BIMteractive: Pushing the Boundaries of Traditional Walkthroughs

Leo Salcé, Intl Assoc AIA, LEED AP – Senior Consultant, Microdesk
Peter Marchese, LEED Green Associate – Senior Consultant, Microdesk

AV3752 Today clients expect to be "wowed." This means that you as a designer need to find ways to do more with your work and help your audience or clients better understand your designs and intentions. Generating compelling, interactive 3D in presentations is a way to really engage and help clients and stakeholders visualize spaces and designs, and in turn help you win more work. We'll use real-life project examples to show you how to effectively integrate a workflow that takes advantage of your existing Autodesk® Revit or AutoCAD® Civil 3D® software models to create an interactive walkthrough that is as easy and smooth to navigate as a modern video game and can be used for client, real estate, or town board presentations. We will discuss some of the challenges that designers may encounter when getting started, but how ultimately, anyone can take advantage of technology to produce professional presentations that can help build more business. Learn about available options for gaming engines to get the most value from your project intelligent models.

Learning Objectives

At the end of this class, you will be able to:

- Use your production models for marketing and presentation purposes through less traditional workflows
- Explain the benefits of having a more interactive method of communicating designs and winning new business
- Describe a practical workflow for integrating your intelligent models in a game engine environment for producing interactive navigation
- Explain the challenges and overcome them to effectively take full advantage of various 3D visualizations tools

About the Speakers:

Leo Salcé, Int Assoc AIA, LEED AP, is a Senior Consultant with Microdesk. He specializes in Building Information Modeling technology Implementation for Architecture and Engineering firms both nationally and internationally and focuses on Strategic CAD/BIM planning, Standards Development and Multi-platform integration to better leverage existing and emerging Design and Construction technology. He is a designer with over 12 years of experience in Building Design and BIM/CAD Management and has served as a technology advisor partner to hundreds of AEC firms, providing Planning, training and technical methodology strategies on a wide range of AEC platform solutions. He holds degrees in Architecture and Computer Science from Universidad Católica Madre y Maestra in the Dominican Republic, where he is also a Registered Architect. He is an active educator, speaker and published writer on Building information modeling and sustainable design techniques.

linkedIn *http://www.linkedin.com/in/leosalce*
e-mail *lsalce@microdesk.com*

Peter Marchese, LEED Assoc, is an Senior Consultant with Microdesk, providing support to national AEC firms in implementing BIM. He also specializes in providing implementation, custom content creation, consulting services, customized training, and leading firms through the process of creating standards and workflows based on building information modeling technology. Prior to joining Microdesk, Peter worked at Architecture firms working on residential, institutional, liturgical, and commercial projects. He has managed projects throughout all phases from design through construction documentation. Completed projects include several laboratory/research buildings, residences, libraries, movie theaters, and a new processing and distribution center for the USPS. Peter has extensive working knowledge of AutoCAD® and Revit for construction documents, presentations, animations, and renderings. He holds a bachelor of science degree in Architecture from Drexel University in Philadelphia, Pennsylvania

linkedIn www.linkedin.com/in/petermarchese
e-mail pmarchese@microdesk.com

Introduction

Over the last decade we have seen great advances in the way technology can help visualize a design concept from scanning a sketch and building a model out of that or using photogrammetry to generate a model out of photographs, to laser scanning and generating point cloud usable 3D data, but no matter how the 3D model is created, it's the way you experience the 3D scene in front of you what really conveys the idea, concept or story.

In the AEC Industry there have been several ways to navigate and explore a 3D BIM model through publicly available methods like Rendered Animations, Navigation in Real Time Rendering, 360 Panoramic Navigation and even Augmented Reality.

What has been lagging in the way we navigate a 3D model is the way we interact with the objects in a scene in Real Time by having actions triggered by the user and a reaction as a consequence, making the experience more immersive.

Through the use of Game Engine technology we will explore how a BIM model from any discipline can be explored and Navigated in a more immersive way and how to push the boundaries of Traditional Walkthroughs.

The amazing thing is not so much that this is available now, but that it is just starting to catch on. The first time I used a game engine to show an architectural design was in 1996. That's a much longer time than I would like to think about, but the tools have been around for a while, similar to how BIM was around for manufacturing long before the AEC adopted it. With Games becoming a huge business and being seen as less of a kids toy and more legitimate more people are willing to see what they can be used for as opposed to just something to spend a few hours playing. The technology behind games are being used for medicine and rehabilitation, military training, education, and many more innovative ways. As for AEC a very large and high profile example may be the new Dallas Cowboys stadium designed by HKS, and visualized using the Unreal engine.

Table of Contents

What is a Game Engine? 4
Why Visualization with Game Engines? 4
Before we begin - Understanding Unity 5
Setting up a Project in Unity 6
Assets 7
Importing your BIM Model from Revit as an Asset 7
Importing your BIM Model from AutoCAD Architecture/MEP/Civil3D as an Asset 10
Best Practices for Exporting out of Revit and Recommended Workflow 11
Working with the Site 14
Adding a Sky 18
Adding Sun light and Shadows 20
Setting up your Character Controller to Navigate your scene 22
Publishing the Game 27
Working with Lighting 30
Working with Audio 32
Working with Materials and Shaders 33
Prefabs 40
Setting up the Controllers 41
Conclusion 48
Glossary and Links: 48

What is a Game Engine?

A game engine is a system designed for the creation and development of video games. The leading game engines provide a software framework that developers use to create games for video game consoles and personal computers. The core functionality typically provided by a game engine includes a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, and a scene graph.

Commercially Popular available engines

Why Visualization with Game Engines?

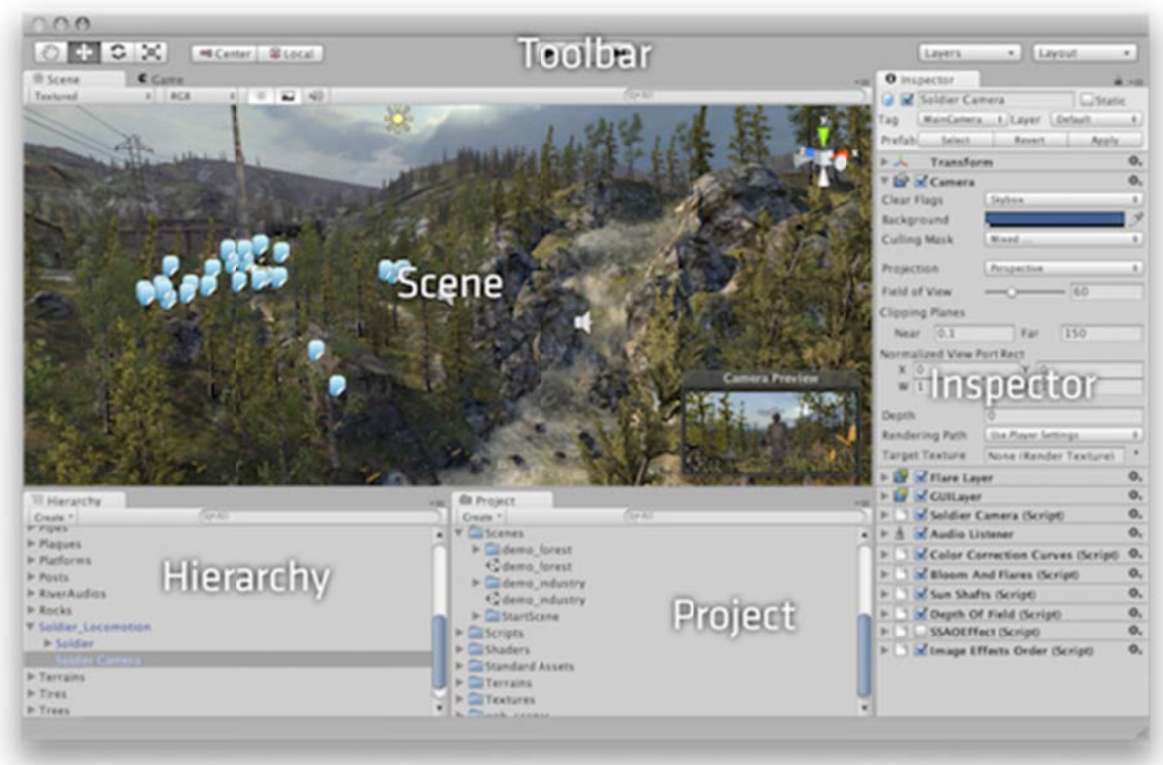
Real Time Visualization and Exploration
Built-in Interactivity in the Presentation
Flexible Interaction by Combining Assets: Videos, Sound, Textures, Lighting to act when triggered by an action
Easy Sharing of the final output (web based or .exe viewing)
Suitable First and Third Person Perspectives
Based on the Complexity Little to None Scripting Knowledge (Assets)
Supports model reference for constant design iteration from the BIM models

Why Unity?

It is asset-centric rather than code-centric, placing the focus on the assets in much the same way as a 3D modeling application and not so much in a Code Editor.
Affordable and easy to use right out of the box.
Ability to import and utilize CAD and BIM models as well as Support update Iterations.
Format Support
Strong Asset Store and Lots of Documentation

Before we begin - Understanding Unity

The Main Editor Window is made up of several Tabbed Windows, called Views. They all have a specific purpose.



The Project View is where you store all the assets that make up your game (Project), like scenes, scripts, 3D models, textures, audio files, and Prefabs (we'll discuss this one later)

The Hierarchy View contains every Game Object in the current Scene

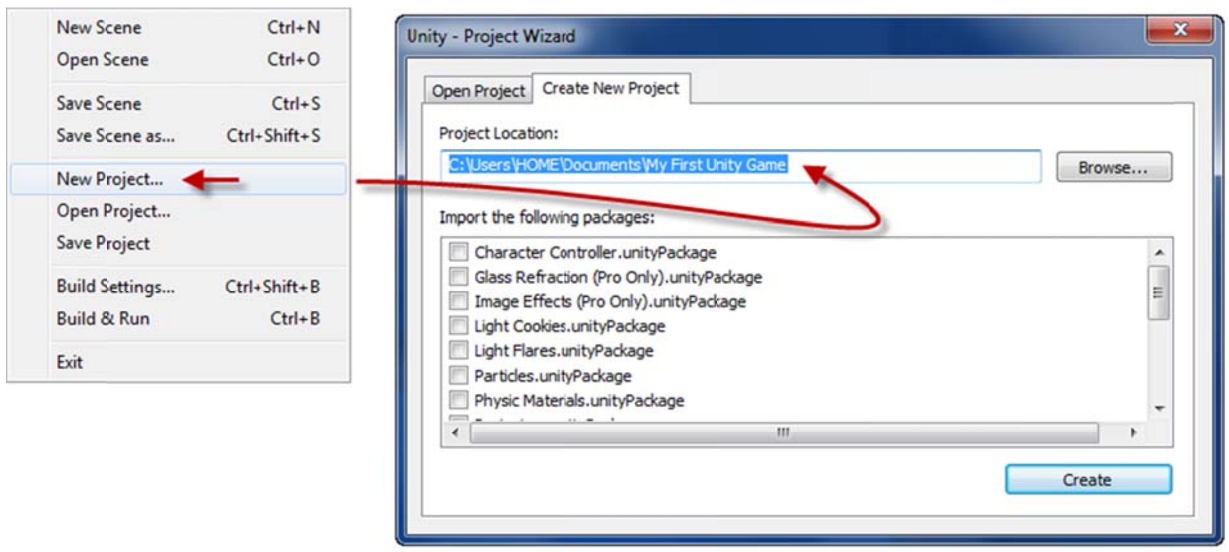
The Scene View is your interactive space. You will use the Scene View to select and position environments, the player, the camera, and all other Game Objects

The Inspector displays detailed information about your currently selected Game Object, including all attached Components and their properties

The Toolbar consists of five basic controls. Each relate to different parts of the Editor.

Setting up a Project in Unity

To setup a project in your file menu click on open project

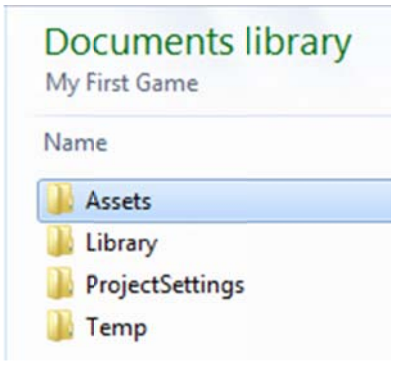


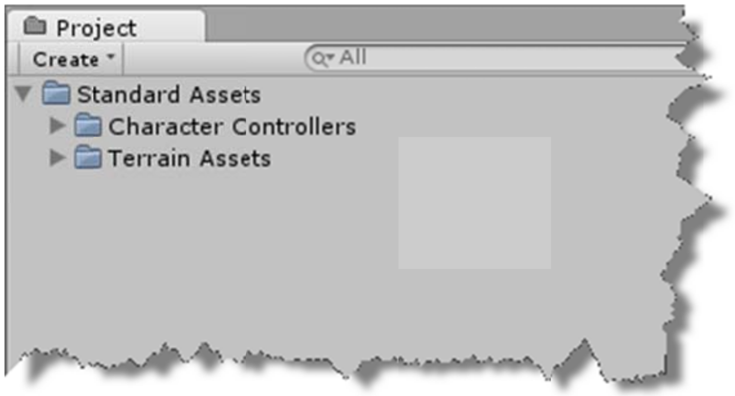
Project Location:
C:\Users\HOME\Documents\My First Unity Game

Import the following packages:
- Character Controller.unityPackage
- Glass Refraction (Pro Only).unityPackage
- Image Effects (Pro Only).unityPackage
- Light Cookies.unityPackage
- Light Flares.unityPackage
- Particles.unityPackage
- Physic Materials.unityPackage

You may include some packages now (will discuss this in a bit), or you can import as needed later on. We will include the Character Controller and Terrain Assets for now.

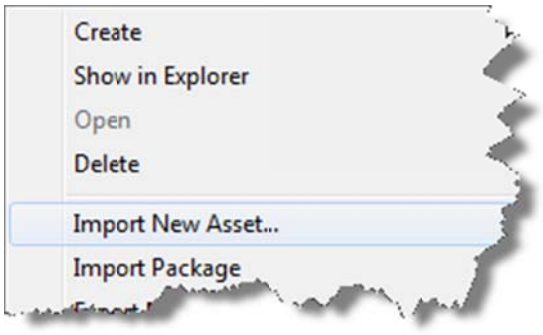
Every Unity project contains an **Assets** as well as other project folders.





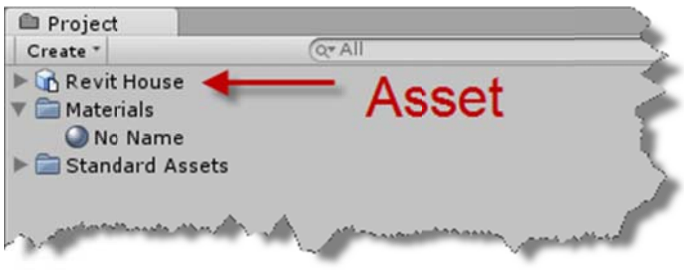
Assets

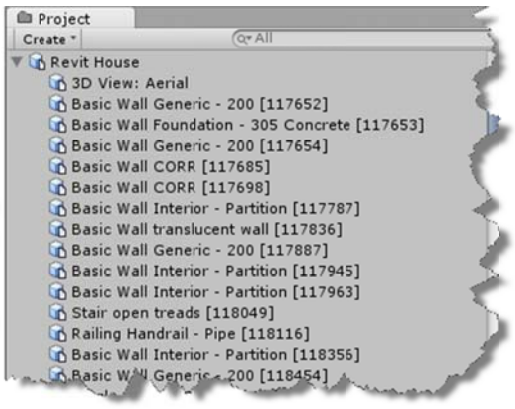
All your imported Assets appear in the Project Pane and they can be almost anything:
Your BIM model, a simple material or texture, audio files, or even a complete, prefabricated Game Objects (known as a "Prefab").
Unity gives you the freedom to organize your project's assets in the way that makes more sense to you. You may want to organize by asset type with folders like Textures, Models, Sound, etc or by function.
Unity can read **.FBX, .dae** (Collada), **.3DS, .dxf and .obj** files, so any software that that can export any of these formats should work with Unity
To Import an asset just go to the Assets menu and click on the Import New Asset.



Importing your BIM Model from Revit as an Asset

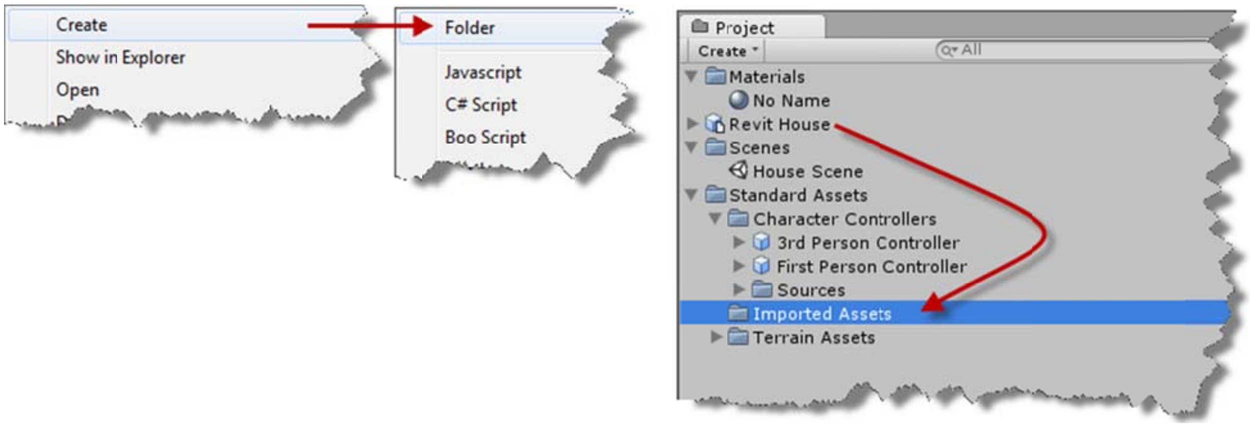
When you save your asset initially, you should save it normally to the Assets folder in your Project folder. When you open the Unity project, the asset will be detected and imported into the project. When you look in the Project View, you'll see the asset located there, right where you saved it. Please note that Unity uses the FBX exporter provided by your modeling package to convert your models to the FBX file format. You will need to have the FBX exporter of your modeling package available for Unity to use. Alternatively, you can directly export as FBX from your application and save in the Projects folder.
As you can see below the Imported Asset (Revit model) carries the geometry information.



