# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Editor Overview</td>
<td>3</td>
</tr>
<tr>
<td>New Level Creation</td>
<td>4</td>
</tr>
<tr>
<td>Lumberyard Objects</td>
<td>5</td>
</tr>
<tr>
<td>Editor Layout</td>
<td>6</td>
</tr>
<tr>
<td>Essential Tools</td>
<td>7</td>
</tr>
<tr>
<td>Select, Move, Rotate, and Scale</td>
<td>7</td>
</tr>
<tr>
<td>Snap Grid</td>
<td>11</td>
</tr>
<tr>
<td>Snap Angle</td>
<td>11</td>
</tr>
<tr>
<td>Follow Terrain and Snap to Objects</td>
<td>12</td>
</tr>
<tr>
<td>Display Options and Settings</td>
<td>15</td>
</tr>
<tr>
<td>Perspective Viewport Options</td>
<td>15</td>
</tr>
<tr>
<td>Show/Hide Helpers</td>
<td>16</td>
</tr>
<tr>
<td>Toggle Display Information</td>
<td>16</td>
</tr>
<tr>
<td>Navigation Speed Settings</td>
<td>17</td>
</tr>
<tr>
<td>AI/Physics Toggle</td>
<td>17</td>
</tr>
<tr>
<td>Layers</td>
<td>17</td>
</tr>
<tr>
<td>Collaboration with Multiple Users</td>
<td>18</td>
</tr>
<tr>
<td>Moving Assets Between Layers</td>
<td>19</td>
</tr>
<tr>
<td>Auto Backup</td>
<td>20</td>
</tr>
<tr>
<td>Object Placement</td>
<td>22</td>
</tr>
<tr>
<td>Place Brushes</td>
<td>22</td>
</tr>
<tr>
<td>Prefabs</td>
<td>25</td>
</tr>
<tr>
<td>Terrain</td>
<td>29</td>
</tr>
<tr>
<td>Terrain Painting</td>
<td>29</td>
</tr>
<tr>
<td>Terrain Height</td>
<td>38</td>
</tr>
<tr>
<td>Terrain Vegetation</td>
<td>40</td>
</tr>
<tr>
<td>Lighting</td>
<td>45</td>
</tr>
<tr>
<td>Environment Probes</td>
<td>45</td>
</tr>
<tr>
<td>Time of Day</td>
<td>48</td>
</tr>
<tr>
<td>Lights</td>
<td>50</td>
</tr>
<tr>
<td>Camera and Player Start</td>
<td>57</td>
</tr>
<tr>
<td>Place a Game Camera</td>
<td>57</td>
</tr>
<tr>
<td>Switch to Game Mode</td>
<td>59</td>
</tr>
<tr>
<td>Designer Objects</td>
<td>60</td>
</tr>
<tr>
<td>Creating Designer Objects</td>
<td>60</td>
</tr>
<tr>
<td>Materials</td>
<td>65</td>
</tr>
<tr>
<td>Create a New Material</td>
<td>66</td>
</tr>
<tr>
<td>Multi-Material</td>
<td>71</td>
</tr>
<tr>
<td>Assigning Material to Objects</td>
<td>75</td>
</tr>
<tr>
<td>Apply a Multi-Material to an Object</td>
<td>78</td>
</tr>
<tr>
<td>Exporting Objects</td>
<td>81</td>
</tr>
<tr>
<td>Physics</td>
<td>82</td>
</tr>
<tr>
<td>Create a physics block</td>
<td>83</td>
</tr>
<tr>
<td>Create an Archetype Entity</td>
<td>87</td>
</tr>
<tr>
<td>Flow Graph Scripting</td>
<td>91</td>
</tr>
<tr>
<td>Prepare the scene</td>
<td>92</td>
</tr>
<tr>
<td>Place Tag Points</td>
<td>92</td>
</tr>
<tr>
<td>Add Block and Link</td>
<td>94</td>
</tr>
<tr>
<td>Stack Physics Blocks</td>
<td>97</td>
</tr>
<tr>
<td>Add A Trigger Volume</td>
<td>98</td>
</tr>
<tr>
<td>Add A Camera</td>
<td>99</td>
</tr>
<tr>
<td>Create a Flow Graph Script</td>
<td>102</td>
</tr>
</tbody>
</table>
Welcome to Lumberyard Editor’s Getting Started Guide. Lumberyard is a fully-connected, multiplayer-obsessed technology platform that enables professional game developers to build games with vast and vibrant communities.

This Getting Started Guide familiarizes you with the basics of Lumberyard Editor. You will be guided through nine tutorials that cover the most commonly used tools and features of this editor. This tutorial assumes you have some previous experience with game engines or 3D modeling tools. At the end of this tutorial, you will have created an environment with buildings, trees, and rolling hills. You’ll also have a character that you can move around in third person and shoot balls to knock over a block wall based on a script you’ve created.

After you complete this tutorial, you will have enough knowledge to further explore Lumberyard’s wide range of tools and features. You can complete additional tutorials to help you learn more about specific tools and features to put you well on your way to building your next game.

To view the Getting Started Guide online, see
http://docs.aws.amazon.com/lumberyard/latest/gettingstartedguide
The Getting Started Guide is divided into the following sections:

- Editor Overview (p. 3)
- Object Placement (p. 22)
- Terrain (p. 29)
- Lighting (p. 45)
- Camera and Player Start (p. 57)
- Designer Objects (p. 60)
- Materials (p. 65)
- Physics (p. 82)
- Flow Graph Scripting (p. 91)
Editor Overview

The Editor Overview covers Lumberyard Editor's most common and essential elements. This guide describes basic features that will help you complete the Getting Started Guide tutorials and put you on your way to making great games.

For more detailed information about the features described in this overview, see our documentation pages.

Features and functions described in this overview include:

• New Level Creation (p. 4)
• Lumberyard Objects (p. 5)
• Editor Layout (p. 6)
• Essential Tools (p. 7)
• 3D Level Navigation (p. 13)
• Editors (p. 14)
• Display Options and Settings (p. 15)
• Layers (p. 17)
• Auto Backup (p. 20)
New Level Creation

To create a new level:

1. When you first launch Lumberyard Editor, the Welcome to Lumberyard Editor dialog box appears. From here, you can create a new level, open an existing level, view and open recent levels, and access the Lumberyard support pages on our website. You can also choose to stop showing this dialog on startup.

2. Choose New Level.

3. In the Name field, type a name for your level.
4. In the **Generate Terrain Texture** dialog box, keep the default settings for this tutorial.

This dialog box generates your level's terrain using texture dimensions.

![Generate Terrain Texture dialog box](image)

---

**Lumberyard Objects**

Lumberyard has three object types, which encompass every object that can be placed in a level:

**Entities**
- Objects with behavior properties. The behavior properties enable objects to respond to game events via game scripts or code. Entities are subdivided into the following types:
  - **Entities** - General objects used to set up and create game play conditions or visual settings (such as lights, volumes, cameras, physics objects).
  - **Geometry Entities** - Entities with an attached geometry mesh.
  - **Particle Entities** - Particle systems created and placed within a level.
  - **Archetype Entities** - Custom-defined entities defined by the user based on existing entity properties.

**Brushes**
- Objects with 3D mesh data only. Brushes do not contain behavior properties of an entity.

**Designer Objects**
- Objects that have been created with the Designer modeling tool (these are similar to brushes).
Editor Layout

Lumberyard Editor interface is comprised of the following areas:

- **1: Main menu** - All functions and settings
- **2: Editor toolbar** - Most commonly used tools and editors
- **3: Viewport header** - Search bar and display options for Perspective Viewport
- **4: Perspective Viewport** - 3D environment view of level
- **5: Viewport controls** - Controls for selected objects; Options for navigation speeds; Other Viewport functionality
- **6: Console** - Input and output of editor and game data
- **7: Rollup Bar** - Access to objects/entities and tools for building and managing content in the Perspective Viewport

The Rollup Bar contains the following tabs:

- **Objects** (brushes, entities, volumes, prefabs, etc.)
- **Terrain** (terrain, vegetation, and environment tools)
- **Modeling** (obsolete)
- **Display** (render settings, 3D settings, hide settings)
- **Layers** (organize and manage assets by layers)

You will find it helpful to learn these terms and become familiar with each area, as they are referred to often in Lumberyard tutorials.
Essential Tools

Lumberyard Editor features many robust tools, settings, and options to help you build high quality games. The most essential tools are described in the following sections.

