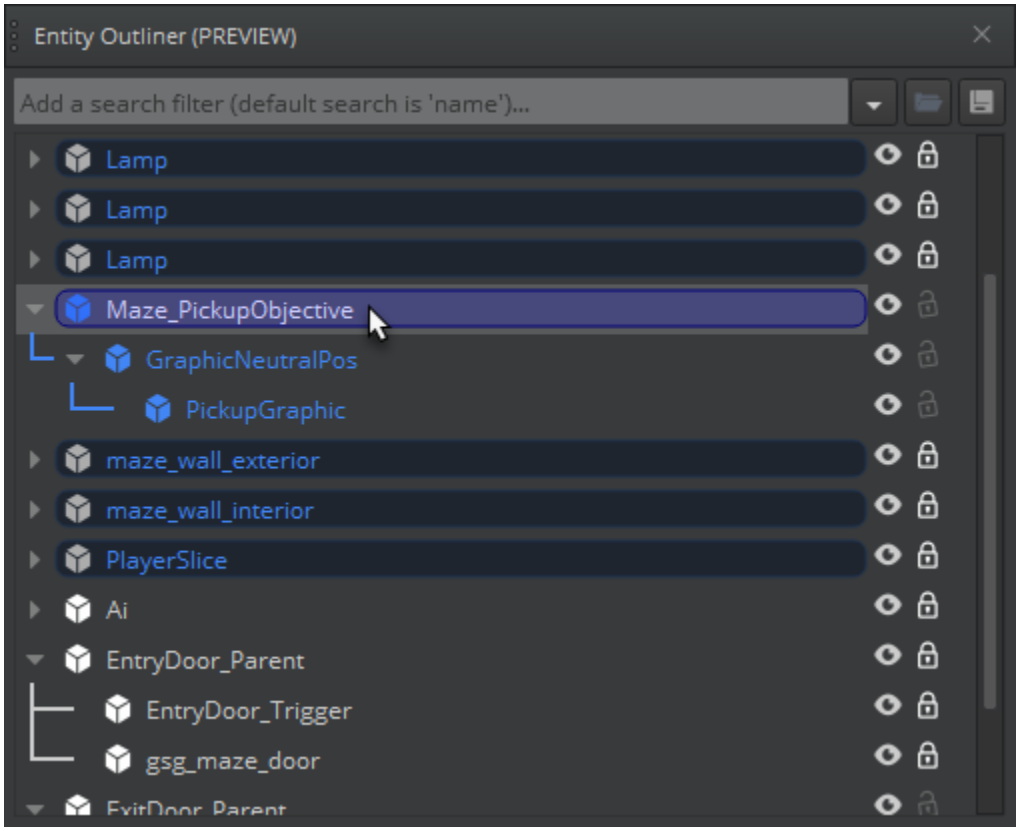
1. In the **Asset Browser**, navigate to `Game\Slices\GSG`.

   Select `maze_pickupobjective.slice` and drag it into your viewport, into the far corner of the maze.

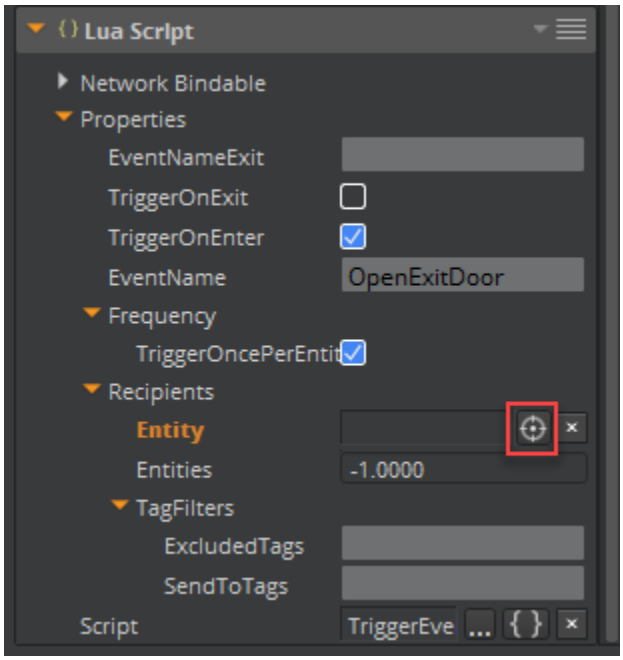


2. The `maze_pickupobjective.slice` has a number of components and several entities. You modify the properties on one of these components.

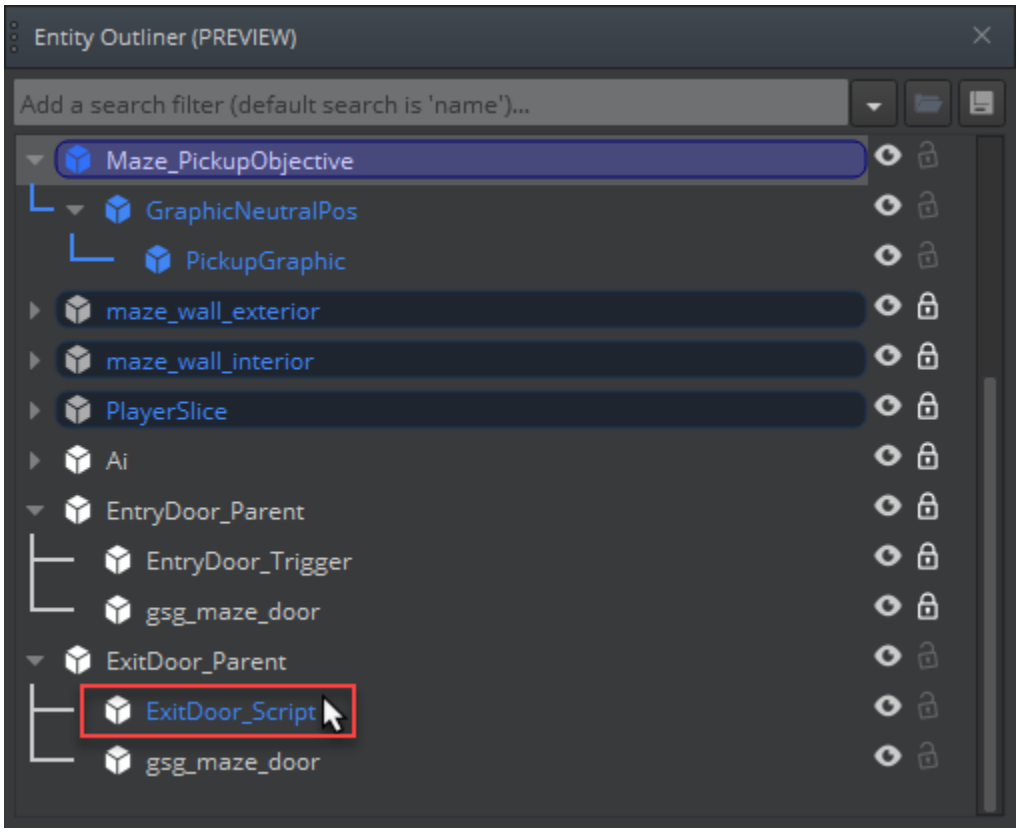
   In the **Entity Outliner**, select the **Maze_PickupObjective** slice.



3. In the **Entity Inspector**, in the bottommost **Lua Script** component, under **Recipients**, next to **Entity**, click the picker  .



4. In the **Entity Outliner**, under **ExitDoor_Parent**, select **Exitdoor_Script**.



You now have a pickup that, when retrieved, allows the player to exit the maze.

5. Press **Ctrl+S** to save your level.

Next: Adding a Timer (p. 130)

Adding a Timer

To add more challenge to the game experience, you'll add a timer that starts when the player enters the maze and stops upon exit. To do this, you'll set up a Lua script with supporting entity components that trigger when the player enters the maze and stops upon exiting. Timing how long it takes to complete a particular task is an element of gameplay that adds tension and excitement.

To set up your timer, you do the following:

- Create a parent entity (**TimerParent**) and add a child entity (**TimerStart**).
- To **TimerStart**, add and configure the following components:
  - **Box Shape**
  - **Trigger Area**
  - **UI Canvas Asset Ref**
- To **TimerStart**, add and configure three **Lua Script** components.
- Adjust the position of the **TimerStart** entity so that it starts when the player crosses the maze threshold.

You complete a similar set of steps to set up the stop trigger, which stops the timer when your player exits the maze.

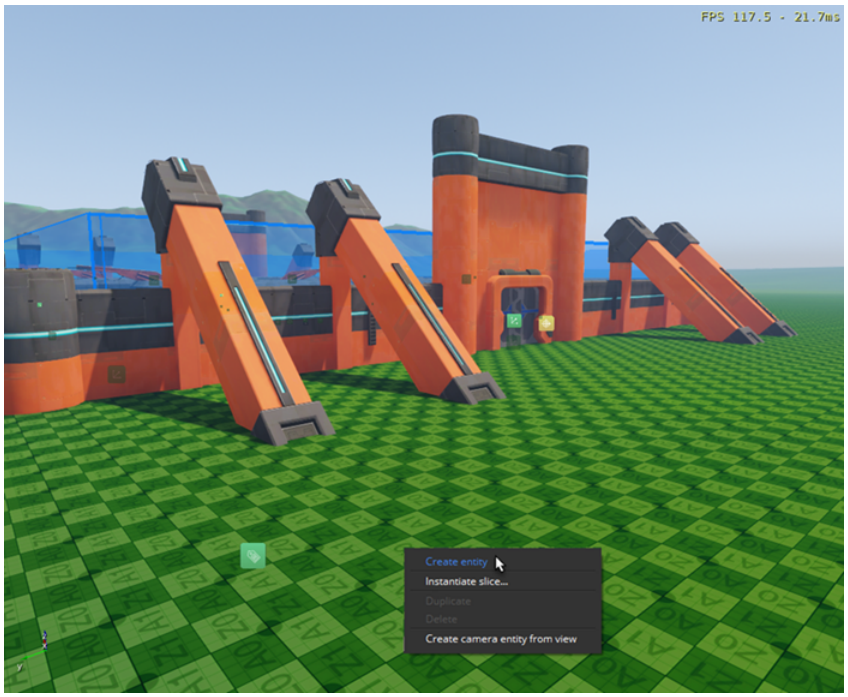
Next: Creating the Timer Parent and Child Entities (p. 130)

Creating the Timer Parent and Child Entities

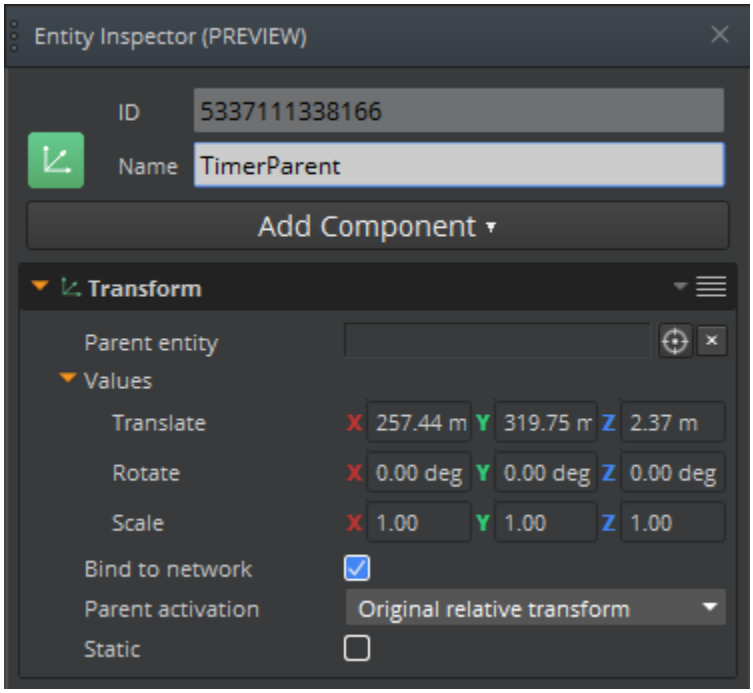
You create a parent entity and name it **TimerParent**. Inside **TimerParent**, you create a child entity and name it **TimerStart**.

To create the timer parent and child entities

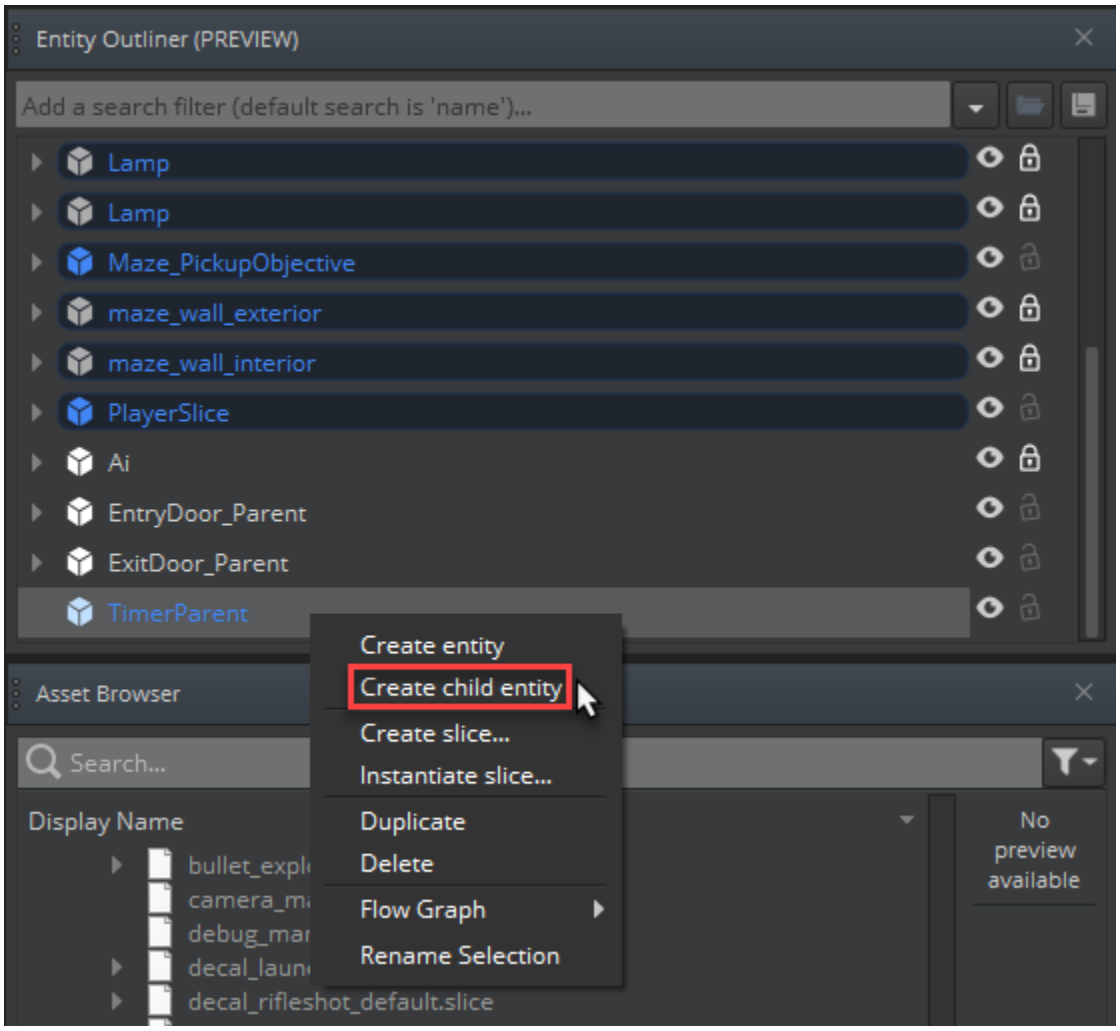
1. In the viewport, right-click just outside the maze and choose **Create entity**.



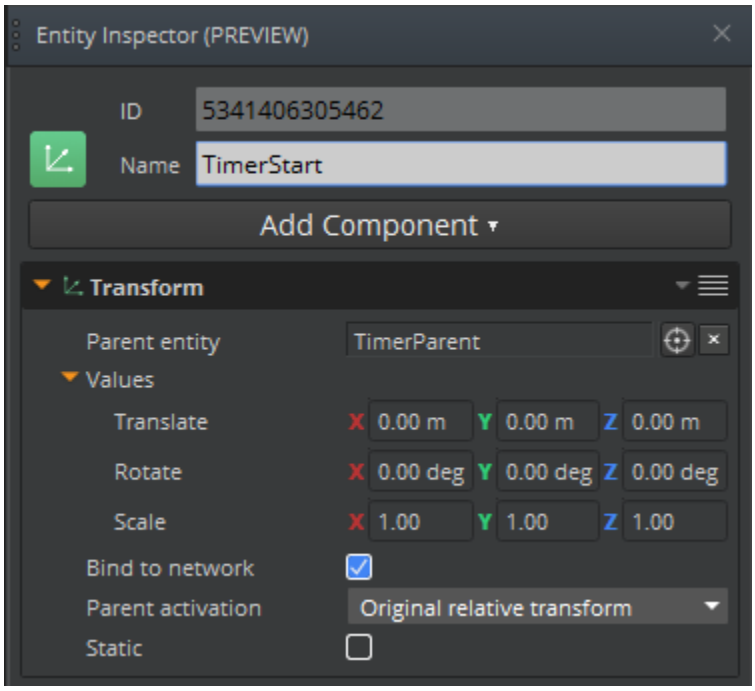
2. In the **Entity Outliner**, select the new entity and name it **`TimerParent`**.



3. Right-click **TimerParent** and choose **Create child entity**.



4. In the **Entity Inspector**, rename the new child entity **TimerStart**.



Next, you add components to the child entity **TimerStart**.

Next: Adding Components to TimerStart (p. 133)

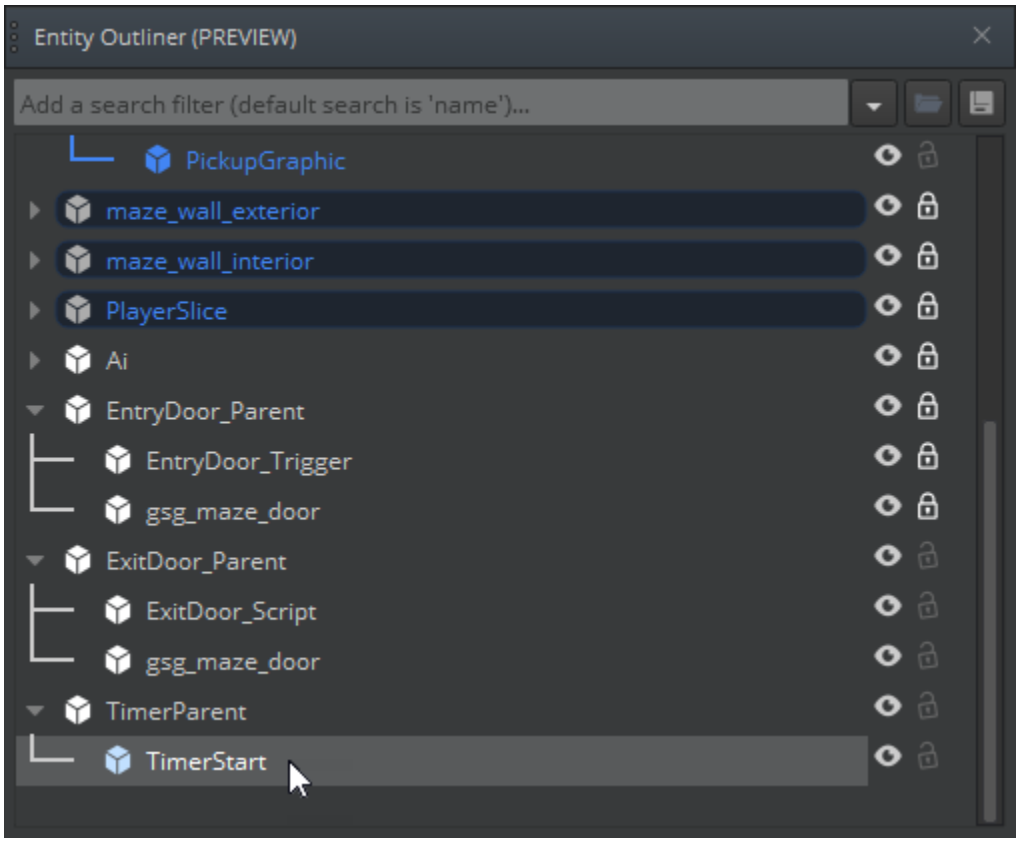
Adding Components to TimerStart

You add the following three components to **TimerStart** and configure them:

- **Box Shape** – Gives the trigger area its shape and volume
- **Trigger Area** – Works in tandem with the box shape to create a trigger
- **UI Canvas Asset Ref** – Associates a prebuilt UI canvas with the **TimerStart** entity to display the timer

To add and configure the box shape component

1. In the **Entity Outliner**, select the **TimerStart** entity.

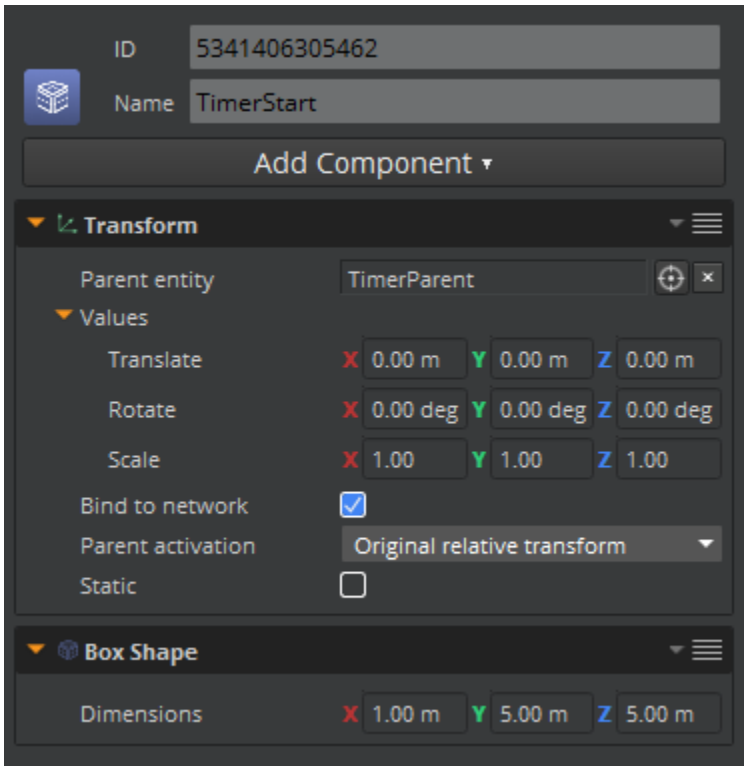


2. In the **Entity Inspector**, click **Add Component**.

   Select the **Box Shape** component.

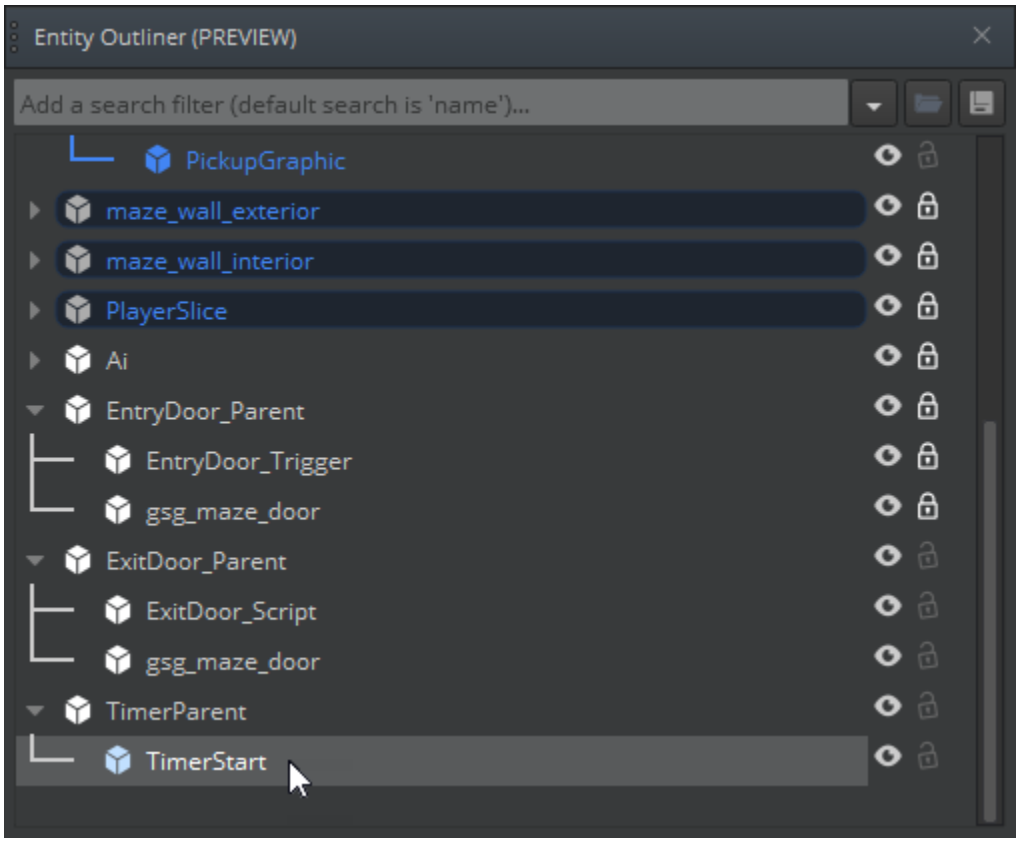
3. In the **Box Shape** component properties, set the **Dimensions** as follows:

   - **X** : `1.00`
   - **Y** : `5.00`
   - **Z** : `5.00`



To add and configure the trigger area component

1. In the **Entity Outliner**, select the **TimerStart** entity.

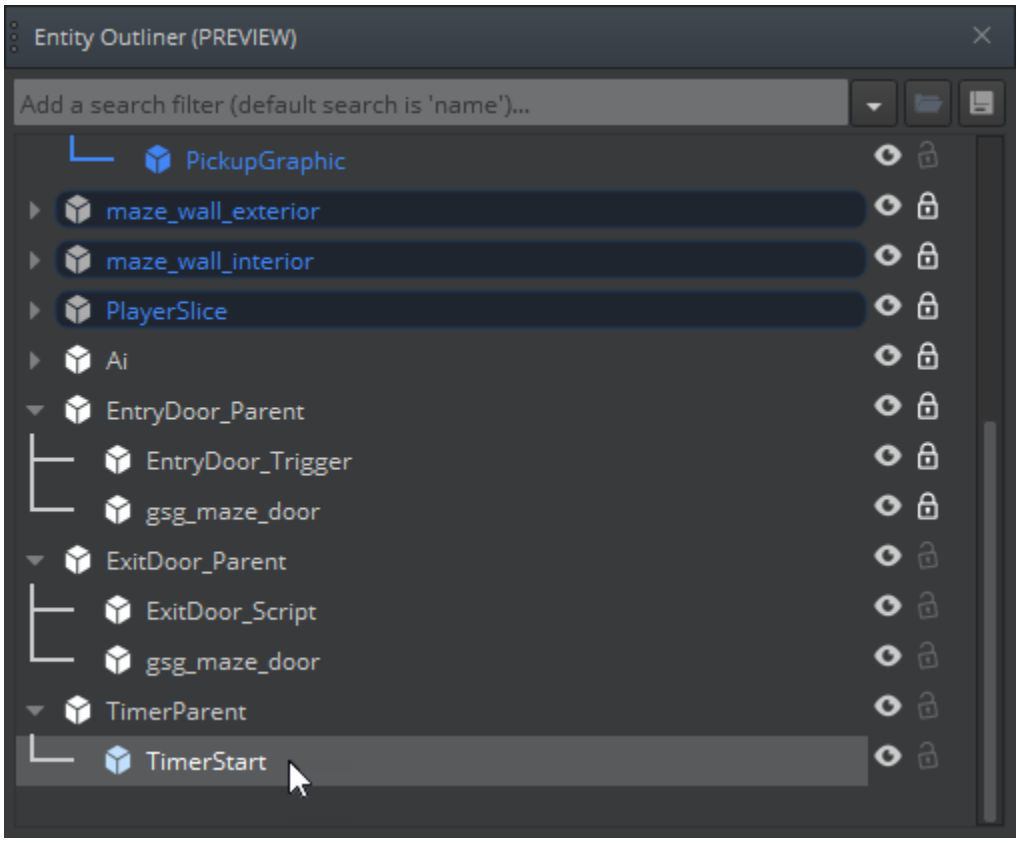


2. In the **Entity Inspector**, click **Add Component**.

   Select the **Trigger Area** component.

To add and configure the UI canvas asset ref component

1. In the **Entity Outliner**, select the **TimerStart** entity.

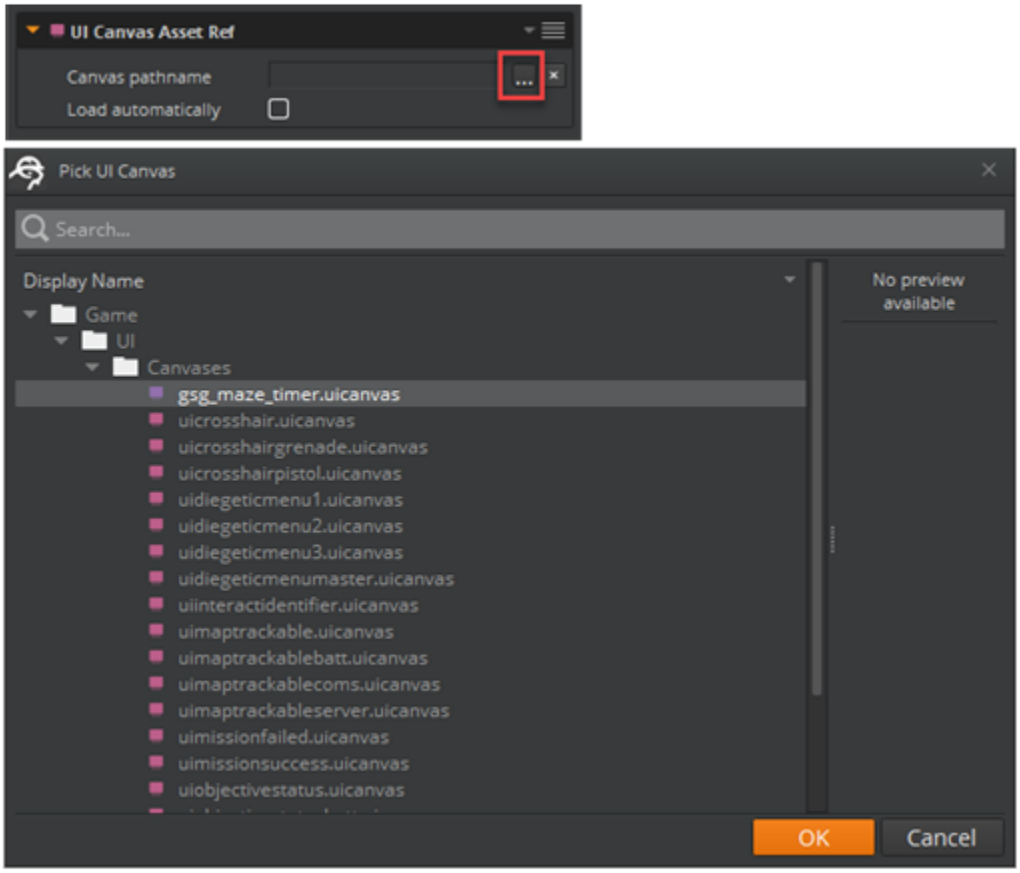


2. In the **Entity Inspector**, click **Add Component**.

   Select the **UI Canvas Asset Ref** component.

3. In the **UI Canvas Asset Ref** properties, click **Browse (...)** and navigate to `Game\UI\Canvases\`.

   Select `gsg_maze_timer.uicanvas`. Click **OK**.



Next, you add three **Lua Script** components to the same entity, **TimerStart**.