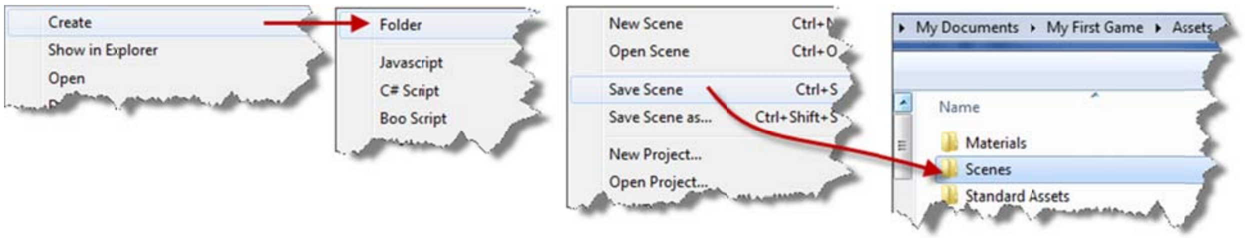


Revit House
3D View: Aerial
Basic Wall Generic - 200 [117652]
Basic Wall Foundation - 305 Concrete [117653]
Basic Wall Generic - 200 [117654]
Basic Wall CORR [117685]
Basic Wall CORR [117698]
Basic Wall Interior - Partition [117787]
Basic Wall translucent wall [117836]
Basic Wall Generic - 200 [117887]
Basic Wall Interior - Partition [117945]
Basic Wall Interior - Partition [117963]
Stair open treads [118049]
Railing Handrail - Pipe [118116]
Basic Wall Interior - Partition [118356]
Basic Wall Generic - 200 [118454]

To organize your imported assets in one location for easy instancing you can create a folder and name it something like Imported assets. To do this right click on Standard Assets and create a new folder.

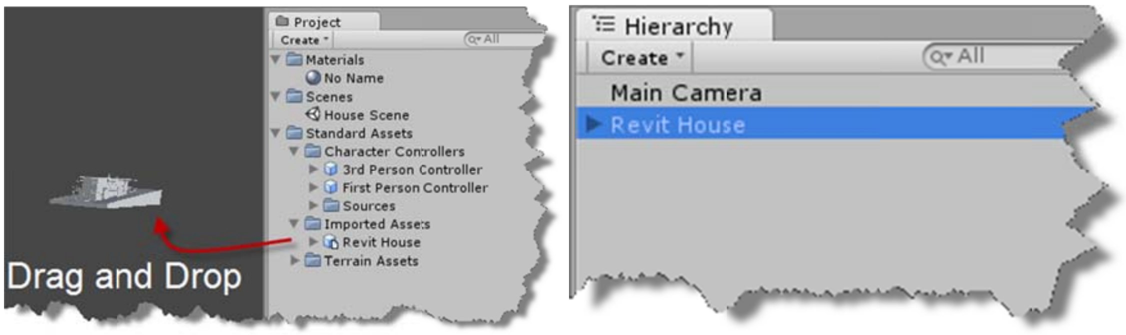


Before moving further, We recommend organizing your project in either a scene or scenes. To do this create a new folder and name it scenes and name according to what it's representing, just the Site, the building or the whole project. In gaming a single Scene will contain a single game level. **Scenes** are also stored in the Project View.

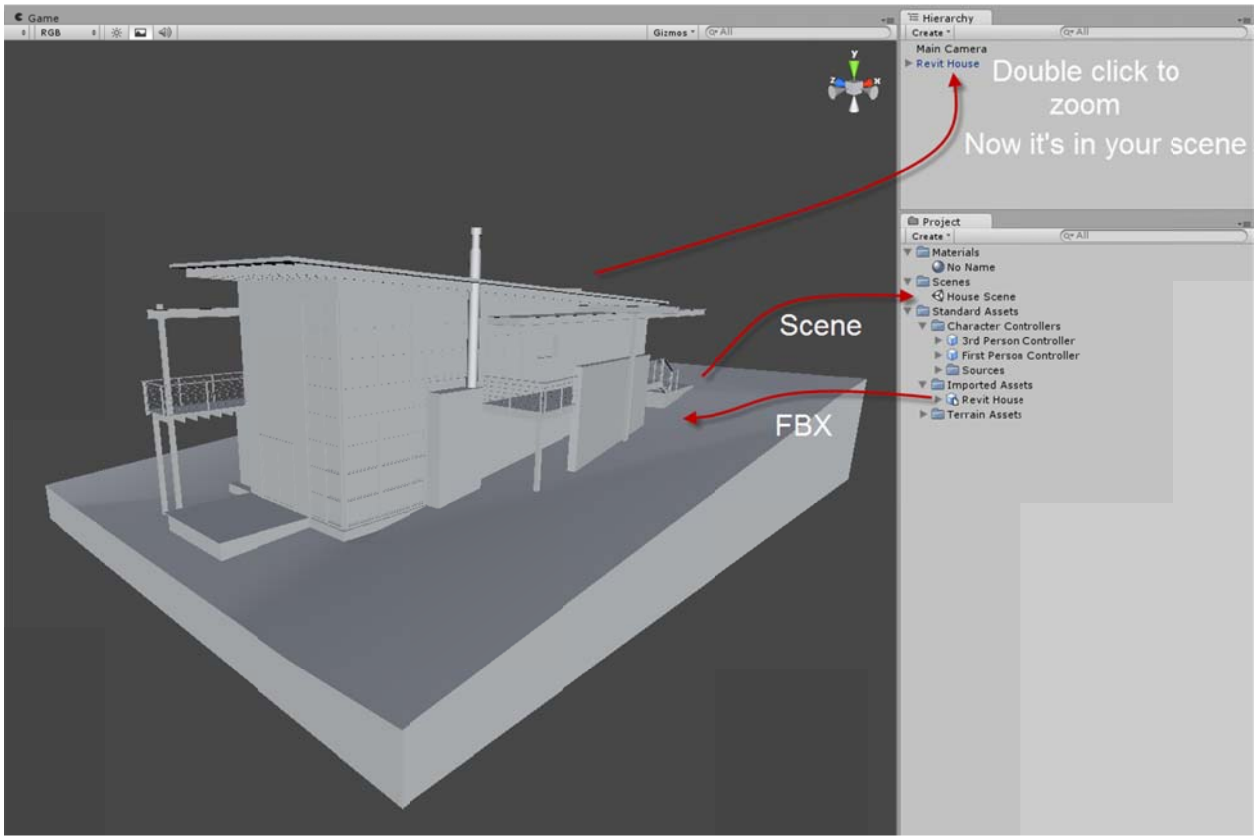


Adding an Asset to the Scene

Simply click and drag the mesh from the Project View to the Hierarchy or Scene View to add it to the Scene. When you drag a mesh to the scene, you are creating a Game Object that has a Mesh Renderer Component. If you are working with a texture or a sound file, you will have to add it to a Game Object that already exists in the Scene or Project.



The Hierarchy contains every Game Object in the current Scene. Some of these are direct instances of asset files like 3D models, and others are instances of Prefabs (we'll discuss this soon), custom objects that will make up much of your game.



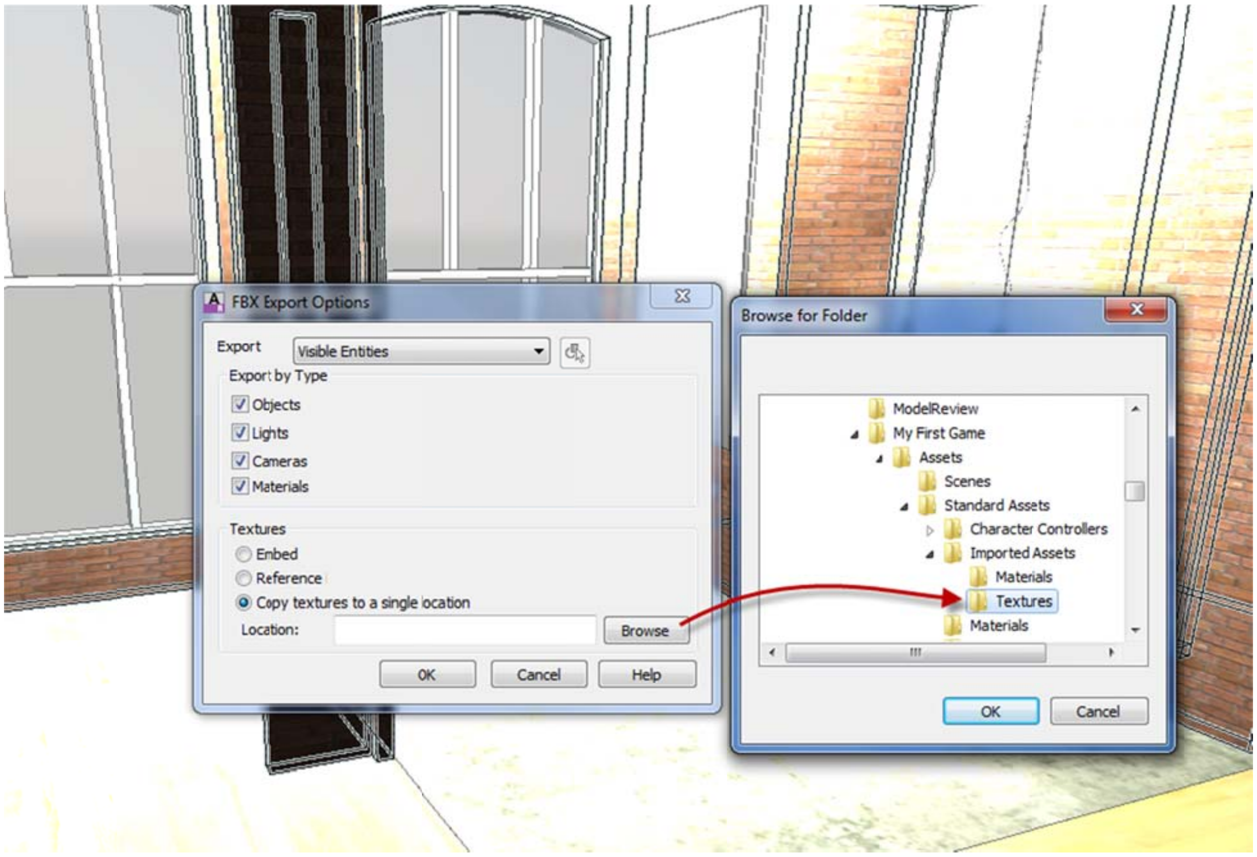
You should **never** move project assets around using the OS since this will break any metadata associated with the asset. **Always** use the Project View to organize your assets.

NOTE:
Importing a Revit model straight into Unity is not really the proper workflow if you are using Revit 2011 and newer. If you noticed textures coming out of Revit did no come in, this is due partly to the updated Autodesk Material libraries which are not the same as the material libraries in other previous versions and because of this Unity is not able to recognize such. This can be taken care of in a couple of ways which will describe shortly.

Exporting your BIM Model from AutoCAD Architecture/MEP/Civil3D as an Asset

Similar to how you would do this in Revit, please export out of the AutoCAD based platform application.
To export out of AutoCAD Architecture/MEP we recommend the export command **FBXEXPORT** rather than using the file menu to export as fbx.

Include all objects you want to export out from ACA/MEP so they get converted and for the textures, if you want to skip the 3ds max recommended workflow and have the textures show for manual texture assignment then specify the Project Texture folder as shown below.

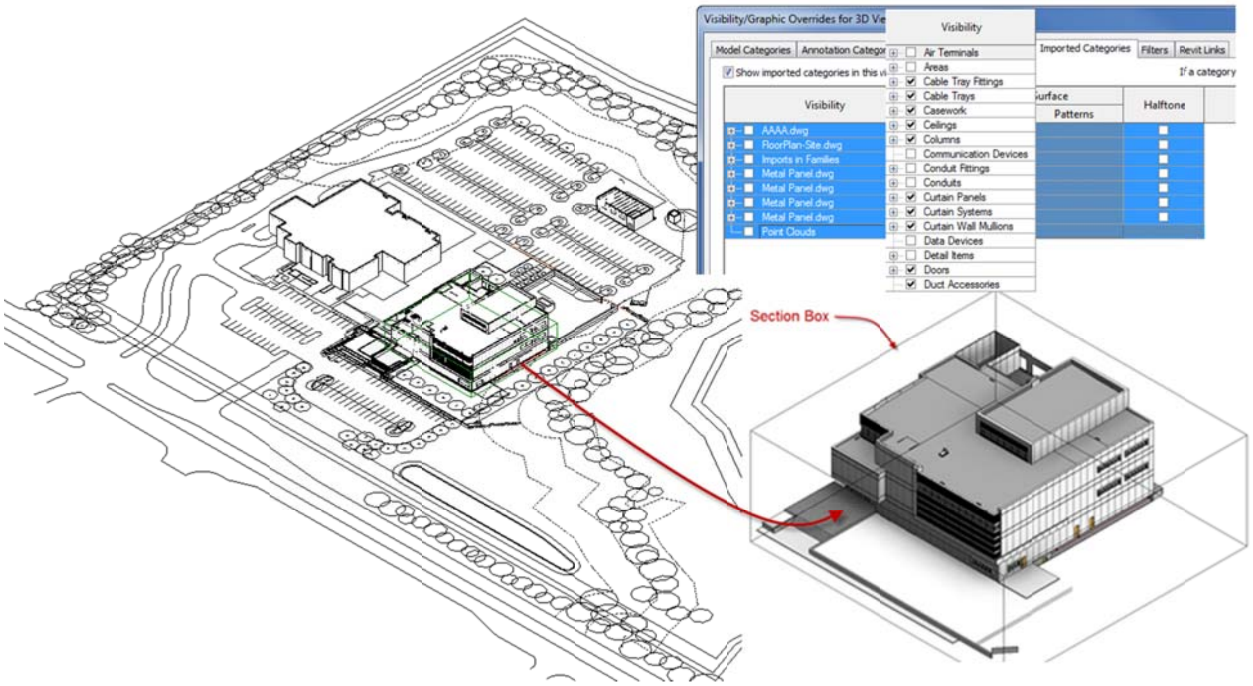


Best Practices for Exporting out of Revit and Recommended Workflow

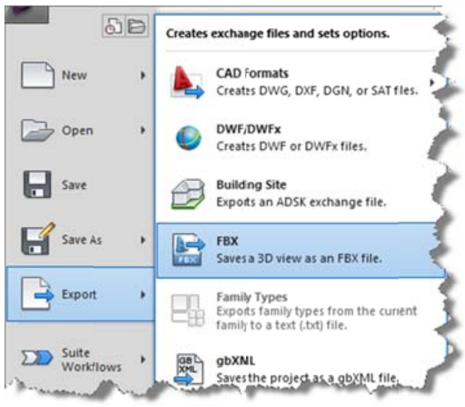
Below is a common recommended workflow to get your Revit Models as assets into Unity.

A. Model Preparation

- Visibility/ Graphics Settings
- Section Box
- Getting rid of unnecessary details



B. Export as FBX from Revit



Creates exchange files and sets options.

New
Open
Save
Save As
Export
Suite Workflows

CAD Formats
Creates DWG, DXF, DGN, or SAT files.

DWF/DWFx
Creates DWF or DWFx files.

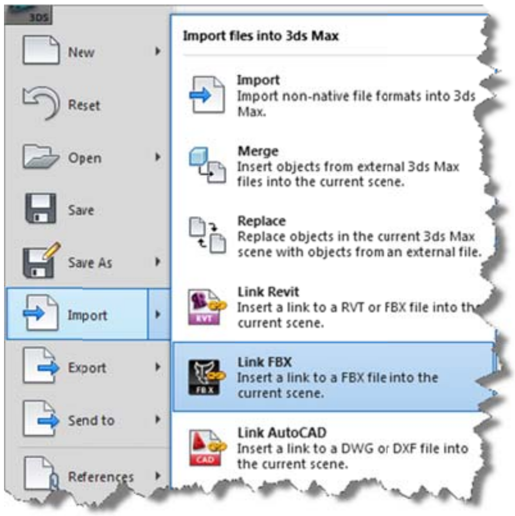
Building Site
Exports an ADSK exchange file.

FBX
Saves a 3D view as an FBX file.

Family Types
Exports family types from the current family to a text (.txt) file.

gbXML
Saves the project as a gbXML file.

C. Import into Autodesk 3ds max



Import files into 3ds Max

New
Reset
Open
Save
Save As
Import
Export
Send to
References

Import
Import non-native file formats into 3ds Max.

Merge
Insert objects from external 3ds Max files into the current scene.

Replace
Replace objects in the current 3ds Max scene with objects from an external file.

Link Revit
Insert a link to a RVT or FBX file into th current scene.