Select, Move, Rotate, and Scale

Select, Move, Rotate, and Scale are the primary tools for interacting with objects. These tools control the selection, location, rotation, and size of objects.

When an object is selected, an XYZ gizmo overlays it, which identifies the currently active tool and shows how the object can be manipulated.

Select

Using Select, you can choose any object in the Perspective Viewport. The gizmo for Select is a set of three lines – one for each direction: X, Y, Z.

To select, move your cursor over the object you wish to select. When the object is highlighted yellow and the cursor changes to a +, left-click to select the object.
Move

The Move tool selects and moves an object within the 3D space of the Perspective Viewport. The Move gizmo is a set of three lines with arrowheads on the X, Y, and Z lines.

To move your selected object along a fixed line, click the X, Y, or Z line, which becomes highlighted in yellow. You can then click and drag your object along that line.

The Move gizmo also features three small right angle squares along the XY, ZY, and XZ planes. To move your object along a plane, click to select one of the small squares. You can then click and drag your object along that plane.
**Rotate**

The Rotate tool selects and rotates an object. The rotate gizmo is a set of circles around the object along the X, Y, and Z axes.

To rotate an object, select one of the small inner circles. You can then click and drag to rotate around that rotational plane.

A larger outer circle also surrounds the entire gizmo. Select and drag this circle to rotate the object in relation to the screen display.
Scale

The Scale tool can select an object and change its size. Scale's gizmo has cubes on the X, Y, and Z lines.

To scale an object, select the X, Y, or Z line, then click and drag in the direction you wish to scale your object.

Selection Hotkeys

Hotkeys enable you to quickly switch to the Select, Move, Rotate, or Scale tools.

Press the following numbers on your keyboard to use these hotkeys:

- Select – 1
- Move – 2
- Rotate – 3
- Scale – 4
Snap Grid

The **Snap Grid** lets you snap to points along a grid when moving an object. The Snap Grid is on by default, and can be turned off or on by clicking its icon on the Editor toolbar.

![Snap Grid Icon](image)

Adjusting Grid Snap

To customize the size of the Snap Grid, select the small down arrow to the right of the grid icon. Adjust this value to increase or decrease the snap steps of the object you are moving.

![Adjusting Grid Snap](image)

Snap Angle

**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, select the icon.

![Snap Angle Icon](image)
Adjusting Snap Angle

To customize the Snap Angle degrees of rotation, select the down arrow to the right of the Snap Angle icon. Adjust this value to increase or decrease the degree of rotation with each snap.

Follow Terrain and Snap to Objects

Follow Terrain and Snap to Objects enables you to move an object along terrain features rather than along the X, Y, Z axes or planes. With the Follow Terrain and Snap to Objects option selected, you can freely move your object in any direction along your terrain, and the object automatically adjusts to terrain features.

In levels with a terrain mesh, this tool can be extremely useful, as you can easily keep your objects sitting directly on the terrain, or in whatever relation you already have it to the terrain, rather than having to adjust it manually to peaks and valleys.
3D Level Navigation

The level navigation within the Perspective Viewport is similar to that of other 3D modeling tools in that the control scheme uses First Person Shooter (FPS) controls. Because of this similarity, those users who are familiar with playing FPS games should find it easy to navigate within their Perspective Viewport.

To navigate within your level in the Perspective Viewport, use the following click and drag actions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Mouse Button(s) (click and drag) or Keystroke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select multiple objects</td>
<td>Left mouse button</td>
</tr>
<tr>
<td>Turn left/right, look up/down</td>
<td>Right mouse button</td>
</tr>
<tr>
<td>Pan left/right, pan up/down</td>
<td>Middle mouse button</td>
</tr>
<tr>
<td>Zoom in, zoom out</td>
<td>Right mouse + middle mouse button OR Mouse wheel</td>
</tr>
<tr>
<td>Strafe forward</td>
<td>W</td>
</tr>
<tr>
<td>Strafe backward</td>
<td>S</td>
</tr>
<tr>
<td>Strafe left</td>
<td>A</td>
</tr>
<tr>
<td>Strafe right</td>
<td>D</td>
</tr>
</tbody>
</table>
Editors

Lumberyard Editor features a collection of editor tools that enables game developers to build specific categories of content. The Getting Started Guide refers to many of these editors and their uses.

You can open all the editors from the View, Open View Pane menu.
The most commonly used editors can be accessed from the Editor Toolbar.

Display Options and Settings

Display Options and Settings allow you to customize what you see within Lumberyard Editor so that you can set up your view to be most useful for you. The following sections describe the display options and settings available to configure.

Perspective Viewport Options

To configure display options for the Perspective Viewport, right-click on the Viewport Header. Check or clear options to best suit your individual workflow preferences.
Show/Hide Helpers

The right-hand side of the Viewport Header shows additional display settings. Toggle the H "Helper" icon to show or hide all the entity icons and their visual guidelines. Hiding these elements can de-clutter your view when you wish to focus on other components.

Toggle Display Information

To change the amount of debug/display information that is displayed in the Perspective Viewport, click the "i" icon. Click this icon multiple times to choose the level of information you'd like to see.
Navigation Speed Settings

You can adjust your Perspective Viewport navigation speed. The Speed setting shows the current movement speed setting. You can type a number into the Speed field, or click .1 (Slow), 1 (Normal), or 10 (Fast).

AI/Physics Toggle

The AI/Physics button turns on and off the movement events for physics, AI, and particles in edit mode. With these options, you can test and view these events without entering game mode.

Layers

The Layers tab in the Rollup Bar helps you organize the large amount of content that is created with building a level.

You can create new, delete, rename, save, and export your layers from within this tool using the icons above the layer list.
Additionally, each layer has its own Eye and Arrow icons to help you manage your objects:

- **Eye icon**: Temporarily hides that layer in order to focus on a specific layer. Click the Eye icon on each layer you wish to hide. Click it again to make the layer(s) visible.

- **Arrow icon**: Disables the ability to select objects in that layer. This feature can be useful if you are having trouble selecting an object that is overlapped by objects in other layers.

Furthermore, you can organize your layers into nested groups by using Ctrl + drag to place them to precisely where you’d like them.

**Collaboration with Multiple Users**

The **Layers** tool also enables multiple users to work within the same level file. If multiple users need to work within the same level, they can do this by creating their own layer file and building all their content within that layer. Using a source control tool such as Perforce (while not required) is a useful way to manage these different layers with its ability to check files in/out and get latest updates from other team members.

When a new layer is created in a level, it's stored within the level folder in a directory called Layers. These layer files appear in that folder with the extension .lyr.

To work within a specific layer file, select the layer tool, and then select the layer file that you want to work in. With that layer selected, you can create and add content, all of which are automatically created as a part of that layer.

When working within a specific layer, the level file doesn't need to be saved, but the layer file being worked does. To save the layer, click the icon Save External Layers, as shown in the following image.
Moving Assets Between Layers

Each entity, brush, or whitebox designer object you place in the level is assigned to the currently selected layer. If you have not created any additional layers, objects are placed in the default Main layer.

To assign an object to a different layer, select the object in the Perspective Viewport. In the Rollup Bar, on the Objects tab, you can view the name of the layer the object is currently in.

Click the Layers icon to display a list of the current layers for that level. To switch the object to a particular layer, select that layer from the drop-down list.
# Auto Backup

Lumberyard has an Auto Backup feature that is on by default. Auto Backup saves your level file incrementally. This helps prevent loss of your progress in case an unexpected event causes a crash/freeze while you are working in your level.