Next: Adding Lua Script Components (p. 138)

Adding Lua Script Components

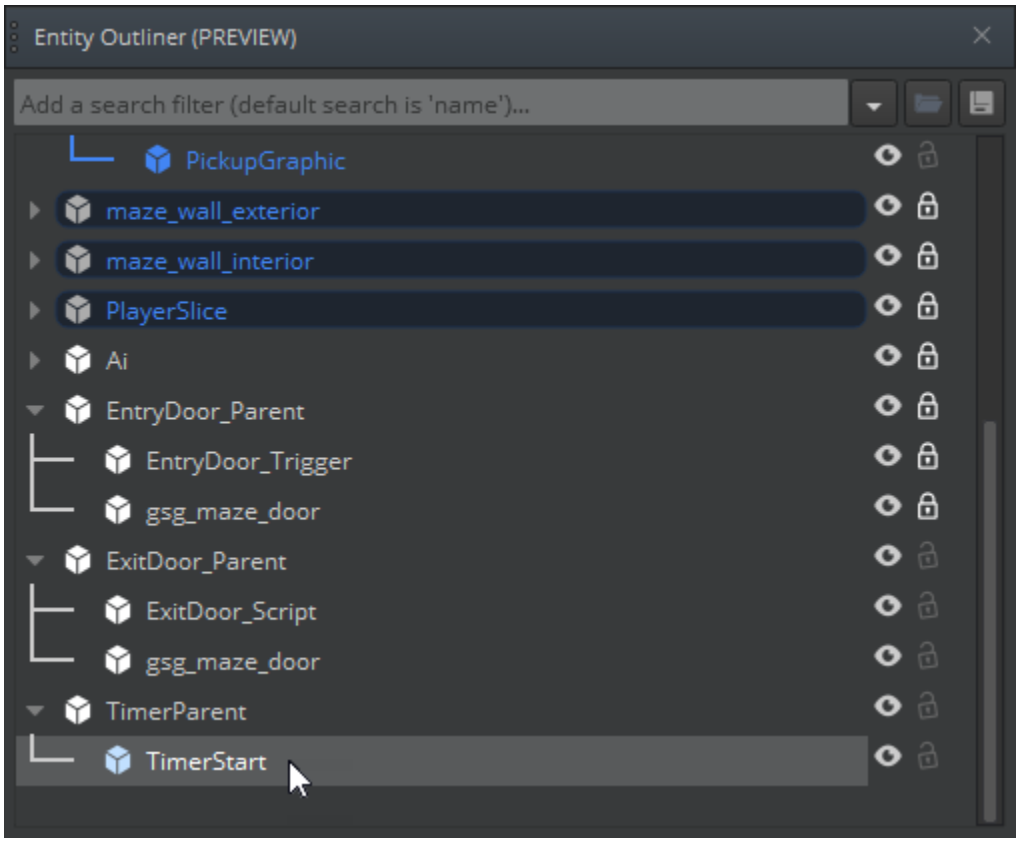
You add three more components, all **Lua Script** components, to the **TimerStart** entity. You then add the following scripts and make the following changes:

- Lua Script component #1:
  - **Script** – `Timer.lua`
  - **TimerElementName** – **`Timer`**
- Lua Script component #2:
  - **Script** – `TriggerEvent.lua`
  - **EventName** – **`TimerStart`**
- Lua Script component #3:
  - **Script** – `TriggerEvent.lua`
  - **EventName** – **`TimerStop`**

You then use the **Move** tool to adjust the **TimerStart** entity's position in the viewport so that it is just inside the maze entrance.

To add and configure Lua Script components

1. In the **Entity Outliner**, select the **TimerStart** entity.



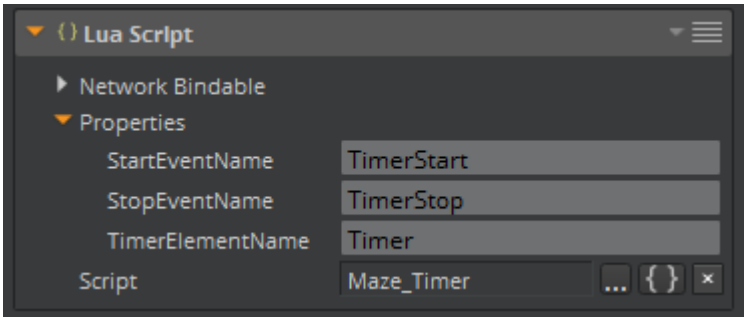
2. In the **Entity Inspector**, click **Add Component**.

   Select **Lua Script**.

   Repeat twice more so that you have a total of three **Lua Script** components on the **TimerStart** entity.

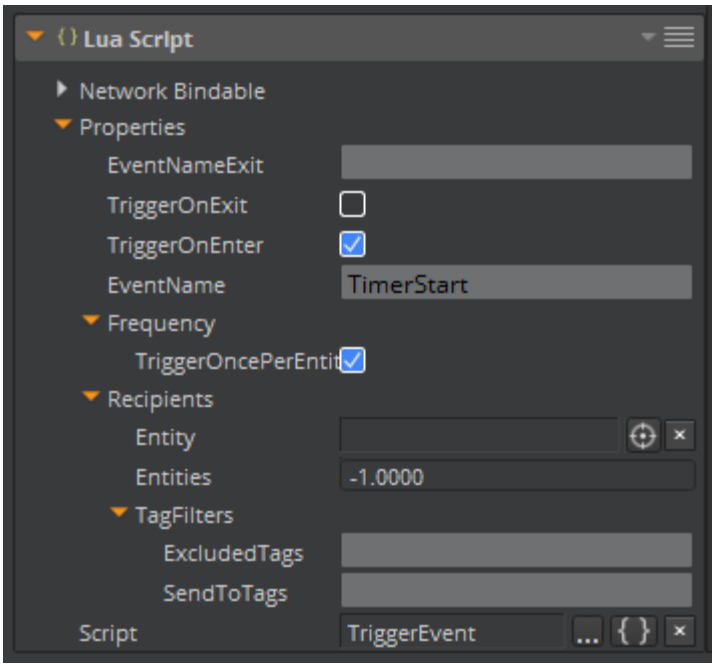
3. On the first **Lua Script** component, do the following:

   - Click **Browse (...)** next to the **Script** box. Navigate to `Game\Scripts\General` and select `Timer.lua`.
   - In the **TimerElementName** property, type **`Timer`**.



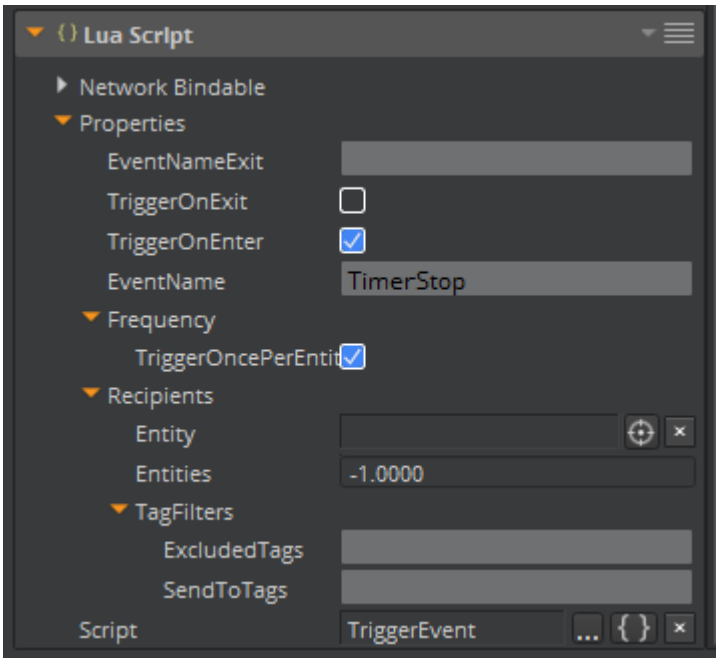
4. On the second **Lua Script** component:

- Click **Browse (...)** next to the **Script** box. Navigate to `Game\Scripts\Triggers` and select `TriggerEvent.lua`.
- In the **EventName** property, type **`TimerStart`**.

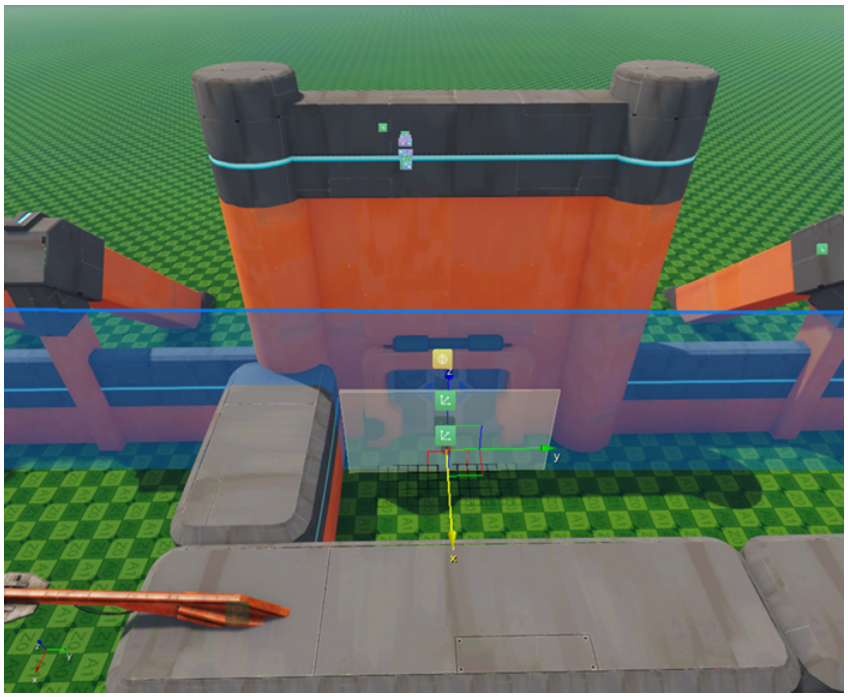


5. On the third **Lua Script** component:

- Click **Browse (...)** next to the **Script** box. Navigate to `Game\Scripts\Triggers` and select `TriggerEvent.lua`.
- In the **EventName** property, type **`TimerStop`**.



6. In the viewport, use the **Move** tool to adjust the position of the **TimerStart** entity so that it is just inside the maze entrance. That way, the timer starts when the player enters the maze.



7. Press **Ctrl+S** to save your level.

Next, you add and configure a similar set of entities and components at the maze's exit to trigger the timer to stop.

Next: Adding and Configuring the Timer's Stop Trigger (p. 141)

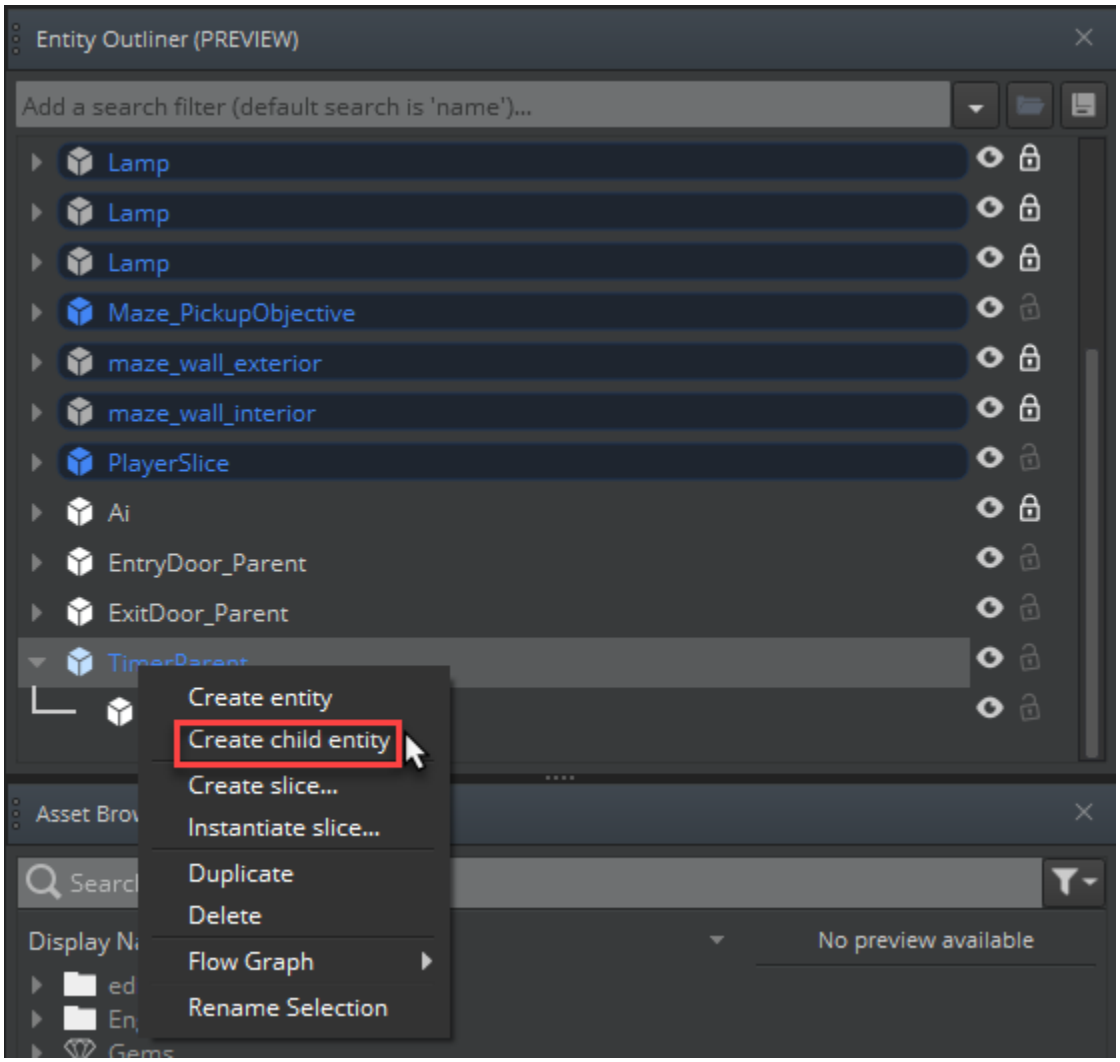
Adding and Configuring the Timer's Stop Trigger

To set up the timer's stop trigger, you do the following:

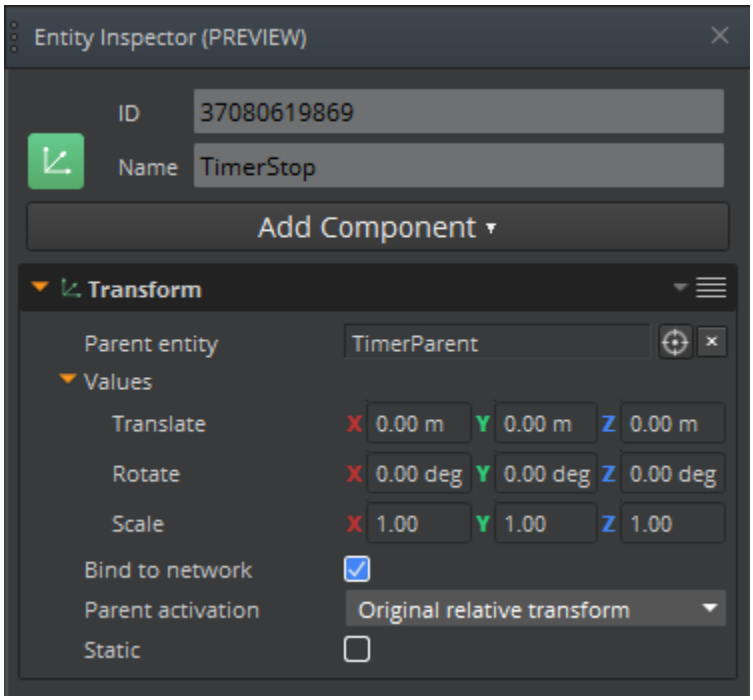
- Create a child entity under **TimerParent** and name it **TimerStop**.
- To **TimerStop**, add and configure a **Box Shape**, a **Trigger Area**, and a **Lua Script** component.
- Adjust the position of the **TimerStop** entity so that it is at the maze's exit.

To create the TimerStop child entity

1. In the **Entity Outliner**, right-click **TimerParent** and select **Create child entity**.



2. In the **Entity Inspector**, rename the new child entity **TimerStop**.

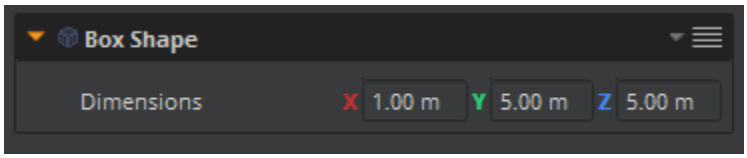


To add and configure the box shape component

1. In the **Entity Outliner**, select the **TimerStop** entity.
2. In the **Entity Inspector**, click **Add Component**.

   Select the **Box Shape** component.
3. In the **Box Shape** component properties, set the **Dimensions** as follows:

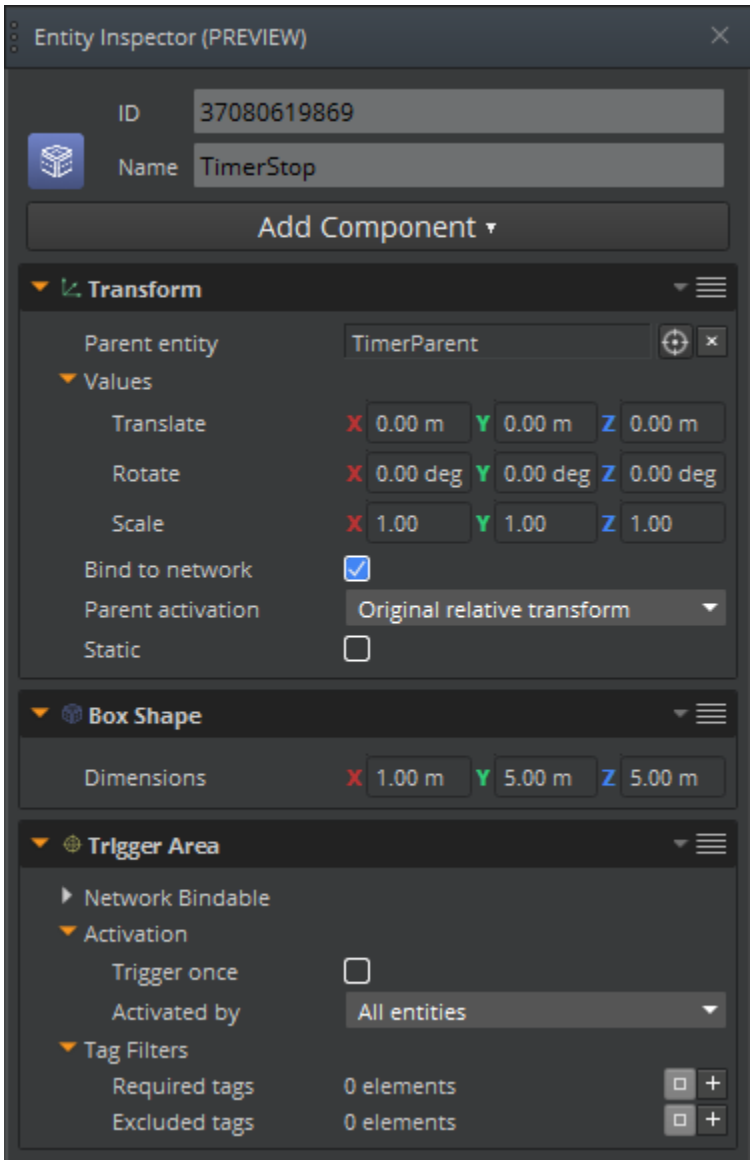
   - **X**: `1.00`
   - **Y**: `5.00`
   - **Z**: `5.00`



To add and configure the trigger area component

1. In the **Entity Outliner**, select the **TimerStop** entity.
2. In the **Entity Inspector**, click **Add Component**.

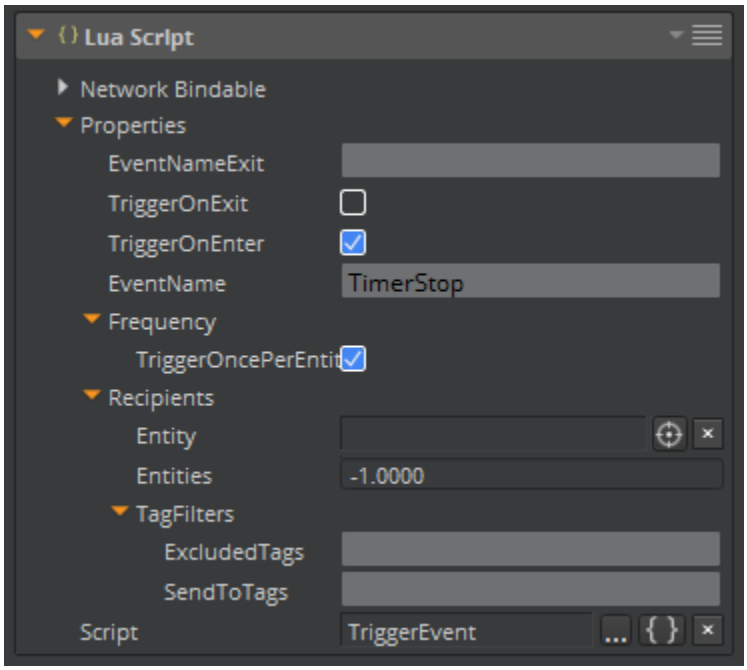
   Select the **Trigger Area** component.



To add and configure the Lua Script components

1. In the **Entity Outliner**, select the **TimerStop** entity.
2. In the **Entity Inspector**, click **Add Component**.

   Select **Lua Script**.
3. Click **Browse (...)** next to the **Script** box. Navigate to `Game\Scripts\Triggers` and select `TriggerEvent.lua`. Click **OK**.
4. In the **Lua Script** component's **EventName** property, type **`TimerStop`**.

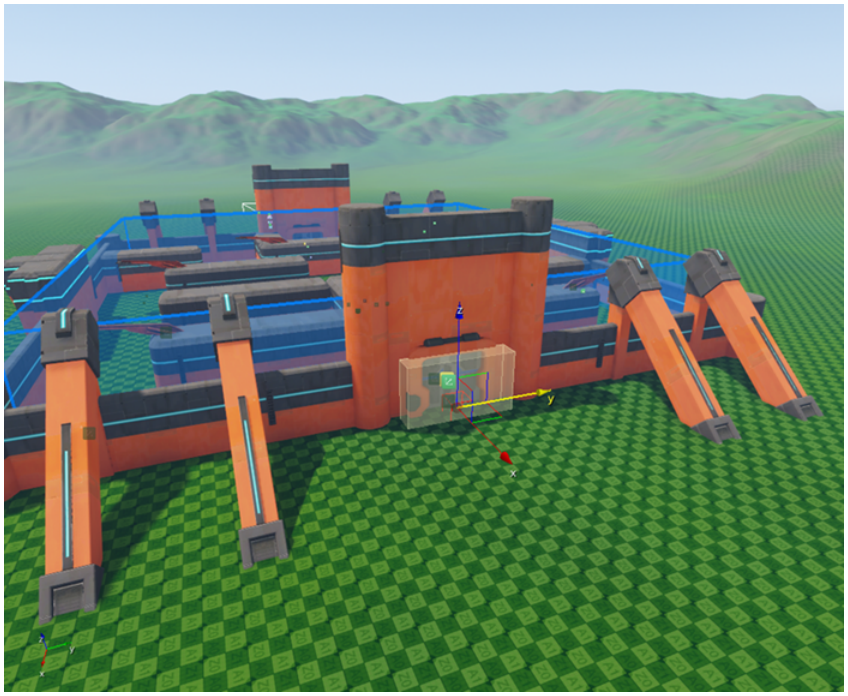


5. Under **Recipients**, next to the **Entity** box, click the picker icon .

   In the **Entity Outliner**, select the **TimerStart** entity.

To adjust the position of the timer's stop trigger

1. In the **Entity Outliner**, select the **TimerStop** entity.
2. In the viewport, use the **Move** tool to adjust the position of **TimerStop** so that it is at the maze's exit door.



3. Press **Ctrl+S** to save your level.
4. Press **Ctrl+G** to play your level.

   The timer should start when you enter the level. Collect the pickup item. The timer should stop when you exit the level.

   Press **Esc** to quit.

Next: 7: Applying Physics to Stacked Crates (p. 147)

7: Applying Physics to Stacked Crates

Lumberyard features two types of physics components: **Static Physics** and **Rigid Body Physics**. Earlier, you applied the **Static Physics** component to several stationary (immovable) objects. The **Rigid Body Physics** component is used for objects that can move.

You place crates and barrels in your level and then apply and configure the **Rigid Body Physics** component. With this component, you can customize the density of the crates and barrels. This affects how much they move and roll when struck by the player or weapon fire.

You then duplicate the crates and barrels and stack them up to build a wall to block the passageway in an area of your maze.

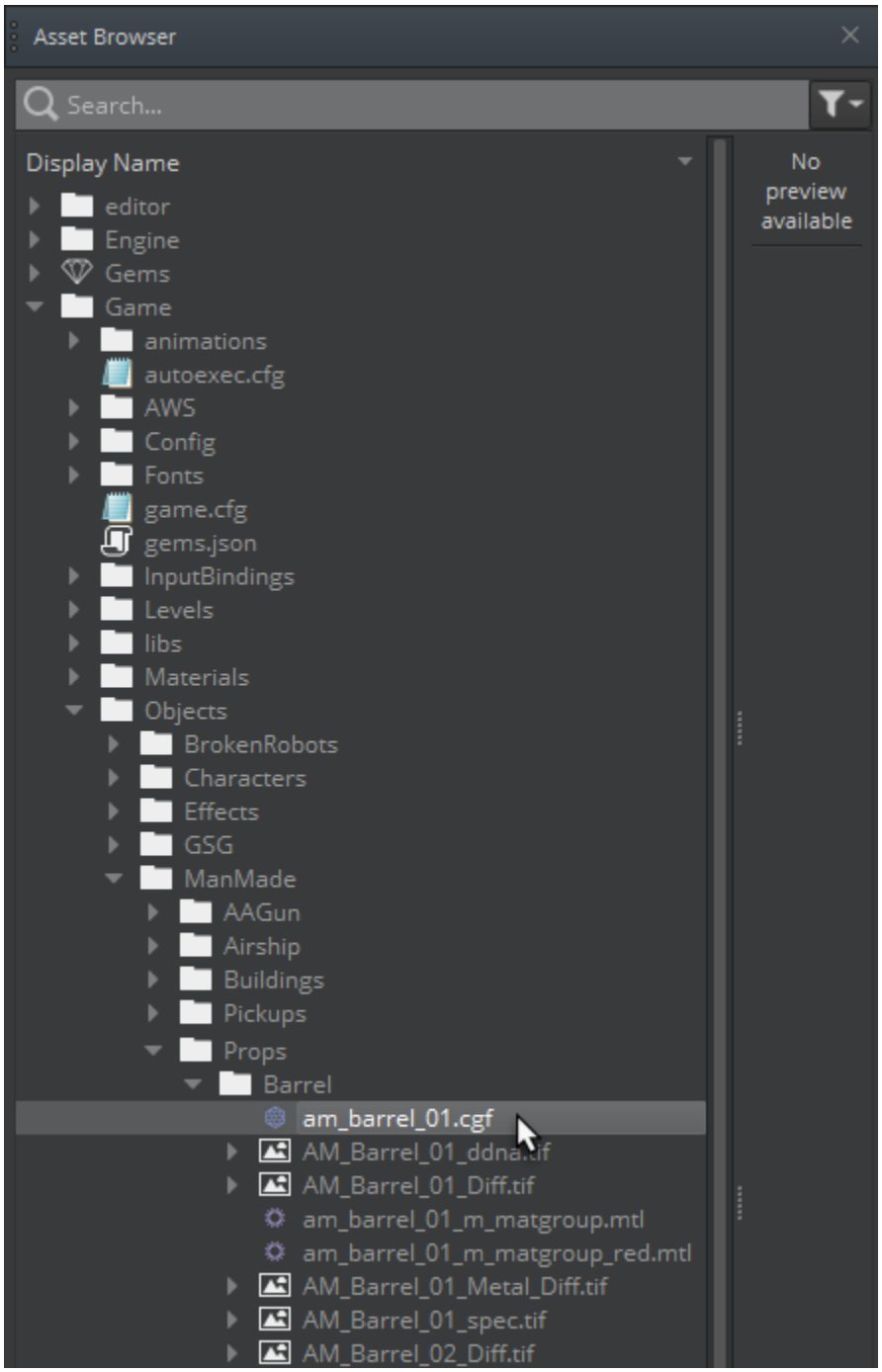
Next: Placing Crates and Configuring Physics (p. 147)

Placing Crates and Configuring Physics

You use the **Asset Browser** to find the barrel and crates. You drag them into your viewport and then add the **Mesh Collider** and **Rigid Body Physics** components to each, and then configure their density.

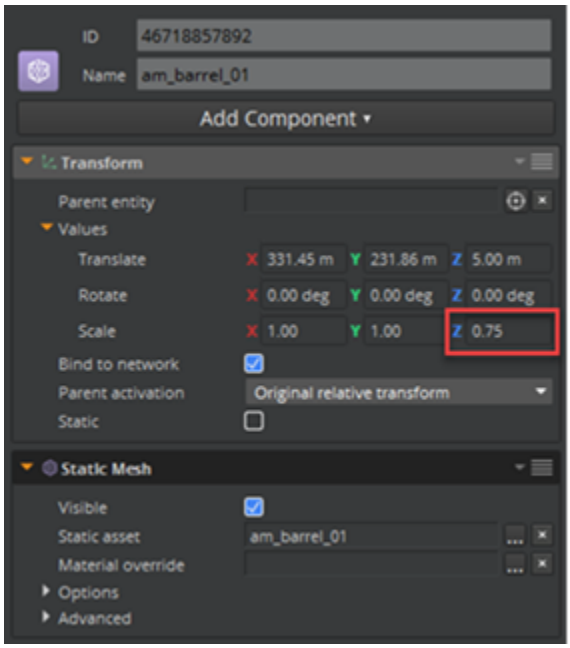
To place crates and barrels

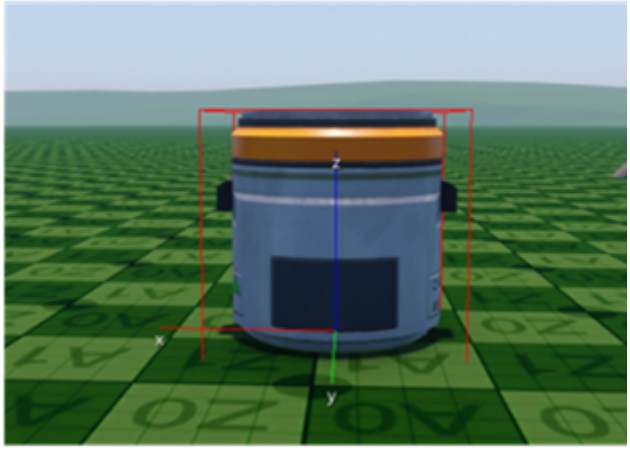
1. In the **Asset Browser**, navigate to `Game\Objects\ManMade\Props\Barrel`. Drag `am_barrel_01.cgf` into the viewport.