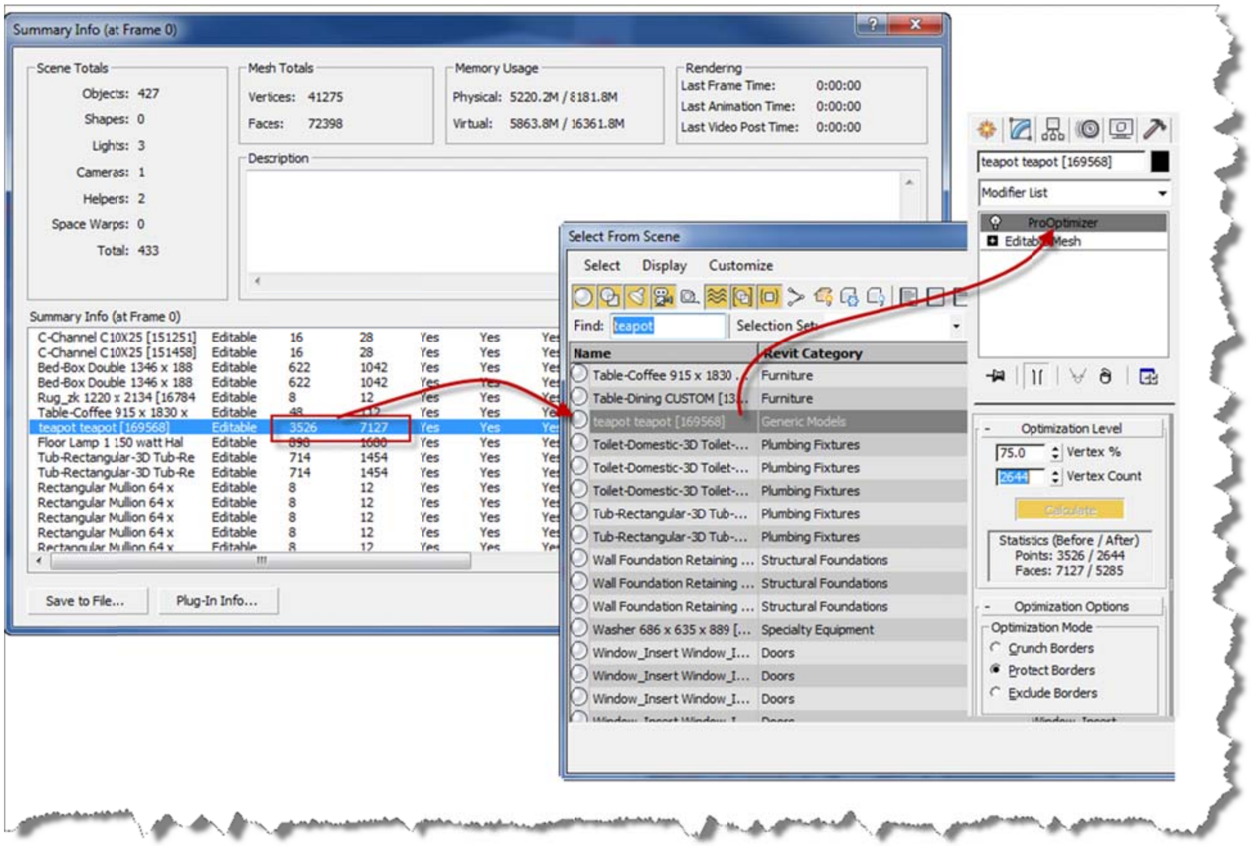
Link FBX
Insert a link to a FBX file into the current scene.

Link AutoCAD
Insert a link to a DWG or DXF file into the current scene.

Import into Autodesk 3ds max Cont.

1.Reduce Polygon Count.

There are several methods to detect high polygon count objects in your model and reduce its complexity without compromising quality. One method is applying a Pro optimizer modifier.


Summary Info (at Frame 0)

Scene Totals: Objects: 427; Shapes: 0; Lights: 3; Cameras: 1; Helpers: 2; Space Warps: 0; Total: 433

Mesh Totals: Vertices: 41275; Faces: 72398

Memory Usage: Physical: 5220.2M / 8181.8M; Virtual: 5863.8M / 16361.8M

Rendering: Last Frame Time: 0:00:00; Last Animation Time: 0:00:00; Last Video Post Time: 0:00:00

Summary Info (at Frame 0)

C-Channel C10X25 [151251]	Editable	16	28	Yes	Yes	Yes
C-Channel C10X25 [151458]	Editable	16	28	Yes	Yes	Yes
Bed-Box Double 1346 x 188	Editable	622	1042	Yes	Yes	Yes
Bed-Box Double 1346 x 188	Editable	622	1042	Yes	Yes	Yes
Rug_zk 1220 x 2134 [16784	Editable	8	12	Yes	Yes	Yes
Table-Coffee 915 x 1830 x	Editable	48	112	Yes	Yes	Yes
teapot teapot [169568]	Editable	3526	7127	Yes	Yes	Yes
Floor Lamp 1 150 watt Hal	Editable	898	1680	Yes	Yes	Yes
Tub-Rectangular-3D Tub-Re	Editable	714	1454	Yes	Yes	Yes
Tub-Rectangular-3D Tub-Re	Editable	714	1454	Yes	Yes	Yes
Rectangular Mullion 64 x	Editable	8	12	Yes	Yes	Yes
Rectangular Mullion 64 x	Editable	8	12	Yes	Yes	Yes
Rectangular Mullion 64 x	Editable	8	12	Yes	Yes	Yes
Rectangular Mullion 64 x	Editable	8	12	Yes	Yes	Yes
Rectangular Mullion 64 x	Editable	8	12	Yes	Yes	Yes

Save to File... Plug-In Info...

Select From Scene

Find: teapot

Name	Revit Category
Table-Coffee 915 x 1830 ...	Furniture
Table-Dining CUSTOM [13...	Furniture
teapot teapot [169568]	Generic Models
Toilet-Domestic-3D Toilet-...	Plumbing Fixtures
Toilet-Domestic-3D Toilet-...	Plumbing Fixtures
Toilet-Domestic-3D Toilet-...	Plumbing Fixtures
Tub-Rectangular-3D Tub-...	Plumbing Fixtures
Tub-Rectangular-3D Tub-...	Plumbing Fixtures
Wall Foundation Retaining ...	Structural Foundations
Wall Foundation Retaining ...	Structural Foundations
Wall Foundation Retaining ...	Structural Foundations
Washer 686 x 635 x 889 [...	Specialty Equipment
Window_Insert Window_I...	Doors
Window_Insert Window_I...	Doors
Window_Insert Window_I...	Doors

teapot teapot [169568]

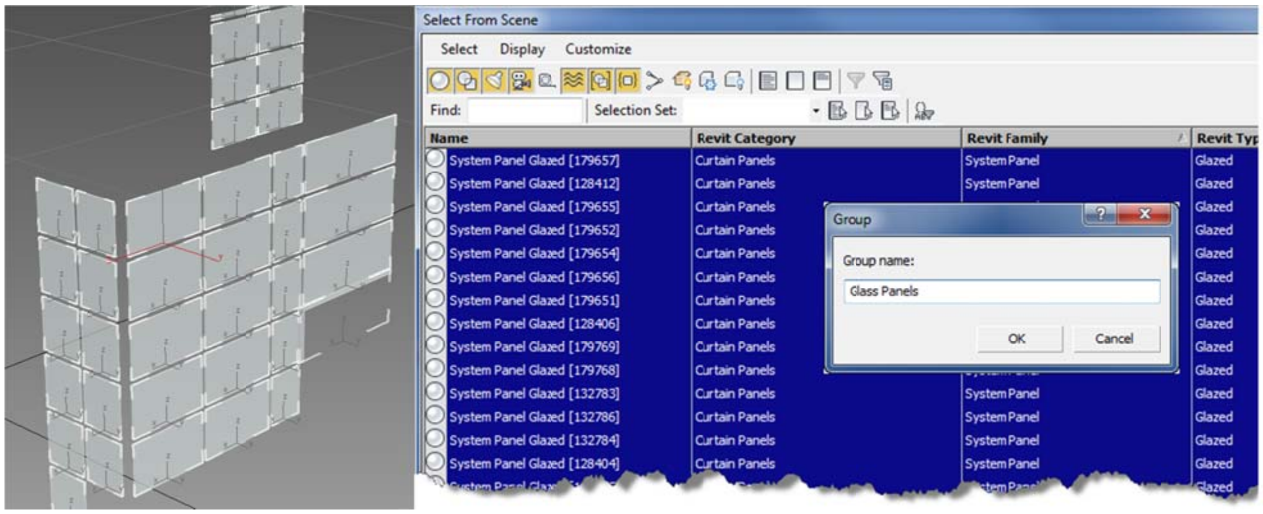
Modifier List: ProOptimizer, Editable Mesh

Optimization Level: Vertex % 75.0; Vertex Count 2644

Statistics (Before / After)
Points: 3526 / 2644
Faces: 7127 / 5285

Optimization Options – Optimization Mode: Crunch Borders [✓ Protect Borders] Exclude Borders

2.Group objects as systems


Select From Scene

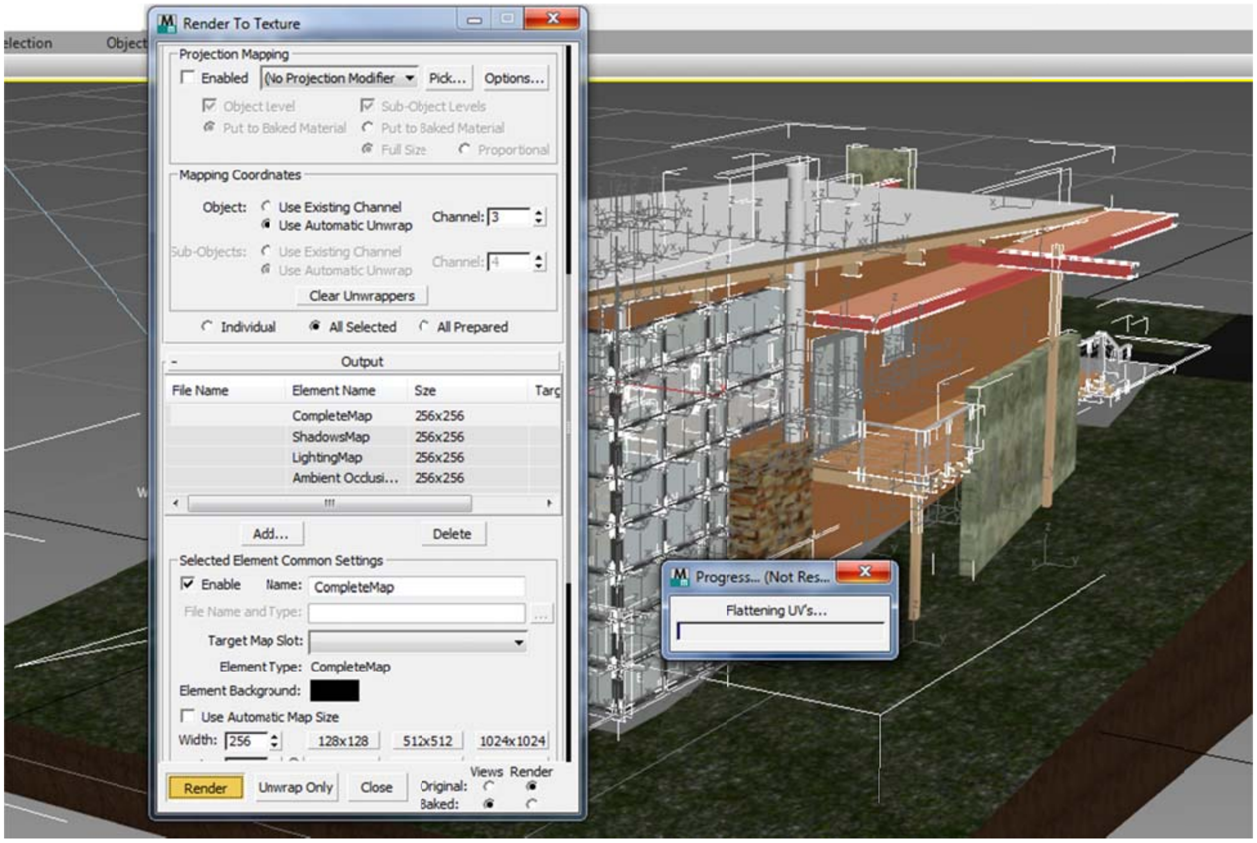
Name	Revit Category	Revit Family	Revit Typ
System Panel Glazed [179657]	Curtain Panels	System Panel	Glazed
System Panel Glazed [128412]	Curtain Panels	System Panel	Glazed
System Panel Glazed [179655]	Curtain Panels		Glazed
System Panel Glazed [179652]	Curtain Panels		Glazed
System Panel Glazed [179654]	Curtain Panels		Glazed
System Panel Glazed [179656]	Curtain Panels		Glazed
System Panel Glazed [179651]	Curtain Panels		Glazed
System Panel Glazed [128406]	Curtain Panels		Glazed
System Panel Glazed [179769]	Curtain Panels		Glazed
System Panel Glazed [179768]	Curtain Panels		Glazed
System Panel Glazed [132783]	Curtain Panels	System Panel	Glazed
System Panel Glazed [132786]	Curtain Panels	System Panel	Glazed
System Panel Glazed [132784]	Curtain Panels	System Panel	Glazed
System Panel Glazed [128404]	Curtain Panels	System Panel	Glazed

Group

Group name: Glass Panels

C. Render to Texture (If you want to bake texture as well as lighting into your geometry)

Rendering to texture, or "texture baking," allows you to create texture maps based on an object's appearance in the rendered scene. The textures are then "baked" into the object: that is, they become part of the object via mapping, and can be used to display the textured object rapidly on Direct3D devices such as graphics display cards or game engines.



Render To Texture

Projection Mapping: Enabled [ ] (No Projection Modifier) — Pick... — Options...
Object Level [✓] — Sub-Object Levels [✓]
[✓ Put to Baked Material] — Put to Baked Material — [✓ Full Size] Proportional

Mapping Coordnates
Object: Use Existing Channel [✓ Use Automatic Unwrap] — Channel: 3
Sub-Objects: Use Existing Channel [✓ Use Automatic Unwrap] — Channel: 4
Clear Unwrappers
Individual [✓ All Selected] All Prepared

Output

File Name	Element Name	Size
	CompleteMap	256x256
	ShadowsMap	256x256
	LightingMap	256x256
	Ambient Occlusi...	256x256

Add... — Delete

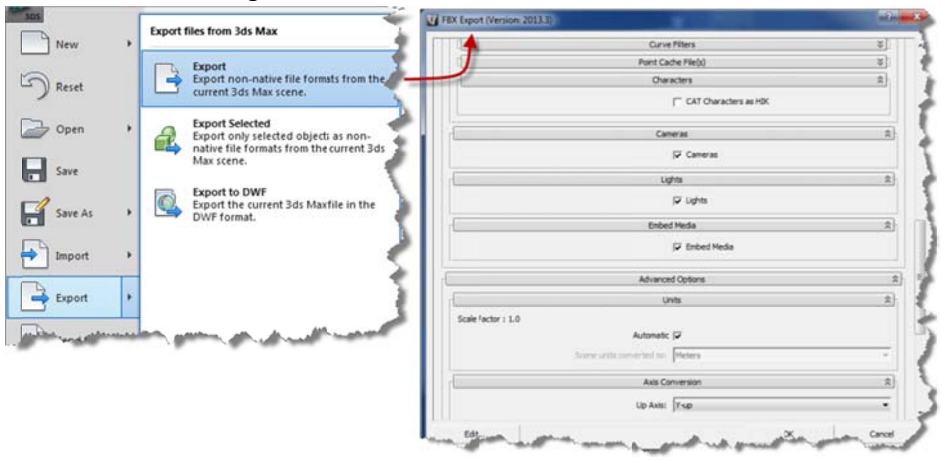
Selected Element Common Settings
Enable [✓] — Name: CompleteMap
File Name and Type:
Target Map Slot:
Element Type: CompleteMap
Element Background:
Use Automatic Map Size [ ]
Width: 256 — 128x128 — 512x512 — 1024x1024

Render — Unwrap Only — Close
Views / Render — Original: Views [✓ Render] — Baked: [✓ Views] Render

Progress... (Not Res...
Flattening UV's...

D. Export as FBX

- Review FBX settings



Export files from 3ds Max

Export Export non-native file formats from the current 3ds Max scene.

Export Selected Export only selected objects as non-native file formats from the current 3ds Max scene.

Export to DWF Export the current 3ds Maxfile in the DWF format.

FBX Export (Version: 2013.3)

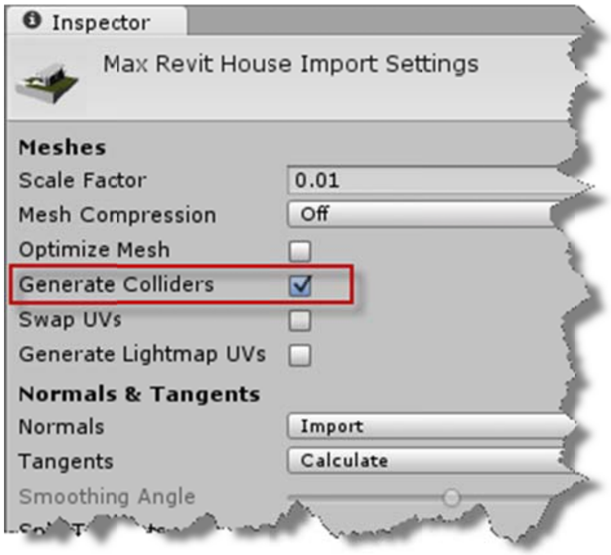
Curve Filters
Point Cache File(s)
Characters: CAT Characters as HIK [ ]
Cameras: Cameras [✓]
Lights: Lights [✓]
Embed Media: Embed Media [✓]
Advanced Options
Units: Scale factor : 1.0 — Automatic [✓] — Scene units converted to: Meters
Axis Conversion: Up Axis: Y-up

E. Import into Unity

- Import as new asset
- Change settings in Inspector

Use the method above to import the Max exported FBX as an Asset.

Note:
Once you bring in the new imported Asset. When you select the model and enable **Generate Colliders** in the Import Settings, a Mesh collider is automatically added when the mesh is added to the Scene. It will be considered solid as far as the physics system is concerned.


Inspector
Max Revit House Import Settings
Meshes
Scale Factor 0.01
Mesh Compression Off
Optimize Mesh ☐
Generate Colliders ✓
Swap UVs ☐
Generate Lightmap UVs ☐
Normals & Tangents
Normals Import
Tangents Calculate
Smoothing Angle

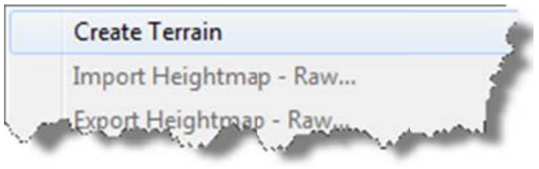
Tip:
Merge your meshes together as much as possible. Make them share materials and textures. This has a huge performance benefit.

Working with the Site

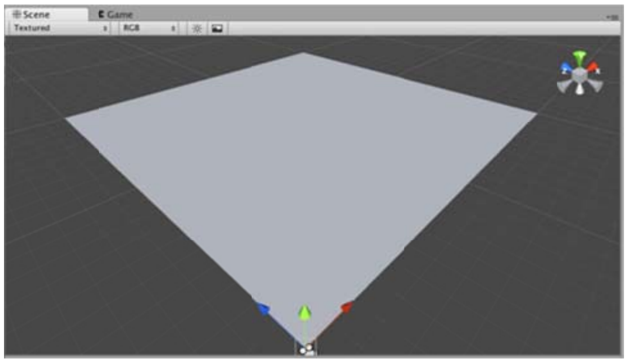
When working with a bigger land area which has not been modeled and when not working with a Site from tools like Civil 3D, then the Terrain tool may be very beneficial and quick to generate.