**To customize your Auto Backup settings:**

1. From the Main Menu, under **File**, select **Global Preferences, Editor Settings**.
2. Under **General Settings**, choose **Files**. In this dialog box, you can customize your **Auto Backup** settings.
Object Placement

This tutorial teaches you how to start building your scene by adding brush objects and placing prefabs.
- Place Brushes (p. 22)
- Prefabs (p. 25)

Place Brushes

To place a brush:

1. Open the Lumberyard Editor. Choose to create a new level. Keep the default new level file settings for this tutorial. Name your new level 'MyFirstLevel'.
2. In the Rollup Bar on the right, select Brush in the Objects tab. Under the Browser heading, open Objects/StyleTown/Natural/Terrain and select the object townblock.
3. Click and drag the word townblock into the Perspective Viewport.
4. In the Perspective Viewport, select the townblock brush. When selected, a gizmo appears at the corner of your brush.

5. Press **Ctrl+C** to copy your townblock, and move your second townblock next to your first townblock.
6. From the Brush list, open \StyleTown\Architecture\Buildings. Drag four buildings (or enough to your liking – have fun!) onto one of your townblocks.

**Tips**

- To display the brush list again after placing a previous brush, choose Brush again beneath the Objects heading.
- When clicking and dragging, you will see only the cursor while dragging. After you release the mouse button, your brush appears.

7. Click Brush again. Under Browser, open \StyleTown\Natural\Vegetation. Drag trees and bushes and place them around the buildings. Add as many as you would like.

Your scene should now look similar to the following:
Prefabs

Prefabs are combinations of pre-defined assets that enable you to create content more quickly.

This tutorial teaches you the simplest way to use a prefab—grouping sets of brushes together. This prefab simplifies and speeds up the building of your environment.

To place prefabs:

1. To access previously created prefabs, open the Database View. From the Editor Toolbar, choose the Database icon or, from the Main menu, open View, Open View Pane, Database view.

The Database Editor has the following three areas:

- **1: Database tabs** - Database types you can view and manage:
  - Entity Library
  - Prefabs Library
  - Vegetation
  - Particles
  - Game Tokens

- **2: Editor toolbar** - Tools to open, save, add, and remove prefabs

- **3: File tree view** - Opened prefabs available in your level. This field is empty at first; the next steps show you how to open and access prefab libraries so you can see the NeighborhoodBlock prefabs like the image below.
2. Select the tab Prefabs Library.
3. Click the Load Library icon \(\text{Load Library}\) and select the file `Prefabs\styletown.xml`. Click OK.

4. In the left column (the file tree view), the prefab library `NeighborhoodBlock` appears.

After you add a prefab library to a level, these prefabs appear when you choose the prefab button in the Rollup Bar. The prefabs located in the Rollup Bar are a faster method to deploy prefabs to a level.
5. Select and drag the file StreetSet_A into the Perspective Viewport.

   Using the movement tool, position the street so that it fits around the two town blocks.

   The street prefab comes into the level sitting above the ground plane. Verify that Grid Snap is on and move the street prefab down so it aligns with the ground.

6. From the prefab list, select and drag Block_A into the Perspective Viewport.

   Move and align this block of houses and trees over the empty block in the level.

7. Select and drag the prefab StreetLight_A into the Perspective Viewport. Position this along the center street.

8. Save your file.

   You should now have something that looks like this:
This tutorial teaches you how to apply materials to the terrain, modify the terrain height, and use the vegetation tool to paint trees.

- Terrain Painting (p. 29)
- Terrain Height (p. 38)
- Terrain Vegetation (p. 40)

**Terrain Painting**

Now that you have the basics of the scene built with brushes, it's time to start building the surrounding terrain environment.
To build your terrain:

1. First, you must open two new editors: The **Terrain Textures Layers** and the **Material Editor**.

   Open the Terrain Texture Layers editor: From the Editor toolbar, select the Terrain Textures icon or, from the main menu bar, select **View, Open View Pane, Terrain Texture Layers**.

   The Terrain Texture Layers editor is used to define all the materials that are used to paint on the level's terrain mesh.
The Terrain Texture Layers editor has the following areas:

- **1: Layer Tasks** – Add, delete, and reorder layers in the Layer List
- **2: Layer Info** – Information about the selected layer, including the layer size and surface type count
- **3: Layer Texture** – Low-detail texture swatch; displays color information for the surface texture; displays far away textures
- **4: Options** – Options related to the Layer List
- **5: Layer List** – Layer textures available for painting onto the terrain (e.g. dirt, grass, rocks)
2. Open the Material Editor: From the Editor toolbar, you can select the Material Editor Dialog icon or, from the main menu bar, select View, Open View Pane, Material Editor.
The Material Editor has the following areas:

- **1: Editor toolbar** - Tool list for applying, deleting, saving, and creating materials
- **2: Material preview** - Display for the selected material’s appearance
- **3: Material folder directory** - Folder tree to navigate through the materials available for use in the level
- **4: Material properties and settings** - Parameters that define the material’s appearance

With these two editors open, you can now assign materials to your terrain mesh.

3. In the Terrain Texture Layers editor, in the Layer Tasks area, click **Add Layer** twice to add two new layers.

4. In the first new layer, double-click on the text **NewLayer** and rename it **grass**.

5. In the second new layer, double-click on the text **NewLayer** and rename it **dirt**.
6. Select the grass layer; it becomes highlighted orange.

7. Go back to the Material Editor and select the material `gs_grass_01` located in the Material Editor's directory path `materials/gettingstartedmaterials/gs_grass_01`.

8. In the Terrain Texture Layers editor, the grass layer should still be selected. In the Layer Tasks area, choose **Assign Material**.

9. Next, select the dirt layer.

Switch back to the Material Editor and select the Getting Started Material `gs_ground_01`. 
10. Switch to the Terrain Texture Layers editor. With the dirt layer still selected, choose Assign Material in the Layer Tasks area.

Note
In each of the terrain material layers is a small material preview box. This material preview box displays the assigned layer texture, not the material assigned from the material editor. For this tutorial, we are using the default grey.dds file, so both the grass and dirt layers appear with the grey layer texture.
11. Close Material Editor and Terrain Texture Editors. You are now ready to paint grass and dirt textures onto the terrain. In the Rollup Bar, select the Terrain tab, and then choose the Layer Painter button. This opens Lumberyard’s terrain layer painting tools:

![Layer Painter window](image)

12. In the Layer Painter window, at the bottom, there is a list of the terrain materials that you have created: **grass** and **dirt**. Select **grass**.

13. Just above, there is the Vertex Coloring box with a **Color** box (default is white). Click on the color box and change the RGB color to 145, 180, 75 for a neighborly grass-green color. Click **OK**.
14. Click the **Flood** button at the bottom of the Layer Painter tool. The terrain is now covered in the grass texture and looks similar to this:

![Grass Texture Example](image)

15. With the entire terrain now painted with the grass texture, you can paint some dirt into the scene around the perimeter of the street. Select the dirt material at the bottom of the Layer Painter tool.

16. Adjust the color box to a brown tone: RGB 115, 95, 50. Click OK.

17. In the **Brush Settings** box, set the Radius to 5 and the Hardness to 0.5.

18. Place your mouse in the Perspective Viewport (where your lovely town is displayed) and left click and drag to paint the dirt texture around the perimeter of the street. Do as little or as much as you like. When you are finished, you will have a town similar to the following:

![Dirt Texture Example](image)
Terrain Height

With the terrain now painted, you can manipulate the height of the terrain with the **Modify Terrain** brush tools.

**To manipulate your terrain’s height:**

1. From the same Terrain tool tab in the Rollup Bar, choose the **Modify** button. This displays the **Modify Terrain** tools.