2. In the **Entity Outliner**, select the barrel that you just placed.

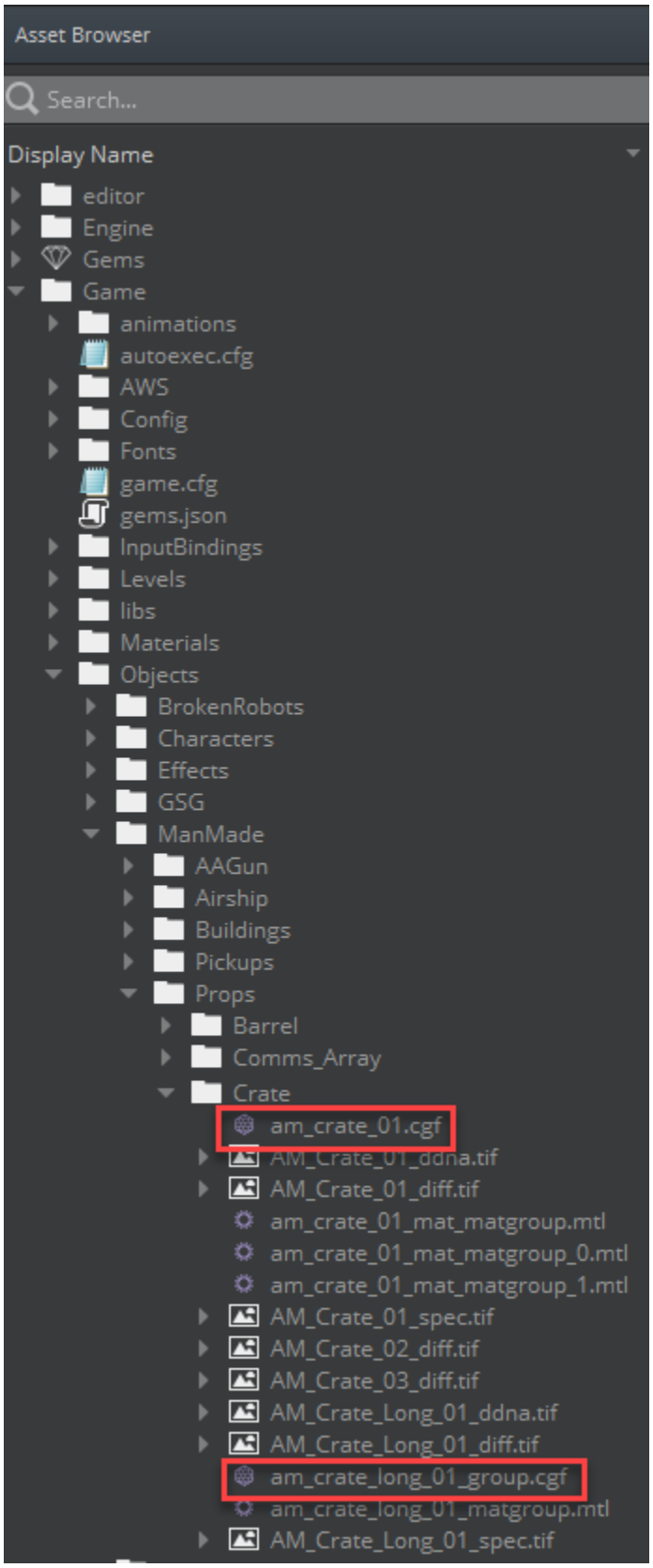
   In the **Entity Inspector**, in the **Transform** component's **Scale** box, set **Z** to `.75`.





3. In the **Asset Browser**, navigate to `Game\Objects\ManMade\Props\Crate`. Drag `am_crate_01.cfg` into the viewport.

   Also drag `am_crate_long_01_group.cfg` into the viewport.

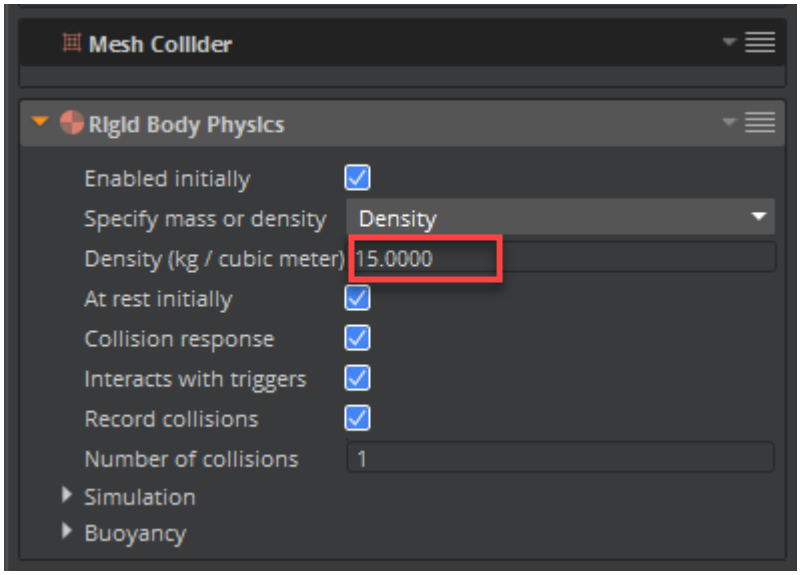

Asset Browser
Search...
Display Name
editor
Engine
Gems
Game
animations
autoexec.cfg
AWS
Config
Fonts
game.cfg
gems.json
InputBindings
Levels
libs
Materials
Objects
BrokenRobots
Characters
Effects
GSG
ManMade
AAGun
Airship
Buildings
Pickups
Props
Barrel
Comms_Array
Crate
am_crate_01.cgf
AM_Crate_01_ddna.tif
AM_Crate_01_diff.tif
am_crate_01_mat_matgroup.mtl
am_crate_01_mat_matgroup_0.mtl
am_crate_01_mat_matgroup_1.mtl
AM_Crate_01_spec.tif
AM_Crate_02_diff.tif
AM_Crate_03_diff.tif
AM_Crate_Long_01_ddna.tif
AM_Crate_Long_01_diff.tif
am_crate_long_01_group.cgf
am_crate_long_01_matgroup.mtl
AM_Crate_Long_01_spec.tif

4. Add a **Mesh Collider** component and a **Rigid Body Physics** component to each of the barrel and crates you just placed. And then set the **Rigid Body Physics** property **Density** to **15**.

   To do this, perform the following for each of the three barrel and crates:

   a. In the **Entity Outliner**, select the barrel or crates.
   b. In the **Entity Inspector**, click **Add Component**. Select **Mesh Collider**.
   c. In the **Entity Inspector**, click **Add Component**. Select **Rigid Body Physics**.
   d. In the **Entity Inspector**, in the **Rigid Body Physics** component, set the **Density (kg/cubic meter)** property to a value between **15** and **25**.

      The **Density** value defines the object's weight and affects how much it rolls when struck by the player or weapon fire.



Next: Stacking the Crates and Barrels to Make a Wall (p. 151)

Stacking the Crates and Barrels to Make a Wall

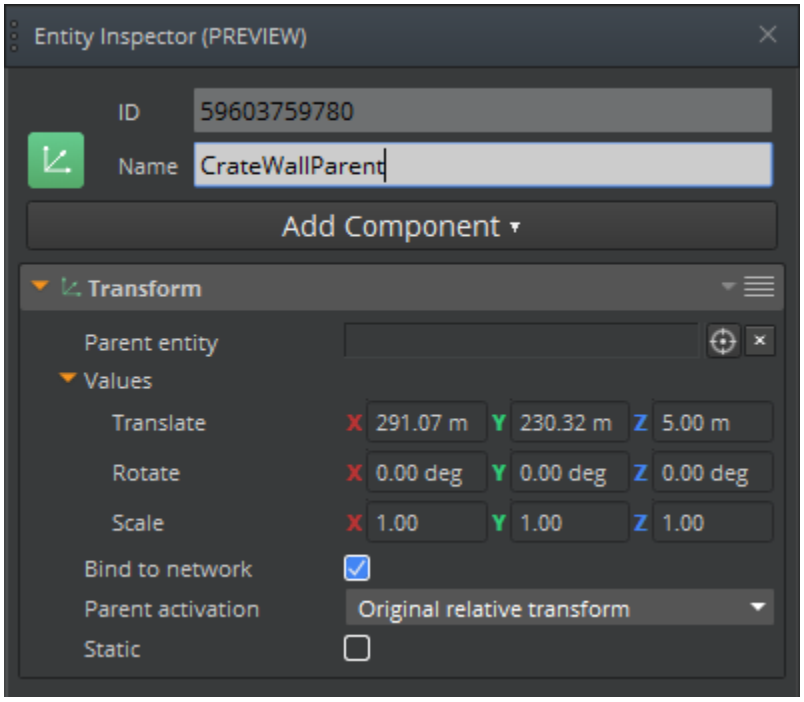
First you organize the crate and barrels by placing them into a parent entity and then moving the parent entity to a place in the maze where you build the wall.

Then you duplicate the crate and barrels and stack them upon each other to create the wall.

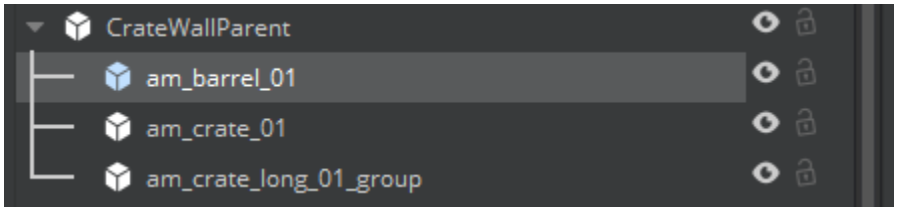
To parent your crate and barrels

1. In the viewport, near where you put your crate and barrels, right-click and choose **Create entity**.

2. In the **Entity Inspector**, name the new entity **`CrateWallParent`**.



3. In the **Entity Outliner**, drag the barrel and crates entities into **CrateWallParent**.



4. In the viewport, use the **Move** tool to drag **CrateWallParent** to where you want to place your wall within the maze.

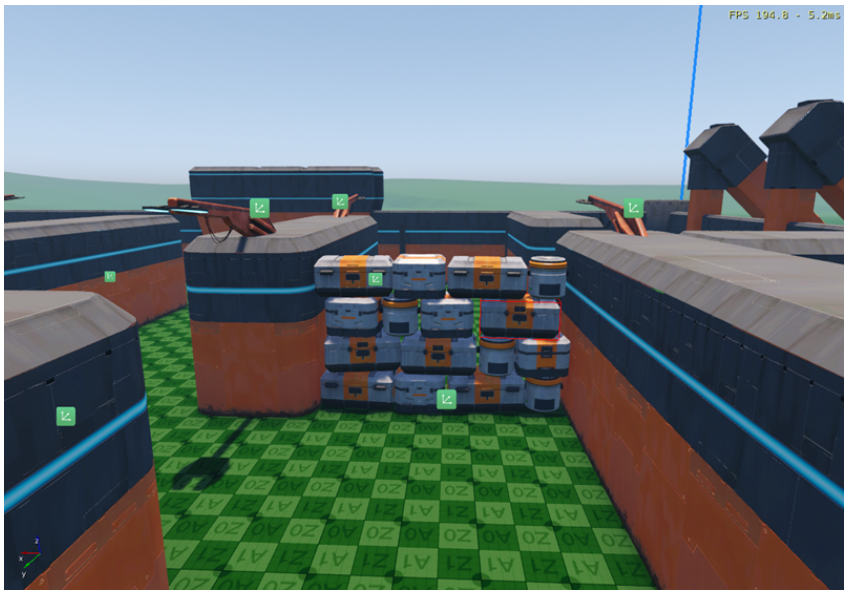
5. In the **Entity Outliner**, right-click on one of your crates and choose **Duplicate**. This creates a duplicate of the object on top of the original object.

   In the viewport, use the **Move** tool to position the duplicated object.

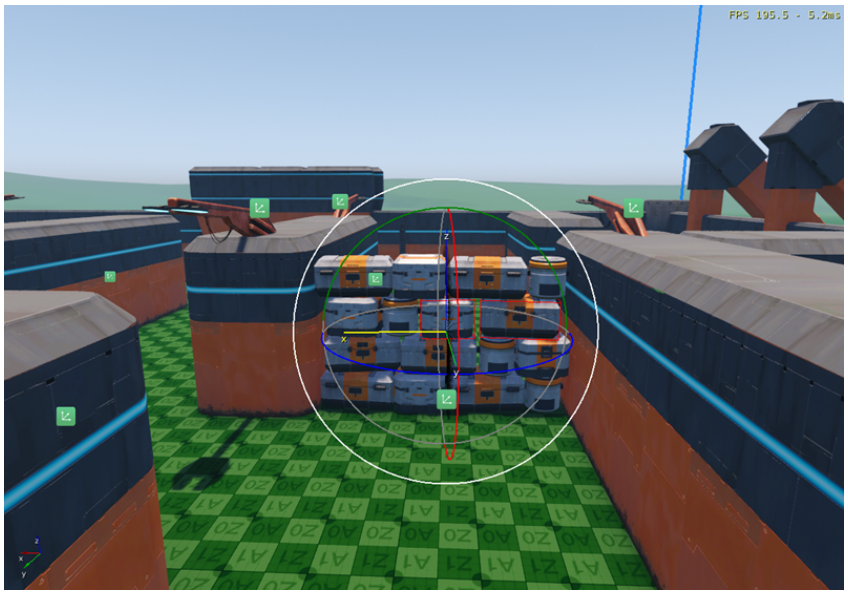
   Repeat this step to duplicate and position the crate and barrels until you have a wall four layers high that blocks the passageway.

   Tip
   You can also duplicate an entity by doing any of the following:

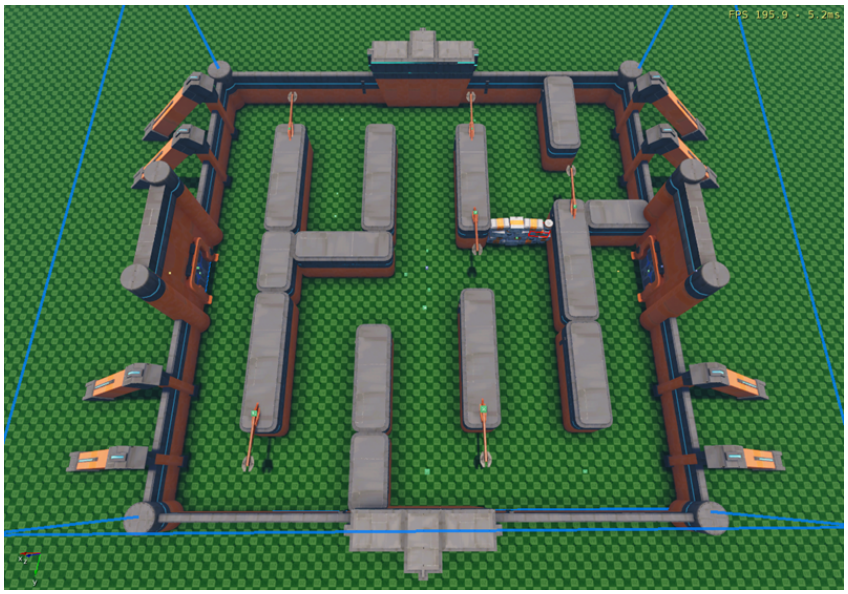
   - Right-clicking it in the viewport and choosing **Duplicate**.
   - Selecting the entity and then pressing **Ctrl+D**. Use your mouse pointer to position it in the viewport and then click to place.



6. Once you have stacked your wall, use the **Rotate** tool to nudge each object a few degrees along the **Z** rotation to give the wall the look of randomness.



When complete, your level should look similar to the following image.



7. Press **Ctrl+S** to save your level.
8. Press **Ctrl+G** to play your level.

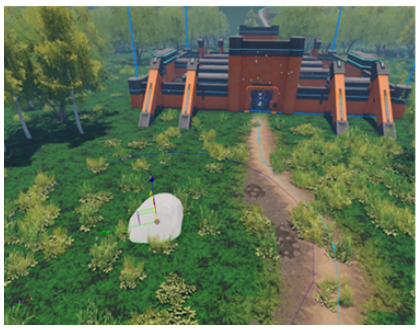
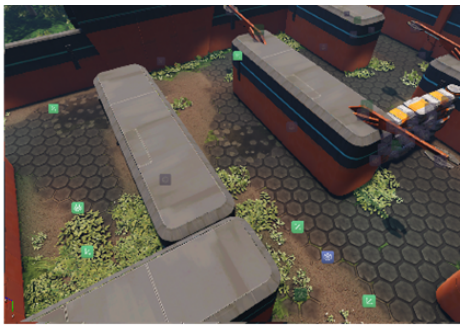
   Find the stack of crates and shoot at them or run into them. Notice how the crates fall and roll based on their density settings.

   Press **Esc** to quit.

Next: 8: Sculpting the Terrain (p. 155)

8: Sculpting the Terrain

A lushly detailed environment with varying features and movement creates an immersive player experience. In this tutorial, you use a variety of terrain and vegetation tools to make a verdant green landscape outside your maze and within, nature creeping in upon a cement world.

To create the textures and colors of this environment, you learn how to do the following:

- Create texture layers (p. 155) – Set the stage for texture layer painting.
- Paint pavement, grass, dirt, and rocks (p. 161) – Use the texture layers and color settings to paint in ground features.
- Draw a road (p. 166) – Create a meandering footpath using Lumberyard's road tool.
- Place trees (p. 172) – Position trees of varying size, shape, and type.
- Add shrubs and grass (p. 178) – Place above-ground vegetation that responds to player movement and environmental effects.
- Insert rocks (p. 184) – Position boulders and groups of smaller rocks; learn how to temporarily hide certain elements of your scene to reduce clutter.
- Customize your landscape (p. 191) – Practice the skills you learned by adding optional special touches such as flowers, shrubbery, and distant trees.

Next: Creating Texture Layers (p. 155)

Creating Texture Layers

To set the stage for painting pavement, grass, dirt, and rocks, you first create the texture layers. Think of this as choosing the consistency and texture of the paint.

You use the **Terrain Texture Layers** tool to create your texture layers. You also use the **Material Editor** to assign the appropriate textures to those layers.

To create texture layers

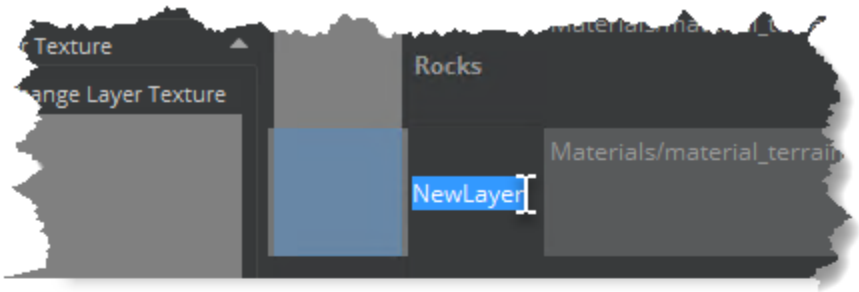
1. Open the **Terrain Texture Layers** editor. To do this, on the main menu, choose **Tools**, **Other**, **Terrain Texture Layers**.

   The **Terrain Texture Layers** editor displays the **Default** layer.

2. To create four more layers, in the **Layer Tasks** panel, click **Add Layer** four times.

   Click in the **Layer** column name and rename the four new layers with the following names:

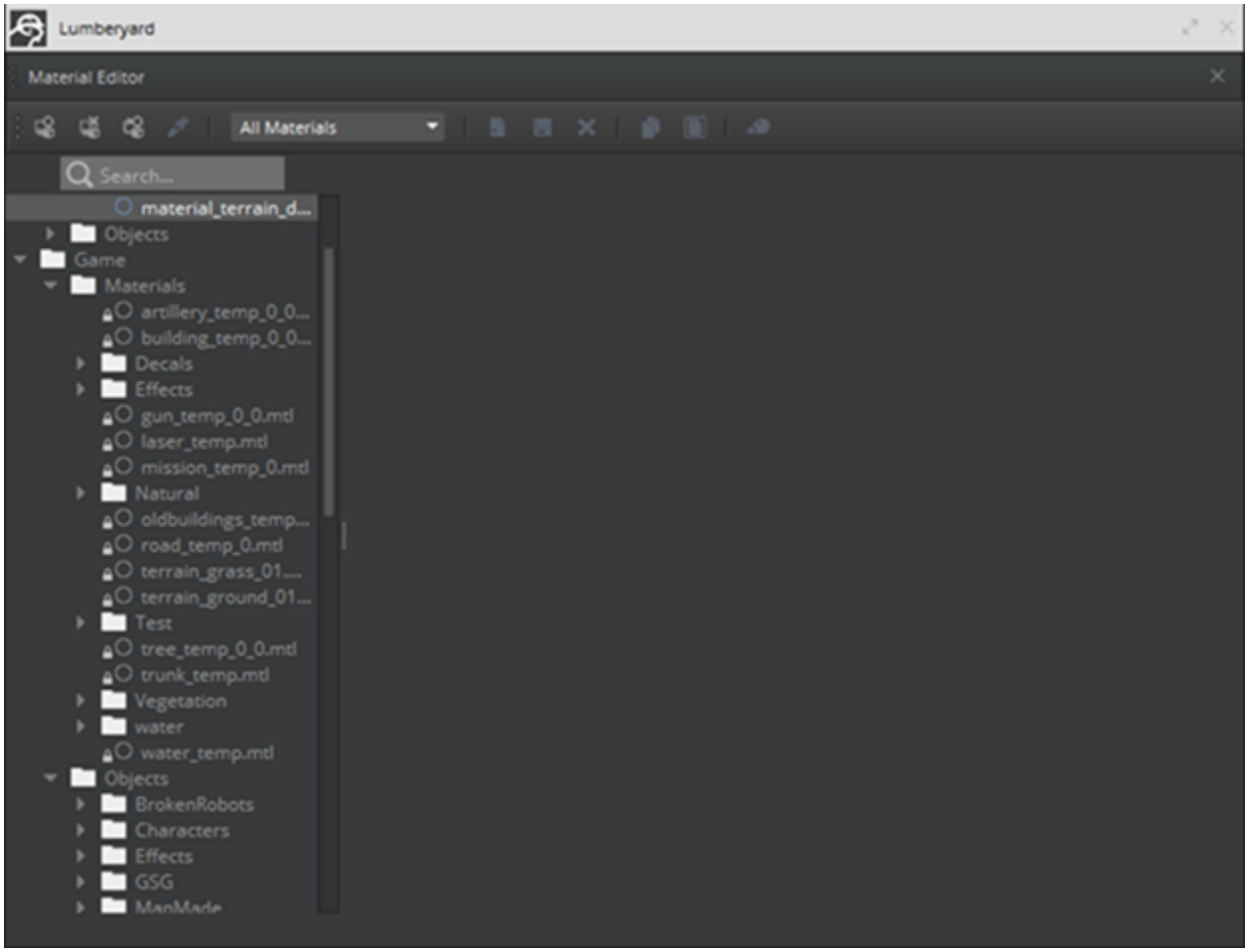
   - **Grass**
   - **Dirt**
   - **Rocks**
   - **HexGround**



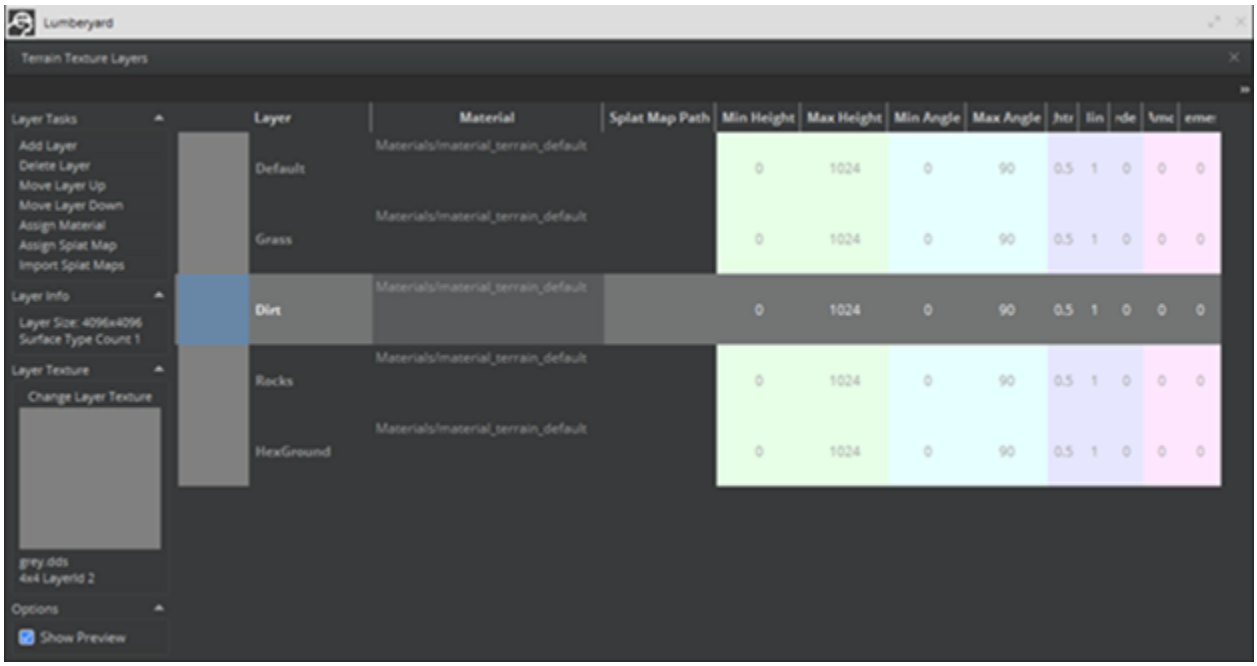
Next, you assign materials to each of the four texture layers you created. In this next procedure, you work with both the **Terrain Texture Layers** editor and the **Material Editor**, switching between the two editors to complete the tasks.

To assign materials to texture layers

1. Open the **Material Editor** by pressing **M** or, from the main menu choose **Tools**, **Material Editor**.

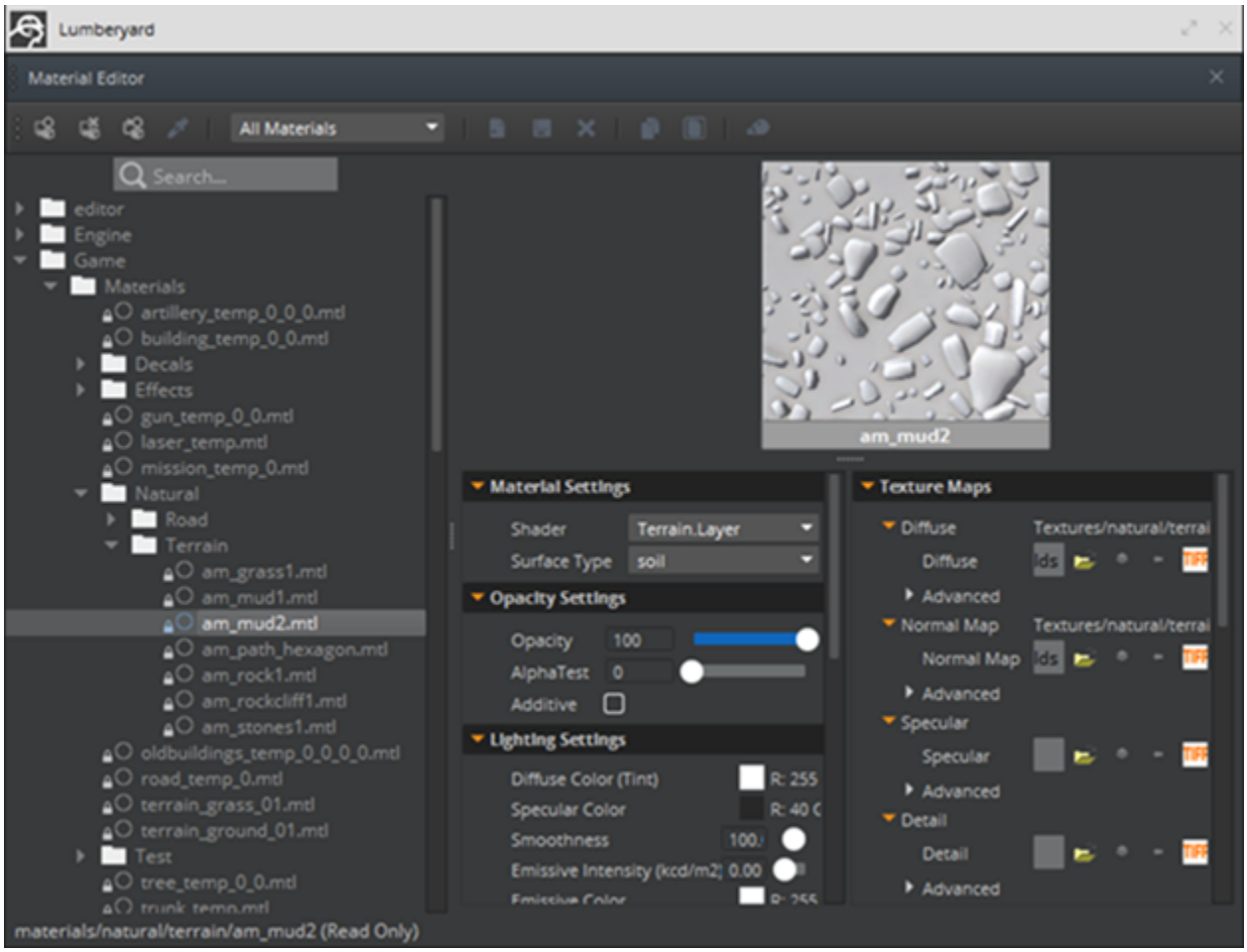


2. In the **Terrain Texture Layers** editor, select the **Dirt** layer.



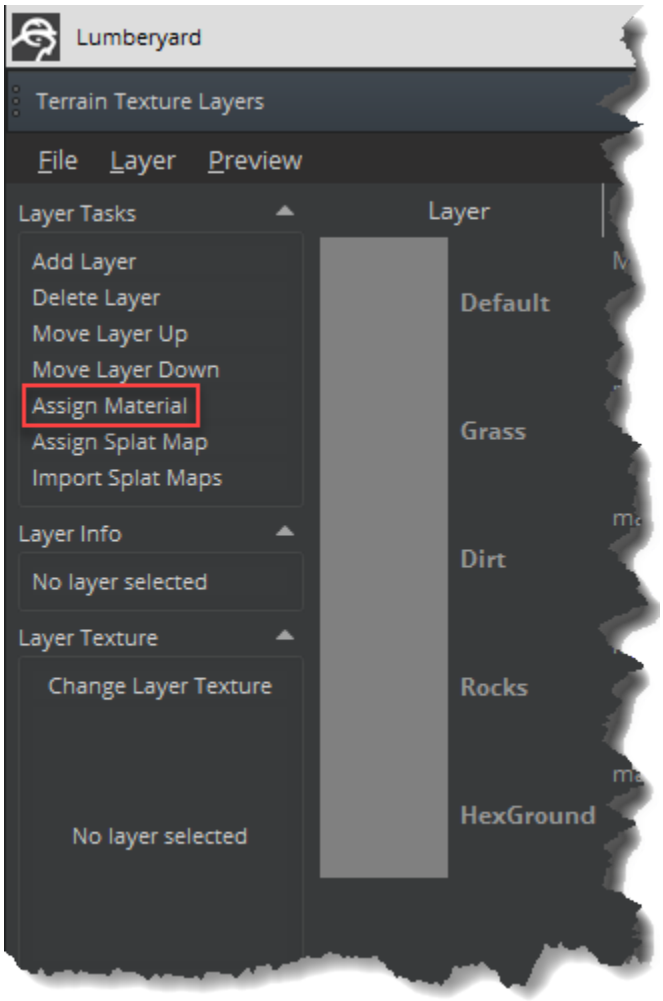
3. In the **Material Editor**, in the left panel, navigate to `Game\Materials\Natural\Terrain`.

   Select `am_mud2.mtl`.



4. Return again to the **Terrain Texture Layers** editor.

   The **Dirt** layer should still be selected. In the upper left panel, **Layer Tasks**, click **Assign Material**.

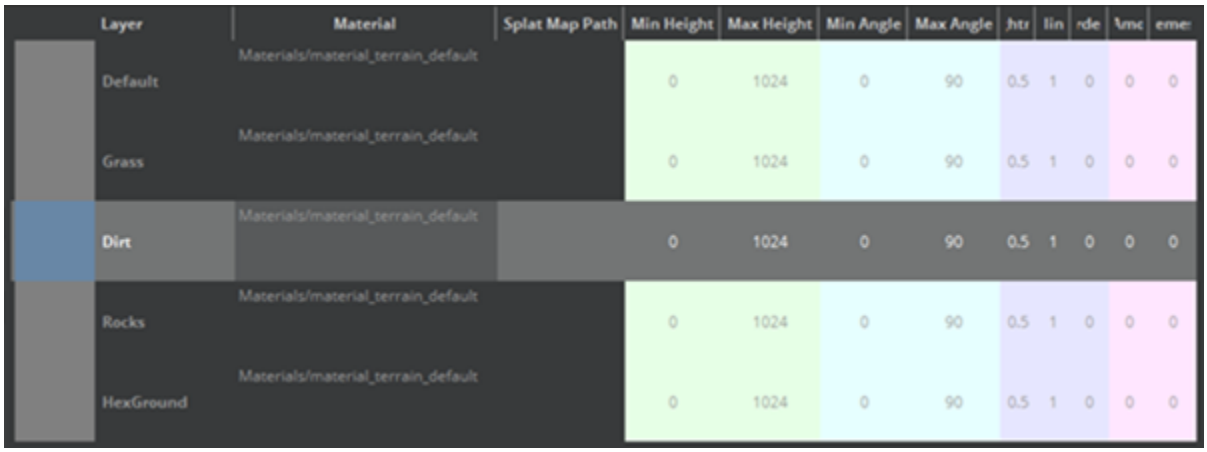


You have just assigned the **am_mud2.mtl** material to the **Dirt** layer.