To create a Terrain in Unity do the following:

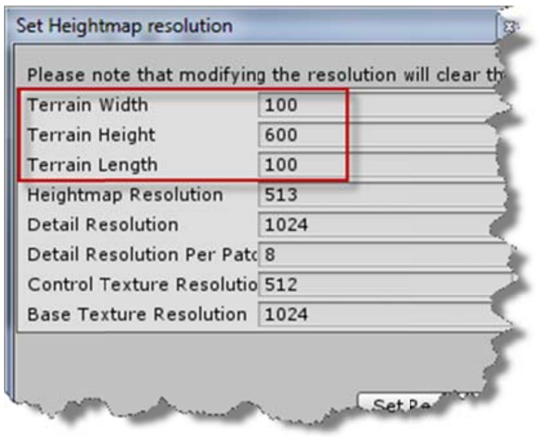
1. Click on the Terrain Menu and chose CreateTerrain



2. Your terrain will look like the scene below.



3. To adjust the size of the terrain and the Resolution of your terrain click on the Terrain Menu and chose Set Resolution

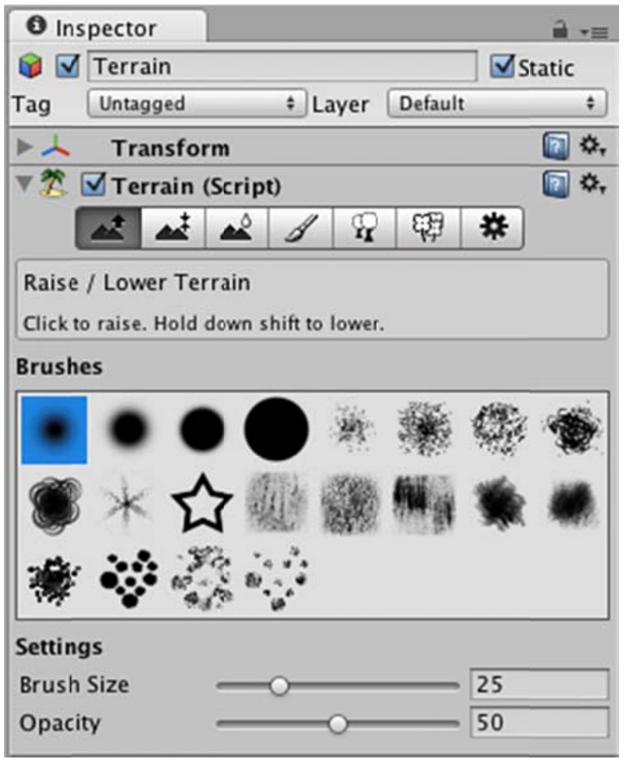


Set Heightmap resolution

Please note that modifying the resolution will clear th

Terrain Width	100
Terrain Height	600
Terrain Length	100
Heightmap Resolution	513
Detail Resolution	1024
Detail Resolution Per Patc	8
Control Texture Resolutio	512
Base Texture Resolution	1024

4. You can use Brushes to paint and manipulate your Terrain. If you want to reposition a Terrain, you can modify its Transform Position values in the Inspector.



Inspector

Terrain ✓ Static

Tag Untagged Layer Default

Transform

✓ Terrain (Script)

Raise / Lower Terrain

Click to raise. Hold down shift to lower.

Brushes

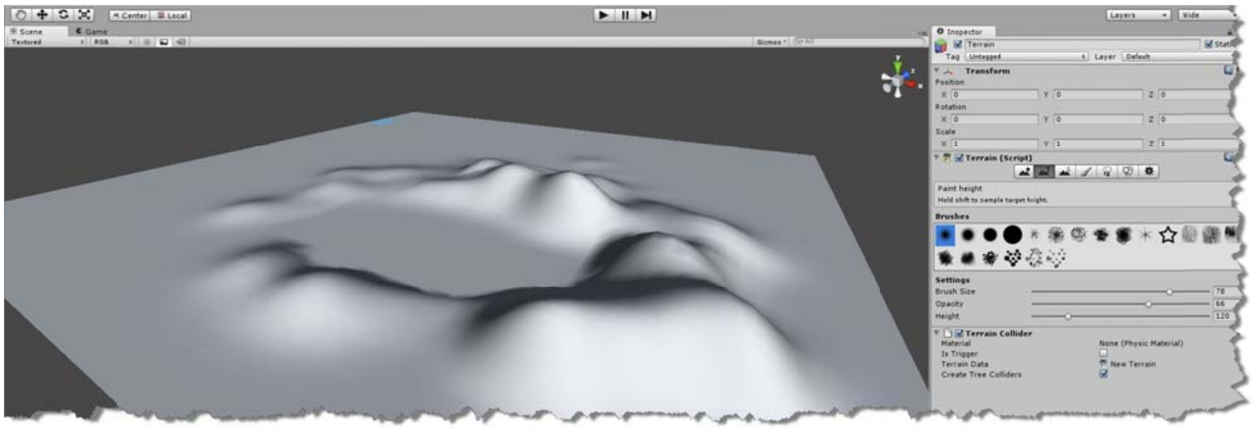
Settings

Brush Size 25

Opacity 50

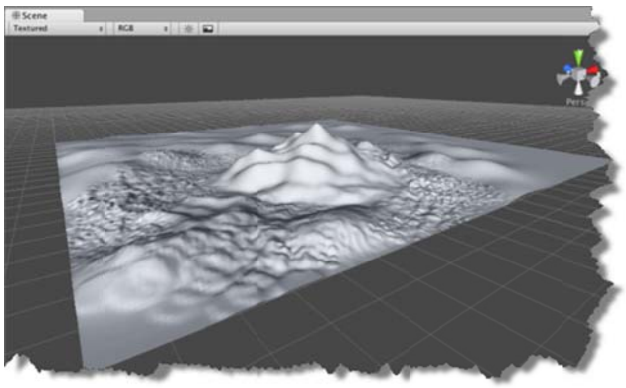
You can Raise and lower the height of your terrain with different brush types and strokes

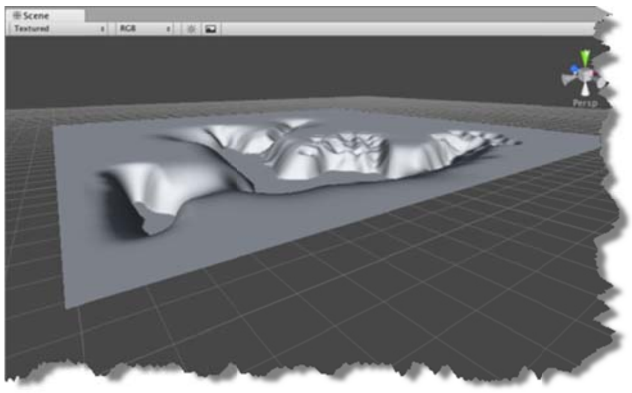
And use any of the brushes to achieve different results



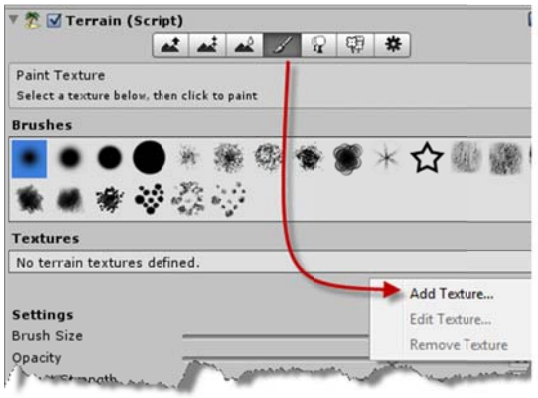
5. Chose a different stroke for a different effect

6. Hold the shift key and press to invert the brush push





7. Before you can begin painting Terrain Textures, you will add at least one to the Terrain from your Project folder.



Terrain (Script)

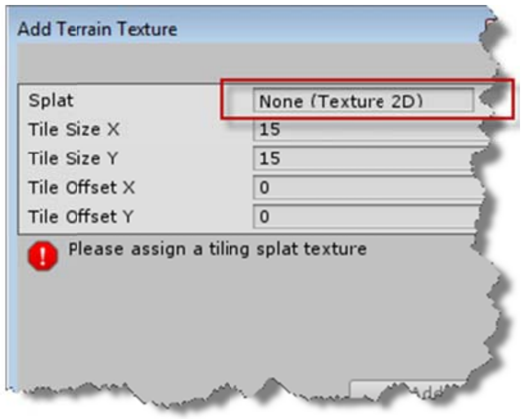
Paint Texture
Select a texture below, then click to paint

Brushes

Textures
No terrain textures defined.

Settings
Brush Size
Opacity

8. Add your texture from the

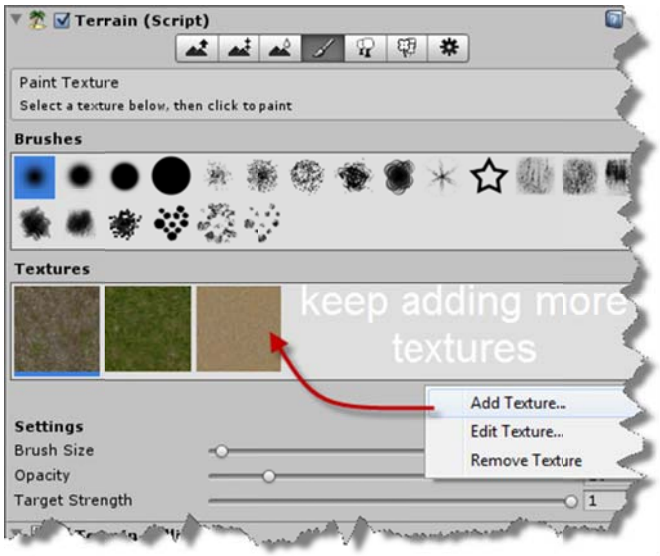


Add Terrain Texture

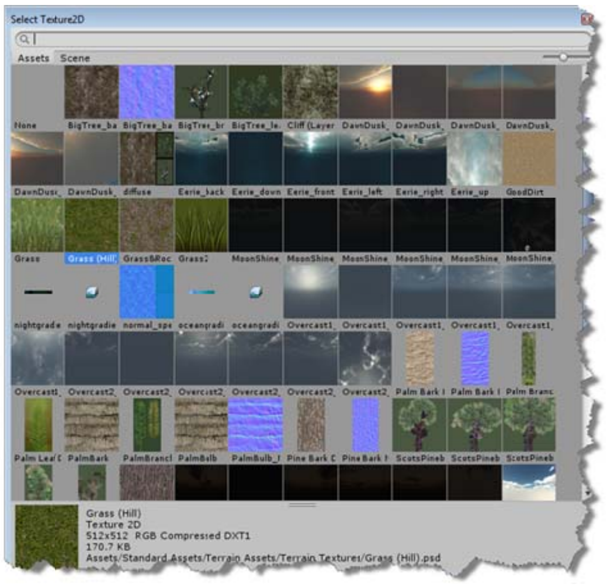
Splat	None (Texture 2D)
Tile Size X	15
Tile Size Y	15
Tile Offset X	0
Tile Offset Y	0

Please assign a tiling splat texture

9. Add more textures and make sure you activate the texture prior to painting.

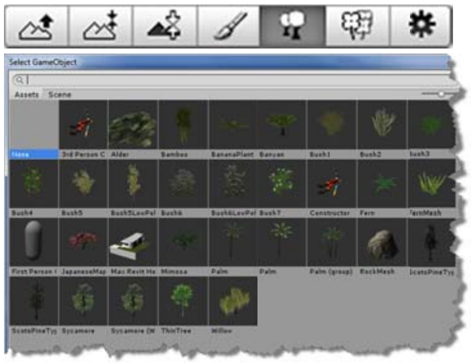


10. Add more textures from the libraries

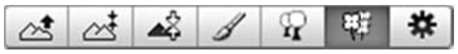

Select Texture2D
Assets
Scene
Grass (Hill)
Texture 2D
512x512 RGB Compressed DXT1
170.7 KB
Assets/Standard Assets/Terrain Assets/Terrain Textures/Grass (Hill).psd

Incorporate your terrain with your Model.

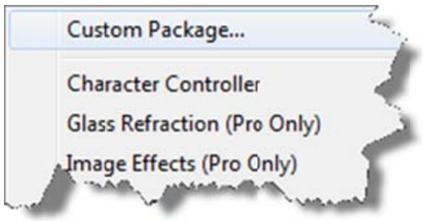
11. Add Trees and paint as with the terrain

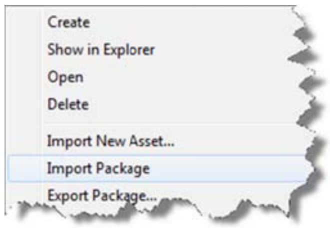


12. Add Grass and paint as with the terrain



To have a bigger library of plants, trees and textures download and load the Terrain Assets. You can download such here: http://unity3d.com/support/resources/assets/terrain-assets



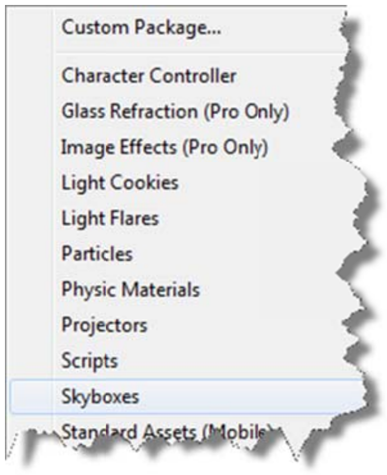


You should be able to populate your scene fast and easily with the above tools to achieve.

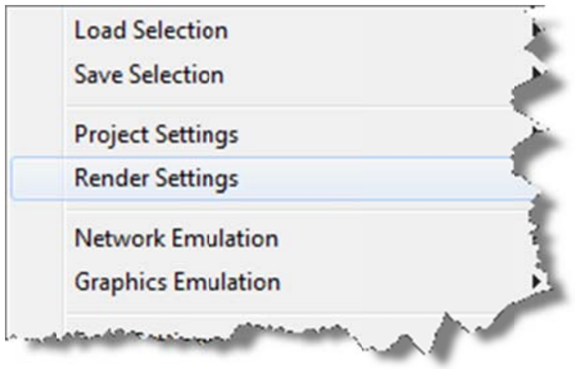
Adding a Sky

To add a 360 sky environment you need a skybox. Skyboxes are a wrapper around your entire scene that display the vast beyond of your world.
You can create your own Skyboxes or you can use a Unity skybox packages. To do that please do the following:

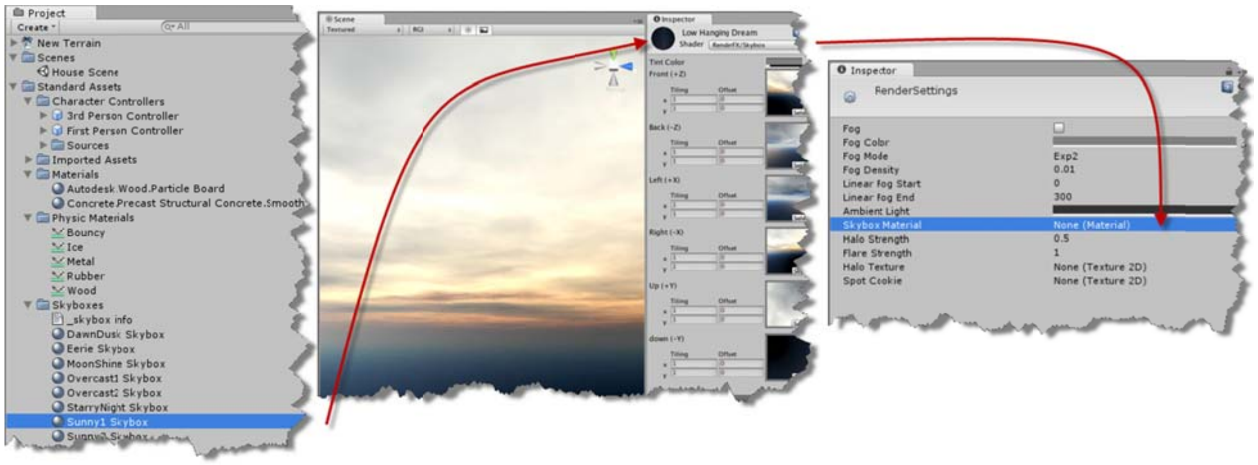
1. In the Assets menu import the Skybox package



2. In the Edit menu click on Render settings



You should see the Skybox Assets in the Project View and should be able to preview it in the Inspector View.