The Modify Terrain tool has the following areas:

**Brush Settings**

- **Flatten** – Flatten the terrain to the designated height setting
- **Smooth** – Soften the terrain down to a smoother surface
- **Rise/Lower** – Raise or lower the terrain based on brush size settings
- **Pick Height** – Find and set heights based on existing terrain geometry
- **Outside Radius** – Set how big your brush is when painting
- **Inside Radius** – Set how round or flat the brush is in relation to the outside radius setting
- **Hardness** – Soften or harden the outer brush settings
- **Height** – Sets brush height

**Noise Settings**

- **Scale** – Strength of the noise effect; higher value produces more noise
- **Frequency** – How often the effect is applied
2. Select the **Rise/Lower** button, set the **Outside Radius** = 25, **Inside Radius** = 1, **Hardness** = 0.25 and **Height** = 3. These settings help you create some gentle hills around your neighborhood.

3. In the Perspective Viewport, navigate towards the outer perimeter of the terrain map and left click to paint on the terrain. Experiment with holding and clicking along the terrain to manipulate the terrain to different heights. Build some larger hills of different sizes and shapes.

4. Adjust the brush settings so the **Inside Radius** = 20, **Hardness** = 1 and **Height** = 1. Paint again on the terrain. Notice how the terrain rises up more straight and rigid.

5. Choose the **Smooth** tool, set the **Outside Radius** = 25 and **Hardness** = 0.2. Paint with the smooth brush over the last area of terrain you created. Notice the smoothing of the terrain.

6. Select the **Pick Height** tool and click on a high point on the terrain. Notice the Height setting in the tool adjusts to the height selected. Select a point on the terrain where the height hasn't been changed. The height setting in the Modify Terrain tool changes to 32. Notice that this tool doesn't change your terrain, but simply adjust the settings for the next step.

7. Select the **Flatten** tool, set **Outside Radius** = 25, **Inside Radius** = 0 and **Hardness** = 1. Paint with the flatten tool over the area you just smoothed. The terrain should now flatten to the same height as the rest of the default terrain height.

8. Using these tools, create a range of high and long hills in the distant background from the neighborhood block area you created.

9. Adjust your brush settings to create smaller rolling hills closer to the neighborhood block. Use the **Smooth** tool to soften where you like. With a few minutes' work, you have something like this:
Terrain Vegetation

With the Vegetation tool, you can paint 3d mesh objects like trees, shrubs, and grasses onto the terrain. Various settings help you to build beautiful organic environments using any type of 3d models you define.

In the previous steps, you textured and modified your terrain. Now it's time to add some trees using the Vegetation tool.

To add trees:

1. In the Rollup Bar, on the Terrain tab, choose the Vegetation button.
In the Vegetation tool, you can modify the following settings:

- **Toolbar** – (Set of small icons) Displays tools to create, modify, and organize vegetation types
- **Brush Radius** – Sets the size of the brush used to paint vegetation into the level
- **Paint Objects** – Enables you to paint in the level
- **Objects** – Lists vegetation objects that have been created
- **Table of attributes** – List the attributes that can be modified for each vegetation object
2. In the **New Category** dialog, enter **Trees**. Click **OK**.

![New Category Dialog](image)

3. In the Objects list, select the Trees category you just created.

4. In the tool list, click the icon ![Vegetation Tool](image)

5. In the **Preview** dialog, choose the **Objects** folder from the left side.

6. Open the folder **StyleTown**.

![Preview Dialog](image)
7. Open **Natural**, then **Vegetation**. In the list of .cfg files, use **Ctrl+click** to select the following files: tree_01, tree_02, tree_04, and tree_06. Click **Open**.

8. You could paint trees into the environment now, but every tree would appear with the default brush settings, which provides no variation on the size, rotation, or spread of the trees.

9. Select the first tree on the list and change the following settings in the attributes list. Do this for each of the trees in your list.

- **+- Size Var = 0.2** – Randomizes the tree’s size
- **Random Rotation = On (checked)** – Rotates the tree randomly
- **Density = 100** – Defines the density of the trees
- **Sprite Distance = 50** – Defines distance from the camera view that vegetation transitions from a mesh to a sprite of that object
10. Select the Trees group name and then adjust the **Brush Radius** to 50 (this size is appropriate for filling the terrain space quickly).

11. Select the **Paint Objects** button, then place your cursor in the Perspective Viewport and click to paint your trees.

Depending on the tree density you want, you can click once and place a random group of trees, or you can click and drag through the space and paint them along a path. Adjust the Brush Radius and the Density settings to change the number of trees painted. Your neighborhood scene should look something like the following image.

![Image of a neighborhood scene with trees]

12. Save your file.

Congratulations, you’ve created your first level environment with a small neighborhood between rolling hills, amidst a lush green forest.
Lighting

This tutorial teaches you about the tools and features used to light a scene, including Environment Probes, Time of Day, and Lights.

- Environment Probes (p. 45)
- Time of Day (p. 48)
- Lights (p. 50)

Environment Probes

You must place an environment probe before you place lights. Environment probes are critical in achieving great-looking lighting. Probes are important for many features, including reflections, ambient diffuse values, particle diffuse values, and shadow colors. When building a level file, place multiple environment probes to help adjust the space to the right visual quality. Always start with a default global environment probe. This tutorial helps you set up the global probe.
To set up a global probe:

1. In the Rollup Bar, select the Objects tab. Under the Objects heading, select Misc.
2. Under the Object Type heading, select EnvironmentProbe. Move your mouse cursor into the Perspective Viewport. An EnvironmentProbe actor moves in the scene. Click on the ground terrain to place the probe.
3. Position the probe roughly in the center of the level.
4. Under the heading EnvironmentProbe Properties, set the BoxSizeX, BoxSizeY, and BoxSizeZ values to the size of the entire map. In this file, that should be X=1024, Y=1024 and Z=250.
5. With the probe selected, zoom out from the level so that you can see the entire map. Adjust the X,Y position of the probe so that it covers the entire area of the level.

6. Now that the probe's properties are set, it's time to turn on the probe and generate the cubemap. To do this, put a check mark in the **Active** attribute box in the EnvironmentProbe Properties.

7. Under the **Probe Functions** header (above EnvironmentProbe Params), click **Generate Cubemap**. The scene is briefly colored fuchsia while the probe is rendering the scene. The most visible change will be that the shadows appear less dark.

**Tip:** Experiment with adjusting the values for Color: Diffuse (RGB values) and Color: DiffuseMultiplier to see how it affects your scene.
Time of Day

Lumberyard can quickly change and even animate the time of day in your scene using the Time of Day editor.

To set the time of day:

1. Open the Time of Day editor by selecting the icon on the Editor Toolbar. You can also open it from the main menu: View, Open View Pane, Time of Day.
The Time of Day Editor has the following areas:

- **1: Editor toolbar** - Icon toolbar for most common functions: Undo, Redo, Import, Export.
- **2: HDR Settings** – Settings to manage HDR lighting
- **3: Time of Day Tasks** – Management of basic tasks within the time of day editor
- **4: Current Time** – Display of times, start/end, and play speed
- **5: Update Tasks** – Controls to update the play or stop of time of day, based on play speed setting
- **6: Timeline editor** – Management of light settings along a 24-hour time cycle
- **7: Parameters** – Lighting settings to adjust time of day conditions

The Time of Day editor features a variety of tools to adjust and manage time of day settings. This tutorial focuses only on changing the time of day.

2. In the **Tasks** area, choose **Import from File**. Navigate to SamplesProject/Levels/GettingStartedFiles and open **TimeOfDay.xml**.

   This loads a set of time of day settings created for this tutorial. Notice the changes in light, fog, and sky colors

3. In the timeline, slide the time line marker (set at 12:00 by default) to 21:00. Your scene now shows night.

   Experiment with changing the timeline marker to different times of day to see the lighting changes in your scene. Observe how this time adjustment also changes the settings in the Parameters area.

   Before moving to the next part of this tutorial, close the Time of Day editor.
Now, add lights to the streetlights:

1. In the Rollup Bar, in the Objects tab, select the Entity button. Beneath the Browser heading, open the Lights folder and select the Light object within the folder. Drag and drop "Light" into the Perspective Viewport.
2. Position the newly placed light entity just under one of the streetlights.