5. Repeat the preceding steps for the other three layers, assigning the following materials:

   - **Grass** layer – `am_grass1.mtl` material.
   - **Rocks** layer – `am_mud1.mtl` material.
   - **HexGround** layer – `am_path_hexagon.mtl` material.

   When you have assigned the materials to the layers, your **Terrain Texture Layers** editor looks like the following image.



Layer	Material	Splat Map Path	Min Height	Max Height	Min Angle	Max Angle	htr	lin	rde	\mc	eme
Default	Materials/material_terrain_default		0	1024	0	90	0.5	1	0	0	0
Grass	Materials/material_terrain_default		0	1024	0	90	0.5	1	0	0	0
Dirt	Materials/material_terrain_default		0	1024	0	90	0.5	1	0	0	0
Rocks	Materials/material_terrain_default		0	1024	0	90	0.5	1	0	0	0
HexGround	Materials/material_terrain_default		0	1024	0	90	0.5	1	0	0	0

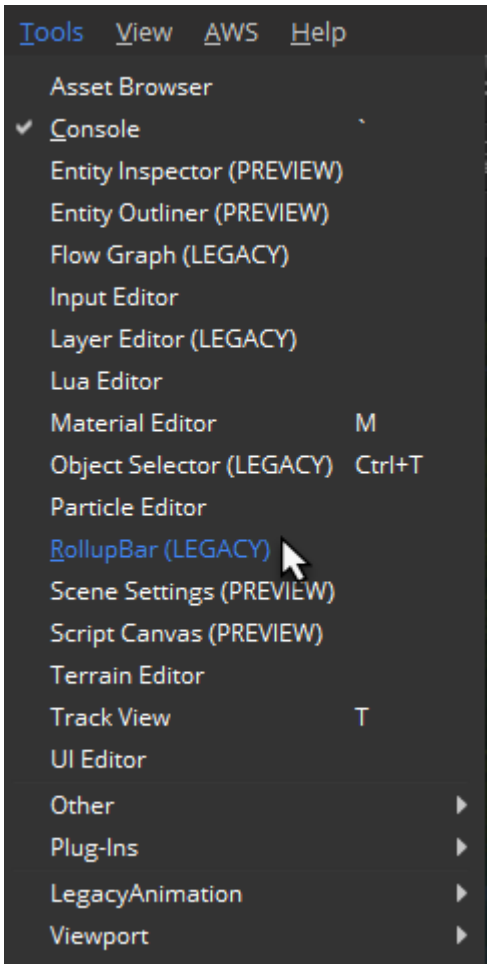
Next: Painting Pavement, Grass, Dirt, and Rocks (p. 161)

Painting Pavement, Grass, Dirt, and Rocks

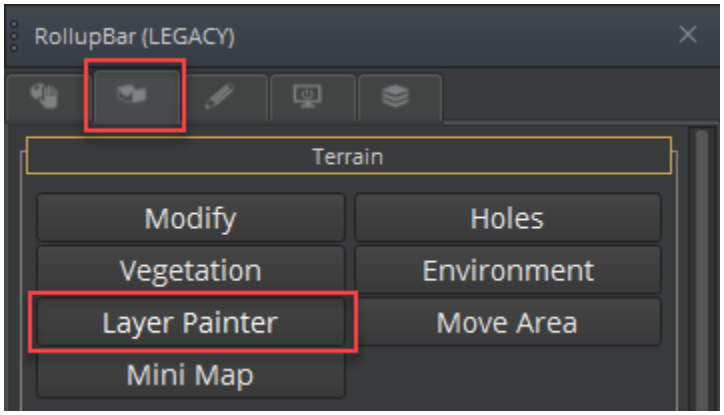
You now use the terrain texture layers that you created to paint those assigned materials onto your terrain. To do this, you use a terrain tool in the **Rollup Bar** called **Layer Painter**. After setting a color for a texture, you use that texture to paint your terrain.

To paint your terrain

1. To open the **Rollup Bar**, from the main menu choose **Tools**, **RollupBar (LEGACY)**.



2. In the **Rollup Bar**, choose the second tab, **Terrain Editing**. Click **Layer Painter**.

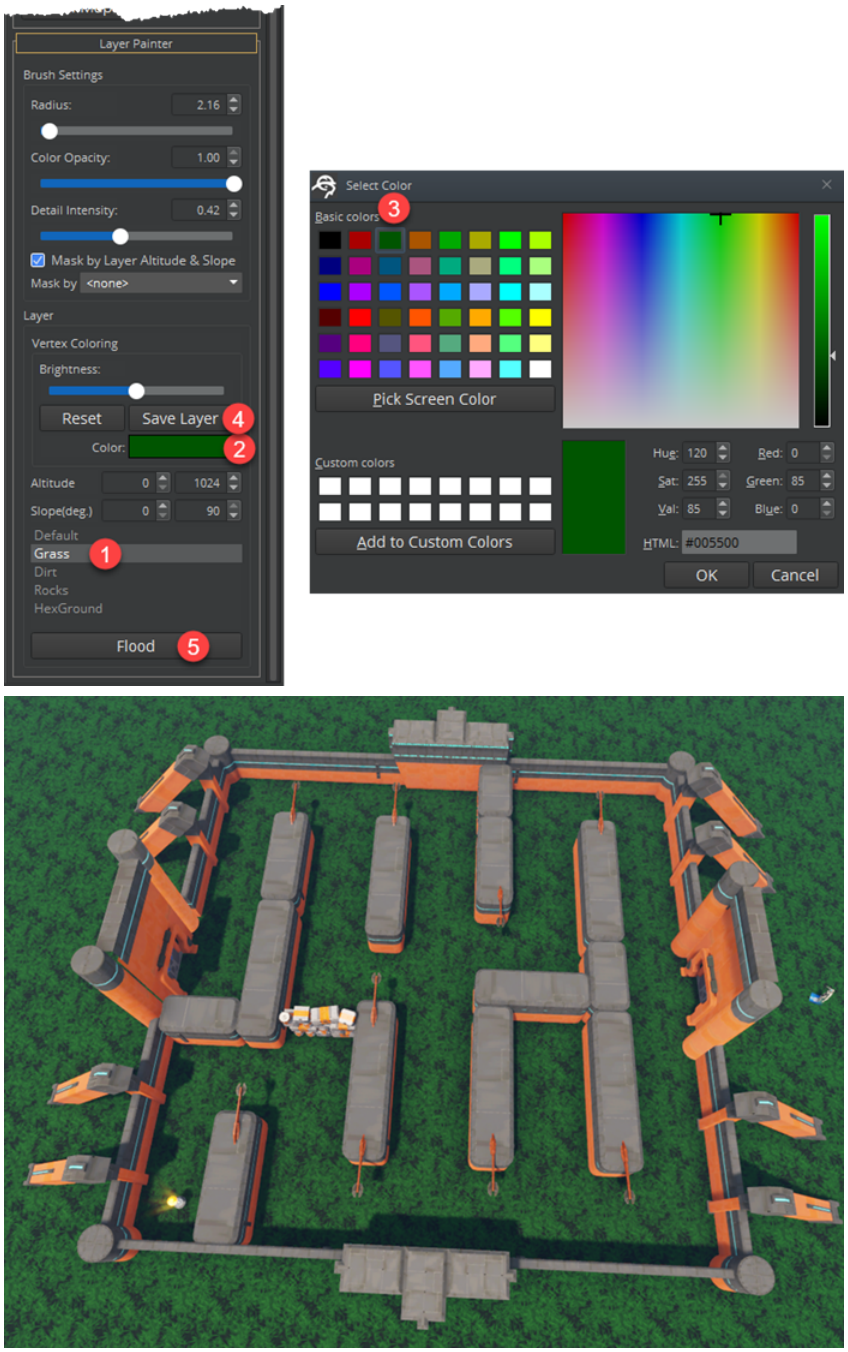


3. At the bottom of the panel is the list of the terrain texture layers that you created. The texture layers contain the texture and the material of the paint, but not the color.

   To define the color, click **Grass** (1), and then click in the colored part of the **Color** box (2). Select one of the green colors (3), and then click **OK**.

   Click **Save Layer** (4) to save this color to the **Grass** layer.

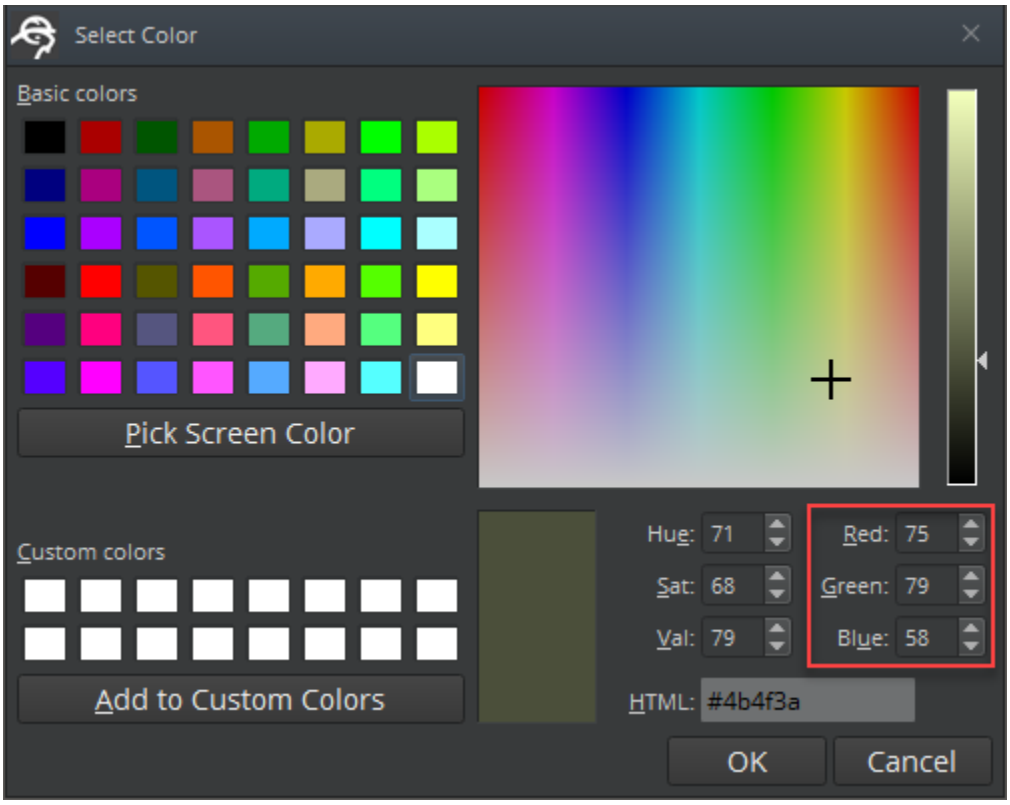
   Click **Flood** (5) to paint the entire landscape with the grass color and texture.



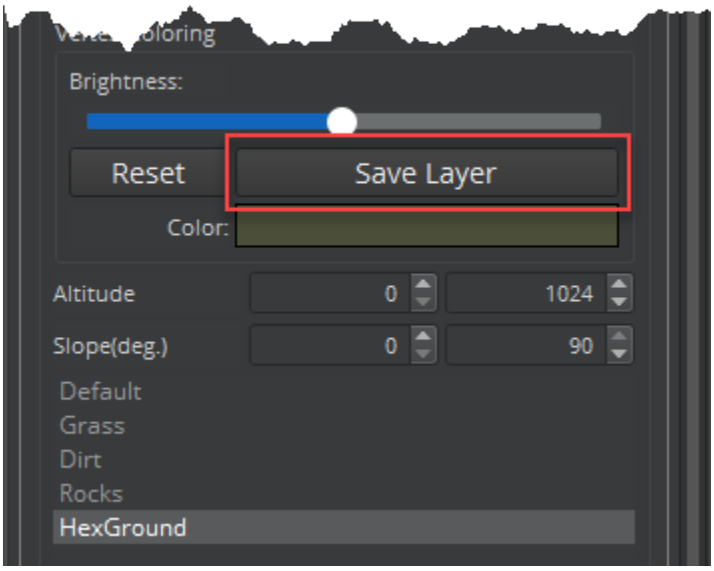
4. Now you can paint inside the maze pathways with a hexagon patterned pavement.

   Select the **HexGround** layer. Click in the colored part of the **Color** box. In the **Select Color** dialog box, set the color to the following values and then click **OK**:

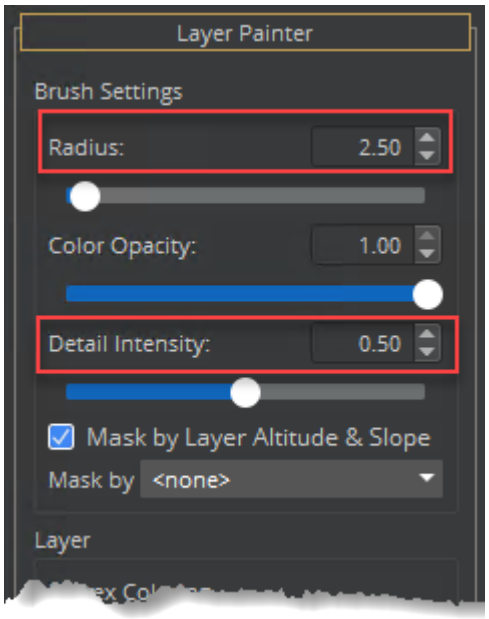
   - **Red**: `75`
   - **Green**: `79`
   - **Blue**: `58`



5. To save the color to the layer, click **Save Layer**.



6. In the **Layer Painter** tool's **Brush Settings**, set the **Radius** to `2.50` and the **Detail Intensity** to `0.50`.



Modify the brush radius (size) to suit the level of detail you want.

The detail intensity settings control the brush edge's softness. A value of **`0.1`** create a soft edge, while a value of **`1`** creates a hard, solid edge.

7. In the viewport, inside the maze, drag to paint the **HexGround** texture into the maze. Leave a bit of grass showing around the inside edges.

   Your maze should look similar to the following image.

8. Now you set the colors for the other texture layers.

   For the **Dirt** layer, do the following:

   - Set the color to **Red**: **`140`**, **Green**: **`113`**, and **Blue**: **`87`**.

- Click **Save Layer** to save the color settings.

For the **Rocks** layer, do the following:

- Set the color to **Red**: **126**, **Green**: **115**, and **Blue**: **102**.
- Click **Save Layer** to save the color settings.

9. Paint the maze pathways with the dirt and rock layers, adjusting the radius and detail intensity as necessary.

   When finished, your maze should look similar to the following image.

10. Press **Ctrl+S** to save your level.

Next: Drawing a Road (p. 166)

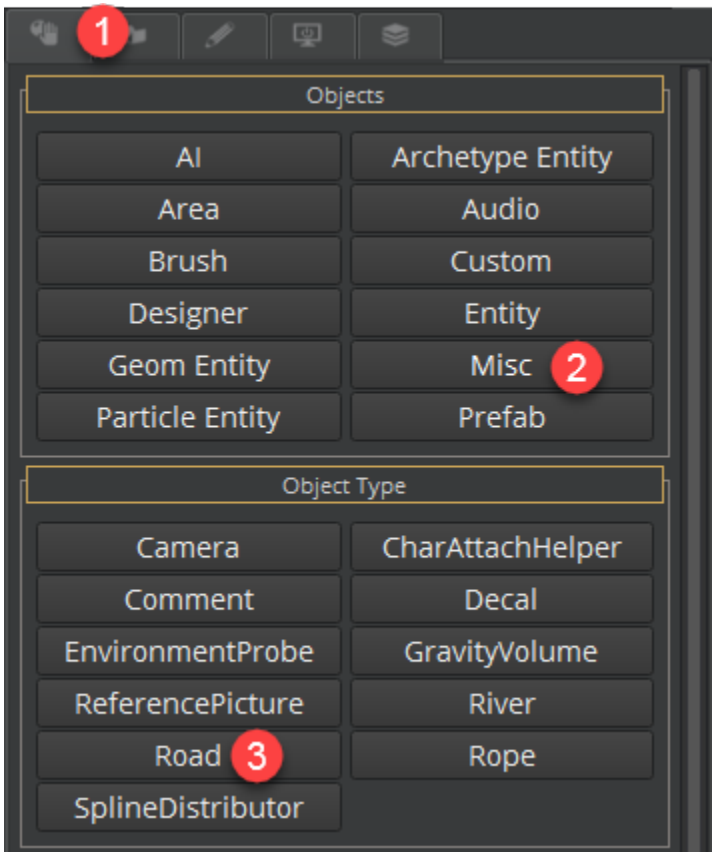
Drawing a Road

You use Lumberyard's **Road** tool in the **Rollup Bar** to create a worn-looking footpath that leads to the maze entrance and away from the maze exit.

You first draw out the meandering line of the road and adjust it to give some variation in the road's height. You then use the **Material Editor** to assign to the road a dirt texture. To make the surrounding landscape slope up to the height variations of the road, you use a height aligning tool. For the final touch, you use the terrain texture layers that you set up previously to add some dirt and rocks to the path's borders.

To create a path using the road tool

1. If the **Rollup Bar** is not already open, open it from the main menu by choosing **Tools**, **RollupBar (LEGACY)**.
2. On the **Objects** tab, click **Misc**, and then click **Road** to open the road tool.

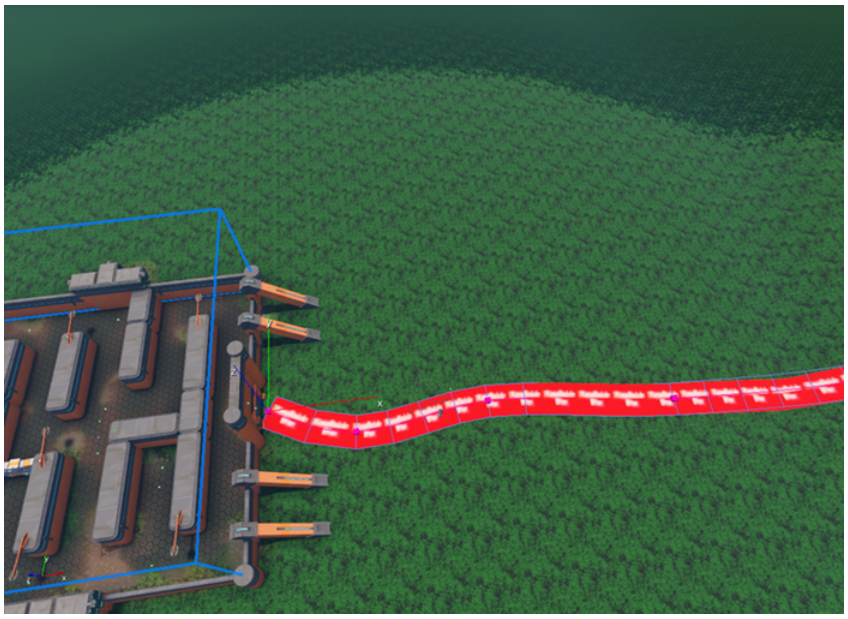


3. The **Road** tool is a simple spline-based tool that draws a decal texture along the terrain.

   Click just outside the entrance to the maze to start your road. Move your mouse pointer a small distance away and click again. Repeat.

   Notice how the shape of the path changes depending on where you place subsequent points. Experiment with the different ways of changing the shape of the path. If you need to expose more of your landscape while creating the path, you can use the typical navigational controls to change your view without breaking the path.

   To stop or end the path, double-click on the last spline point.

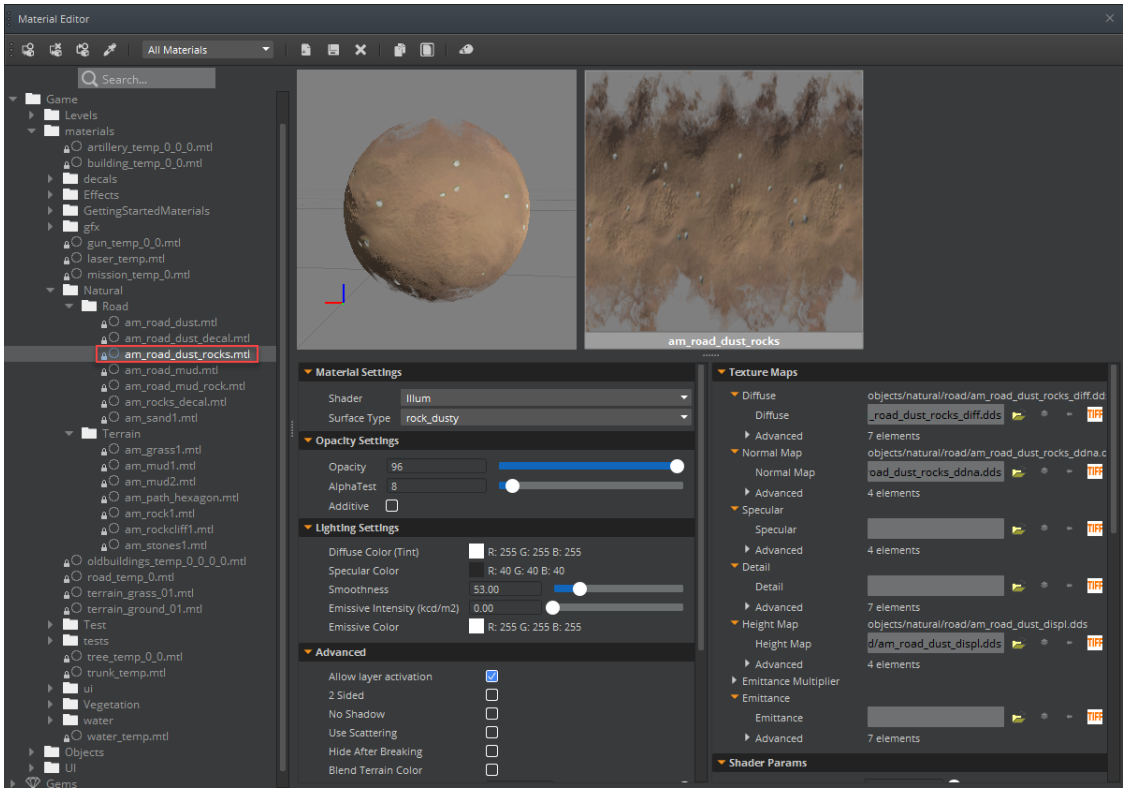


4. The road's default texture displays **Replace Me**. To replace the texture, first make sure that the road is still selected.

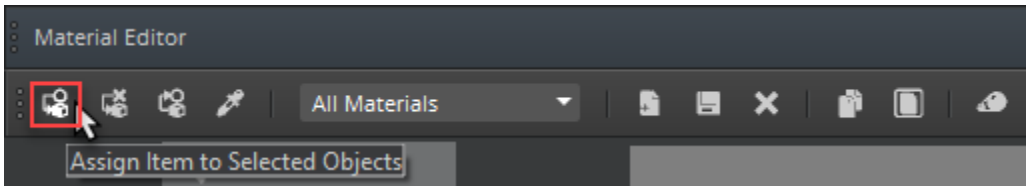
   Open the **Material Editor** by pressing **M**.

   In the left pane, navigate to `Game\Materials\Natural\Road`.

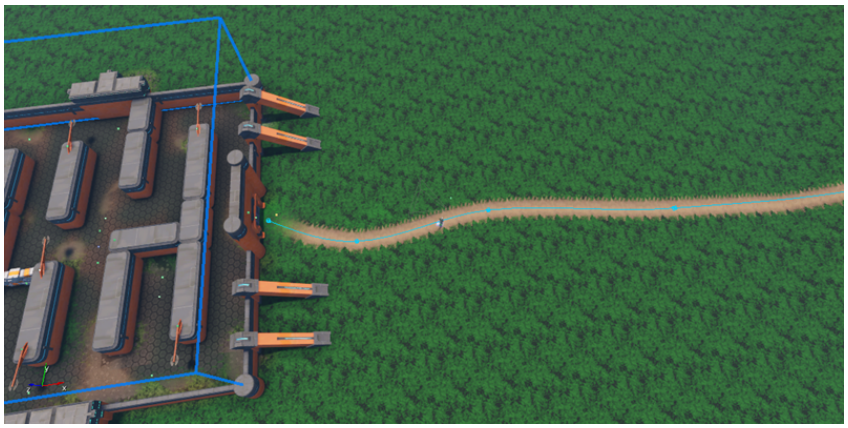
   Select `am_road_dust_rocks.mtl`.



5. Click the **Assign Item to Selected Objects** icon in the **Material Editor**'s toolbar.



Close the **Material Editor**. In the viewport, you see that the road's **Replace Me** texture has been replaced with the dirt and rocks texture that you just assigned.



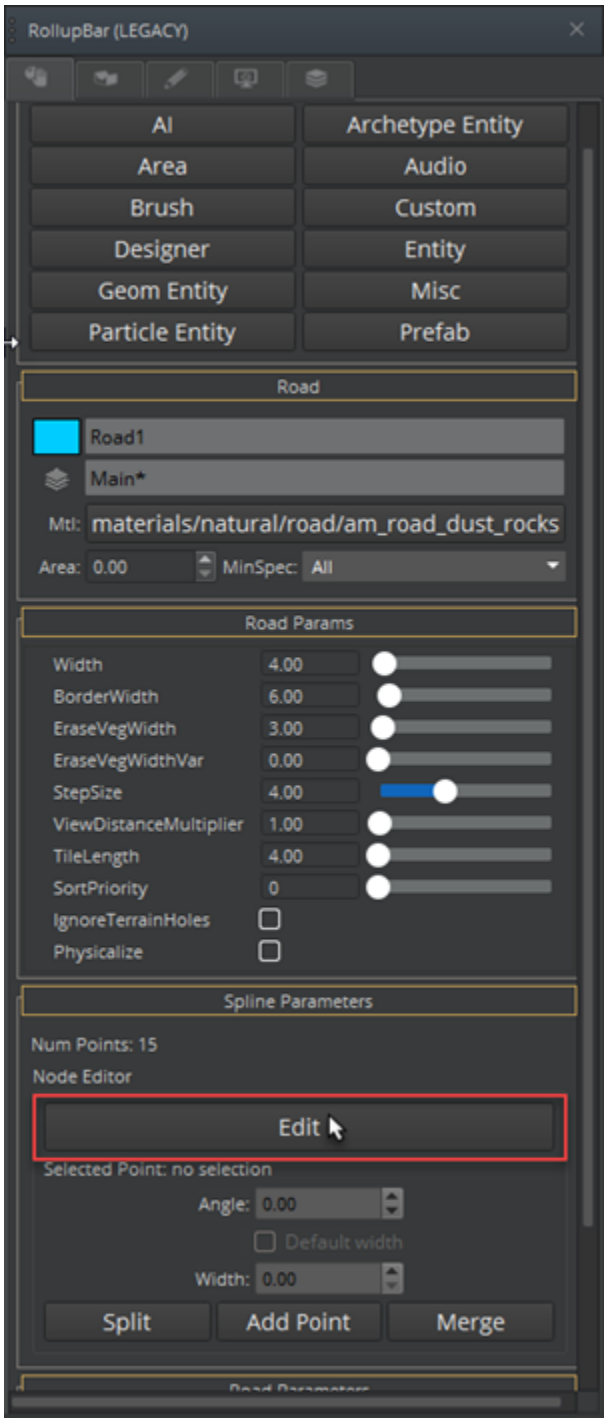
6. Using the same steps, create another road leading away from the maze's exit. Replace the default texture with the dirt and rocks texture.

You can edit your path by moving any of the points. You create some up and down variations in the road you've drawn, and then align the height of the terrain to gradually slope up or down to that point on the path.

To edit the road and align height

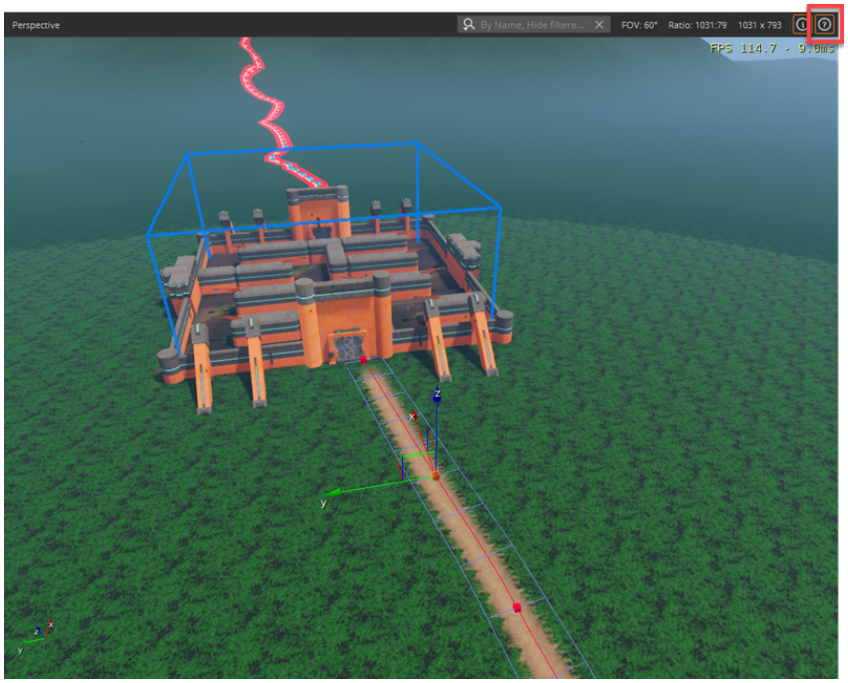
1. In the viewport, make sure that one of your roads is selected.

   In the road tool, under **Spline Parameters**, click **Edit**.

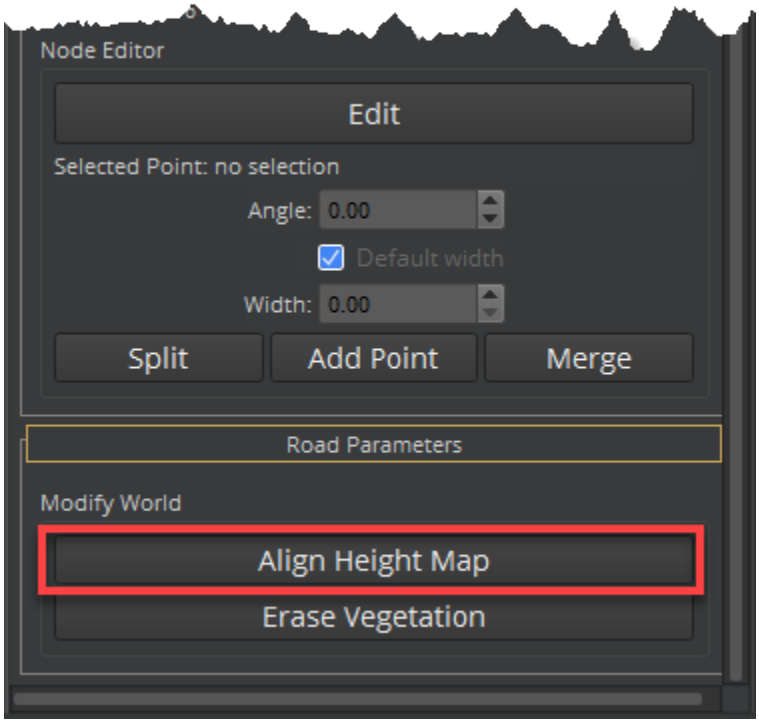


The spline points appear as movable points along the path. Use the **Move** tool to adjust the points. Make some points lower than the terrain and other points higher.