Inspector

RenderSettings

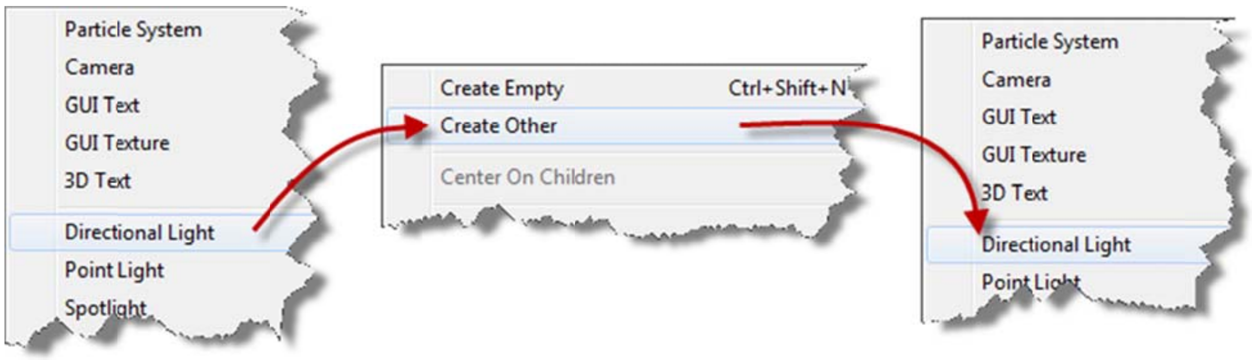
Fog	☐
Fog Color	
Fog Mode	Exp2
Fog Density	0.01
Linear fog Start	0
Linear fog End	300
Ambient Light	
Skybox Material	None (Material)
Halo Strength	0.5
Flare Strength	1
Halo Texture	None (Texture 2D)
Spot Cookie	None (Texture 2D)

You should see your sky and should be able to orbit and see it in a 360 environment

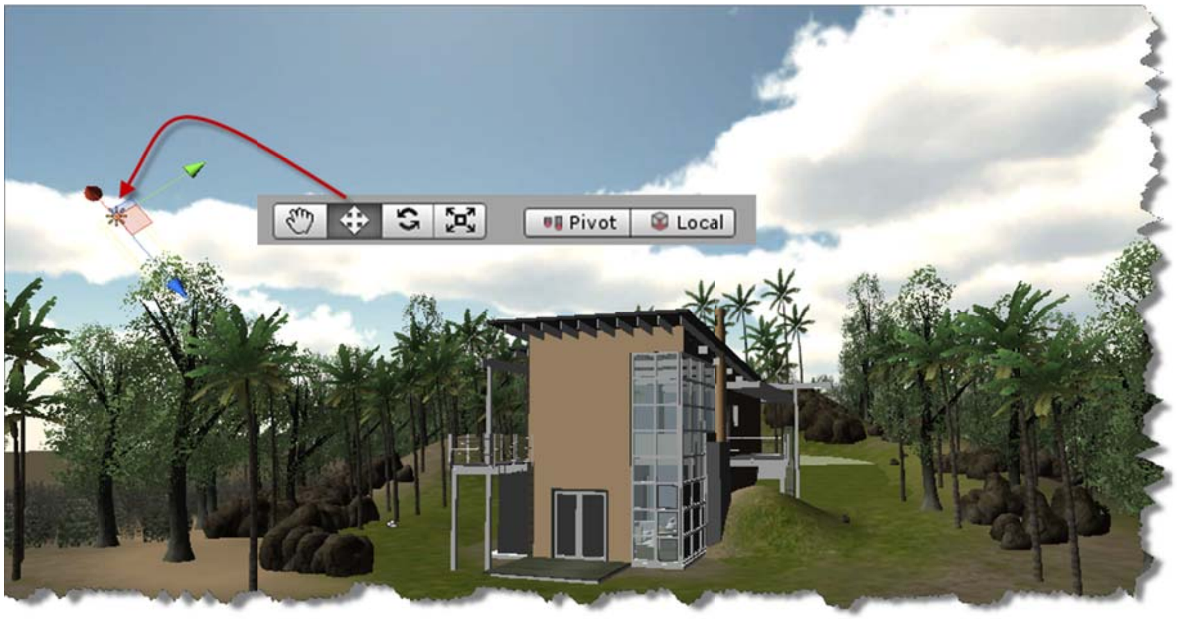
Adding Sun light and Shadows

Directional lights are used mainly in outdoor scenes for sun & moonlight. The light affect all surfaces of objects in your scene. They are the least expensive on the graphics processor.

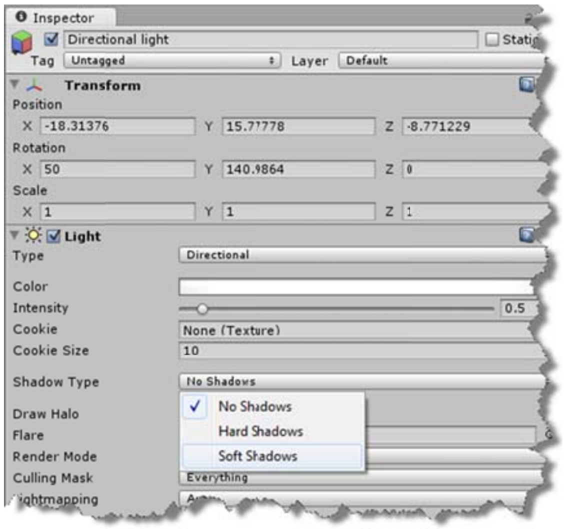
In the Game Object menu select Directional Light and there click on Create Other – Directional Light



You can adjust the light to be raised and rotated.

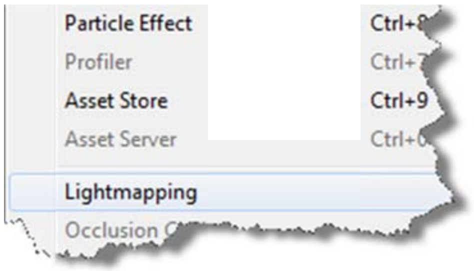


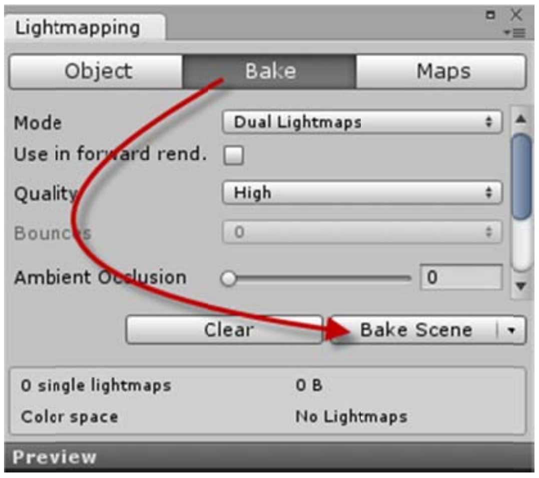
1. To get the shadows to show select the light and in the Shadow Type option chose soft shadows.



Inspector
Directional light
Static
Tag Untagged Layer Default
Transform
Position X -18.31376 Y 15.77778 Z -8.771229
Rotation X 50 Y 140.9864 Z 0
Scale X 1 Y 1 Z 1
Light
Type Directional
Color
Intensity 0.5
Cookie None (Texture)
Cookie Size 10
Shadow Type No Shadows
Draw Halo
Flare
Render Mode
Culling Mask Everything
Lightmapping

2. In the window menu select Lightmapping and bake the scene





Lightmapping
Object | Bake | Maps
Mode Dual Lightmaps
Use in forward rend.
Quality High
Bounces 0
Ambient Occlusion 0
Clear | Bake Scene
0 single lightmaps 0 B
Color space No Lightmaps
Preview

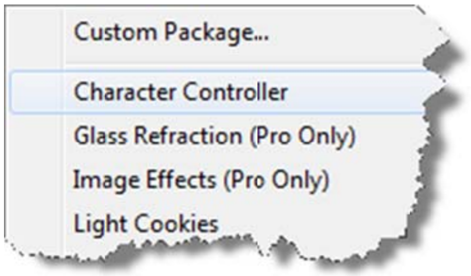
The results should now be visible. You should see shadows.
We encourage you to check out more light properties from the inspector like the light cookies for effects like cloud shadows projecting on the ground.

Setting up your Character Controller to Navigate your scene

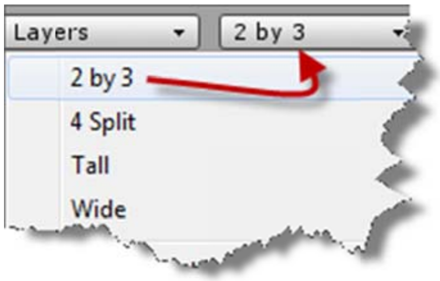
The Character Controller is mainly used for third-person or first-person player control.
A CharacterController is not affected by forces and will only move when you call the Move funtion. It will then carry out the movement but be constrained by collisions.

To setup a First person controller please do the following:

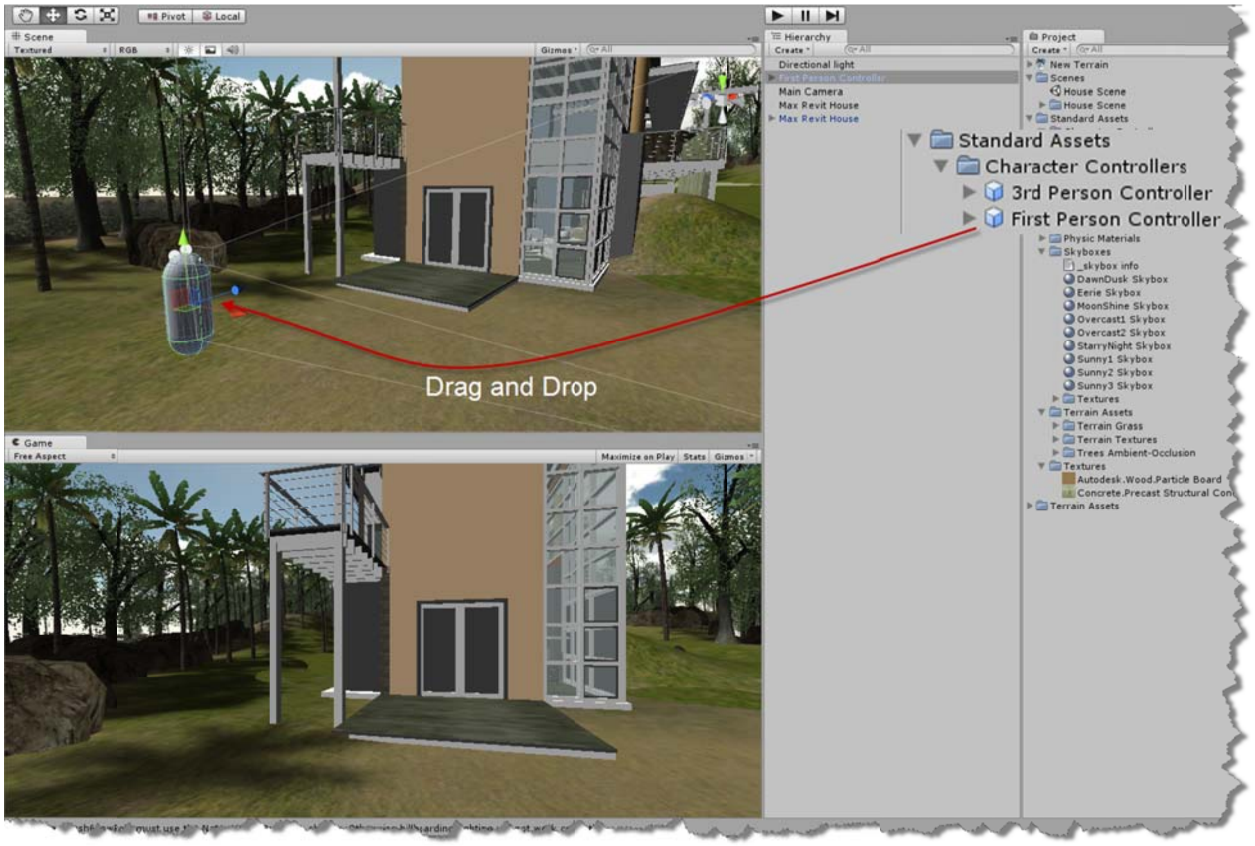
1. In the Assets Menu, chose Import Package and select Character Controller



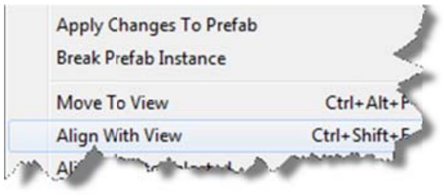
2. For easier viewing switch the View layout from Wide to 2 by 3



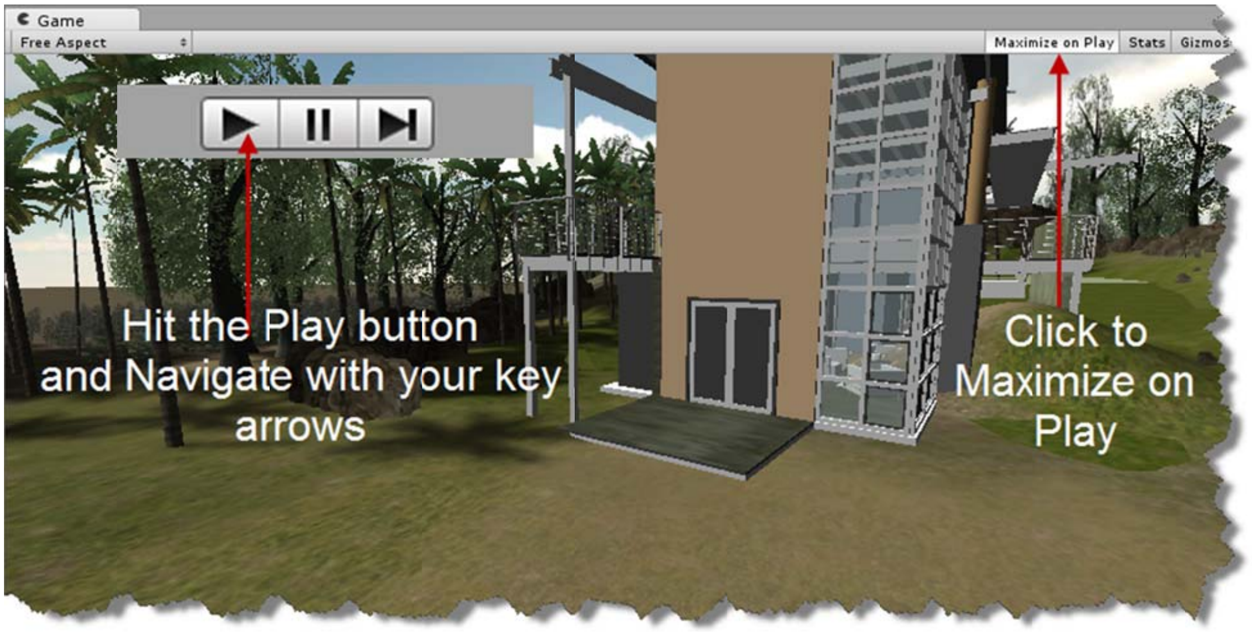
Then From the Standard Assets in the Project view select First Person Controller and Drag and Drop it into your scene.



Then in the Game View click on Game Object and Chose Align With View.

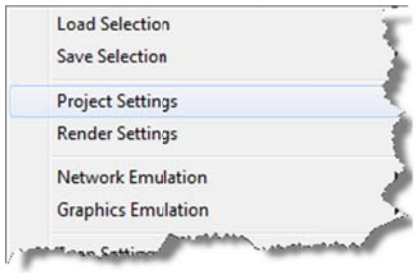


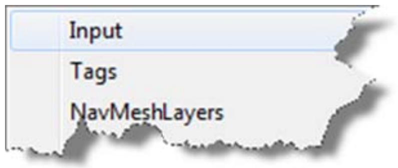
To Start Navigating click on the play button and use your arrow keyboard to navigate.


Game
Free Aspect
Maximize on Play
Stats
Gizmos
Hit the Play button and Navigate with your key arrows
Click to Maximize on Play

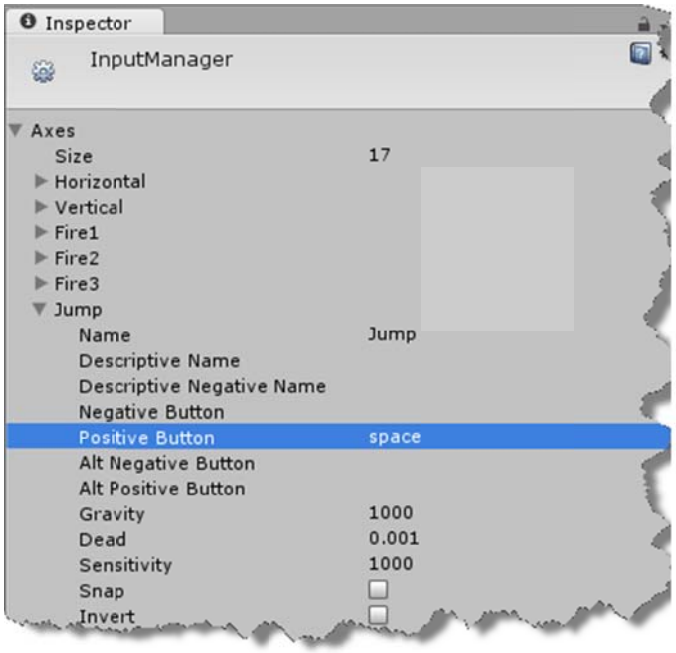
To control the way you navigate your game you can:

1. In the Edit Menu chose Project Settings / Input





2. In the Inspector click on the action to assign a key



```
Inspector
InputManager
▼ Axes
    Size                          17
  ▶ Horizontal
  ▶ Vertical
  ▶ Fire1
  ▶ Fire2
  ▶ Fire3
  ▼ Jump
      Name                        Jump
      Descriptive Name
      Descriptive Negative Name
      Negative Button
      Positive Button             space
      Alt Negative Button
      Alt Positive Button
      Gravity                     1000
      Dead                        0.001
      Sensitivity                 1000
      Snap                        ☐
      Invert                      ☐
```

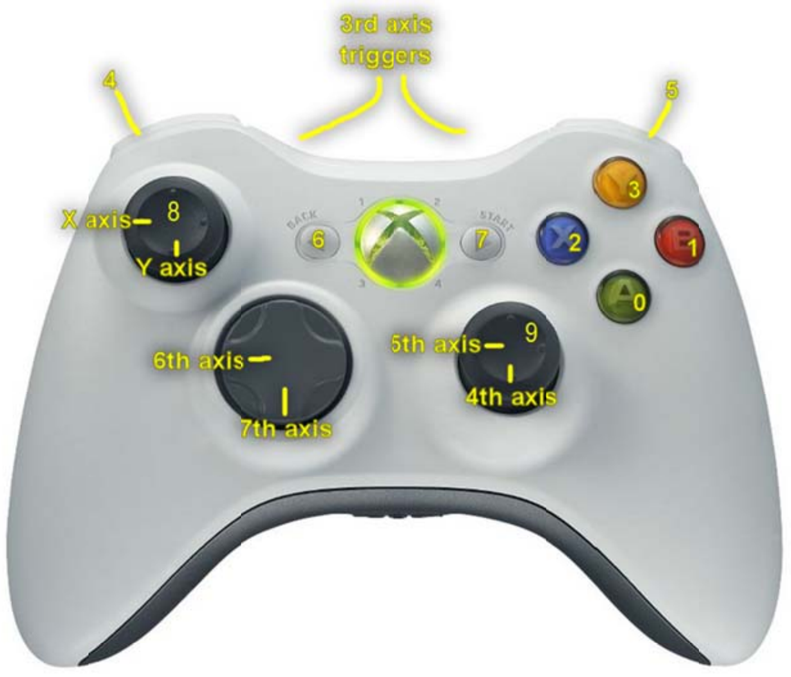
The number of input devices is 17. The maximum number of inputs Unity will allow is 80 inputs. (4 joysticks, each with 20 buttons)

The Microsoft XBOX 360 Controller supports mapping.