3. Because the light source is a streetlamp, it should be a spot light (rather than a fill light). To change the light from fill to spot: Beneath the **Entity Properties** header, find **Projector**. Under **Projector**, select **Texture**, and then click the folder icon.
4. Open the directory \SamplesProject\textures\lights\generic. Open spot_075.tif.
5. The light changes to a spotlight and is oriented sideways. Use the rotate tool to select and rotate the light so that it points down.

In the attributes table (beneath the header **Entity Properties**), you can modify a variety of settings to customize the light. This tutorial teaches you how the following settings affect the light:

- **AttenuationBulbSize** – Light bulb size; this is the starting point for where light begins to fall off exponentially. A value of 1 sets the light at full intensity for one meter before it begins to fall off. Adjust this size in relation to the diffuse multiplier to manage the brightness of the light source without entering unmanageable numbers.
- **Radius** – Distance from the source that the light affects the surrounding area.
- **Diffuse** – Color (RGB value) of the light.
- **DiffuseMultiplier** – Intensity of the diffuse color. Balance this value with the AttenuationBulbSize to define the balance of natural light levels.
- **ProjectorFov** – Field of view for the projection light.
- **CastShadows** – Makes the light cast a shadow based on the minimum selected config spec. "High" won't work on Low/Medium, for example. To ensure shadows are always cast, set this to 'Low Spec'. **Note:** The CastShadows setting is not a 'quality' setting; it's a performance spec setting on the type of machine running the level.
6. For this tutorial, use the following settings:

- **AttenuationBulbSize** = 6
- **Radius** = 20
- **Diffuse Color** = 250,250,150
- **Diffuse Multiplier** = 25
- **ProjectorFov** = 70
- **CastShadows** = Lowspec

Experiment with these settings to get a feel for how the differences between bulb size, radius, and diffuse multiplier change based on the input values you use.
7. With the light selected, press **Ctrl+C** and drag your mouse to make a copy of the light. Position the light under the next street light. Repeat this for the rest of the streetlamps in the scene.
8. You can place some additional fill lights to help light the scene, which is still a bit dark. To do this, repeat the previous steps and place lights between the street lamps, similar to the following image. For these lights, use the following settings:

- **AttenuationBulbSize** = 5
- **Radius** = 25
- **Diffuse Color** = 250,250,150
- **Diffuse Multiplier** = 8
- **CastShadows** = Never

9. Experiment with these settings and observe how the lights affect the scene. Find a setting you like, or keep the suggested values above.

10. Try using the **Hide** option (H icon on the right side of the Viewport Header) to hide the entity icons. You can view the scene without the visual clutter of entity icons. Toggle it off and on to see how this affects what the scene looks like.
Camera and Player Start

This tutorial teaches you how to place a prefab Game Camera with Character control input and how to run the game level.

• Place a Game Camera (p. 57)
• Switch to Game Mode (p. 59)

Place a Game Camera

Before you can play the level, you must create and place a game play camera with character control and specify a start point. Lumberyard contains default game play cameras (with supporting input controls) to help you ‘quickstart’ building a game play level.

To place a game play camera:

1. From the main menu, open the Database Editor by selecting View, Open View Pane, Database View. You can also select the Database View Editor icon on the Editor toolbar.
The Database Editor for Prefabs has the following areas:

1. **Database tabs** – Database types to view and manage:
2. **Editor toolbar** – Tools to open, save, add, remove Prefabs
3. **File tree view** – View of the Prefabs available for use

2. Select the **Prefabs Library** tab.
3. Select the **Load Library** icon on the toolbar.
4. From the directory listing, select `prefabs\character_controllers.xml`. 
5. In the file tree view, select **Sphere_Controller(1)** and drag it into your Perspective Viewport.

6. This prefab (Sphere_Controller) provides the level with a 3rd person camera, character, and input control. The location of the Sphere_Controller in the map is also determines the default start point within the level.

**Switch to Game Mode**

Now that you have placed a game camera, you can run the level and move around in the space.

To switch to Game Mode, do one of the following:

- From the main menu, choose **Game, Switch to Game**.
- On your keyboard, press **Ctrl+G**.

You are now running your first Lumberyard level.

To navigate, use the following controls:

- **Move forward/back** – W and S keys
- **Strafe left/right** – A and D keys
- **Turn left/right** – Mouse slide left/right
- **Jump** – Space bar
- **Camera Look** – Mouse movement

To exit game mode and return to edit mode, press **Esc** on your keyboard.
This tutorial steps you through the Designer tool and shows you how to create two objects for use in a level.

- Creating Designer Objects (p. 60)

Creating Designer Objects

To create designer objects:

1. In the Rollup Bar, click the Objects tab. Select the Designer button.
With the Designer tool, you can create and edit meshes, and perform UV Mapping and Exporting.

**The Designer tool has the following areas:**

- **1: Standard Parameters** – Change standard Designer object settings like name and layer location.

- **2: Settings** – Toggles various conditions for object creation. Usually, you can keep the default settings.

- **3: Selection** – Selects vertex, edge, or face. Also features secondary selection methods.

- **4: Create & Edit** – Features tools, divided among five tabs, to select and edit shapes, modify objects, manage surface material, and export objects.

- **5: Mesh Dimensions** – Customizes subdivision of faces and numerical input of objects size and shape.

Objects created with the designer tool become a third type of mesh object within the editor. Though they are similar to brushes and entities, they have their own unique category. Designer objects do not have the same capabilities as entities, but they work just as well as brush objects when building a level. You can block out entire levels with the Designer tool; this is an effective method for blocking out gameplay environments.

2. For this tutorial, we will create two Designer objects: A box and a sphere. Before you create an object, turn off **Snap to Grid** in the Editor toolbar, or press G on your keyboard.
3. Under the **Designer Menu** heading, select **Box**. In the Perspective Viewport, click and drag to create a rectangle. When you release the mouse button, the length and width are 'locked in,' and the height control appears (see next step).

4. Create a box that is about 2.5 units long by 1 unit wide. The size doesn't need to be exact, but try to come close to that size range for this tutorial.

   **Tip:** It is easier to create the box if you zoom in close to the ground.

5. Move your cursor up to define the height at about 1 unit high. Click the left mouse button again to lock the height. This completes the creation of the object.
6. Under the Designer Menu heading, in the Selection area, select the Object button. This exits the edit mode on the box you just created.

7. Next, create a sphere: Under the Designer Menu heading, in the create & edit section (has tabs SH, ED, MO, SU, MI), select Sphere.
8. Click and drag with the left mouse button to create a sphere that is about 1.5 units around.

**Tip:** Like with the box, zoom in close to the ground to make it easier to create a sphere with these dimensions.

9. To exit creation mode: Under the **Designer Menu** heading, in the **Selection** area, select the **Object** button.

   Now that you have these two objects created, you can apply textures to them and then export them.
This tutorial steps you through the process of creating materials, applying those materials to an Designer object, and exporting that object.

- Create a New Material (p. 66)
- Assigning Material to Objects (p. 75)
- Exporting Objects (p. 81)
Create a New Material

To create a new material:

1. To open the Material Editor: From the Editor toolbar, select the Material editor icon or, from the main menu, select **View, Open View Pane, Material Editor**.
The Material Editor has the following areas:

- **1: Editor toolbar** – Icon tool list to assign, reset, get properties, add and save or remove.
- **2: Material preview window** – Appearance of current material, based on settings
- **3: Material folder directory** – Folder tree to navigate through the materials available for use
- **4: Material properties and settings** – Settings that change the properties and behavior of the selected material

2. In the Material folder directory, open the materials folder, then open `gettingstartedmaterials`. 
3. With the `gettingstartedmaterials` folder selected, in the Editor Toolbar, choose the **Add New Item** button.

![Material Editor](image)

4. Navigate to `Materials/GettingStartedMaterials`. In the **File name** field, type `sphere`. Choose **Save**.

5. Click the **Add New Item** button again. In the **File name** field, type `block`. Choose **Save**.
6. Select the sphere material you just created. Your material editor should look similar to the following:

![Material Editor Interface]

7. In the Materials Properties and Settings (lower right area), scroll down and find the **Texture Maps** heading. Select the ellipsis ‘...’ icon on the **Diffuse** line.