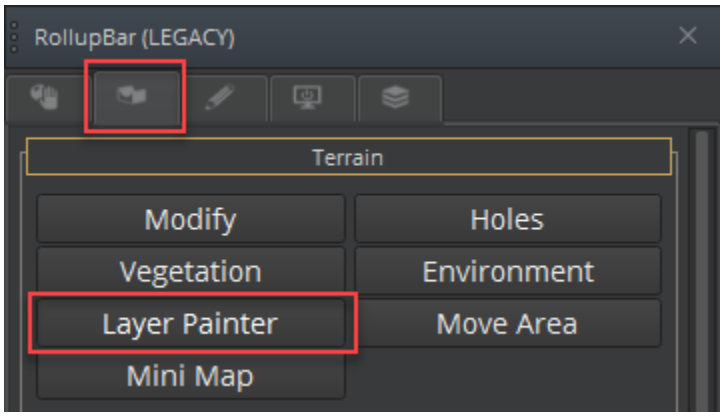
> **Tip**
> If you click **Edit** and the movable points are not displayed, turn on your helpers. To do so, click the question mark icon at the upper right corner of your viewport.



2. To make the height of the terrain slope up or down to the points on your road that do not sit at the same level, click **Align Height Map**. You may have to scroll down in the road tool, as this command is located near the bottom of the tool, under **Road Parameters**.



3. Use the **Layer Painter** tools to paint some dirt and rocks textures onto the borders of the path.



The path should look similar to the following image.

Next: Placing Trees (p. 172)

Placing Trees

You can use the **Vegetation** tool in the **Rollup Bar** to quickly and easily place any number and combination of mesh objects in your level.

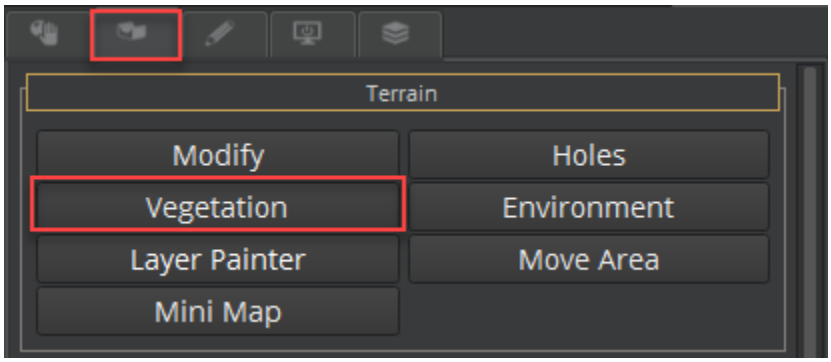
You use this tool to place vegetation such as trees, shrubs, and grasses around and inside your maze.

In the following procedure, you group multiple tree types into a group or category and then paint the trees into your scene. You also learn how to adjust the tool properties to customize the tree placement.

> **Note**
> Be mindful when placing a large number of mesh objects, as performance can suffer if there is too much vegetation.

To paint trees into your scene

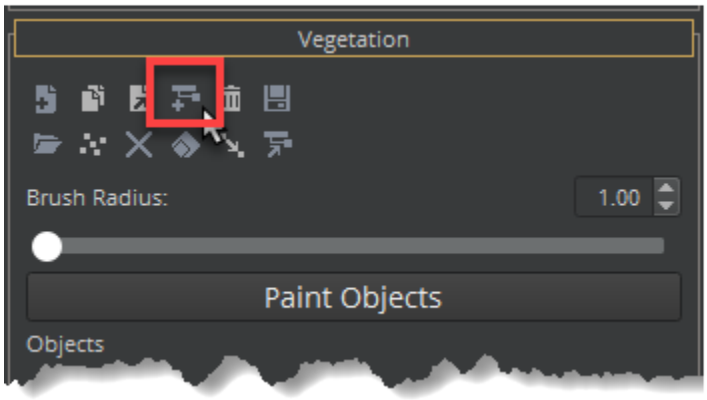
1. If the **Rollup Bar** is not open, open it from the main menu by choosing **Tools**, **RollupBar (LEGACY)**.
2. In the **Rollup Bar**, on the **Terrain** tab, click **Vegetation**.

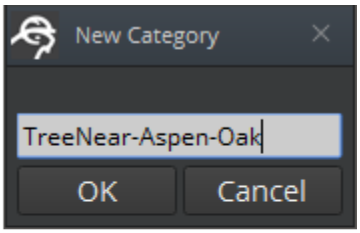


3. To paint different types of trees into the scene with one stroke, you must set up a category.

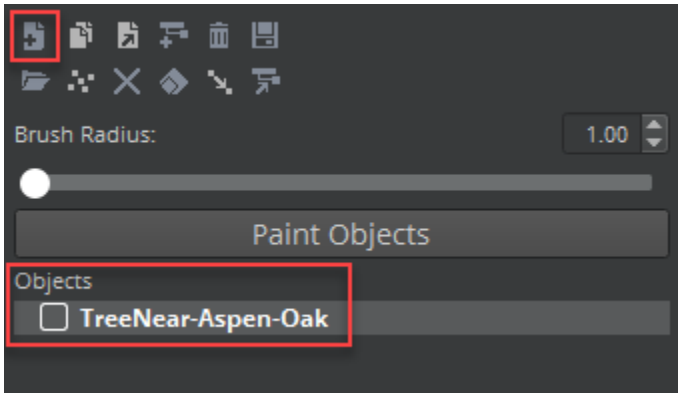
   To do this, under **Vegetation**, click the **Add Vegetation Category** icon.

   In the **New Category** box, type **`TreeNear-Aspen-Oak`**. Click **OK**.



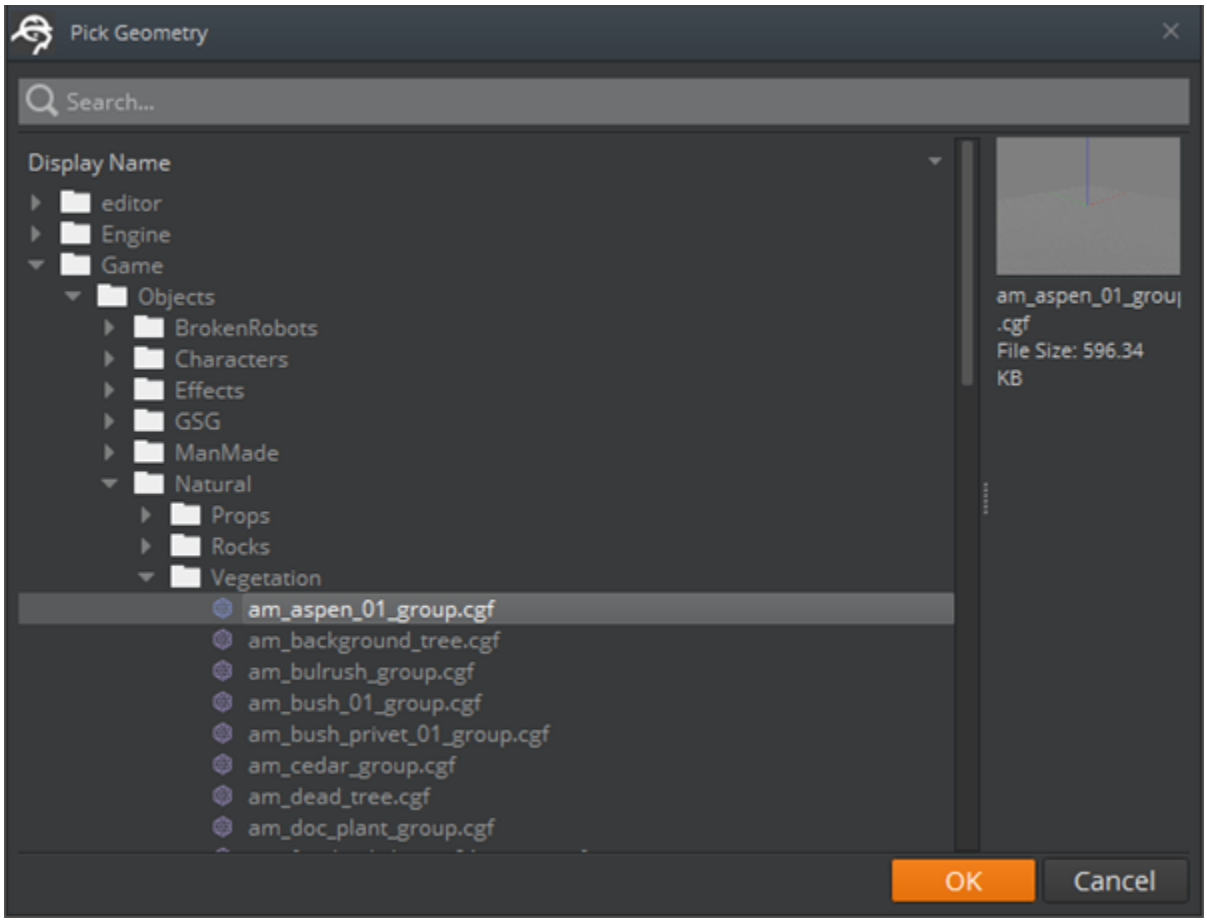


4. Click to select the object category that you just created. Then click the **Add Vegetation Object**  icon.



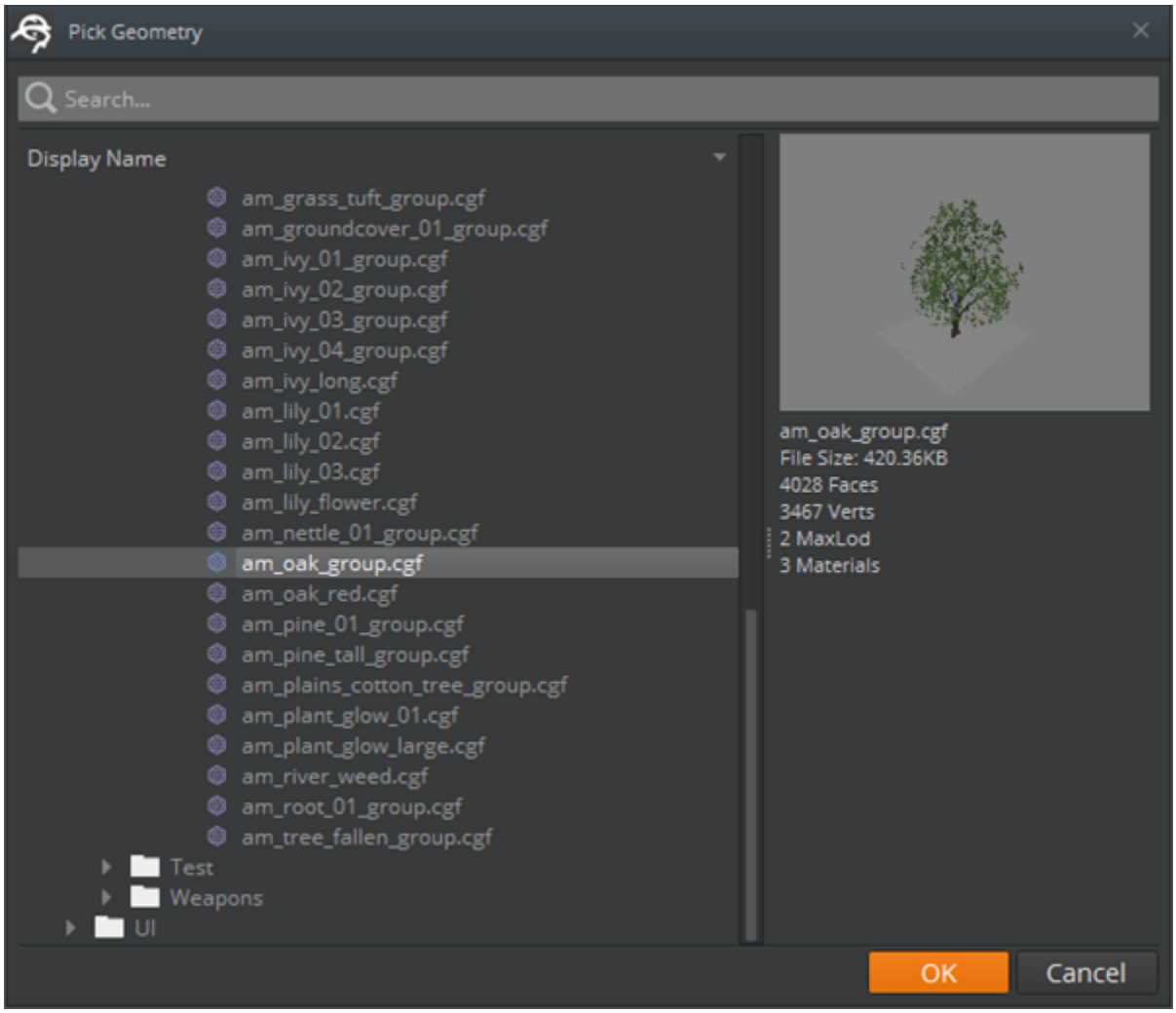
5. In the **Pick Geometry** dialog box, navigate to `Game\Objects\Natural\Vegetation`.

   Select `am_aspen_01_group.cgf`. Click **OK**.



6. To add another tree type to this category, click **Add Vegetation Object**  again.

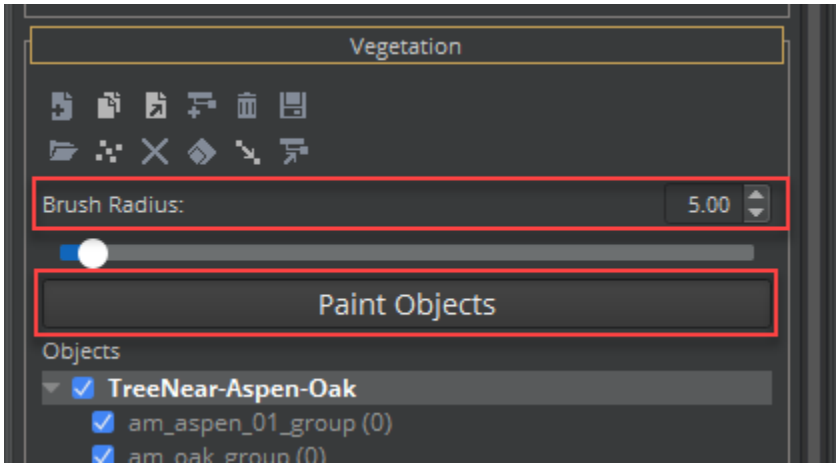
   Select `am_oak_group.cgf`. Click **OK**.



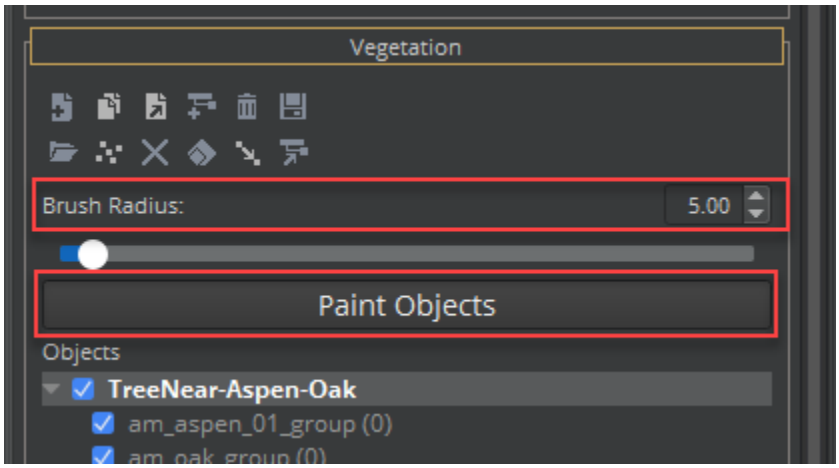
7. To paint with all the objects within the category, select the category name (which is selected by default at this point). You can also choose to paint with specific objects within the category.

   Set the **Brush Radius** to **5**.

   Choose **Paint Objects**.

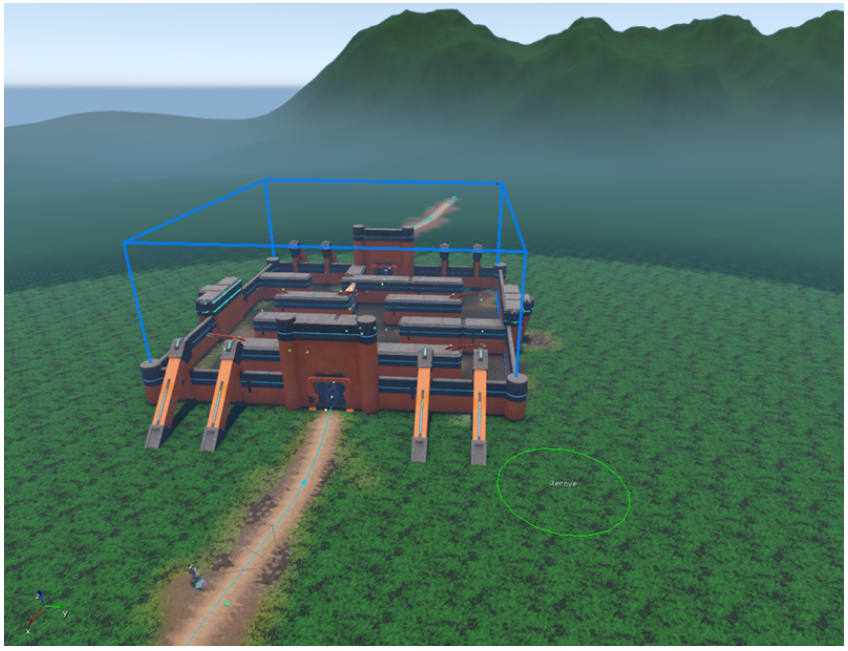


8. In the viewport, drag to place a small grove of trees.



With the current settings, notice that the trees are distributed too densely and have little variation between them.

9. To erase the trees that you just placed, hold down **Ctrl** while dragging over the trees. Continue until you remove all the trees.



10. To change the settings for the placement of the trees, select the group **TreeNear-Aspen-Oak**.

    In the properties settings that appear below the group, set the following properties. Leave all other properties at their default settings.

    - **Size**: `1.25`

      Sets the base size for the objects that you place.
    - **+-SizeVar**: `0.2`

      Varies the size of each object (larger or smaller) by up to this amount.
    - **RandomRotation**: `Selected`

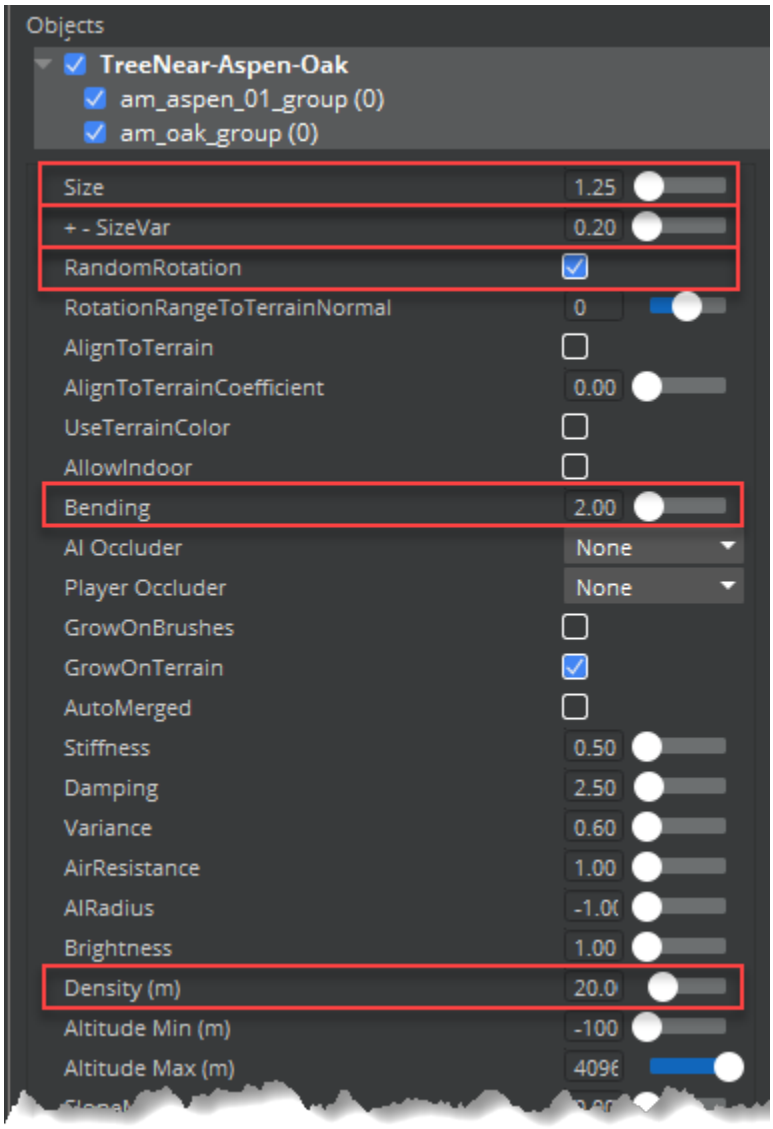
Rotates the objects in random directions.

- **Bending**: **2**

  Sets degree to which wind can affect the object.

- **Density**: **20**

  Defines the maximum distance between objects.

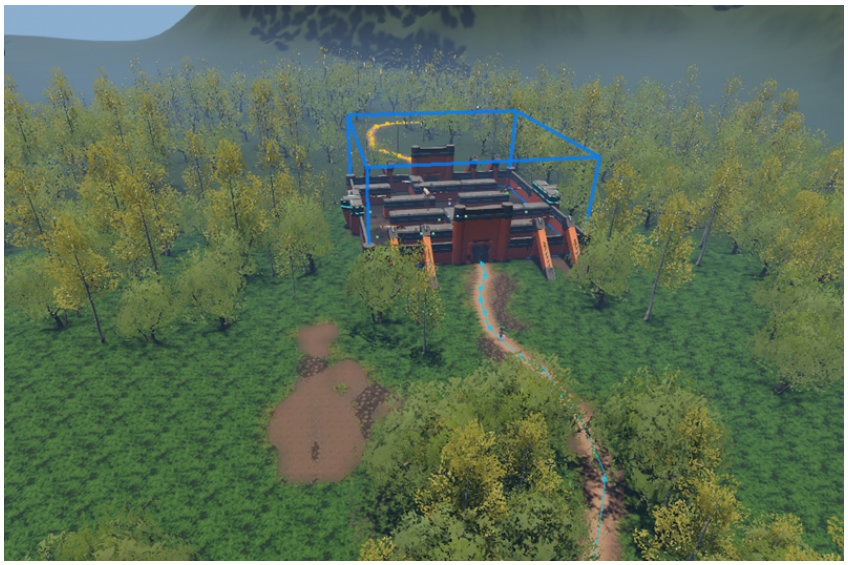


11. Set the **Brush Radius** to **10**.

    Drag in the viewport to paint the trees into the scene.

    Note
    The **Brush Radius** setting cannot be less than half of the **Density** setting. For example, if you attempt to paint with a brush radius of **9**, nothing happens. With a **Density** setting of **20**, your **Brush Radius** must be **10** or larger.

It can be easy to go overboard and paint an overly dense set of trees. Try to achieve a density similar to the following image.



12. Press **Ctrl+S** to save your level.
13. Press **Ctrl+G** to play your level.

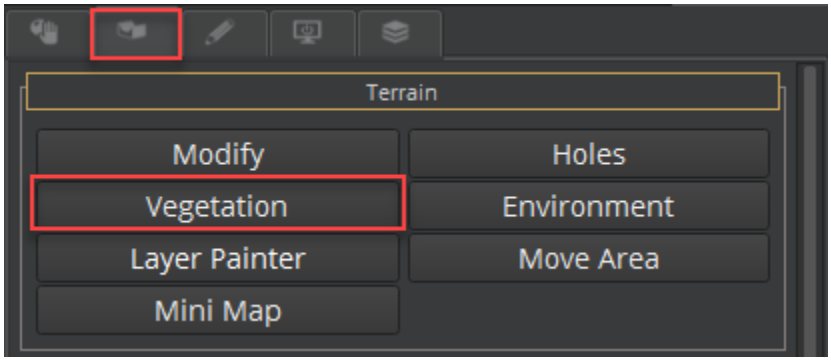
Next: Adding Shrubs and Grass (p. 178)

Adding Shrubs and Grass

The grass texture that you added to your terrain provides a good base color and texture but gives little realism. Painting in shrubs and grasses adds movement, responsiveness, and variation that cannot be achieved with texture alone.

To add shrubs and grasses

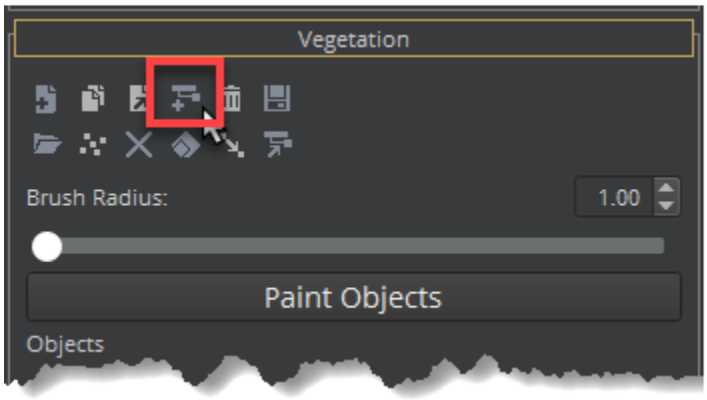
1. If the **Rollup Bar** is not open, open it by clicking in the main menu **Tools**, **RollupBar (LEGACY)**.
2. In the **Rollup Bar**, on the **Terrain** tab, click **Vegetation**.

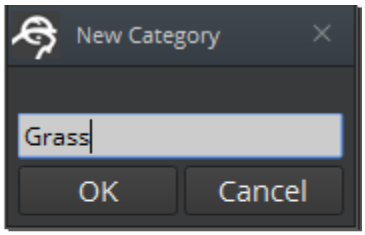


3. To paint different types of grasses and shrubs into the scene with one stroke, you must set up a new category.

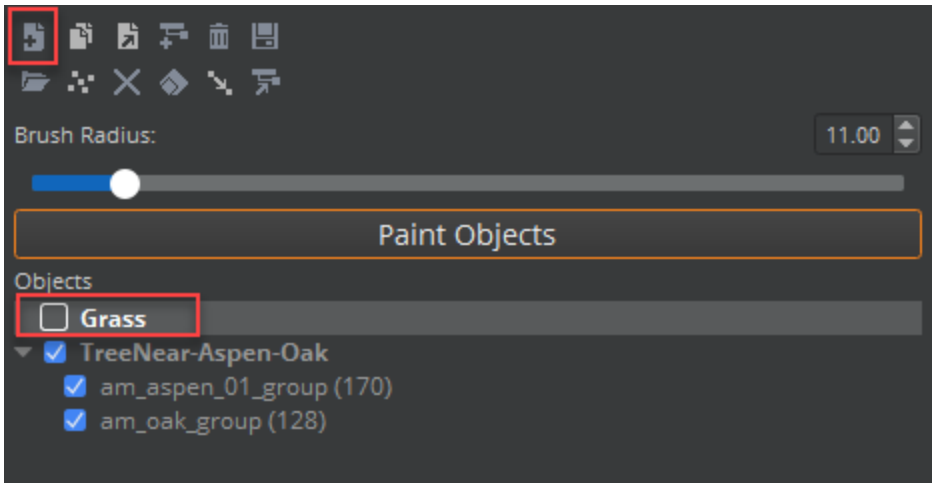
   To do this, under **Vegetation**, click the **Add Vegetation Category** icon.

In the **New Category** box, type **`Grass`**. Click **OK**.



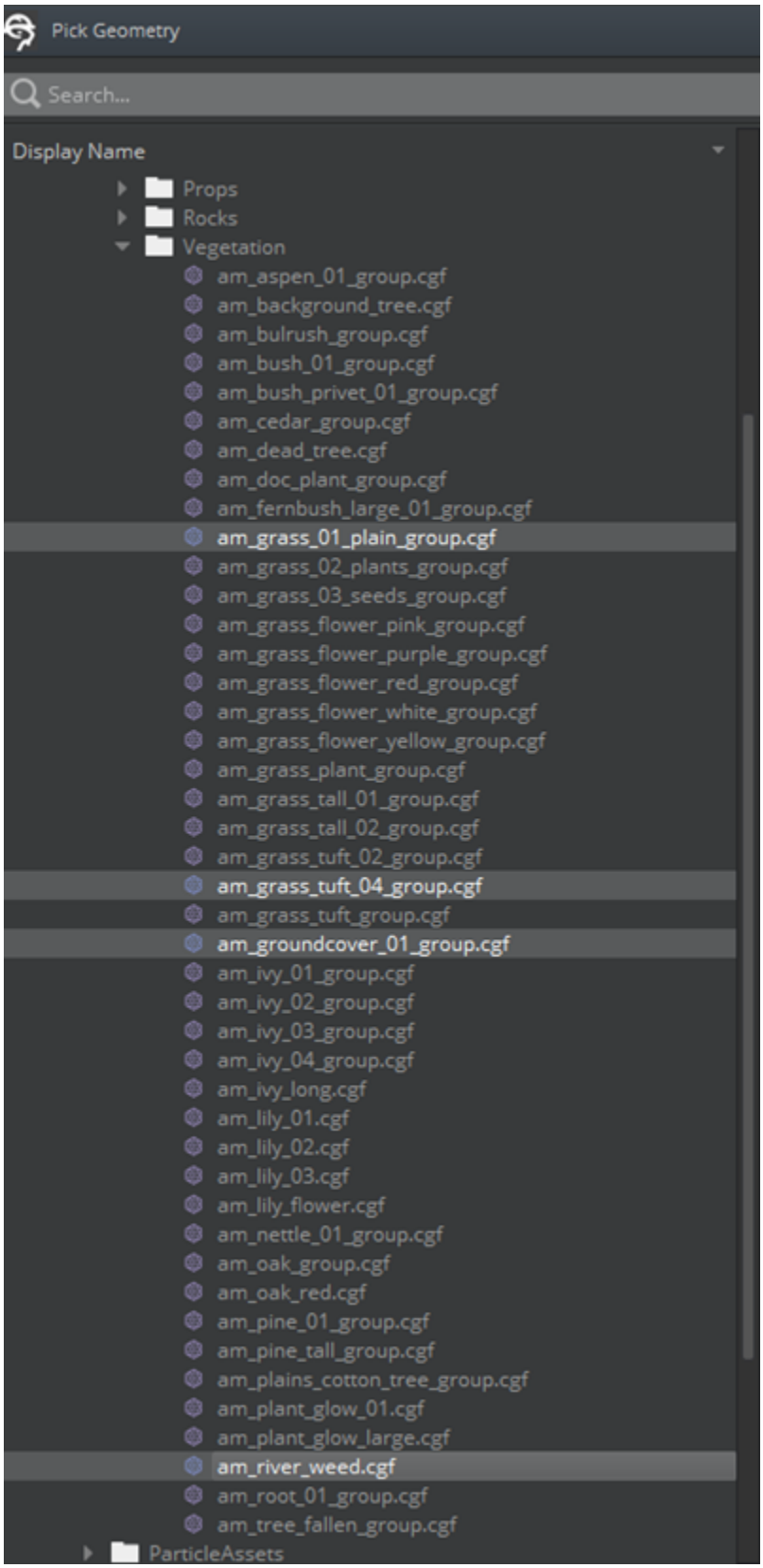


4. Select the **Grass** object category that you just created, and then click the **Add Vegetation Object**  icon.



5. In the **Pick Geometry** dialog box, navigate to `Game\Objects\Natural\Vegetation`. Hold down **Ctrl** while selecting the following vegetation objects:

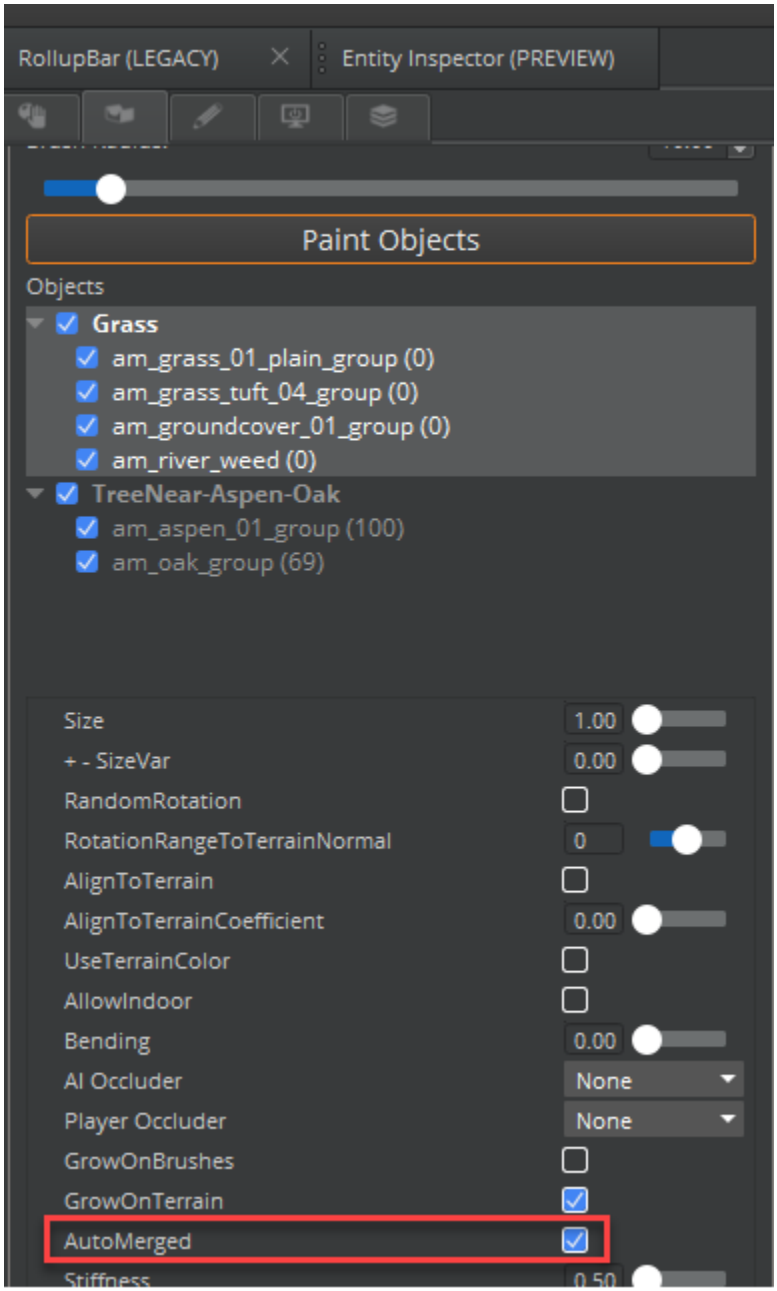
   - `am_grass_01_plain_group.cgf`
   - `am_grass_tuft_04_group.cgf`
   - `am_groundcover_01_group.cgf`
   - `am_river_weed.cgf`


Pick Geometry
Search...
Display Name
Props
Rocks
Vegetation
am_aspen_01_group.cgf
am_background_tree.cgf
am_bulrush_group.cgf
am_bush_01_group.cgf
am_bush_privet_01_group.cgf
am_cedar_group.cgf
am_dead_tree.cgf
am_doc_plant_group.cgf
am_fernbush_large_01_group.cgf
am_grass_01_plain_group.cgf
am_grass_02_plants_group.cgf
am_grass_03_seeds_group.cgf
am_grass_flower_pink_group.cgf
am_grass_flower_purple_group.cgf
am_grass_flower_red_group.cgf
am_grass_flower_white_group.cgf
am_grass_flower_yellow_group.cgf
am_grass_plant_group.cgf
am_grass_tall_01_group.cgf
am_grass_tall_02_group.cgf
am_grass_tuft_02_group.cgf
am_grass_tuft_04_group.cgf
am_grass_tuft_group.cgf
am_groundcover_01_group.cgf
am_ivy_01_group.cgf
am_ivy_02_group.cgf
am_ivy_03_group.cgf
am_ivy_04_group.cgf
am_ivy_long.cgf
am_lily_01.cgf
am_lily_02.cgf
am_lily_03.cgf
am_lily_flower.cgf
am_nettle_01_group.cgf
am_oak_group.cgf
am_oak_red.cgf
am_pine_01_group.cgf
am_pine_tall_group.cgf
am_plains_cotton_tree_group.cgf
am_plant_glow_01.cgf
am_plant_glow_large.cgf
am_river_weed.cgf
am_root_01_group.cgf
am_tree_fallen_group.cgf
ParticleAssets

6. To modify the settings for the placement of the shrubs and grasses, select the group **Grass**.

   Unlike with trees, you want the player to run through the grass and shrubs without colliding with its geometry. Select the **AutoMerged** check box. With this setting enabled, the player can run through the grass and affect its movement.