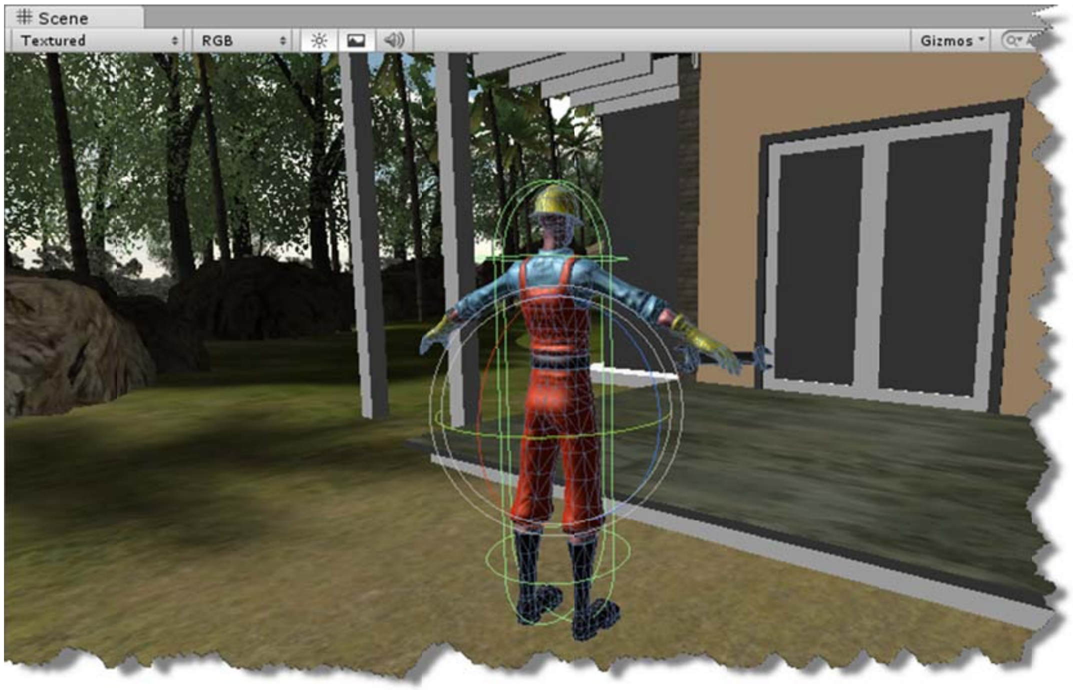
Buttons (Key or Mouse Button)

joystick button 0 = A
joystick button 1 = B
joystick button 2 = X
joystick button 3 = Y
joystick button 4 = L
joystick button 5 = R
joystick button 6 = Back
joystick button 7 = Home
joystick button 8 = Left analog press
joystick button 9 = Right analog press

X axis = Left analog X
Y axis = Left analog Y
3rd axis = LT/RT
4th axis = Right analog X
5th axis = Right analog Y
6th axis = Dpad X
7th axis = Dpad Y



Feel free to explore the Third person character controller.
To insert it into your scene please repeat the same steps taken for the 1st person controller.
Unity has a pre rigged third person controller called the Construction Worker

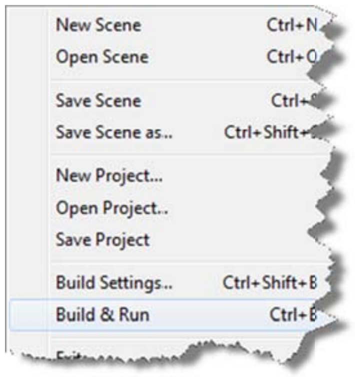


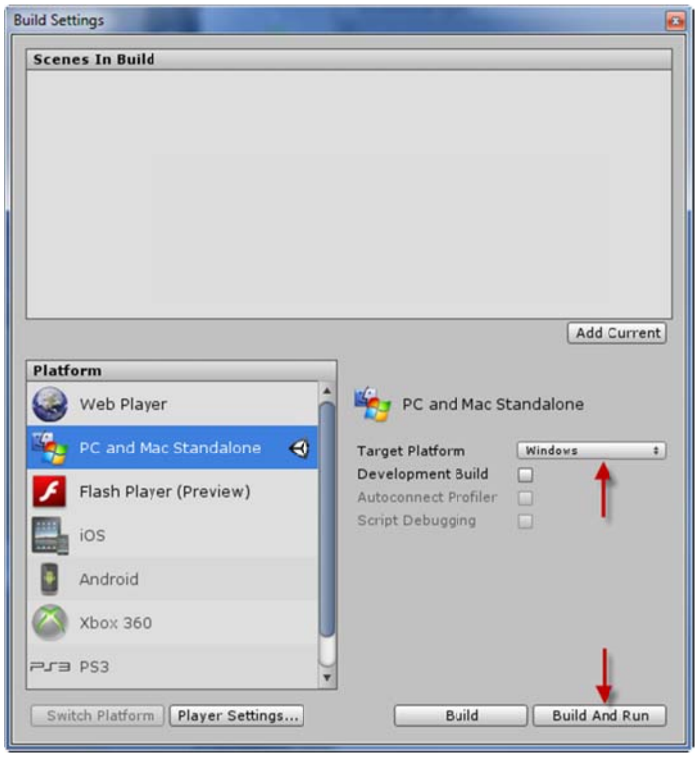
Publishing the Game

At any time while you are creating your game, you might want to see how it looks when you build and run it outside of the editor as a standalone or web player

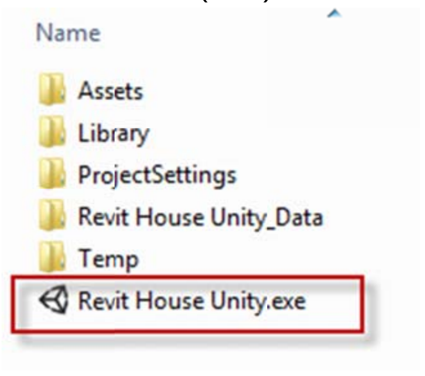
To create the game do the following:

1. In the File menu select Build & Run
2. In the Build Settings window chose PC and Mac Standalone to make an executable portable file.

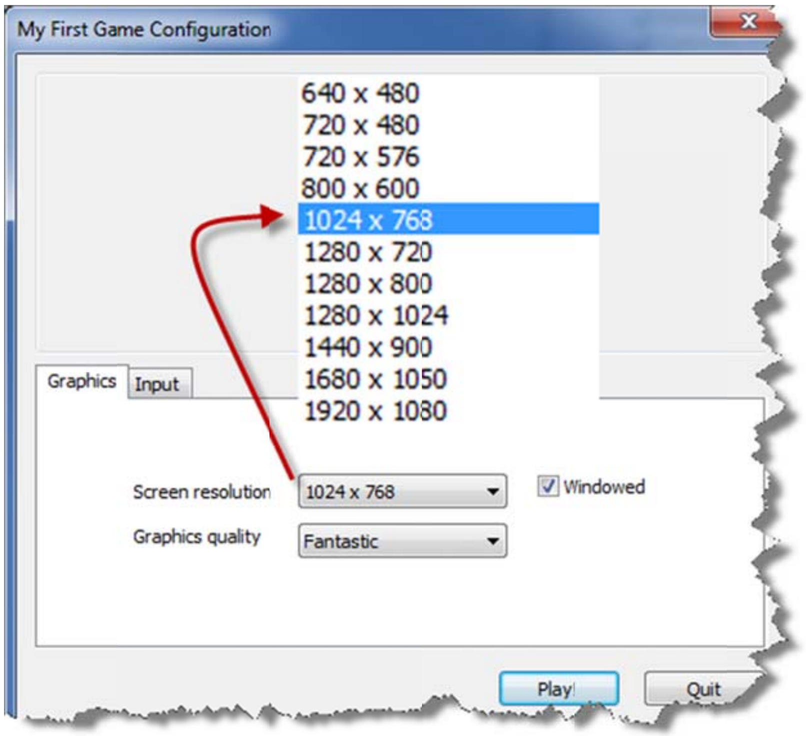




When the person who wants to launch the game they just simply have to double click on the executable file (.exe)

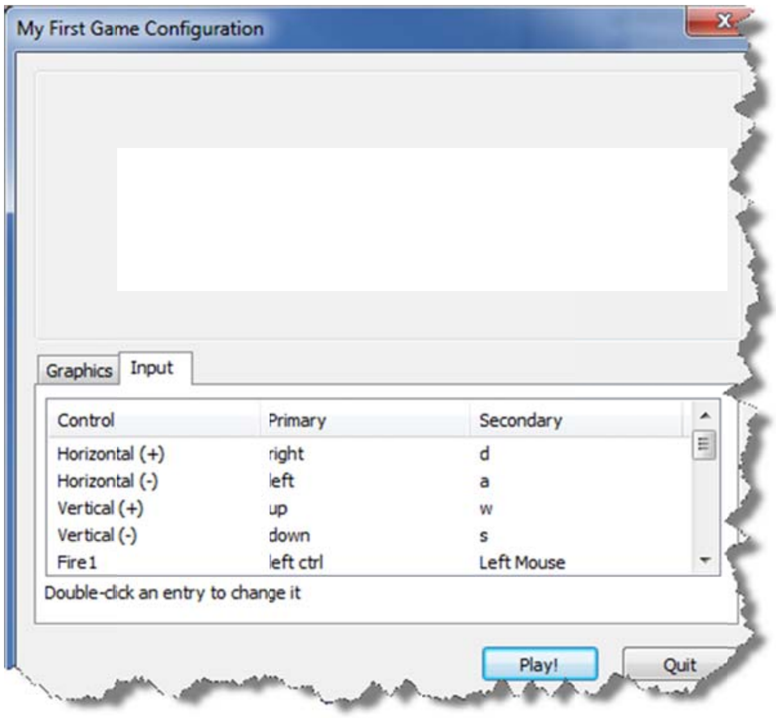


The user then has the ability to change the Resolution and Graphic quality.



Screen resolution: 1024 x 768
Graphics quality: Fantastic
[✓] Windowed

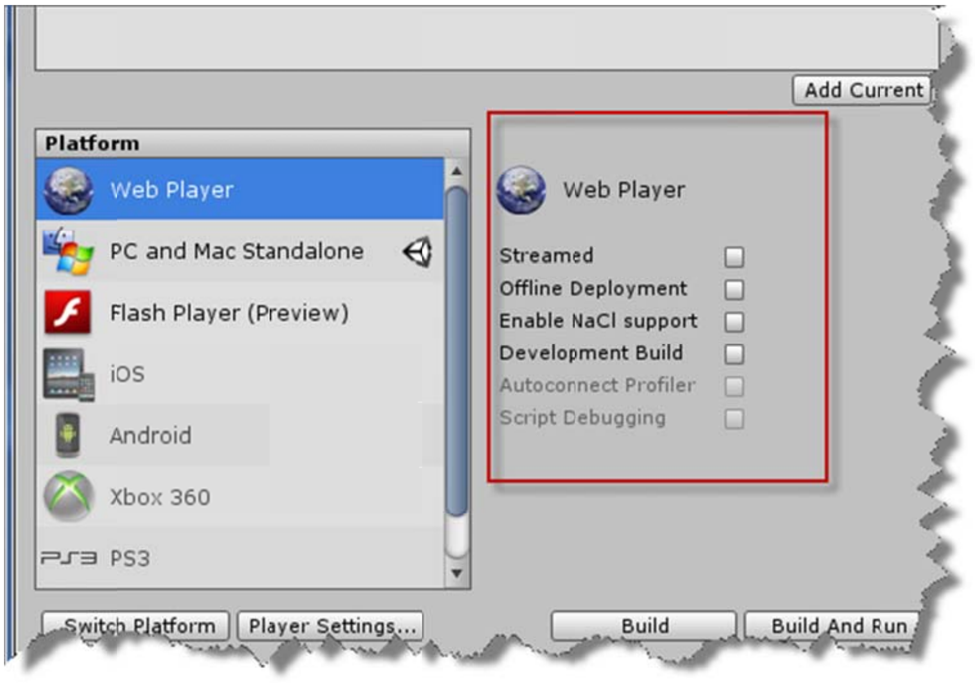
They can also check the assigned keys and controllers for Navigation.



Control	Primary	Secondary
Horizontal (+)	right	d
Horizontal (-)	left	a
Vertical (+)	up	w
Vertical (-)	down	s
Fire1	left ctrl	Left Mouse

Double-click an entry to change it

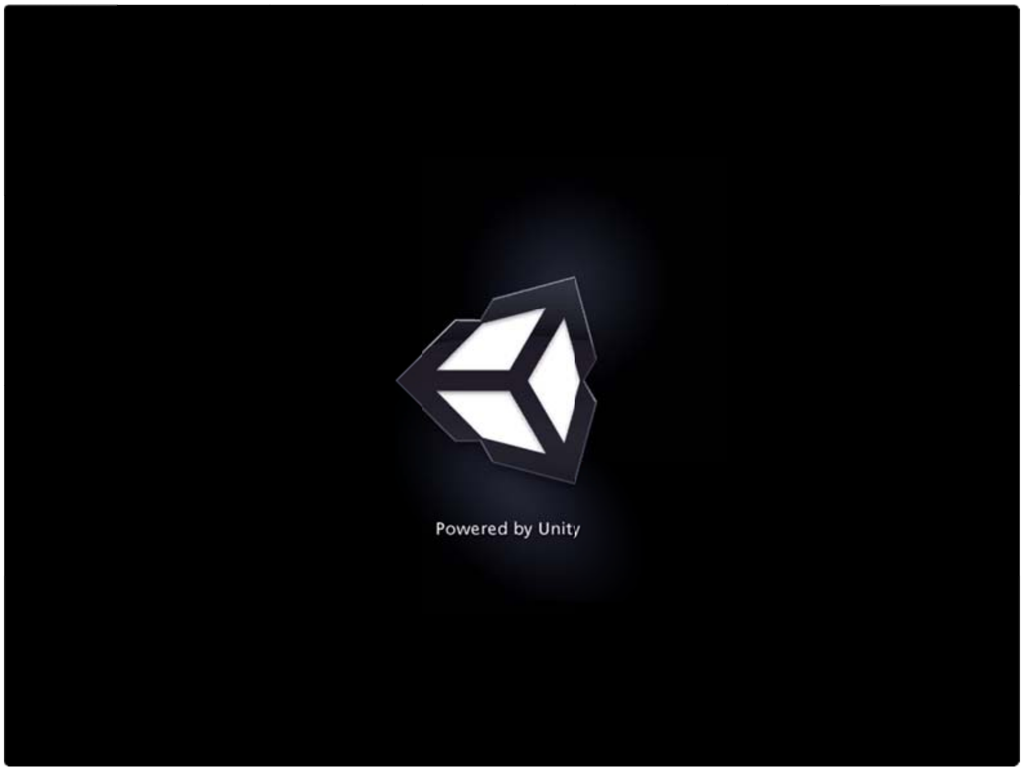
You can also Publish to web to view online and Offline.



Web Player

Streamed ☐
Offline Deployment ☐
Enable NaCl support ☐
Development Build ☐
Autoconnect Profiler ☐
Script Debugging ☐

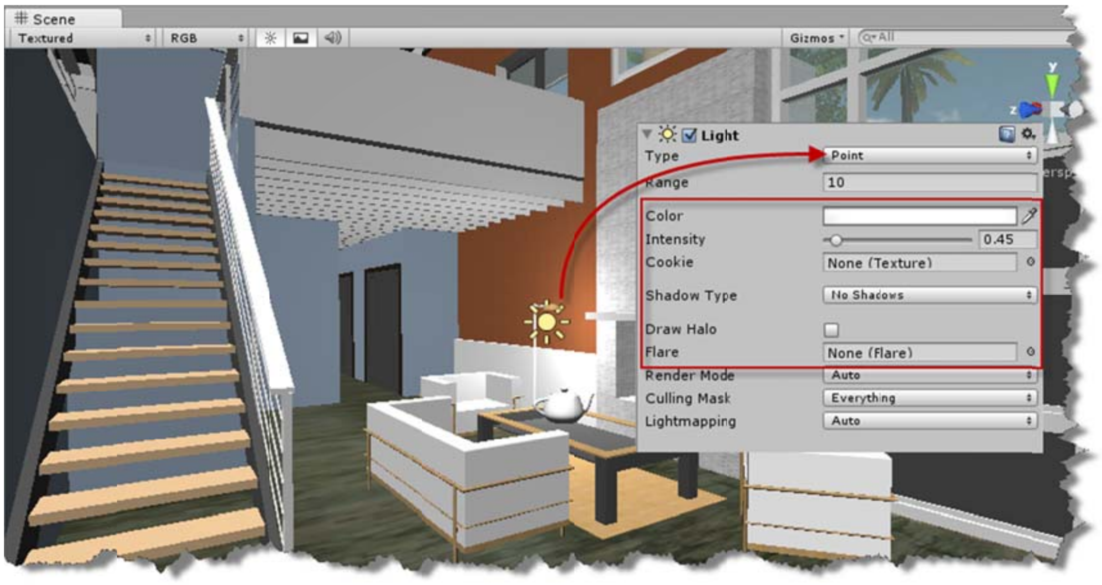
Launch and Play!