![Texture Maps Section]

9. Under the heading **Materials Settings**, on the **Surface Type** line, select **rubber** from the drop-down menu.

   This gives the material object the properties of rubber; any object mapped with rubber bounces a little when hitting another surface.

10. Under the heading **Lighting Settings**, set **Diffuse Color** (Tint) to \(0, 40, 155\).

11. Set **Specular Color** to \(60, 60, 60\).

12. Set **Smoothness** to 175.

13. Save your material setting: In the editor toolbar, choose the Save item icon.

14. Experiment with these various settings to see how adjusting Diffuse color, Specular color and Smoothness quickly changes your material's appearance.
Multi-Material

Now you will take the previously-created 'block' material and turn it into a multi-material. A multi-material is a single material file with multiple materials. If you need to map different materials to a single object (a block for example) creating a multi-material is the only way to do this. Multi-materials are very handy for whitebox prototyping of a scene. For example, if you need to build a neighborhood full of houses, you can build a box with a peaked roof and then cut faces on the side to be windows and doors to represent a house. With a multi-material, you can create the siding, roof, window, and door textures and then assign the correct material id to that single mesh to represent a house.

To create a multi-material:

1. Select and right-click on the 'block' material you previously created.
2. Select **Convert to Multi Material** from the menu pop-up.

Beneath your initial 'block', a '[1] block' appears.
3. Select 'block' (not '[1] block') again; right-click and select **Set Number of Sub-Materials**.

4. In the **Number of Sub Materials** dialog box, set the number to 3. Click **OK**.

   ![Number of Sub Materials dialog box]

   Three new sub-materials appear beneath 'block'.

5. Right-click on each sub-material and rename them **block_01**, **block_02**, and **block_03**.

6. Select **block_01**. On the lower right, under the heading **Texture Maps**, find **Diffuse** and choose the ellipsis '...' button.

7. Navigate to **SamplesProject/textures/GettingStartedTextures**. Open **White.tif**.
8. Do the same for block_02 and block_03.

All sub-materials now have a diffuse texture map assigned to them.

The material preview window shows the three sub-materials on the right; the selected sub-material is also displayed (larger) on the left-hand side.

9. Select sub-material block_01 and set the Diffuse color to 35, 100, 35.
10. Select sub-material block_02 and set Diffuse color to 100, 35, 35.
11. Select sub-material block_03 and set Diffuse color to 35, 35, 100.
12. Under the heading **Material Settings**, set **Surface Type** to wood for all three sub-materials.

The block sub-material should look like the following:
Assigning Material to Objects

Now that you have made your materials, you can assign them to the designer objects you created in the previous tutorial.

To assign materials to objects:

1. First, assign the 'sphere' material to the 'sphere' object you created: In the Perspective Viewport, select the sphere object you created.
2. In the Material Editor, select the sphere material you created.
3. In the Material Editor toolbar, choose the Assign Item to Selected Objects button. You can also right-click the material name and select Assign to Selected Objects.
The sphere you created should now look something like this.

4. The designer sphere’s faces are faceted. You can smooth the faces of the sphere. Select the sphere and open the Designer tool.
5. Beneath the heading Designer Menu, select the tab SU.
6. Press the button **Smoothing Group**, and then click and drag over the faces of the sphere to select them:

![Smoothing Group button](image)

7. Below the SU tab area, a **Smoothing Group** tab appears. Press the button **Auto Smooth with Threshold Angle**.

This smooths out the sphere so that it appears smooth and round.
Apply a Multi-Material to an Object

Next, apply to the block the multi-material you created:

1. Select the block that you created.
2. Assign the block material to the block. Your block should look like the following image:

3. Now you need to assign different material IDs to the faces of the box. Select the Designer tool.
4. Under the heading **Designer Menu**, select the tab **SU**. Choose the **UV Mapping Button**.

5. The faces are now selectable; select the faces on each end of the block.

To select more than one face, click on the first face, then hold down the **Ctrl** key to select additional faces.

6. With selected faces, open the **Sub Mat ID** drop-down menu and select **2.Block_02**.

7. In the **UV mapping** tab at the bottom, choose the **Assign** button.

The sub-material #2 is now assigned to the ends of the block.
8. Select the two sides of the box.
9. Open the Sub Mat ID drop-down menu and select 3.Block_03. Press the Assign button.
10. Choose the Object button in the Selection tab to close the tool.

You now have a block with multi-materials assigned to it.
Exporting Objects

After creating the objects and applying material, you can now export these objects and use them for our physics entities that need to be set up.

To export the objects:

1. Select the block in the Perspective Viewport.
2. Select the Designer tool. Under the heading Designer Menu, select the MI tab.
3. Choose the Export button.
4. In the Export tab at the bottom, choose .CGF.
5. In the Save As dialog box, in the File name field, type block. Save it in the directory SamplesProject/Objects/GettingStartedAssets. Choose Save.
6. Do the same steps for the sphere, but name it sphere.
Physics

This tutorial steps you through how to set up a physics object to work in the level and then activate during a game scripted event.

- Create a physics block (p. 83)
- Create an Archetype Entity (p. 87)
Create a physics block

In this section, you will learn how to add a physics entity to the level and assign it the block mesh created in previous tutorials.

**To add a physics entity:**

1. From the Rollup bar, on the Objects tab, select **Entity**.
2. Under the heading **Browser**, expand the **Physics** folder. Select **BasicEntity** and drag it into the Perspective Viewport.
The BasicEntity object appears as a sphere.

3. Beneath the heading **Entity Properties**, select the attribute **Model** and click on the folder icon.
4. Navigate to the folder `SamplesProject/Objects/GettingStartedAssets` and open `block.cfg`.
The sphere changes to the block that you created in the previous tutorials.

5. In the **Entity Properties**, change the **Mass** setting from -1 to 100.

This turns on the physics object so that the block can respond to events. Any value above 0 turns on the physics object. Higher values indicate greater mass, which affects the object's behavior.

You will do more with this physics block in the scripting tutorial.
Create an Archetype Entity

Now you will create a physics object that spawns into the level when activated through a flow graph scripted event. To do this, you must define an Archetype Entity in the Database editor. Archetype entities are like regular entities, but they have their attributes defined before they are placed in a level. This allows you to create specific entity types that can be placed across multiple level files or spawned into a level based on a scripted game event.

To define an archetype entity:

1. Open the Database View Editor: From the Editor Toolbar select the Database View Editor icon or, from the main menu, select View, Open View Pane, Database View.
The Database Editor for Archetype Entities has the following areas:

- **1: Database tabs** – Database types to view and manage (such as the previously shown Prefabs Library and Vegetation tool set)
- **2: Editor toolbar** – Tools to load, save, add and remove archetype entities
- **3: File tree view** – View of the Archetypes available for use
- **4: Settings and properties** – Settings that can be edited for any archetype entity selected
- **5: Preview window** – Preview window for the selected archetype entity

2. In the Editor toolbar, choose the Add New Item icon.

![Database Editor Screenshot](image-url)
3. Open Scripts/Entities/Physics. Select Basic Entity. Choose OK.

4. In the Group field, enter Physics. In the Name field, enter Ball. Choose OK.

The Database view shows the new archetype entity.

5. Under Class Properties, select Model. Click the folder icon.

6. Navigate to SamplesProject/Objects/GettingStartedAssets and open sphere.cfg.
7. Under **Physics**, select **Mass** and set it to 100.

8. Select the Save icon.

9. Make note of the entity name: **Level.Physics.Ball**. You will use this name in the next tutorial to reference this object to spawn.
Flow Graph Scripting

This tutorial introduces the concept of game play scripting using the Flow graph editor.

You will set up flow graph scripts that do five things:

1. Trigger a change between the game play camera and a fixed camera
2. Move a block left and right with keyboard input
3. Set up same block to launch (spawn) a physics sphere
4. Control the power input of the block launcher through keyboard input
5. Toggle input controls to block based on camera state

Topics in the section include:

- Prepare the scene (p. 92)
- Create a Flow Graph Script (p. 102)
- Debug View (p. 121)
- Conclusion (p. 121)
Prepare the scene

Before you build a script, you will prepare the scene with several entities that you will reference when building out the flow graph scripts.