   Enabling this setting also makes the **Bending** property ineffective for this group. Instead, the settings for **Stiffness**, **Damping**, and **Variance** now affect the movement of the objects. In this tutorial, the default settings are acceptable, so leave them at default.



   Just as you did with the trees, modify size, rotation, and density settings using the following values, or experiment with your own settings:

   - **Size**: `1.25`

Sets the base size for the objects that you place.

- **+-SizeVar**: `0.2`

  Varies the size of each object (larger or smaller) by up to this amount.

- **RandomRotation**: `Selected`

  Rotates the objects in random directions.

7. Modify the **Density** setting for each of the different grasses using the following values, or experiment with your own. To do this, click each of the grasses in the group to set the **Density** settings individually.

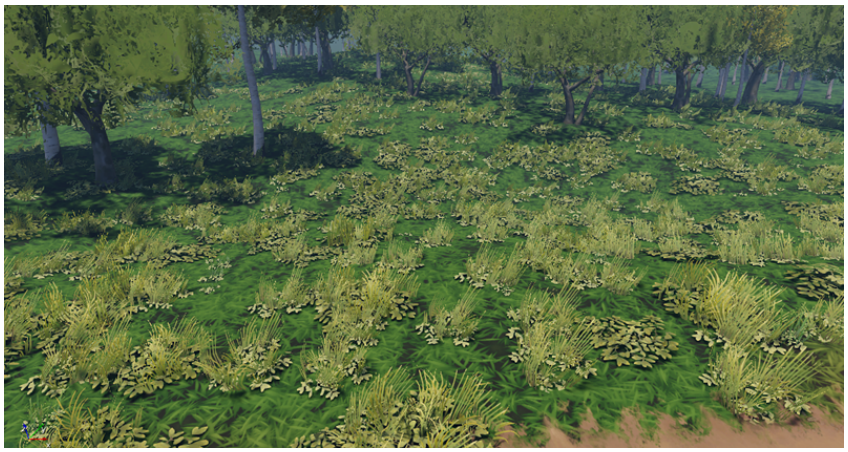
   - **am_grass_01_plain_group**: `16`
   - **am_grass_tuft_04_group**: `10`
   - **am_groundcover_01_group**: `12`
   - **am_river_weed**: `8`

8. Select the **Grass** category.

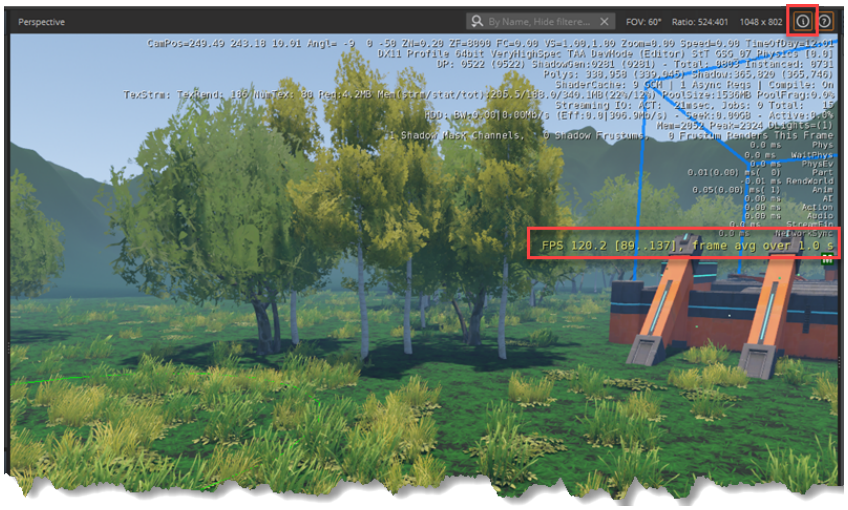
   Set your **Brush Radius** to `8`.

   Paint shrubs and grasses into your scene.

   Note
   It is easy to paint too much grass into a scene, which may cause a lag in performance. Focus primarily around the area that the player walks through. The amount of grass should be similar to the following image.

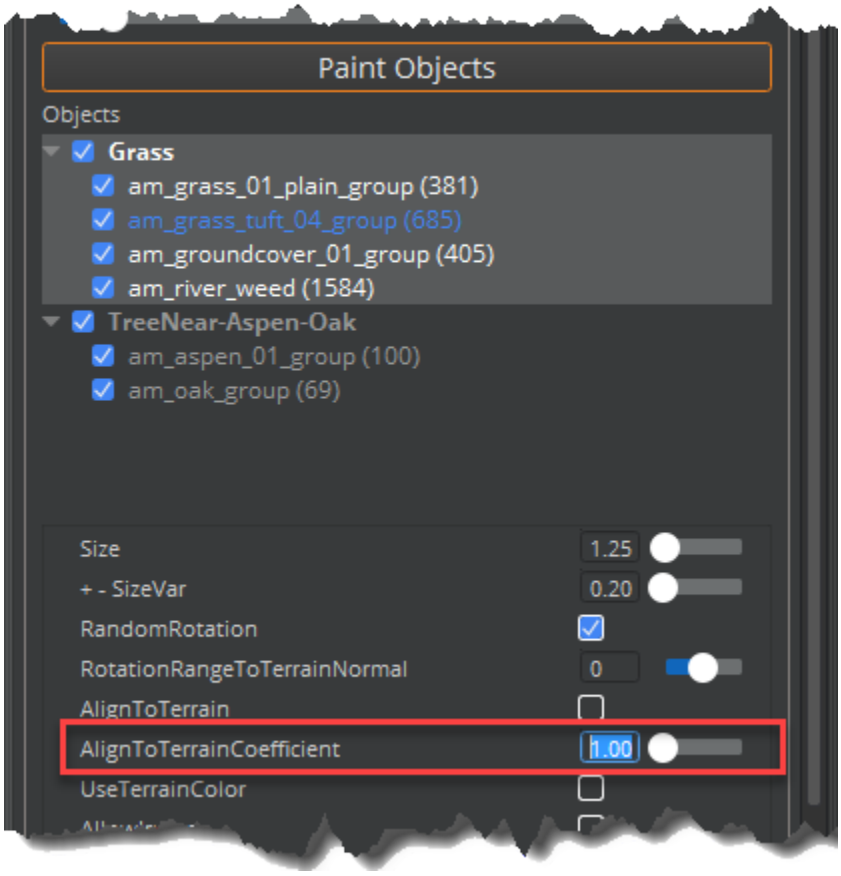
   

9. To check the frame rate in the current scene, repeatedly click the "i" icon in the upper right corner of the viewport until the FPS message and additional render data appear. Ideally, the FPS rate should be over 30 frames per second. If it is below this, you may have too much information in the scene.

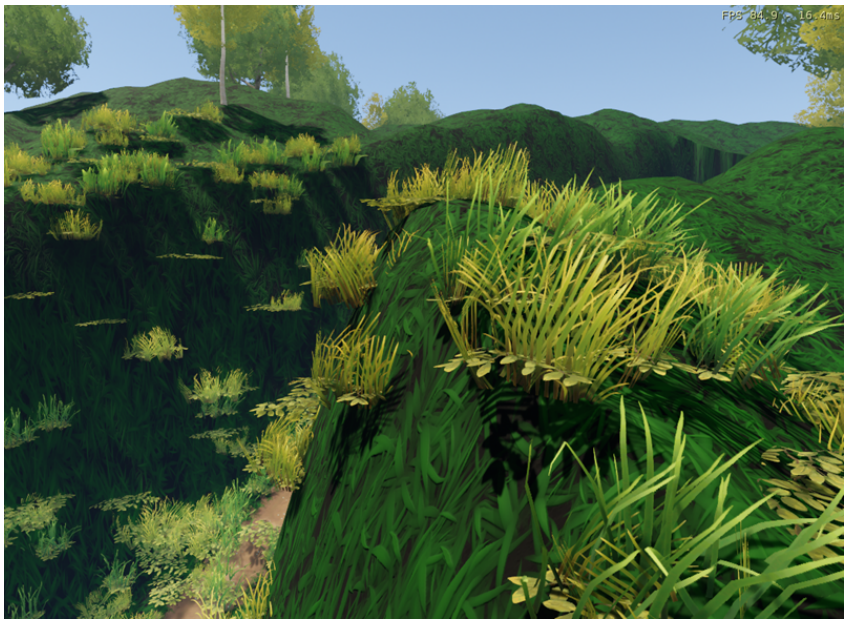


10. For these types of grasses, set the **AlignToTerrainCoefficient** to **`1.0`** to make the objects align to the terrain.

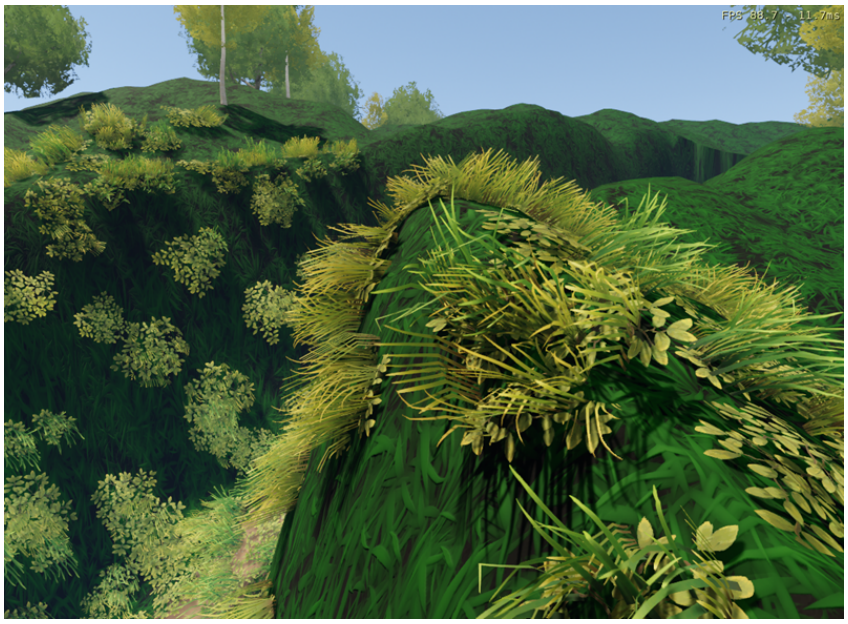
    The **AlignToTerrainCoefficient** controls the alignment of the object with regards to the terrain it is on. A value of **`0`** makes the objects ignore the terrain's angle where it is placed. A value of **`1`**, however, causes the objects to align themselves to the slope of the terrain.



For example, the following image shows the **AlignToTerrainCoefficient** set to **`0`**. Notice how the grasses point skyward even though they are on a steep slope.



The following image shows the **AlignToTerrainCoefficient** set to `1`. Notice how the grasses appear to grow perpendicular to the slope, which appears more natural.



Next: Inserting Rocks and Undergrowth (p. 184)

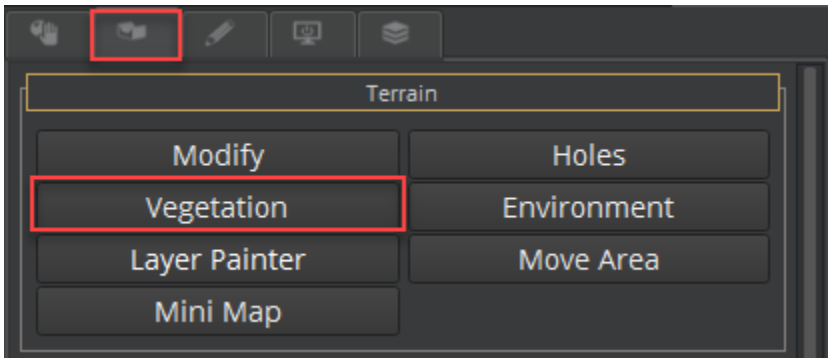
Inserting Rocks and Undergrowth

You can also use the **Vegetation** tool to place individual objects. This is useful when placing large objects such as boulders, tree stumps, or logs.

To better focus on a particular vegetation category, you can temporarily hide groups of vegetation objects. You can also fine-tune the placement of individual objects within a group.

To insert rocks into your landscape

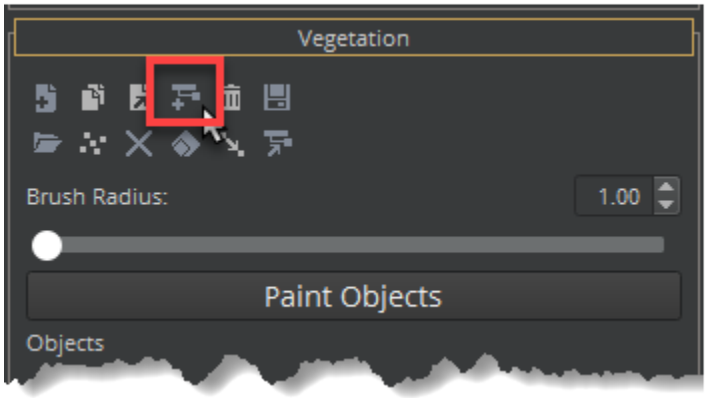
1. If the **Rollup Bar** is not open, open it from the main menu by choosing **Tools**, **RollupBar (LEGACY)**.
2. In the **Rollup Bar**, on the **Terrain** tab, click **Vegetation**.

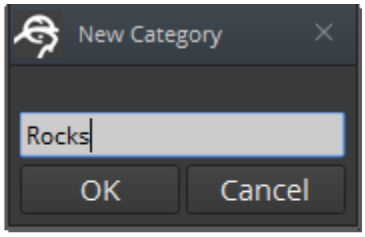


3. Create a new category and name it `Rocks`.

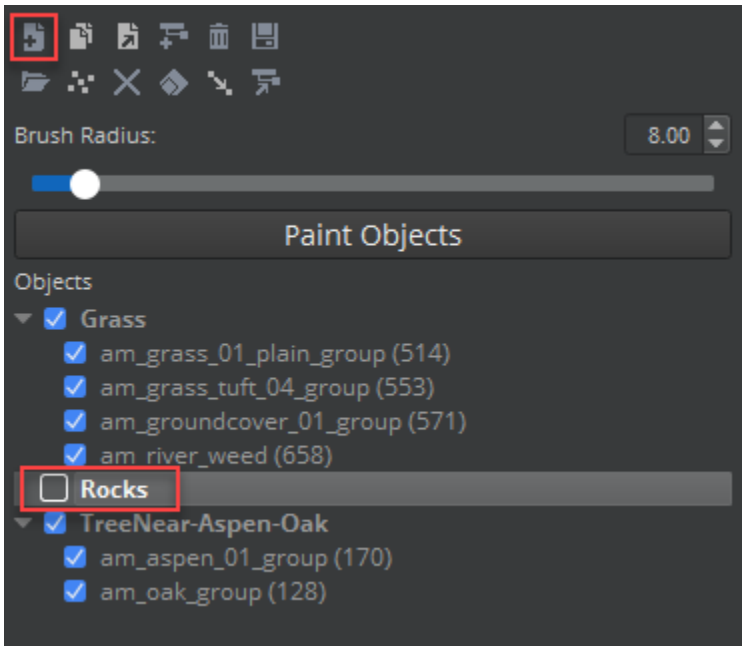
   To do this, under **Vegetation**, click the **Add Vegetation Category** icon.

   In the **New Category** box, type `Rocks`. Click **OK**.



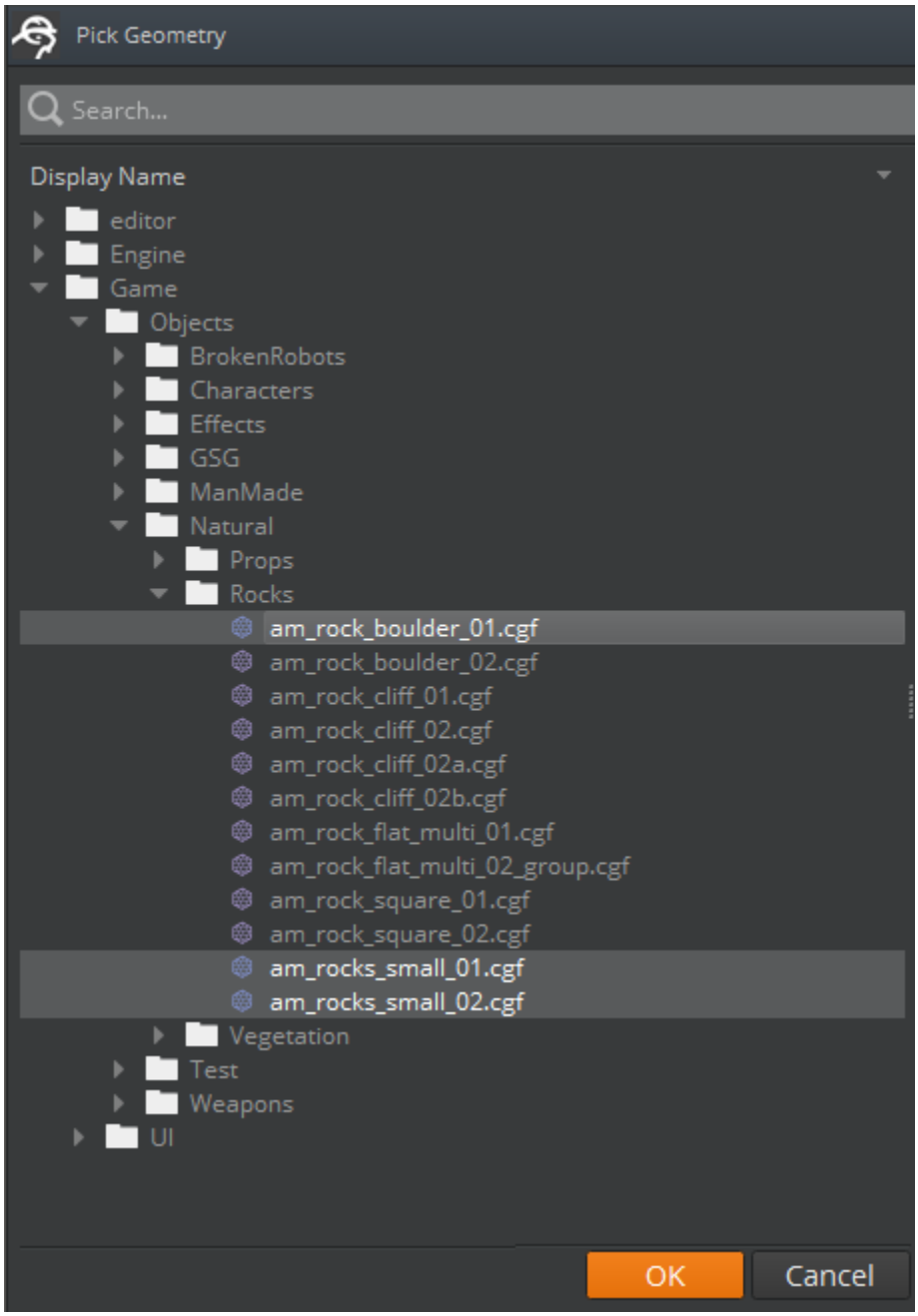


4. Select the **Rocks** object category that you just created, and then click the **Add Vegetation Object**  icon.

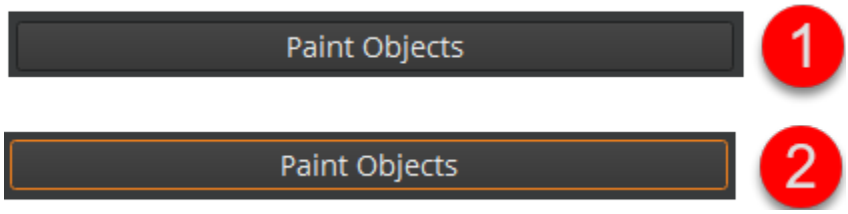


5. In the **Pick Geometry** dialog box, navigate to `Game\Objects\Natural\Rocks`. Hold down **Ctrl** while selecting the following files:

   - `am_rocks_small_01.cgf`
   - `am_rocks_small_02.cgf`
   - `am_rocks_boulder_01.cgf`



6. Select only **am_rocks_boulder_01** and modify the following settings, or experiment with your own.
   - **Size**: `1.50`
   - **+-Sizevar**: `0.20`
   - **RandomRotation**: `Selected`
7. Click **Paint Objects** to disable it. The setting is disabled (1) when it has no orange border. An orange border (2) indicates that the feature is enabled.



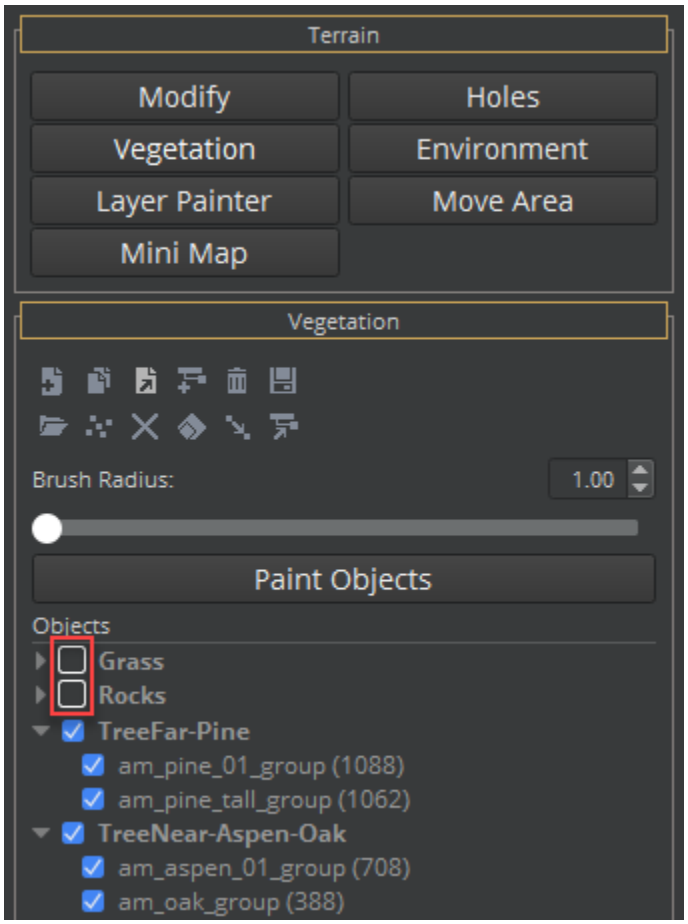
Hold down **Shift** and click once in the viewport. This places a single boulder.

You can place single objects using any object in the vegetation categories. This is useful for placing objects with precision.

You can move individual objects that you previously placed. Because the scene is full of vegetation, it can be difficult to select one particular object. In the vegetation categories and objects lists, you can hide and unhide groups or individual objects.

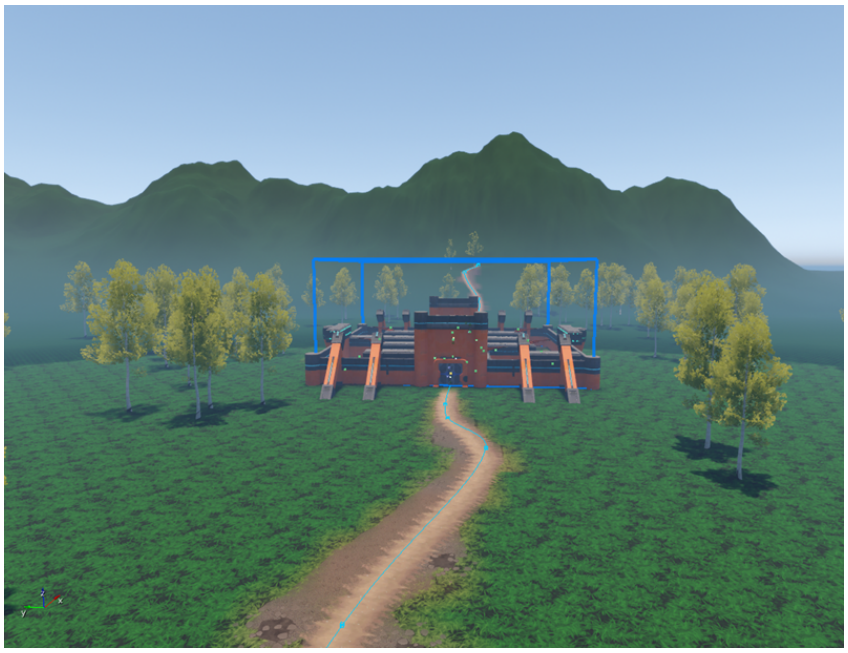
To select individual objects

1. To hide the **Grass** and **Rocks** categories, clear their check boxes in the **Vegetation** tool. This hides all of the grass and rocks in your scene.



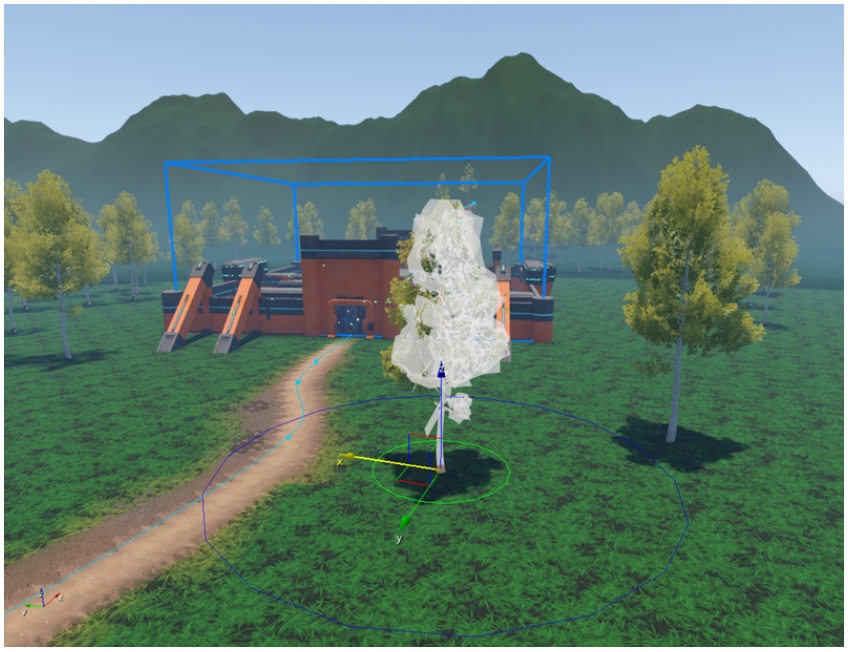
2. In the **TreeNear-Aspen-Oak** category, hide the **am_oak_group**.

   With all categories hidden except **am_aspen_01_group**, your level should look similar to the following image.



3. Make sure **Paint Objects** is still disabled and then click to select one of the trees in the scene.

   Press **2** on your keyboard to select the Move tool. Move the tree to different location.



4. Press **Ctrl+S** to save your level.

Using the techniques you learned in this section, use the **Grass** category to add some undergrowth to the maze pathway. You can do this by setting your **Density** setting to **1** and your **Brush Radius** to **2**, and then painting. Or you can disable the **Paint Objects** button and then hold down **Shift** to place individual plants.

You should now have a lush vegetative environment in your level.

(Optional) You can move on to the next tutorial, which shows you how to add more details to your landscape. Or you can proceed to the Lighting (p. 196) tutorials.

Press **Ctrl+S** to save your level, and then **Ctrl+G** to play your level. Press **Esc** to quit.

Next: Customizing Your Landscape (Optional) (p. 191)

Customizing Your Landscape (Optional)

If you prefer you can add several more vegetation categories. Use the tools and settings you learned to paint in the following landscape elements.

Flowers – Use a high density setting for scattered flowers, or a lower density setting for small pockets of flowers. Navigate to `Game\Objects\Natural\Vegetation`. Use the following files:

- `am_grass_flower_purple_group.cgf`
- `am_grass_flower_red_group.cgf`
- `am_grass_flower_white_group.cgf`

Shrubbery – Place these in small pockets throughout to add detail to the scene. Navigate to `Game\Objects\Natural\Vegetation`. Select `am_bush_01_group.cgf`.

TreeFar-Pine – Place these trees on the mountains and hillsides in the distance. Navigate to `Game\Objects\Natural\Vegetation`. Select the following files:

- `am_pine_01_group.cgf`
- `am_pine_tall_group.cgf`

Next: 9: Enhancing Your Level with Details (p. 192)

9: Enhancing Your Level with Details

You can enhance the look of your level with details that significantly heighten the sense of ambiance and mood in your game.