Working with Lighting

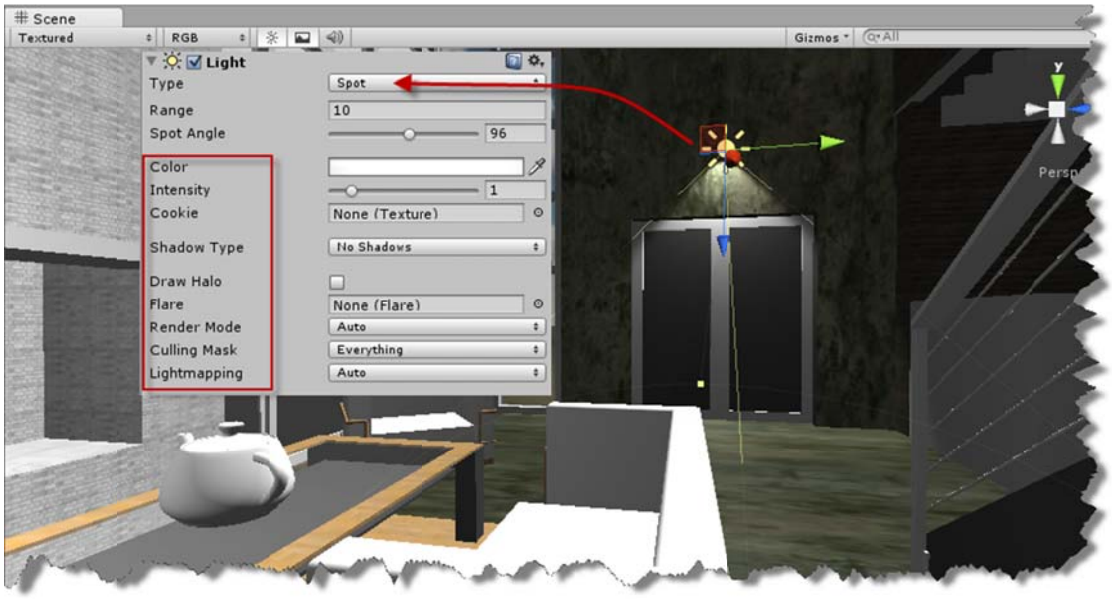
Lights will bring personality and flavor to your project. You use lights to illuminate the scenes and objects to create the perfect visual mood. Lights can be used to simulate the sun, burning match light, flashlights, gun-fire, or explosions, just to name a few

There are four types of lights in Unity:
Point lights shine from a location equally in all directions, like a light bulb.



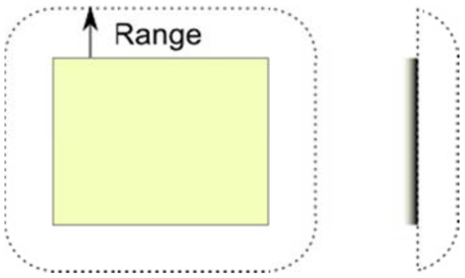
Type	Point
Range	10
Color	
Intensity	0.45
Cookie	None (Texture)
Shadow Type	No Shadows
Draw Halo	☐
Flare	None (Flare)
Render Mode	Auto
Culling Mask	Everything
Lightmapping	Auto

Spot lights shine from a point in a direction and only illuminate objects within a cone - like the headlights of a car.



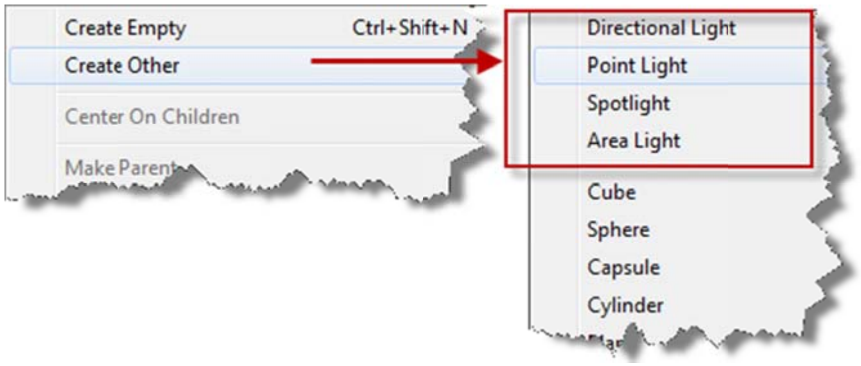
Type	Spot
Range	10
Spot Angle	96
Color	
Intensity	1
Cookie	None (Texture)
Shadow Type	No Shadows
Draw Halo	☐
Flare	None (Flare)
Render Mode	Auto
Culling Mask	Everything
Lightmapping	Auto

Area lights (only available for lightmap baking) shine in all directions to one side of a rectangular section of a plane.
Area lights cast light from one side of a rectangular area of a plane.



Directional lights are placed infinitely far away and affect everything in the scene, like the sun. (See previous Section on Adding Sun light to your scene)

To place any light in Unity just go to the Game Object menu and chose Create Other

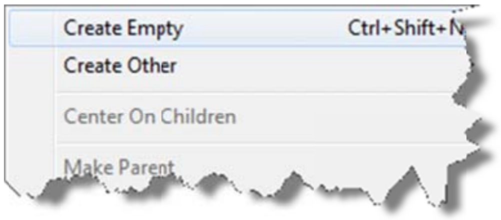


Tip
Spot lights with cookies can be extremely effective for making light coming in from windows.
Low-intensity point lights are good for providing depth to a scene.

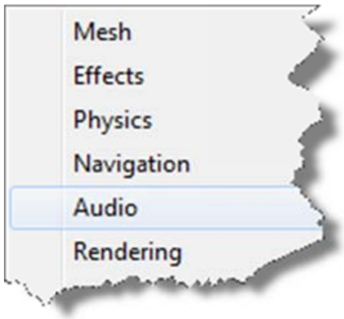
Working with Audio

The Audio Source plays back an Audio Clip in the scene. If the Audio Clip is a 3D clip, the source is played back at a given position and will attenuate over distance

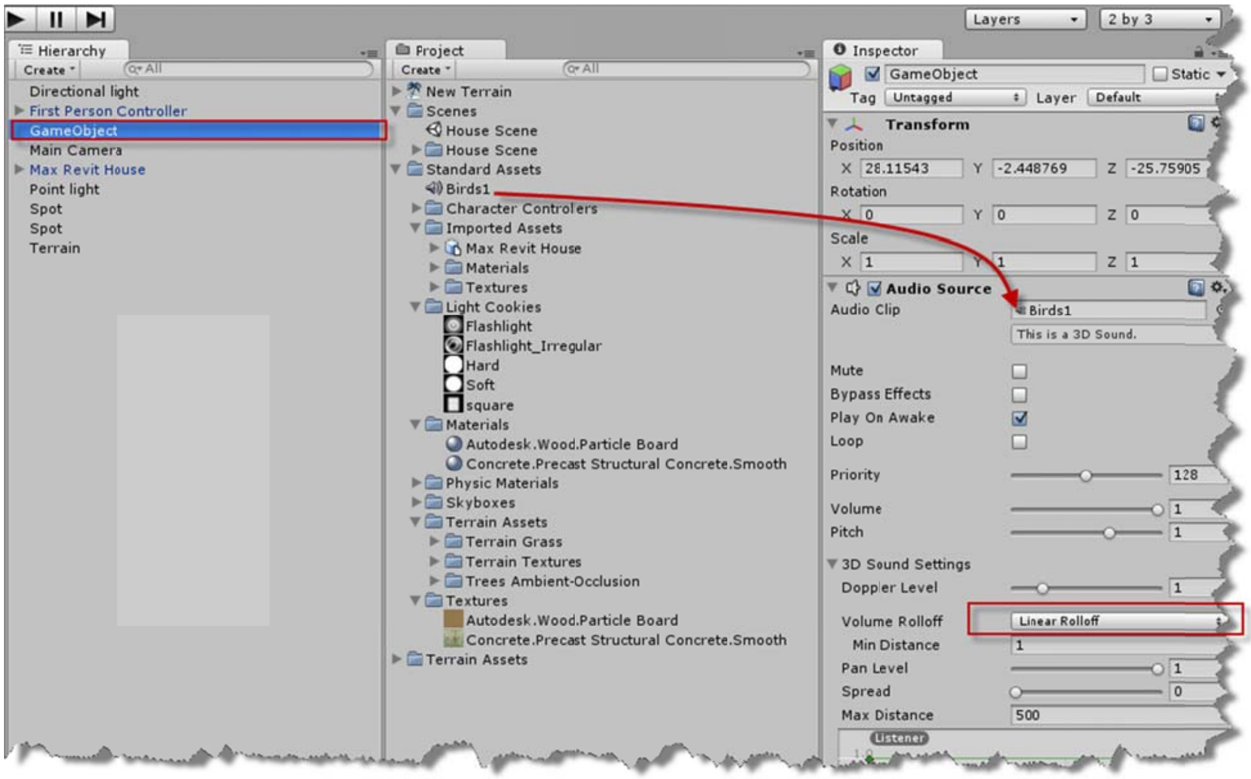
1. Import your Audio Clips by dragging and dropping from your windows explorer or from the Import asset option
2. From the game object menu chose create Empt



3. Select the Game Object from the Project view and From the Component menu Chose Audio



Drag and drop your audio clip from the Project View onto the Audio source property for your Dummy game object.



GameObject Static
Tag Untagged Layer Default

Transform
Position X 28.11543 Y -2.448769 Z -25.75905
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Audio Source
Audio Clip Birds1
This is a 3D Sound.
Mute ☐
Bypass Effects ☐
Play On Awake ☑
Loop ☐
Priority 128
Volume 1
Pitch 1
3D Sound Settings
Doppler Level 1
Volume Rolloff Linear Rolloff
Min Distance 1
Pan Level 1
Spread 0
Max Distance 500

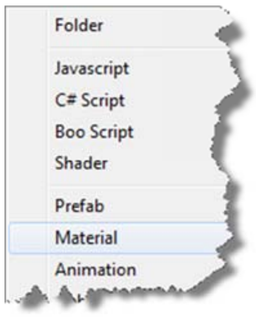
Note: If you want to create an Audio Source just for one Audio Clip that you have in the Assets folder then you can just drag that clip to the scene view - a GameObject with an Audio Source component will be created automatically for it. Dragging a clip onto on existing GameObject will attach the clip along with a new Audio Source if there isn't one already there. If the object does already have an Audio Source then the newly dragged clip will replace the one that the source currently uses.

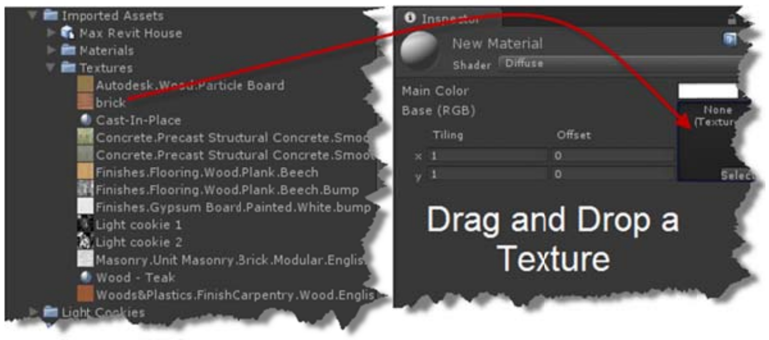
Working with Materials and Shaders

There is a close relationship between Materials and Shaders in Unity. Shaders contain code that defines what kind of properties and assets to use. Materials allow you to adjust properties and assign assets.

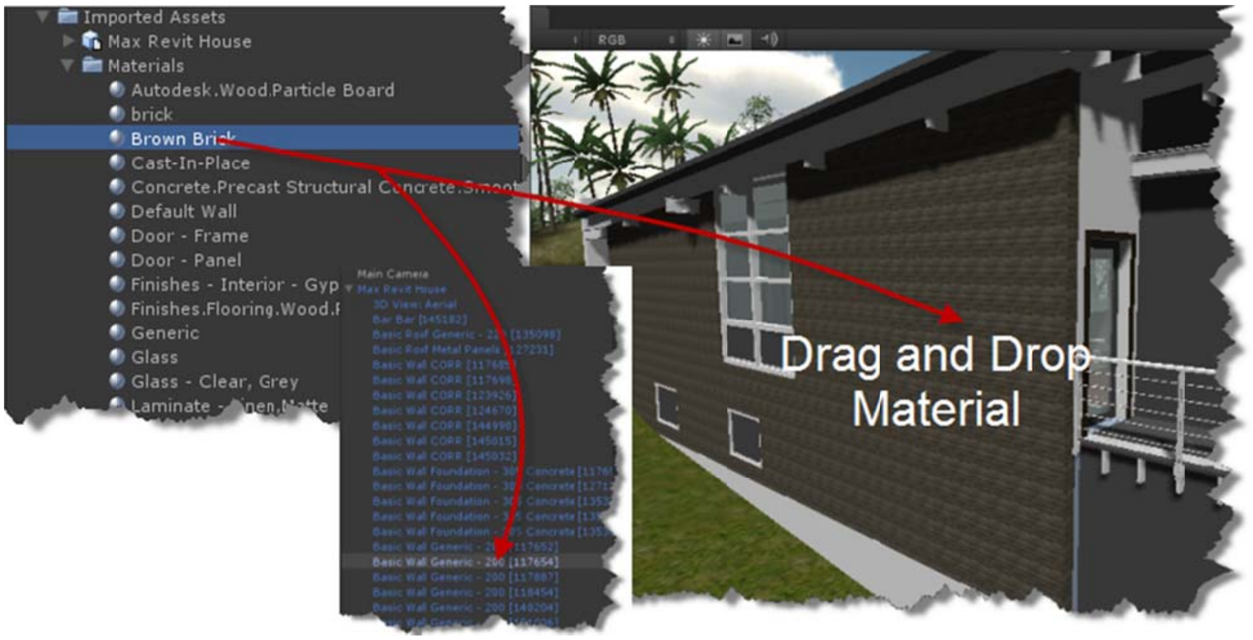
To create a new material do the following:

1. From the Asset menu chose Create and click on Material. The material will now show in the Project View as New Material.
2. We would recommend importing your library of textures upfront that way any custom textures can be applied to the material shader by dragging and dropping from the textures to the material shader texture box.

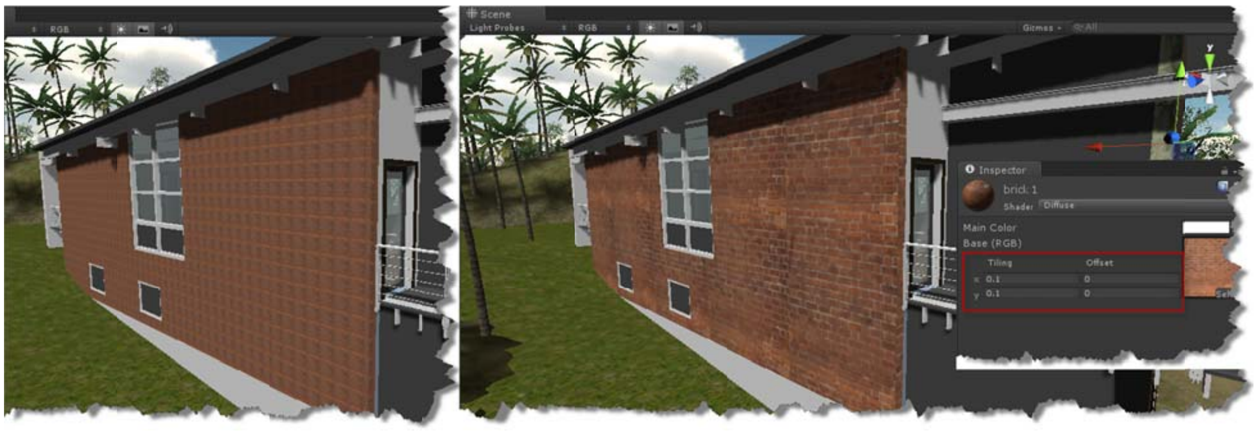



Imported Assets
Max Revit House
Materials
Textures
Autodesk.Wood.Particle Board
brick
Cast-In-Place
Concrete.Precast Structural Concrete.Smoo
Concrete.Precast Structural Concrete.Smoo
Finishes.Flooring.Wood.Plank.Beech
Finishes.Flooring.Wood.Plank.Beech.Bump
Finishes.Gypsum Board.Painted.White.bump
Light cookie 1
Light cookie 2
Masonry.Unit Masonry.Brick.Modular.Englis
Wood - Teak
Woods&Plastics.FinishCarpentry.Wood.Englis
Light Cookies
Inspector
New Material
Shader Diffuse
Main Color
Base (RGB)
Tiling
Offset
x 1 0
y 1 0
None (Textur
Select
Drag and Drop a Texture

Once the Material has been created, you can apply it to an object and tweak all of its properties in the Inspector. To apply it to an object, just drag it from the Project View to any object in the Scene or Hierarchy.



You can change the material scaling and offset from the inspector.



Built-in Shaders
There is a library of built-in Shaders that come standard with every installation of Unity. There are over 30 of these built-in Shaders, and six basic families.

Normal: For opaque textured objects.
Transparent: For partly transparent objects. The texture's alpha channel defines the level of transparency.
TransparentCutOut: For objects that have only fully opaque and fully transparent areas, like fences.
Self-Illuminated: For objects that have light emitting parts.
Reflective: For opaque textured objects that reflect an environment Cubemap.

Adding Interactivity

You can have a game object be controlled by custom properties through scripting
Scripting inside Unity consists of attaching custom script objects called behaviors to game objects. Different functions inside the script objects are called on certain events.
Now do not get intimidated by this and please read on.

Interactivity
There are 2 types of Scripting Interactivity:

Physical Interactivity: Involving objects moving or colliding
Logical Interactivity: Author controls the consequences

Unity Supports various Scripting Languages such as:

- Python
- Boo
- C#
- Java script

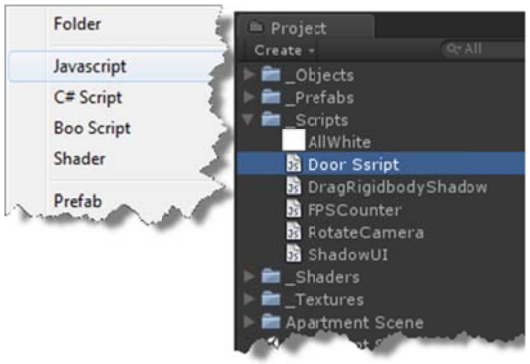
The Most commonly used can be found in unity scripting reference document.
http://docs.unity3d.com/Documentation/ScriptReference/index.html

Note:
Just like filming a movie with a camera you don't need to know what goes on inside the camera to make good use of it.
There is an extensive library of free scripts which can be found online. Below is a good resource:
http://wiki.unity3d.com/index.php?title=Scripts

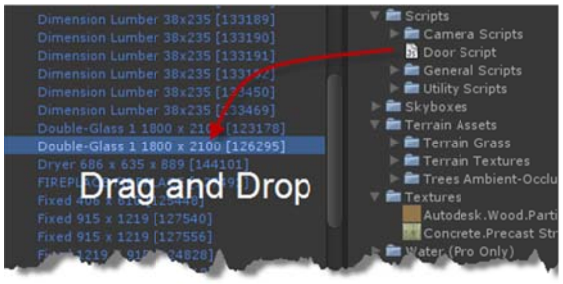
Adding Interactivity to a Door
To have a better idea of what a script can do. Let's say you have a door in your scene and you want the door to open when you trigger it by a mouse click, to do this you would have to create a script which will be sort of like adding a custom set of parameters or controllers to your door.