Place Tag Points

First, you must add some tag points into the scene. Tag points are markers that define an XYX point in a level. They can be used to define a destination, move to position, or start point.

To add tag points:

1. From the Rollup bar, on the Objects tab, select AI.
2. Under the heading Object Type, select Tagpoint. Move your mouse cursor into the Perspective Viewport to place the tag point.

Repeat steps 1 and 2 an additional two times.
3. Select one of the tag points. Notice that the XYZ gizmo—depending on how you’ve built the scene—runs along either the Y or X axis of the street.

In this example scene, the street was built along the Y axis. This is the direction we will set the ball to launch in the level. If your road runs along the X axis, you will use the X forward direction instead.

4. Position two of the tag points about midway on the length of the street, and one on each side of the street. It should look something like this image:
Add Block and Link

Next, you will add the block you created earlier and then link a tag point to it. You will later spawn (shoot) balls from this linked object.

To add the block and link a tag point:

1. In the Rollup bar, on the Objects tab, select the **Geom Entity** button. Under the **Browser** header, open **objects/gettingstartedassets** and select the entity block that you previously created.

   Drag the block into the Perspective Viewport near the tag points.

   **Tip:** If you do not see your block and sphere in the list, click the **Reload** button at the bottom of the **Browser** area.
2. Position the block between the two tag points in the street.

3. Position the third tag point so that it's centered and in front of the block.
4. In the Editor Toolbar, select the Link Object icon. Select the tag point in front of the block, then left-click and drag. The cursor display a link icon as well as the object name it can link to (if present).

When the block name is displayed, release the mouse. The tag point and block should now be linked.

Test this by moving the block around. If the tag point moves with the block, the objects are successfully linked.

5. Rotate the block up about 20 degrees. The linked tag point moves with the block as it rotates.
Stack Physics Blocks

Next, you need to stack the physics blocks into a wall at the end of the street.

To stack the physics blocks:

1. Bring into this location the physics block you created in the previous tutorial.
2. Move the physics block to the end of the street, in front of the block and tag points, as shown in the following image.

3. Position the first block, then press **Ctrl+C** to clone the block and position the next block. Repeat this process to build a wall of blocks similar to the following:
Add A Trigger Volume

To add a trigger volume:

1. From the Rollup bar, select Entity and open the folder Triggers. Select and drag a ProximityTrigger into the Perspective Viewport.

2. Position the trigger behind the block and tagpoints, as shown in the following image.

3. With the proximity trigger selected, under the heading Entity Properties, change the DimX (or DimY) dimensions so that the volume crosses the entire street. Change the DimZ dimension so that it is a little deeper on the street as well.

For example, if Y is your forward direction, the street settings should be approximately DimX=20, DimY=10.
Add A Camera

Now you need to add a camera that the view switches to when controlling the block and shooting at the wall.

To add a camera:

1. Position the view so that you can see the block at the bottom of the screen and the wall of physics blocks down the street. It should look something like the following image:
2. In the upper left corner of the Perspective Viewport, right-click on Perspective and select **Create Camera from Current View**. The current view from your Perspective Viewport is now a fixed camera from that position.
3. Navigate away from that view with your mouse and notice the new camera entity that appears in the scene.
Create a Flow Graph Script

Now that all the objects needed are placed, the next step is to create a flow graph file and begin building the scripts.

To create a flow graph:

1. In the Rollup bar, on the Objects tab, select **Entity**. Under the **Browser** header, open **Default**. Select and drag **FlowgraphEntity** into the Perspective Viewport.

2. With the Flowgraph entity placed, right-click on it and select **Create Flow Graph**.
3. Name the flow graph group **Getting Started** and click **OK**.

4. From the Editor Toolbar, open the Flow Graph Editor. Or, you can open it from the main menu: **View**, **Open View Pane**, **Flow Graph**.
The Flow Graph Editor has the following areas:

- **1: Main Menu** – Tools to manage your Flow Graph files
- **2: Editor Toolbar** – Tools to undo, redo, step through, toggle debug and clear debug
- **3: Component Search Bar** – Search for available components
- **4: Canvas View** – Area where scripts are built
- **5: Component List** – Listing of nodes available to create script events
- **6: Flow Graph File Tree** – Access to Flow Graph files available in level and project
- **7: Node Properties** – Property settings of selected component nodes
- **8: Canvas Node Search** – Search for specific nodes in canvas view
- **9: Canvas Node Search Results** – Results of search for nodes in canvas view
- **10: Debug Breakpoints** – Breakpoint management for debug

5. In the file tree view, open Level Flowgraphs/Entities/GettingStarted and select FlowgraphEntity1. This is the flow graph that you’ll use to create the scripts. Level files can have multiple flow graphs, allowing for separation of script events based on location, function, or specific events.

Flow graph scripts are composed of nodes that contain information about objects in the level or functions to call when activated through the script. Flow graph nodes can be pulled from two places: From an entity object currently in the Perspective Viewport, or from the list of components in the Flow Graph Editor. In this tutorial, you will pull nodes from both places.
Script Camera Event

In this first script, you'll set up a trigger that switches between the game play camera and another camera placed in the level.

To set up a trigger:

1. Select the Proximity trigger that you placed earlier.
2. Return to the Flow Graph Editor and right-click in the Canvas view. Select Add Selected Entity.

The node `entity:ProximityTrigger` is displayed in the canvas view.
3. In the Component list view on the left side, open Camera. Select the node View and drag it into the Canvas view.
4. Repeat the previous step to drag another **View** node into the Canvas view. The canvas should now look like this.

![Canvas view with View and Camera nodes]

5. Now you need to assign a camera to each **Camera:View** node.

To do this, go to the Perspective Viewport and select the camera that you positioned in the middle of the street. This is most likely called **Camera1**.
6. Return to the Flow graph canvas. In the first *Camera:View* node, right-click on *Choose Entity* and select *Assign selected entity*. The red "Choose Entity" is replaced with a blue box containing the name of the assigned camera.

7. Now you need to assign the game play camera to the second camera node. To do this, we need to pull out the Camera ID from the camera prefab using a game token.

Game Tokens are values that exist outside of the existing flow graph script, or outside of the level currently in-work.
8. From the Editor toolbar, open the Database View Editor.

9. Select the tab **Game Tokens** to show the game tokens editor. In the Library Tasks area in the upper left, select Load Library.

10. Select `libs\gametokens\gettingstarted_gt.xml` and select OK.
11. In the component list, open Mission. Select and drag the node GameTokenGet onto the canvas.

12. Double-click Token=, then click the box with the two dots. This opens the GameToken folder.
13. Expand **GettingStarted_GT** and select **3P_CameraiD**. Click **OK**.

14. In the canvas view, on the **Entity:ProximityTrigger** node, click and drag the **Enter** output port to the first camera node (**Camera:View**) and attach it to the **Enable** input port.

This creates a connection spline that links one node to the next. These lines tell the flow graph what to do and when to do it. A flow graph script comes together by connecting nodes to one another.
15. From the `entity:ProximityTrigger` node, drag the \textbf{Leave} output port to the \textbf{Mission:GameTokenGet} node's \textbf{Activate} input port.

Connect the \textbf{Mission:GameTokenGet} node's \textbf{Outvalue} output port to the second \textbf{Camera:View} node's \textit{<Input Entity>} input port as well as its \textbf{Enable} input port.

The Flow graph should look something like this:

![Flow graph](image)

16. With the trigger and cameras set up, test the functionality to make sure the trigger switches to the first camera while in the trigger volume and then returns to the game play camera when leaving the trigger volume.
Create a Mover

In this script, you’ll make a block move between two points with keyboard input control.

To create a script to make the block move between two points:

1. In the same flow graph script file, expose a blank area of the canvas to build your next script node set. Use your mouse wheel to zoom out, and right-click drag to move the canvas.