In this tutorial, you learn how to add some enhancements:

- Decals – Door number decal above the maze's entrance
- Particle effects (FX) – Steam shooting from broken pipes and ambient particles floating in the air
- Distance mesh – Large industrial structures in the distance to create interest

Next: Adding Decals to the Maze (p. 192)

Adding Decals to the Maze

Decals are special texture planes that you can layer on top of mesh assets to give them detail. You use the decals to add a door number decal above the maze's entry door.

1. Open the **Asset Browser** and navigate to `Game/Materials/Decals`. Select `decal_airship_tail_02.mtl`.
2. Drag `decal_airship_tail_02.mtl` into the **Perspective Viewport** just in front of the maze's entry door.

   A *6* appears on the ground, and your entity component automatically has a decal component assigned.
3. In the **Entity Inspector**, rename the decal entity **`door_num_decal`**.
4. In the **Decal** component's properties, under **Decal Settings**, change the **Projection type** to **On Terrain and Static Objects**.

   You can use this setting to adjust the projection display settings of the decal based on its position.

   Move the **door_num_decal** up and down with the **Move** tool to see it fade in and out of the terrain based on distance.
5. Next you map the decal to the space above the entry door using the **Align to Surface** tool. This tool is faster and more precise than the **Move** tool for this purpose.

   In the **Entity Inspector**, unlock the **DoorWay_Parent** group if it is currently locked.
6. With **door_num_decal** still selected, click the **Align to Surface Tool** () on the main toolbar.
7. By default the **Align to Surface Tool** aligns to the terrain only. To align instead to the scene's geometry, hold down **Ctrl** and then click the surface above the door in the **DoorWay_Parent** entity. The decal snaps to that surface.

   You can now move and rotate your decal along that surface. Position it where you like.

   You can also adjust the transparency of the decal. To do this, you can modify the **Decal** component's **Opacity** setting. Or you can move it away from the surface it's aligned to.
8. Use the **Scale** tool to adjust the size of the decal.
9. Press **Ctrl+S** to save your level.

You can find more decals, such as scorch and corrosion decals, in the **Asset Browser** in `Game/Materials/Decals`. Experiment with placing these decals in your maze to give it a damaged and distressed look. To see an example of how you can use these decals, revisit the section called "Opening the Orientation Level" (p. 3).

Next: Using the Particle System to Add Steam (p. 193)

Using the Particle System to Add Steam

The particle system employs a large number of tiny sprites, 3D models, or other graphic objects to simulate certain kinds of fuzzy phenomena. Examples include highly chaotic systems, natural phenomena, or processes caused by chemical reactions. These types of particles are difficult to recreate with conventional rendering techniques.

In this tutorial, you place pipes and use the particle system to add steam erupting from cracks in those pipes.

To place pipes and add steam effects

1. To add pipes to your maze, in the **Asset Browser**, navigate to `Game/Objects/ManMade/Airship/`.

   Drag `am_air_pipes_straight_02.cgf` into the viewport.
2. Use the **Move** tool to position the pipe in the upper half of the inner maze walls, similar to the following image.

   You may need to scale the pipes on the Y axis (p. 29) to make them long enough to span the space between the walls.
3. Next, create an entity to add steam erupting from the pipes. Right-click in the **Perspective** viewport and choose **Create entity**.
4. In the **Entity Inspector**, click **Add component**. Under **Rendering**, click the **Particle** component.
5. In the **Particle** component's properties, next to the **Particle effects library**, click the browse (**...**) button.

   In the **Pick Particle** dialog box, navigate to `Game/libs/particles/`.

   Select `ambientfx.xml` and click **OK**.
6. In the **Entity Inspector**, in the **Particle** component's properties, for **Emitters** select **Damage.SteamDamage** from the particles list.
7. Use the **Move** tool to align the steam particle entity so that it appears to shoot from one of the pipes.

Next: Adding Ambient Particles (p. 193)

Adding Ambient Particles

You can use the particle system to add ambient particles floating through the air. The steps in this procedure are similar to the previous procedure.

To add ambient particles to the air

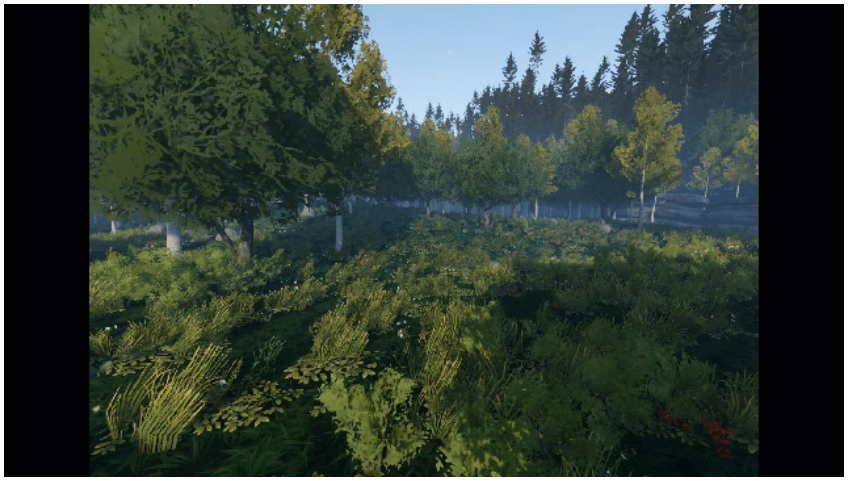
1. On the outside of the maze, near your player character, right-click and choose **Create entity**.
2. In the **Entity Inspector**, click **Add Component**. Under **Rendering**, click **Particle**.

3. In the **Particle** component's properties, next to the **Particle effects library** box, click the browse (**...**) button.

   In the **Pick Particle** dialog box, navigate to `Game/libs/particles/`.

   Select `ambientfx.xml` and click **OK**.
4. In the **Entity Inspector**, in the **Particle** component's properties, for **Emitters** select **FlyingSeeds.FlyingSeeds** from the particles list.

   You now see little seeds floating slowly through the air.



5. Duplicate this entity (**Ctrl+D**) and place them in several locations around the level.
6. Press **Ctrl+S** to save your level.

Next: Adding a Distance Mesh (p. 194)

Adding a Distance Mesh

You can create an enclosed feel to your level by adding a distance mesh. This fills in the empty area in the distance by blocking the view with large industrial structures. This also serves to enclose the game play area.

To add a distance mesh

1. In the **Asset Browser**, navigate to `slices/GSG/`.

   Drag `maze_distance_structure.slice` into your viewport.
2. Move the large wall and buildings to the left of the maze's entrance.

   Adjust the structure so that the walls touch the hillsides on both ends.

   It should look similar to the following image.
3. (Optional) You can add even more details and visual interest. Experiment with adding the following assets:

   - Additional pipes and steam leaks.
   - Vines that dangle from the walls: `slices/GSG/maze_ivy_01.slice`.
   - Crates in the maze path.
   - Destroyed robots: `Game/Objects/BrokenRobots`.

- Any interesting object, decals, and particles that you find in the asset library.

Have fun!

4. Press **Ctrl+S** to save your level.
5. Press **Ctrl+G** to play your level. When you are ready to exit, press **Esc**.

Next: 10: Lighting the Environment (p. 196)

10: Lighting the Environment

To implement global lighting for an entire level, you use a global environment probe. Environment probes control many aspects of the physically based lighting in Lumberyard, including accurate shadow colors, ambient diffuse values, and specular reflections. Probes also provide bounce lighting by taking the colors from the surroundings and applying them directly to the diffuse color of materials inside their radius.

You can also affect the environment's light level by changing the time of day. With the **Time of Day** editor, you can change the sun's position in the sky by specifying any time of day or night. For advanced usage, you can even script times of day to fixed points in time or animate day-to-night cycles at customized speeds.

In this tutorial, you learn a simple method of changing the current time of day. For more information about creating time of day effects, see Creating Time of Day Sky Effects in the *Amazon Lumberyard User Guide*.

Next: Placing a Global Environment Probe (p. 196)

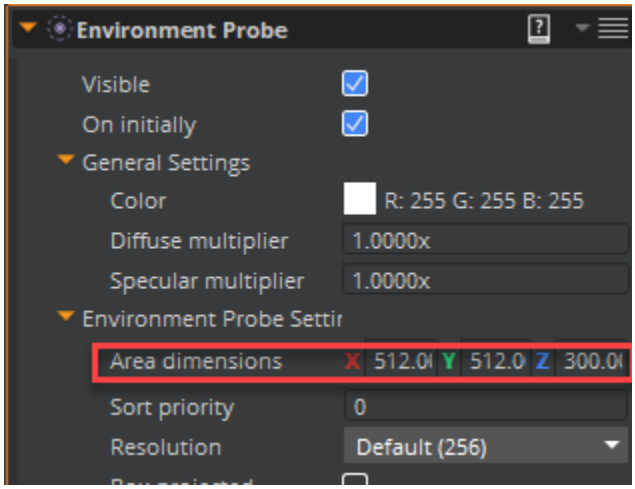
Placing a Global Environment Probe

When you place a global environment probe, the first thing you notice is that the shadows cast by objects look softer and contrast less with the environment.

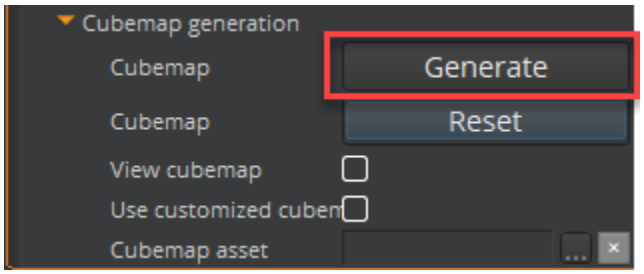
To place an environment probe

1. In the viewport near the center of the level, close to the maze, right-click and choose **Create entity**.
2. In the **Entity Inspector**, name the entity **EnvironmentProbe**.
3. In the **Entity Inspector**, click **Add Component**. Under **Rendering**, click **Environment Probe**.
4. In the environment probe's settings, modify the following values for the **Area dimensions**. Area dimensions control the size of the volume where lighting is applied.

   - **X**: `512`
   - **Y**: `512`
   - **Z**: `300`



5. Under **Cubemap generation**, click **Generate**.



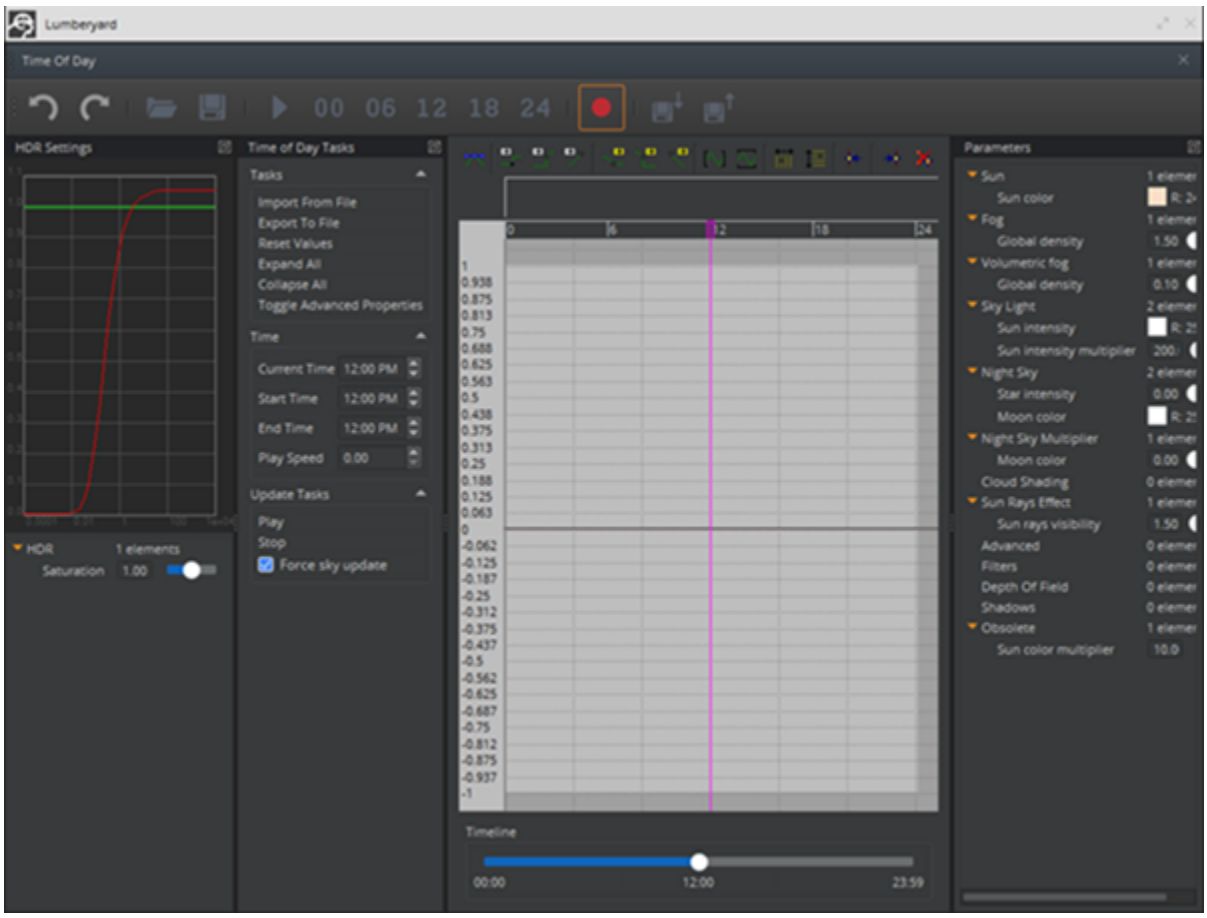
6. Press **Ctrl+S** to save your level.

Next: Changing the Time of Day (p. 197)

Changing the Time of Day

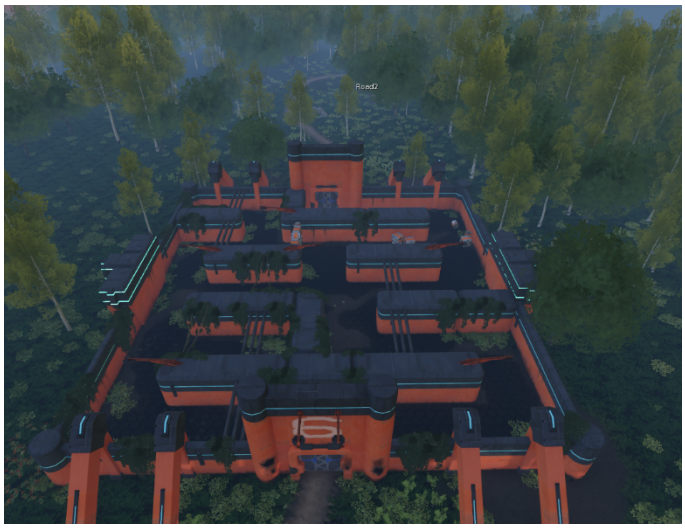
With the **Time of Day** editor, you can change the sun's position in the sky and set day or night.

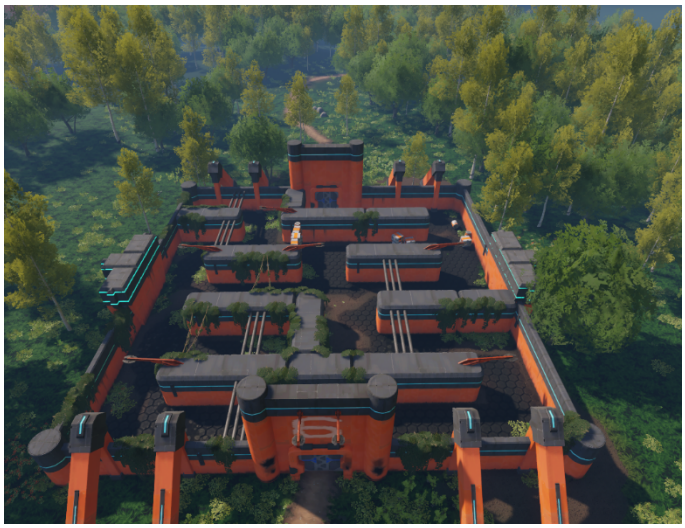
1. Open the **Time of Day** editor by choosing **Tools**, **Other**, **Time Of Day**.
2. At the bottom of the editor window is a section called **Timeline**.



Drag the handle to see the sun change positions and lighting.

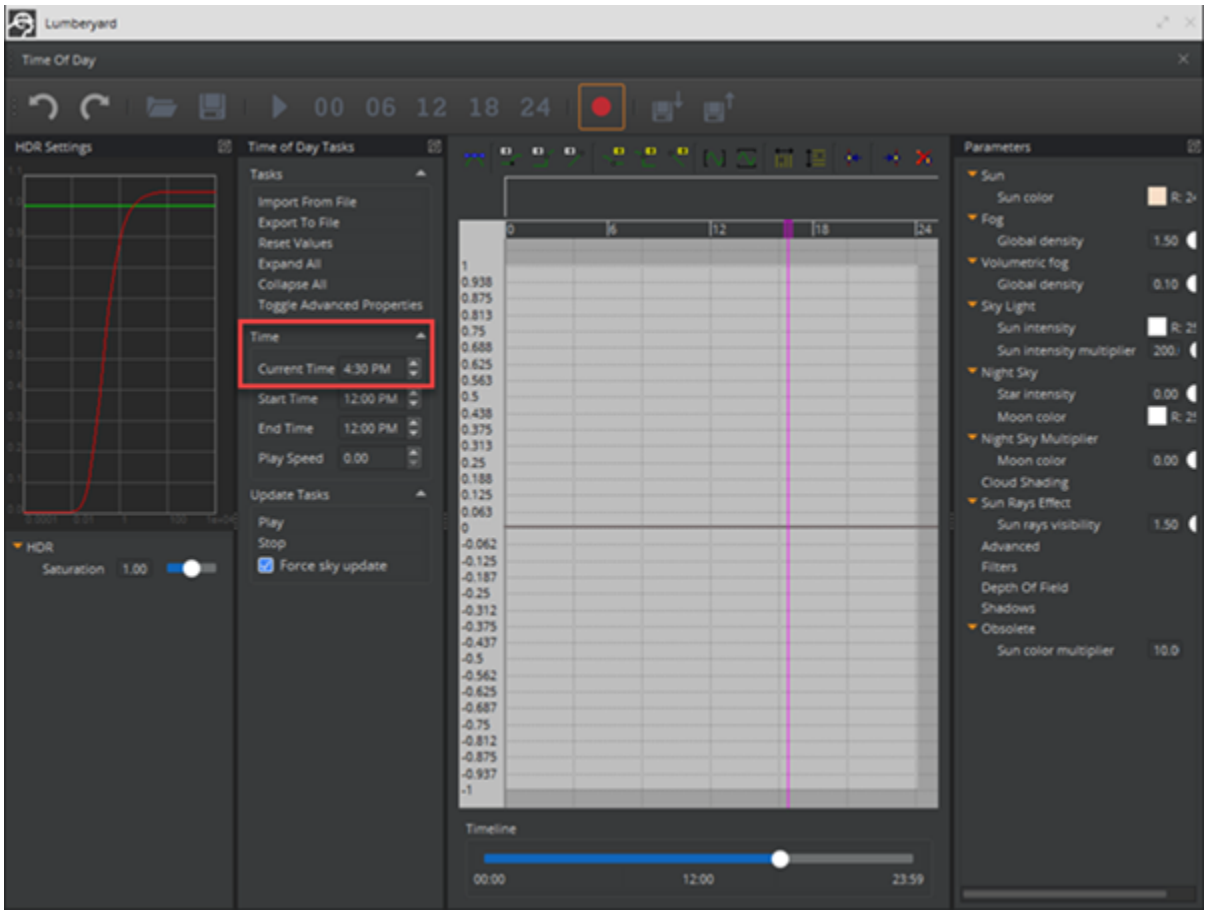
The default setting is 12:00 noon. In the following image the time is set to 6:00 a.m. In the image below that, the time is set to 3:00 p.m. Notice the differences in lighting.





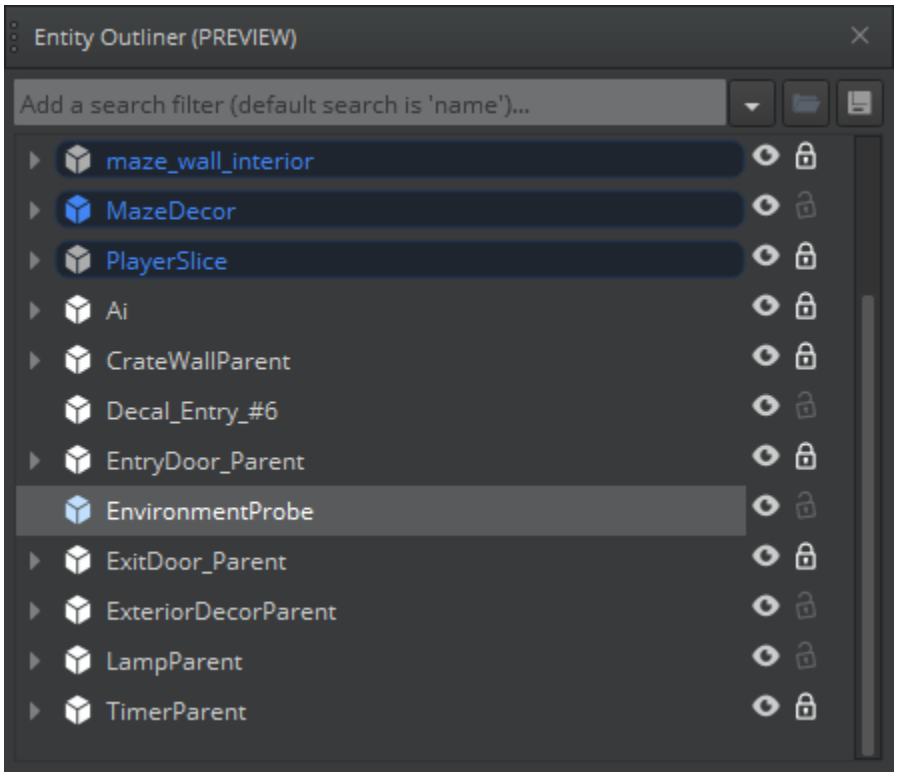
3. For this tutorial, you set the time to 4:30 p.m. To do this, under **Time**, for **Current Time**, specify **`4:30 PM`**.

   This gives you a rosy sunset glow for your scene.

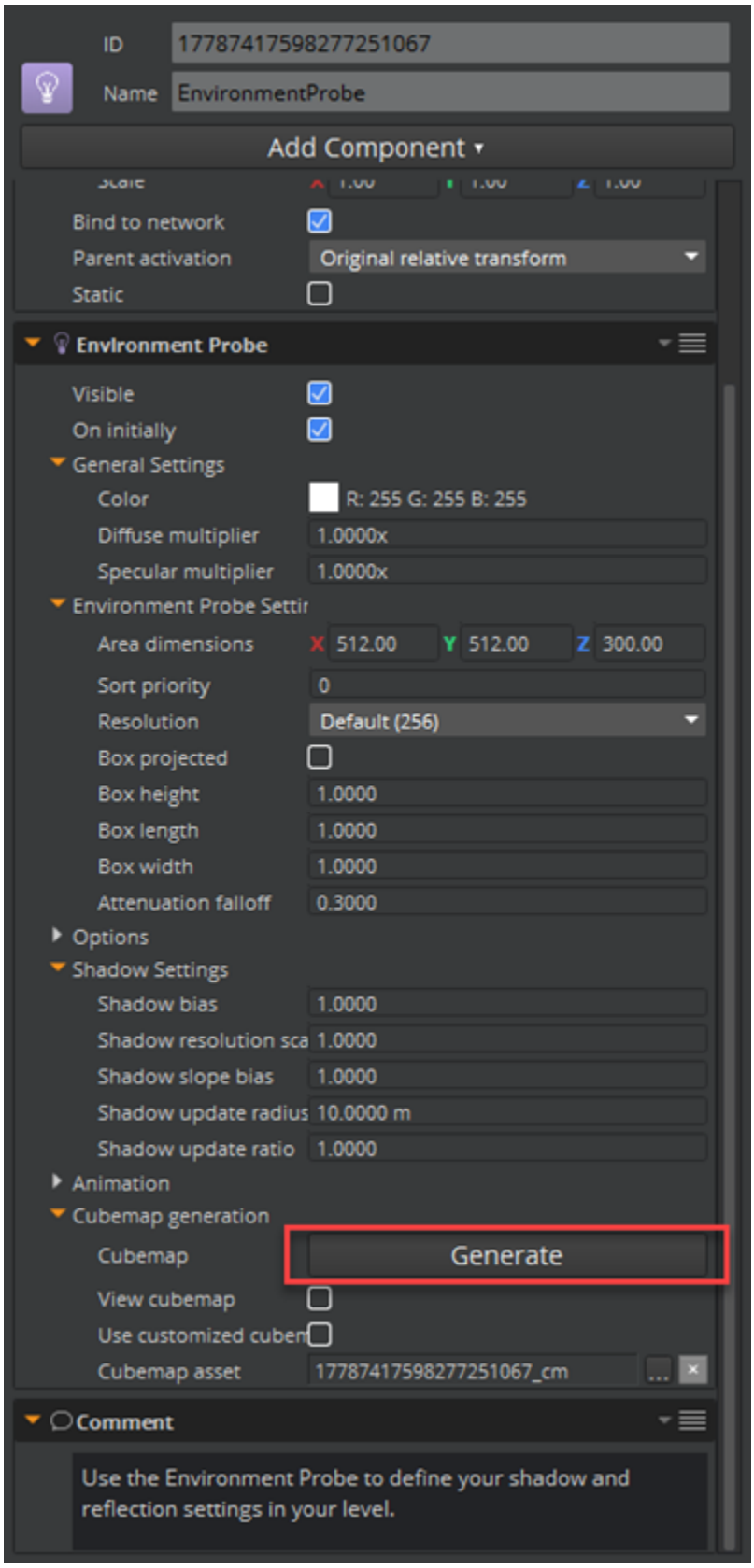


4. Because you changed the time of day, you should re-render your environment probe map in keeping with best practices.

   To do this, in the **Entity Outliner**, select the **EnvironmentProbe** entity.



5. In the **Entity Inspector** on the **Environment Probe** component, under **Cubemap Generation**, next to **Cubemap**, click **Generate**.


ID
17787417598277251067
Name
EnvironmentProbe
Add Component
Bind to network
Parent activation
Original relative transform
Static
Environment Probe
Visible
On initially
General Settings
Color
R: 255 G: 255 B: 255
Diffuse multiplier
1.0000x
Specular multiplier
1.0000x
Environment Probe Settir
Area dimensions
X 512.00
Y 512.00
Z 300.00
Sort priority
0
Resolution
Default (256)
Box projected
Box height
1.0000
Box length
1.0000
Box width
1.0000
Attenuation falloff
0.3000
Options
Shadow Settings
Shadow bias
1.0000
Shadow resolution sca
1.0000
Shadow slope bias
1.0000
Shadow update radius
10.0000 m
Shadow update ratio
1.0000
Animation
Cubemap generation
Cubemap
Generate
View cubemap
Use customized cuben
Cubemap asset
17787417598277251067_cm
Comment
Use the Environment Probe to define your shadow and reflection settings in your level.

This ensures that your cubemap is rendered at the right time of day to get the best visual quality. Small hour to hour increments are not very noticeable in terms of visual differences that occur with cube maps. However, if you compare a 12:00 p.m. (noon) map with a 12:00 a.m. (midnight) map, you notice a significant change in the lighting.

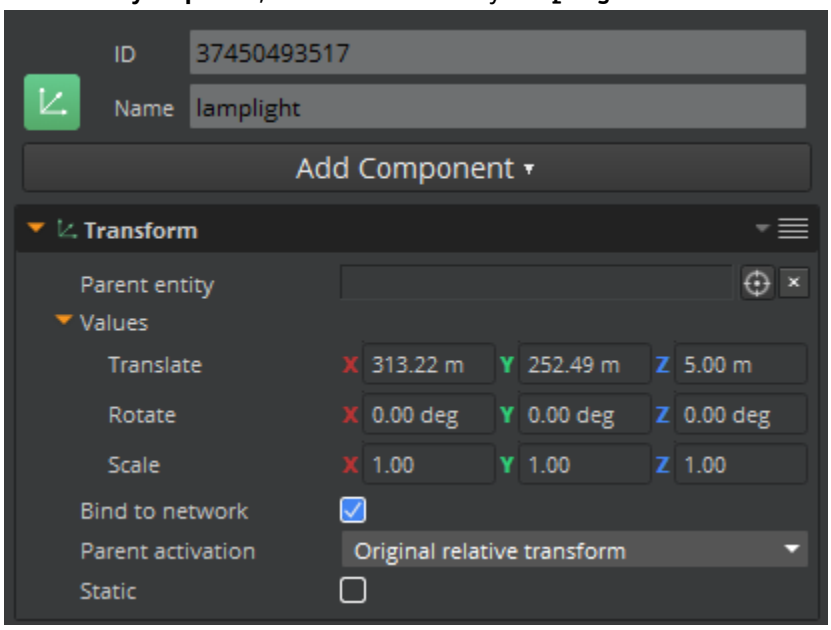
Next: Creating a Light Source (p. 202)

Creating a Light Source

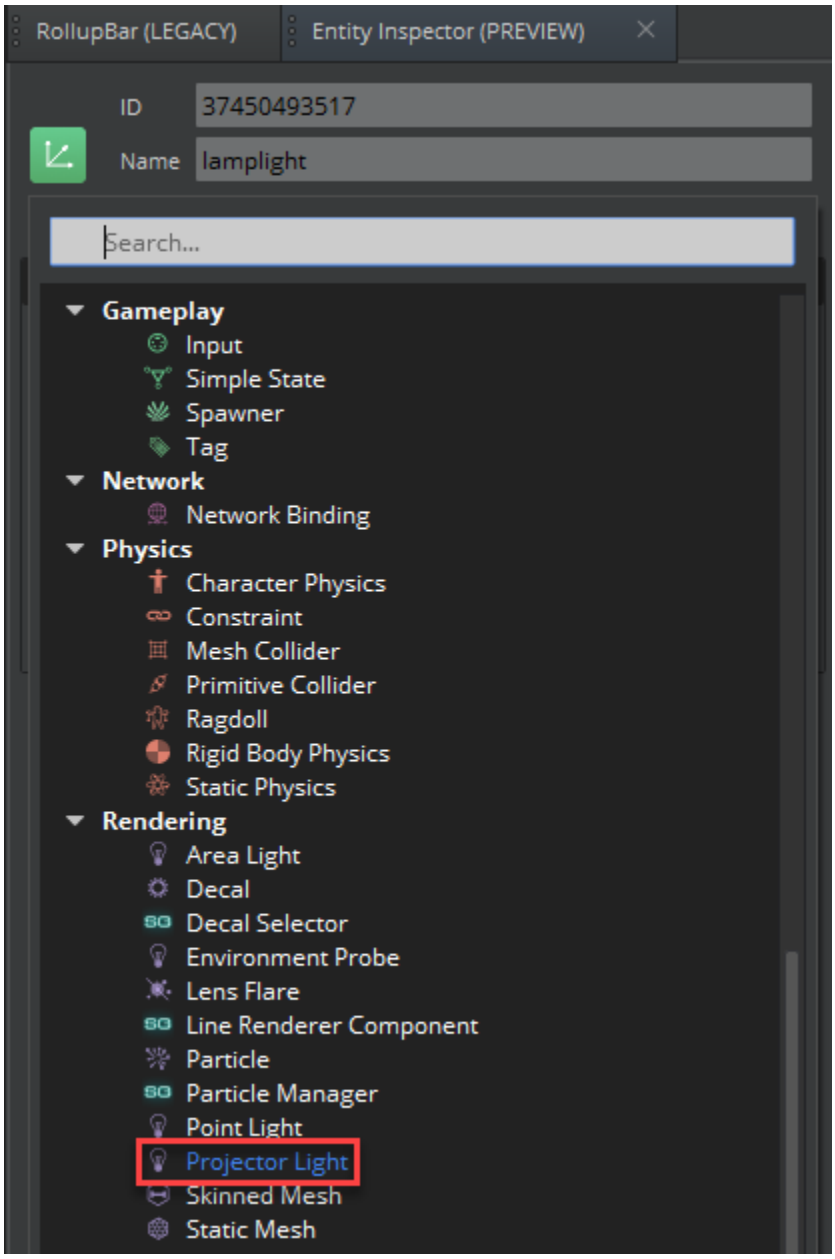
Currently, the lamps in the level don't cast any light. To fix this, you create a light source for the lamps in the maze.

1. In the **Perspective Viewport**, navigate near any lamp post in your level. Right-click and choose **Create entity**.

2. In the **Entity Inspector**, name the new entity **`lamplight`**.

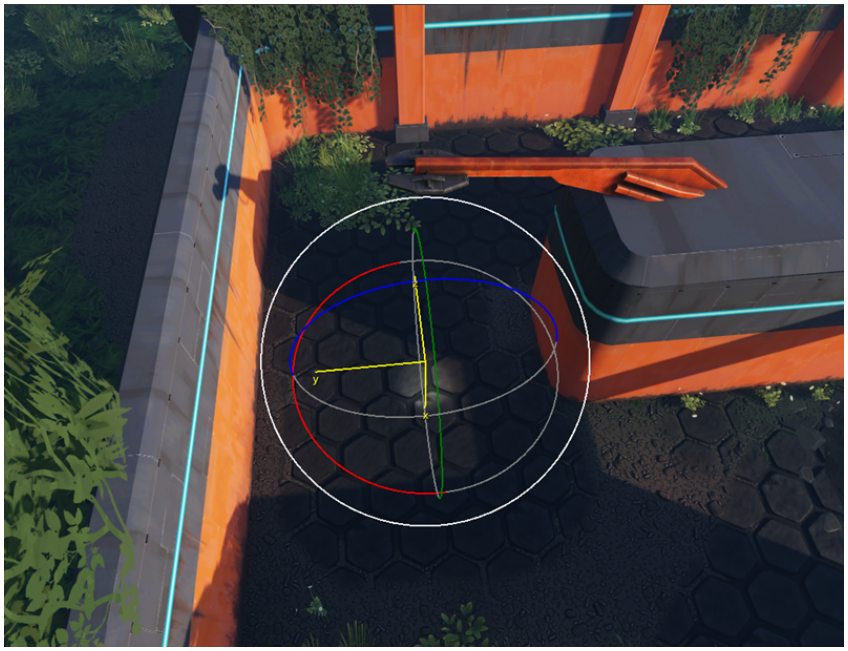


3. Still in the **Entity Inspector**, click **Add Component**. Under **Rendering**, click **Projector Light**.

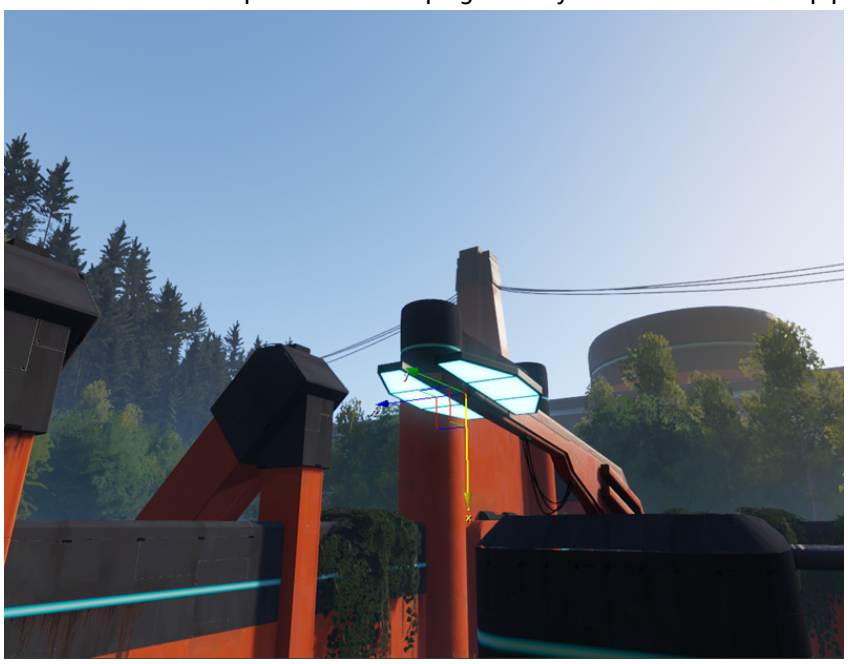


You will notice that components also exist for an area light and a point light, but for this tutorial you use a projection light.

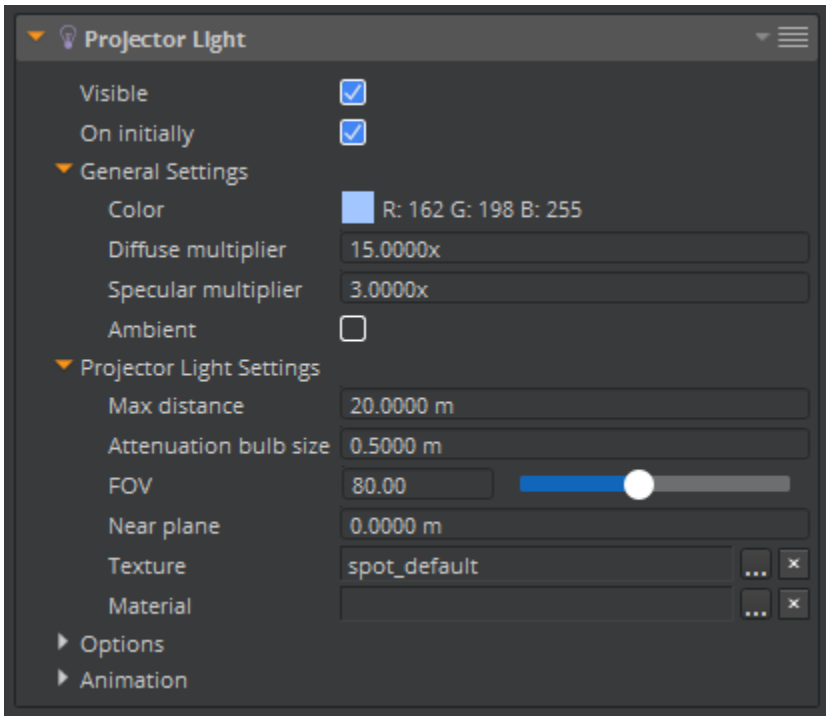
4. In the **Perspective Viewport**, use the **Rotate** tool to rotate the lamp light entity so it points down **Y: 90.00 degrees.**



5. Use the **Move** tool to position the lamp light entity under one of the lamp posts.

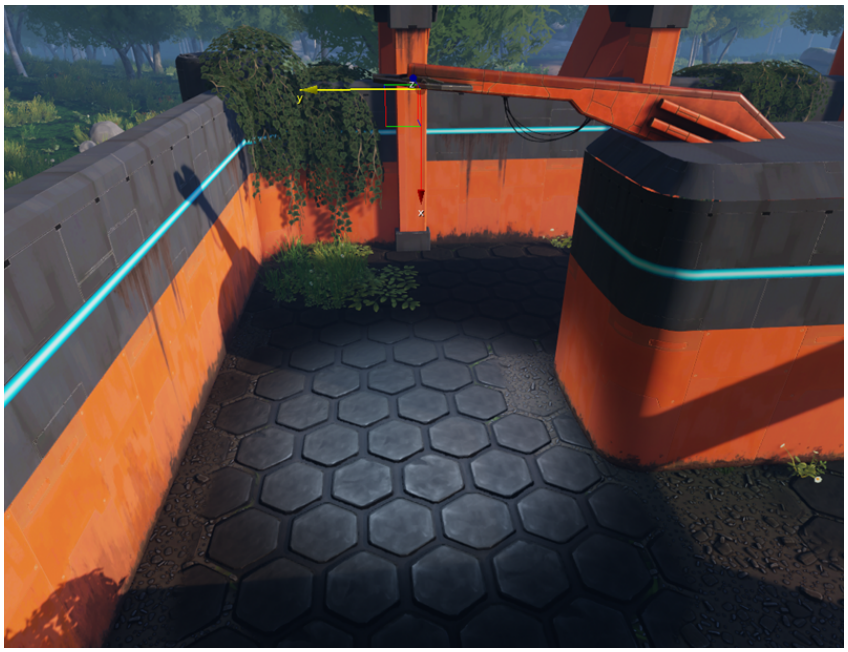


6. In the **Entity Inspector**, use the following settings for the **Projector Light** properties:



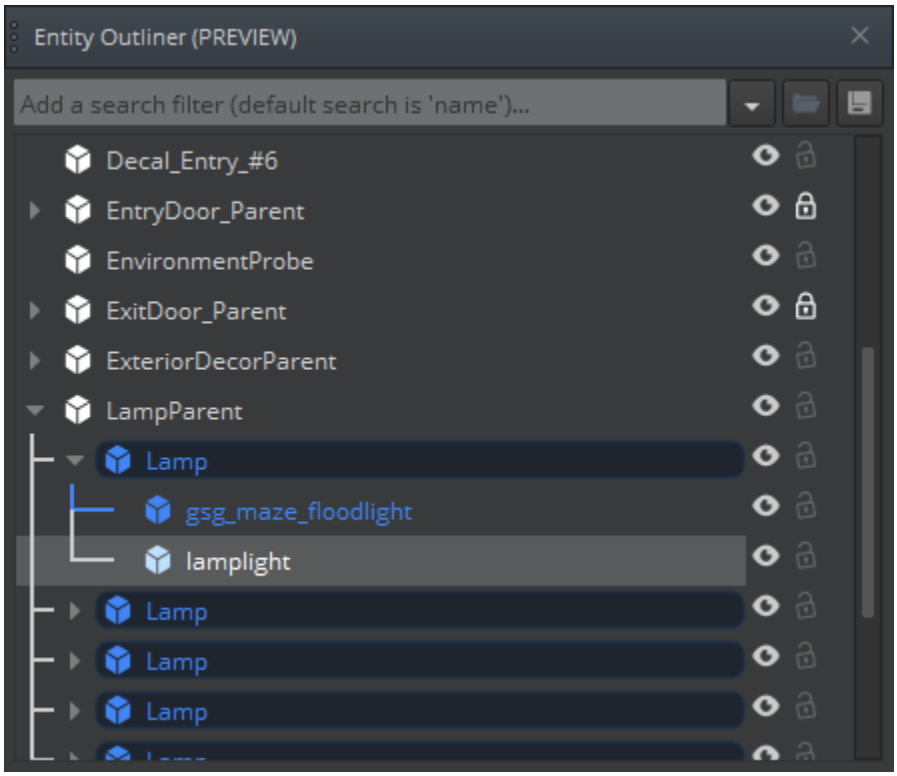
- **Color**: R:`162`, G:`198`, B:`255`
- **Diffuse Multiplier:** `15`
- **Specular Multiplier:** `3`
- **Max Distance:** `20`
- **Attenuation bulb size:** `0.5`
- **FOV**: `80`

Experiment with these settings to find a brightness, color, and size that you like. Your lamp post should look similar to the image below.

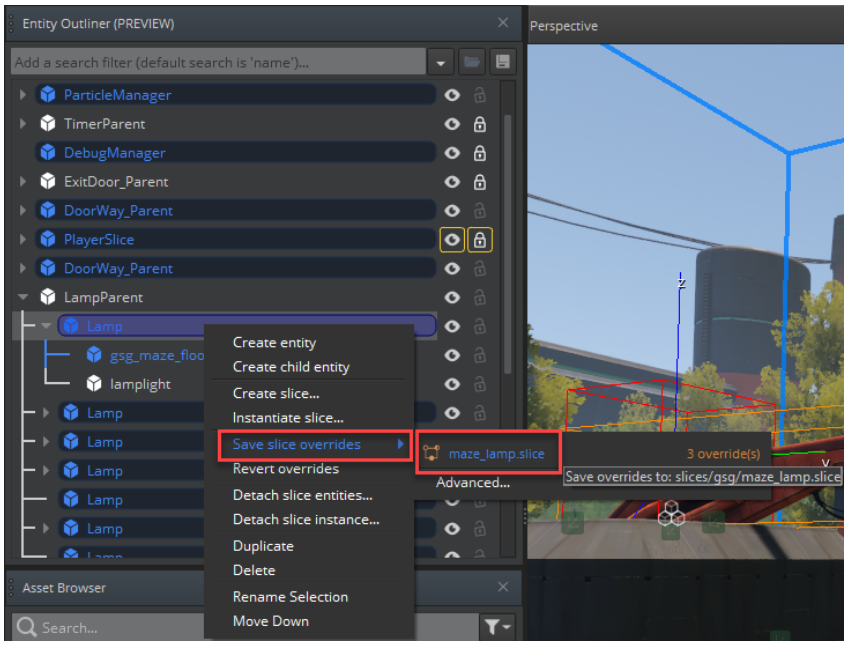


7. Now that the light is positioned and set under the lamp, you can move the light entity into the **Lamp** slice and then save this change to the slice.

   In the **Entity Outliner**, drag the **lamplight** entity into the **`Lamp`** parent slice that you have been working near.



8. Now that the **lamplight** is in one of the lamp entities, under the **LampParent** group, right-click the lamp entity and choose**Save slice overrides**, **maze_lamp.slice**. This saves your changes to the source slice.



9. All the lamp posts in your scene should now be casting light.

10. Press **Ctrl+S** to save your level.
11. Press **Ctrl+G** to play your level. When you are ready to exit, press **Esc**.

Next: 11: Setting up the UI (p. 209)

11: Setting Up the UI

In this tutorial, you'll set up your game's UI. To do this, you'll use a prebuilt slice to set up a title screen and a "mission complete" screen.

To set up the game UI

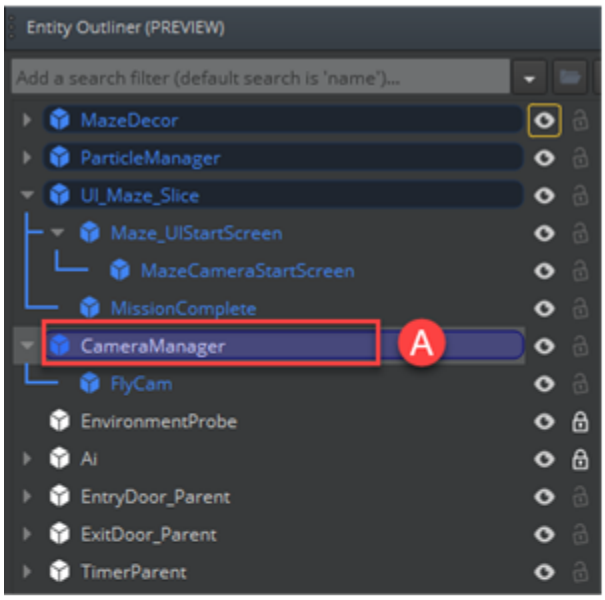
1. In the **Asset Browser**, navigate to `Game\Slices\GSG`.

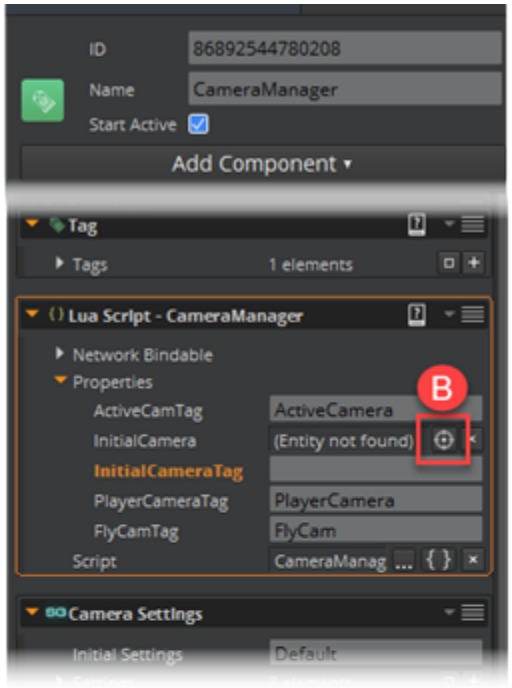
   Drag `maze_ui.slice` into the viewport.
2. Align the slice for this level's purposes. To do this, make sure either the **maze_wall_exterior** or **maze_wall_interior** parent entity is visible.

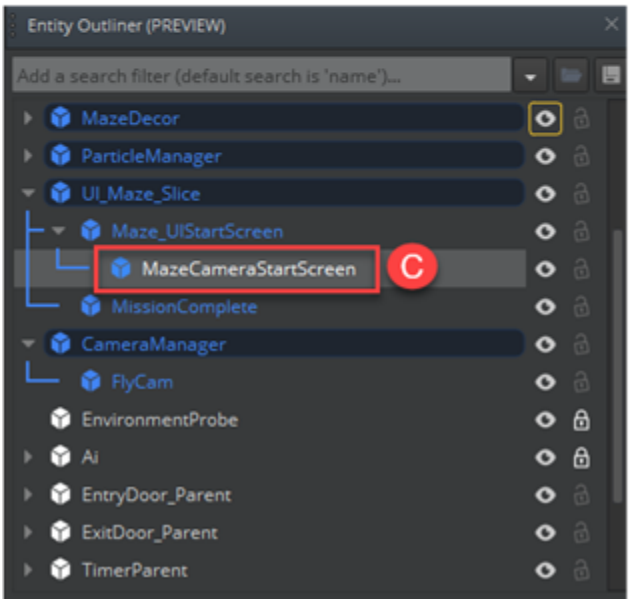
   In the viewport or the **Entity Outliner**, select `UI_Maze_Slice`.

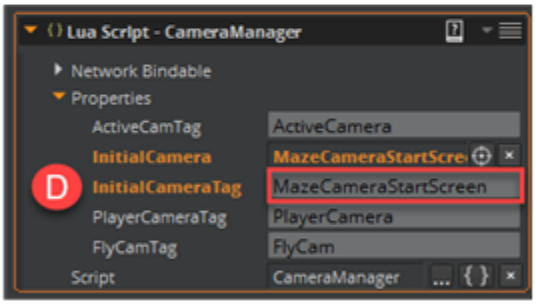
   Click the **Align to Object** icon  , and then click either **maze_wall_exterior** or **maze_wall_interior**.
3. Set the **CameraManager**'s starting camera to the UI slice's camera. To do this:

   - A – In the **Entity Outliner**, select **CameraManager**.
   - B – In **Entity Inspector**, under **Lua Script - CameraManager**, next to the **InitialCamera** box, click the picker icon  .
   - C – In **Entity Outliner**, under **UI_Maze_Slice**, choose **MazeCameraStartScreen** (C).
   - D – In **Entity Inspector**, under **Lua Script - CameraManager**, in the **InitialCameraTag** box, type **`MazeCameraStartScreen`** (D).









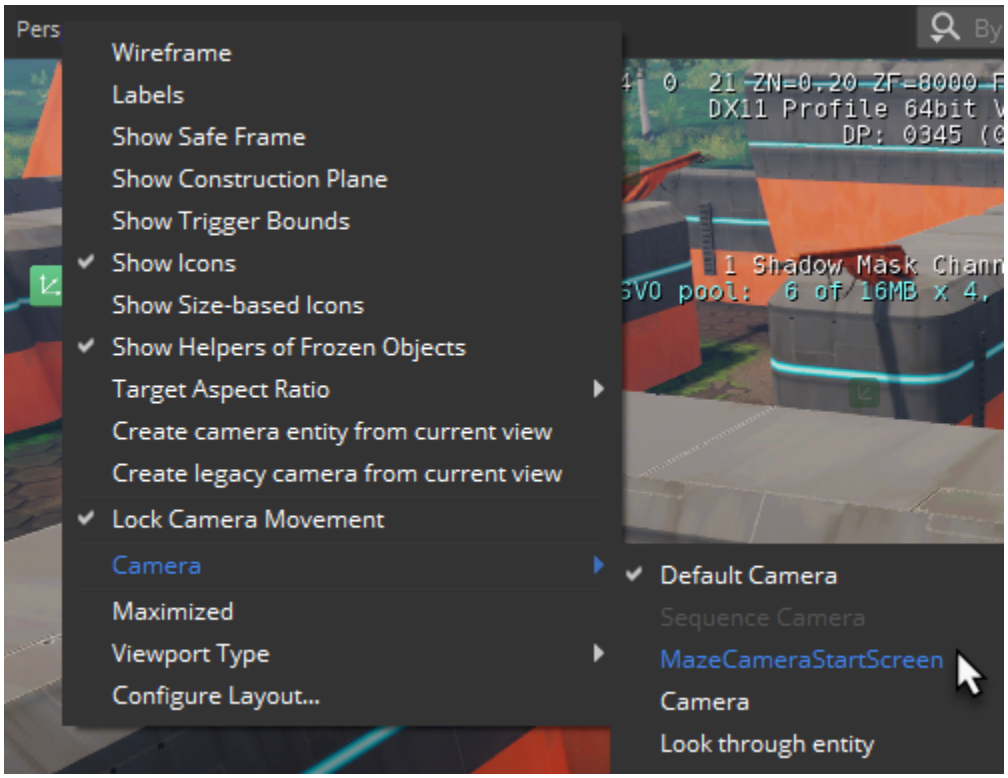
To see the title screen and test your camera, press **Ctrl+G** to play the game.

Press **Esc** to quit.

4. Move the title screen camera to a position that you like for the start of the game. To do this, in the **Entity Outliner**, expand the **UI_Maze_Slice** parent entity.

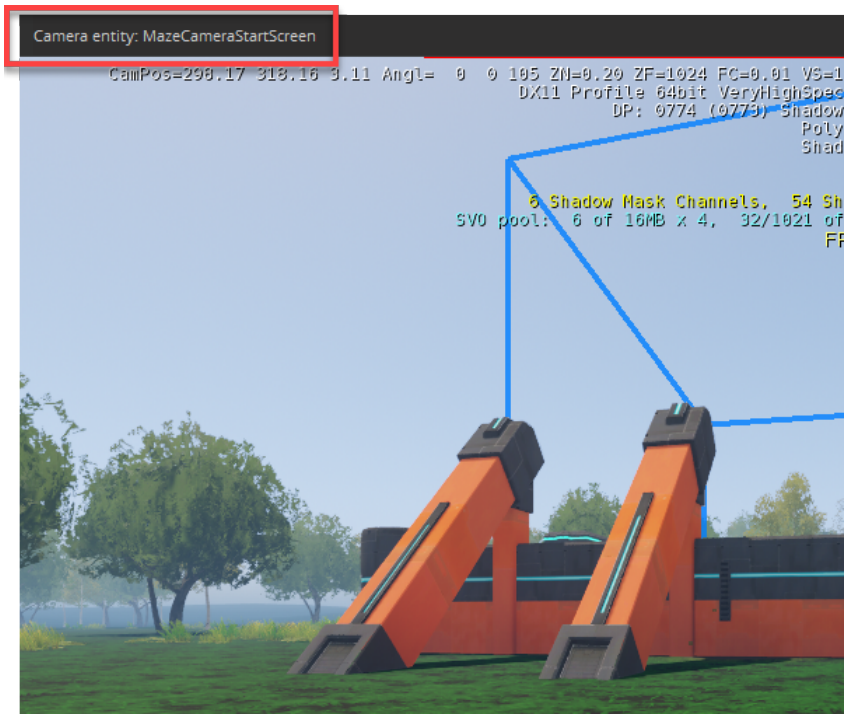
   Select the **MazeCameraStartScreen** entity.

   In the viewport's upper left corner, right-click the **Perspective** title bar. Choose **Camera**, **MazeCameraStartScreen**.



Notice that your view changes when you click this option. That's because you are now viewing the level through this particular entity.

Using viewport navigation tools, position the camera with the view that you want the player to see when starting the game.



5. When you have the view that you want, switch back to the default camera. To do this, right-click the viewport (**Perspective**) title bar and choose **Camera**, **Default Camera**.
6. Press **Ctrl+S** to save your level.
7. Press **Ctrl+G** to play your level. Notice that the start screen camera view starts where you positioned it, and then switches to the third person player camera when you press a key to start.

   Press **Esc** to quit.

Next: 12: Exporting Your Game (p. 213)

12: Exporting Your Game

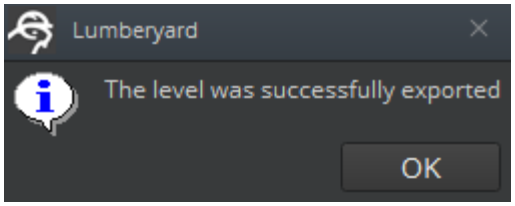
Exporting an executable file for your game is a critical steps in game creation.

You'll export your game and then play it as an executable file. Exporting your game packages up the data files so that you can run the level file as a standalone level.

To export and run your game

1. From the Lumberyard Editor main menu, choose **Game**, **Export to Engine**.

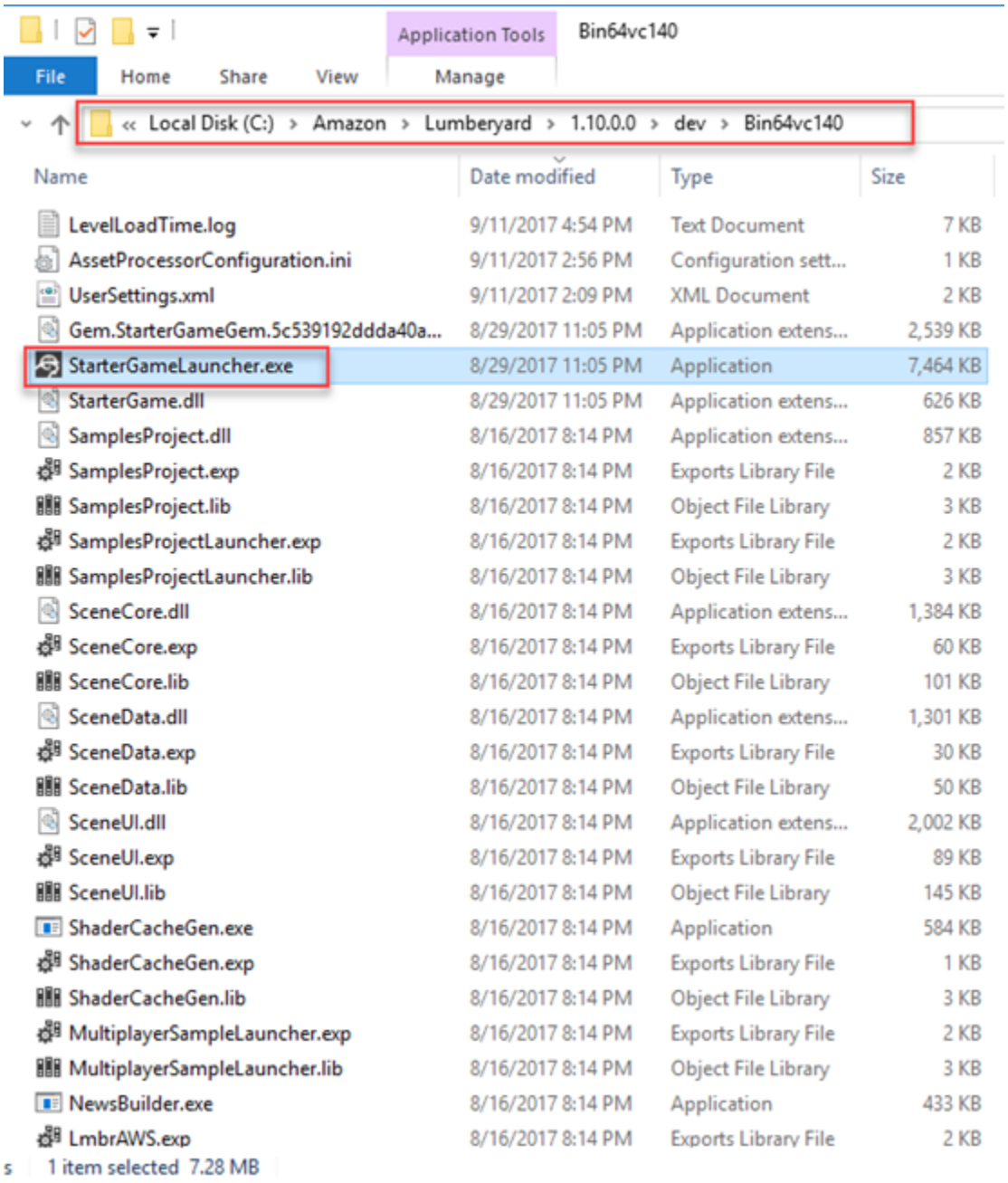
   This creates a launcher file for your level. When the process is finished, a success message is displayed.

2. Open a file manager such as Windows Explorer to find your game launcher, `StarterGameLauncher.exe`. The launcher is saved in the same directory as the Lumberyard editor executable.

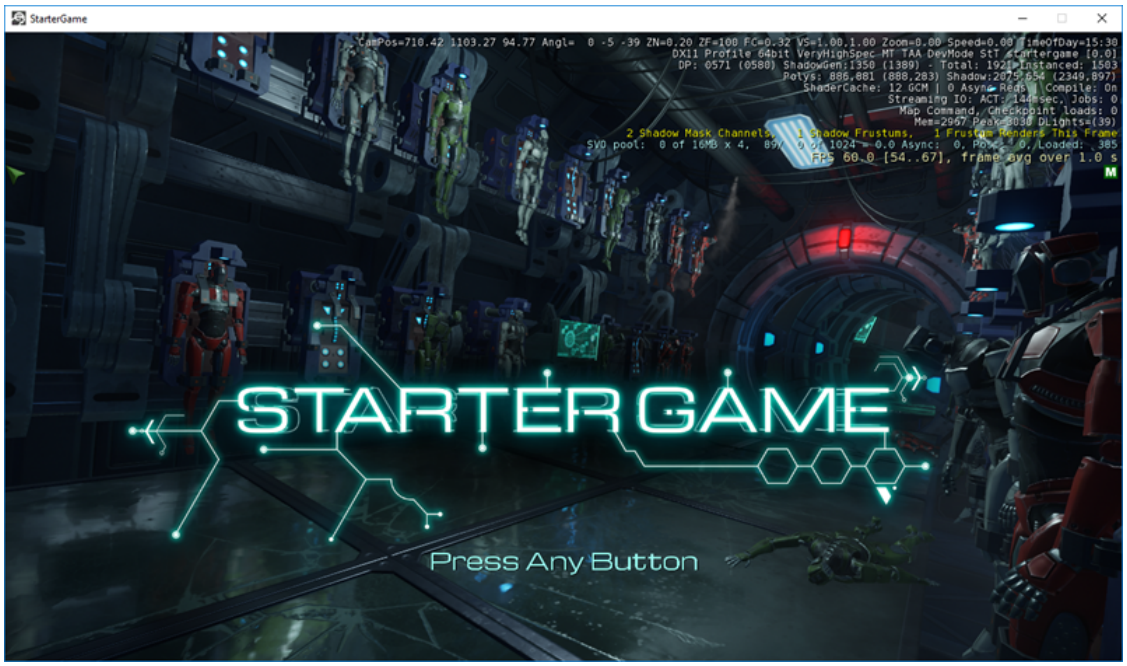
   For example:

   - If you run Visual Studio 2015, this might be `C:\Amazon\Lumberyard\1.x.x.x\dev\Bin64vc140`.
   - If you run Visual Studio 2013, this might be `C:\Amazon\Lumberyard\1.x.x.x\dev\Bin64vc120`.



3. Run `StarterGameLauncher.exe`.

   The game opens in a game window to the default level for the project.



4. Press the tilde (~) key on your keyboard to open the console for your game.

   Type **`map <your level name>`**. For example, **`map MyFirstLevel`**. Press **Enter**.

5. Play and enjoy the game that you just created.

   To quit, press tilde (**~**) again to open the console.

   Type **`quit`**.

Any changes that you make to your game in Lumberyard Editor require you to export your level again in order to see those changes when you use the game launcher.

Accessing Documentation

The Lumberyard Documentation team is continuously writing and improving the official documentation to provide a better help experience:

- Lumberyard online documentation
- Lumberyard tutorials

You can also refer to the `docs` folder in the Lumberyard directory for help topics about using Lumberyard.

Contacting Support

Amazon Web Services provides a combination of tools and expertise to help support your success with Lumberyard. To learn about the variety of resources we offer, see Amazon Lumberyard.

It's day one and we're just getting started. We look forward to your feedback.