   A flow graph can have multiple script events within the same canvas. If desired, you can always create a new script file, which may work best when organizing your level. For this tutorial, keep all the scripts in the same flow graph file.

2. In the component list, open the folder `Movement` and drag into the canvas view two of the `MoveEntityTo` nodes.

3. In the component list, open the folder `Debug` and drag into the canvas view two of the `Debug:InputKey` nodes.

4. In the Perspective Viewport, select the tag point entity on the left side of the street (assuming you are facing the block wall). Add this to the canvas view with the `Add Selected Entity` command. Position the nodes so they look like the image in the next step.

5. Select the second tag point entity on the right side of the street and add it to the canvas view, placing it next to the `Movement:MoveEntityTo` node on the right.

6. Select the block between the two tagpoints. On each of the `Movement:MoveEntityTo` nodes, on the `Choose Entity` line, assign the selected block.
7. From the entity:Tagpoint node on the left, click and drag the pos output to Movement:MoveEntityTo node's Destination input.
8. Assign the same linkage sets with the other Tagpoint and MoveEntityTo nodes on the right.
9. Connect the left Debug:InputKey node's Pressed output to Movement:MoveEntityTo node's Start input.
10. On the same set of nodes, connect the Released output to the Stop input.
11. Assign the same linkage set with the other InputKey and MoveEntityTo nodes on the right.

Your nodes should now look like this:

![Diagram of nodes](image)

12. Select one of the MoveEntityTo nodes. In the Node Properties, change Value = 4. Or, you can double-click on a property directly in the node and enter the value there.

![Node properties](image)

13. Select the other MoveEntityTo node and change Value = 4 as well. With the node's ValueType = Speed, setting the value to 4 sets the speed at which the block moves when activated.
14. Select the Debug:InputKey node on the left. For its property Key, enter the letter 'j'. This sets the MoveEntityTo to start and stop movement when the 'j' key is pressed on the keyboard.
15. Select the Debug:InputKey node on the right, set its Key = 'l' (lower case L).
16. Run the level and test that the block moves left and right with the keyboard inputs of j and l. If necessary, adjust the position of the tag points to which the block moves.
Spawn Object

In this script, you'll spawn a physics object to shoot out of a designated spawn point.

1. In the component list, open the folder Entity and select and bring into the canvas the node SpawnArchetype.
2. In the component list, open the folder String and select and bring into the canvas the node SetString.
3. In the component list, open the folder Physics and select and bring into the canvas the node ActionImpulse.
4. In the component list, open the folder Debug and select and bring into the canvas the node InputKey.
5. Select the tag point that was linked to the block in the scene. Add this entity to the the canvas view.
6. Arrange the nodes to look like the following image.

7. Connect the TagPoint node’s Pos output to the SpawnArchetype node’s Pos input.

You are going to spawn the physics ball that you created in the Archetype Entity Database. The object name is entered into the SetString Node.
8. Open the Database View Editor, select the **Entity Library** tab, and open the **Physics** folder. Select the **Ball** entity.

9. Take note of the string: Level.Physics.Ball. This identifies the ball coming from the **Level** file, under the **Physics** folder, and with the name **Ball**.

10. In the Flow graph canvas view, select the **SetString** node. Double-click the **In** property to enter the string: Level.Physics.Ball.

11. Connect the **SetString** node's **Out** output to the **SpawnArchetype** node’s **Archetype** input.

12. Set the **Inputkey** node's property **Key** = i.
13. Connect the `InputKey` node's `Pressed` output to the `SpawnArchetype` node's `Spawn` input.

14. Connect the `SpawnArchetype` node's `Succeeded` output to the `ActionImpulse` node's `Activate` input as well as its `Input Entity` input.

15. In the `ActionImpulse` node, set the `Impulse` values to `X=0`, `Y=5000`, `Z=1000`. If X is forward in your level, swap the X and Y values.

16. Test the level. You should be able to launch a sphere from the block at the block wall and move the block side to side.

Set Impulse Value

In this script, you'll modify the spawn object to shoot a physics object at different velocities based on how long you hold the keyboard input.

1. In the component list, open the folder `Interpolate` and add the node `Int` to the canvas view.

2. In the component list, open the folder `Vec3` and add the node `ToVec3` to the canvas view.

3. In the `Debug:Inputkey` that spawns the physics ball, disconnect the `pressed` output from the `SpawnArchetype` node's `Spawn` input.

   In the same debug node, also:
   - Connect the `Released` output to the `Spawnarchetype` node's `Spawn` input. The ball should now spawn on release of the key.
   - Connect the `Pressed` output to the `Interpolate:Int` node's `Start` input.
   - Connect the `Released` output to the `Interpolate:Int` node's `Stop` input.

4. In the `Interpolate:Int` node, change the `EndValue` = 6000. Also change the `Time` = 2.

5. In the `Vec3:ToVec3` node, change `Z=1000`.

6. Connect the `Interpolate:Int` node's `Value` output to the `Vec3:ToVec3` node's `Y` input. (Or the `X` input, if that is the direction your street is oriented)
7. Connect the Vec3:ToVec3 node's Result output to the ActionImpulse node's Impulse input. The Flow graph script should now look like this:

![Flow graph diagram]

8. With these script modifications, you have changed how much impulse is applied to the spawn objects depending on how long the i key is pressed down. It also sets the maximum impulse output to 6000. As the script is currently, you will not see an output value displayed when you release the i key. The next few steps enable that output value to be displayed.

9. In the component list, open the folder Debug and add the node DisplayMessage.
11. Run the level again and test the inputs. You will see that the length of input affects the ball's spawn velocity. The impulse output value should also be displayed in the upper left of the screen.

**Any Nodes**

Next, you will create a script that disables the cannon and its inputs until it has been triggered with the Camera switch.

1. In the component list, open the folder Logic and add three of the Any nodes to the canvas view. Place one of these by each of the Debug:InputKey nodes.

2. Down by the Debug:InputKeys that control the block moving left and right, add a Game:Start node by right-clicking and selecting Add Start Node.

3. Connect the Game:Start node's output output to both Logic:Any nodes' In0 inputs.

4. Connect the first Any node's Out output to the Debug:InputKey node's Disable input. Do the same for the second Logic:Any node, as shown in the following image.
5. Connect the **Entity:ProximityTrigger** node’s **Enter** output to each of the two **Debug:InputKey** nodes’ **Enable** inputs.

6. Connect the **ProximityTrigger** node’s **Leave** output each of the two **Logic:Any** node’s **In1** inputs.

The ability to move the block left and right is now configured to be active only when the Proximity Trigger is activated. The inputs are disabled when leaving the proximity triggers.

The **Any** node enables multiple inputs to be routed into a single output. Because any given input port can accept only one input, using the **Any** node allows multiple inputs to be collated into a single output.

7. Assign the same linkages for the **Debug:InputKey** node that controls the launching of the physics ball.
Debug View

If your script does not work as you expect, you can turn on Debug in the Flow graph editor. Debug is a helpful tool to detect where your logic has failed.

To turn on Debug:

1. In the Flow Graph Editor's toolbar, select the Debug icon.

2. Move the Flowgraph window to the side (but still visible).

3. Run the level (Ctrl+G). Move around in the level and trigger the camera switch. Move and shoot from the block. Notice how each action in the level is displayed in the Flowgraph events.

4. Exit the level. Select the Trash icon (next to the Debug icon) to clear the debug events.

5. Save your file.

You have completed the Flow graph scripting of your level.

You now have a fully working level that allows you to move a third person character around an environment that you created, trigger a camera that switches to a new view, and turn on the control of a box that launches a ball at a wall of blocks.

Conclusion

This concludes Lumberyard's Getting Started Guide Tutorial. Using this set of tutorials, you have begun to understand the power of Lumberyard Editor. You learned about and used features that you will use regularly while making your games. When you are comfortable with these basic features, browse our tutorial pages to learn even more about Lumberyard's advanced tools and features. There are thousands of games to be made!