To create a script that will allow us to control the door do the following:

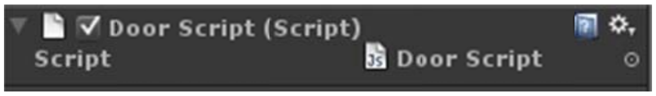
1. From the Asset menu chose Create and click on Javascript (this is a simpler language)



2. Now drag the script from the Project View to the Cube (in the Scene or Hierarchy View, it doesn't matter).

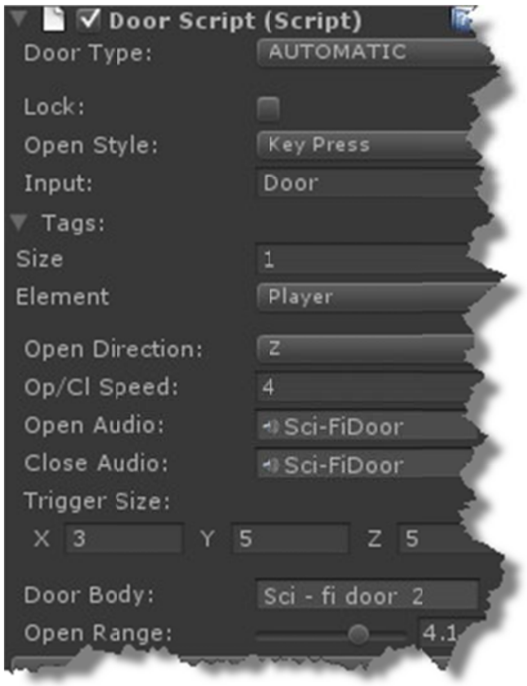

Dimension Lumber 38x235 [133189]
Dimension Lumber 38x235 [133190]
Dimension Lumber 38x235 [133191]
Double-Glass 1 1800 x 2100 [126295]
Dryer 686 x 635 x 889 [144101]
Drag and Drop
Fixed 915 x 1219 [127540]
Fixed 915 x 1219 [127556]
Scripts
Camera Scripts
Door Script
General Scripts
Utility Scripts
Skyboxes
Terrain Assets
Terrain Grass
Terrain Textures
Trees Ambient-Occlu
Textures
Autodesk.Wood.Parti
Concrete.Precast Str
Water (Pro Only)

Your door should have a new property you can see in the inspection view



Door Script (Script)
Script: Door Script

Below is a fully finished script that can be accessed from the Asset Store. (see next section)



Door Script (Script)

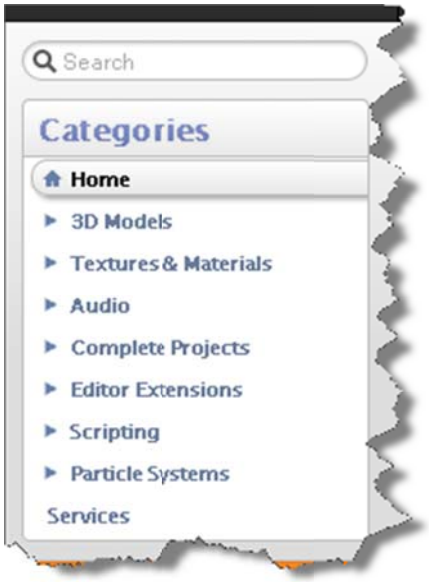
Door Type:	AUTOMATIC
Lock:	☐
Open Style:	Key Press
Input:	Door
Tags:	
Size	1
Element	Player
Open Direction:	Z
Op/Cl Speed:	4
Open Audio:	Sci-FiDoor
Close Audio:	Sci-FiDoor
Trigger Size:	X 3 Y 5 Z 5
Door Body:	Sci - fi door 2
Open Range:	4.1

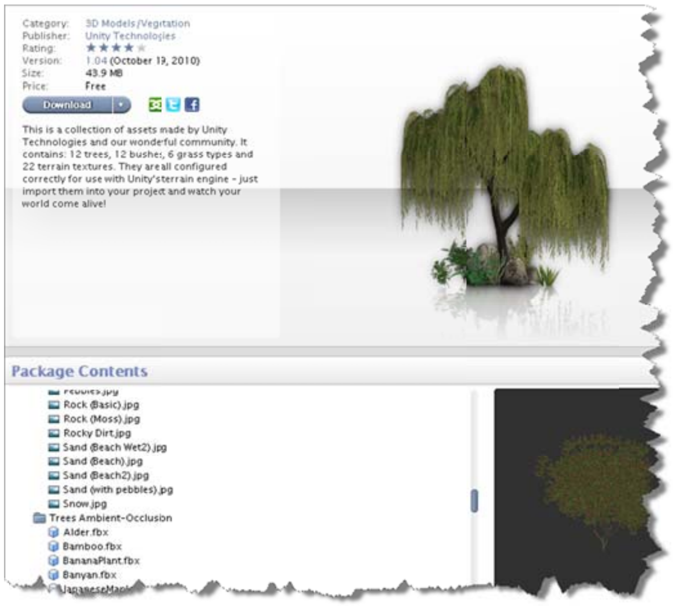
The Asset Store

The Asset Store is a time and effort-saving resource (Portal) when creating your project. Character models, props, materials and textures, landscape painting tools, game creating tools, audio effects and music, visual programming solutions, scripts and editor extensions are all available.

You can search for scripts, audio clips, additional particles, models, etc

You access the Asset Store directly from Unity. Download Unity. Open the Unity Editor, go to Windows and click on Asset Store from the drop down menu. The Asset Store catalogue is constantly growing, and currently comprises over 3000 ready-to-use free and for purchase assets.

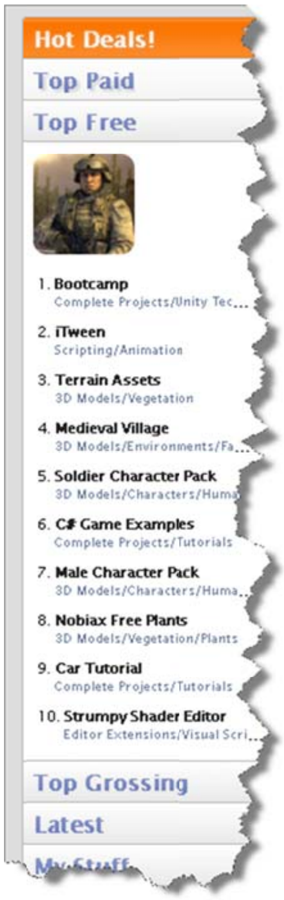



Category: 3D Models/Vegitation
Publisher: Unity Technologies
Rating: ★★★★☆
Version: 1.04 (October 19, 2010)
Size: 43.9 MB
Price: Free

This is a collection of assets made by Unity Technologies and our wonderful community. It contains: 12 trees, 12 bushes, 6 grass types and 22 terrain textures. They are all configured correctly for use with Unity'sterrain engine - just import them into your project and watch your world come alive!

Package Contents

Rock (Basic).jpg
Rock (Moss).jpg
Rocky Dirt.jpg
Sand (Beach Wet2).jpg
Sand (Beach).jpg
Sand (Beach2).jpg
Sand (with pebbles).jpg
Snow.jpg
Trees Ambient-Occlusion
Alder.fbx
Bamboo.fbx
BananaPlant.fbx
Banyan.fbx

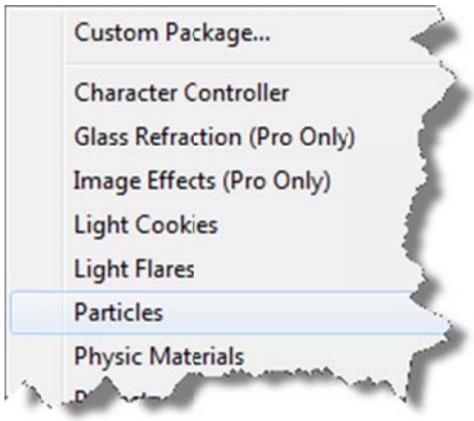


1. Bootcamp
Complete Projects/Unity Tec...
2. iTween
Scripting/Animation
3. Terrain Assets
3D Models/Vegetation
4. Medieval Village
3D Models/Environments/Fa...
5. Soldier Character Pack
3D Models/Characters/Huma
6. C# Game Examples
Complete Projects/Tutorials
7. Male Character Pack
3D Models/Characters/Huma...
8. Nobiax Free Plants
3D Models/Vegetation/Plants
9. Car Tutorial
Complete Projects/Tutorials
10. Strumpy Shader Editor
Editor Extensions/Visual Scri...

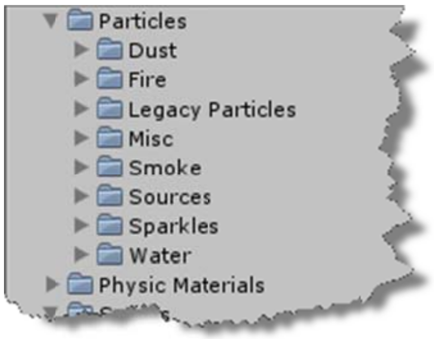
Particles

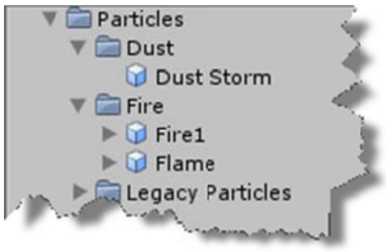
Particle Systems in Unity are used to make clouds of smoke, steam, fire and other atmospheric effects. They can complement your scene.

1. Import the Particles package from the Asset menu

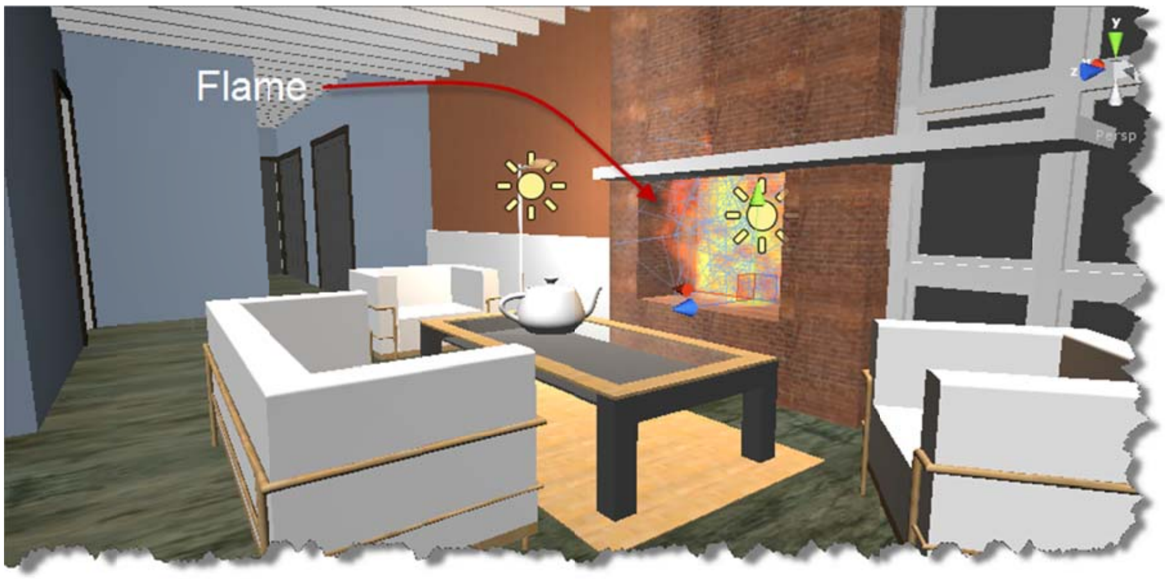


2. From the Project View





Simply Drag and Drop the Particle into your scene and adjust position. You should be able to preview it right on the screen. Feel free to experiment with the out of the box library of particles.

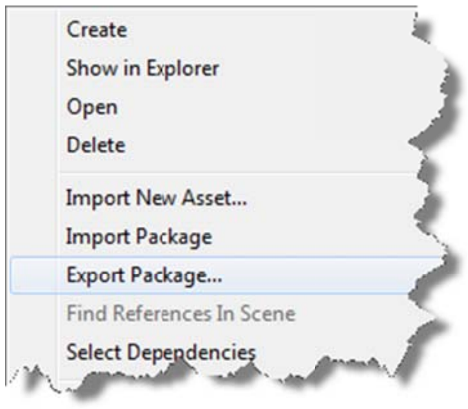


Smoke
Particle

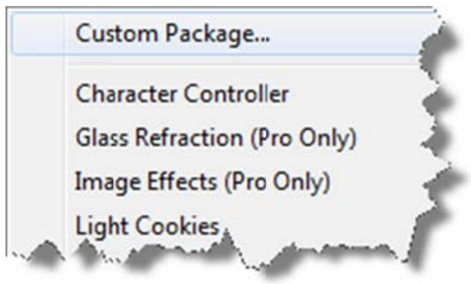
Water Surface
Splash

How do I reuse assets between projects?
As you build your Project, Unity stores a lot of metadata about your assets (import settings, links to other assets, etc.). If you want to take your assets into a different project, there is a specific way to do that. Here's how to easily move assets between projects and still preserve all this info.

1. In the Project View, select all the asset files you want to export.
   Choose Assets and select Export Package from the menu bar.
   Name and save the package anywhere you like.



2. Open the project you want to bring the assets into.
   Choose Assets and click on Import Package / Custom Package from the otpions
   Select your package file saved in the previous step



Setting up the Controllers

Now that we have our models moved into Unity and our environments and interactions setup we will look at how we want to interact with our work. In truth this is something that you will likely want to think about earlier as the method you use to control may change or influence some decisions you make with regards to the camera or interactivity. An example of this is making the model and mapping al the commands you want to perform to a controller you have and are comfortable with, but you send the completed presentation to the client to interact with and they either do not have the comfort level with the controller you have setup, or do not have that kind of controller at all!

The controllers we will be looking at for the purpose of this class will cover several different control styles, to better examine how these models can be interacted with, but are in no means the only ways possible. The options we have chosen to focus on are the traditional mouse and/ or keyboard, a 3d mouse, a game controller, a Kinect peripheral, and a mobile device. Each of these methods of interaction has their own benefits and possible drawbacks to your presentation.

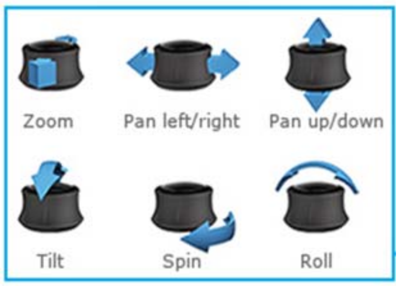
Mouse and Keyboard

This is probably the most “old school” in many ways for controlling a game, and is essentially always available and requires no additional hardware as any computer will have these. In addition to that, most users that are perhaps not comfortable with a game controller in their hands will at least be familiar with the usage of them in their daily work so they will only need to understand how to use them in the context of what we have created.

3D Mouse

Those who work in 3D modeling or design will likely know about 3D Mice, while many others may not have heard of them. The primary makers of these are 3D Connexion and they offer several versions with differing levels of abilities. The benefit of this is that the person moving through the space can easily control their direction in all axis. If they would like to turn to the right they just twist the dial to the right, if they want to fly upward to look down on the model then they just lift up on the dial. When you map the buttons on this to perform actions you can really make a presentation jump.

The downside to these controllers is that they may take a little while to get used to and will require drivers to be installed. Not a biggie but if you are sending a model to a client to interact with, it is something to keep in mind.



Video Game Controller

While the image of a modern game controller will be familiar by all, its usage may not. However because they are ubiquitous those who already use them will be able to navigate a scene intuitively and in many cases, with more ease than a keyboard and mouse.

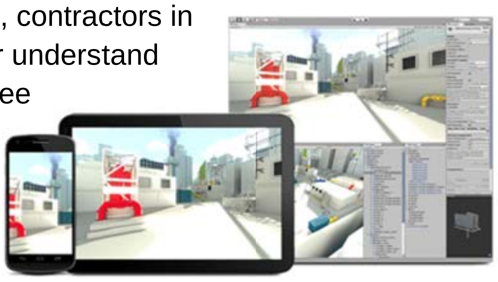
Depending on which controller you decide to use there may be more or less effort to get it connected. We will be focusing on the Microsoft Xbox 360 Wireless controller. Please note that this typically requires an adapter to connect to a pc but you can purchase a wired controller that will plug into an available usb port as well.

Microsoft Kinect

For those looking to not have a controller entirely or at least appear as to not have one and make the experience more interactive or transparent you can use a tool like the Microsoft Kinect. This is the most divergent from the options we will look at because there is no physical interaction with anything as the device responds to your speaking or your physical movements. In some ways it's the most "whizbang" but also would likely require some getting used to for those who have never done anything like this before. What we have seen though is that for those who try it and get past what seems like novelty, it engages them more fully than any of the other controllers. And for those situations where you feel you need to have more specific interaction you can combine the Kinect with other controllers, such as a mouse that is designed to be worn like a ring from Genius.

Mobile Device / Web

While there are still people who do not play games or may not feel comfortable using a controller, almost everyone has a mobile device. Be it a tablet or a phone, almost everyone has something that can run these models on them and be interacted with. This allows us to showcase our designs in a fashion that was unheard of before, With a simple tool we can enable perspective buyers to interact with real estate space, contractors in the field can fly around the structure of our building to better understand the connections, clients on the other side of the world can see and interact with our work and swap design options on the fly, first responders can review floor plans quickly en route to a building, the list goes on.

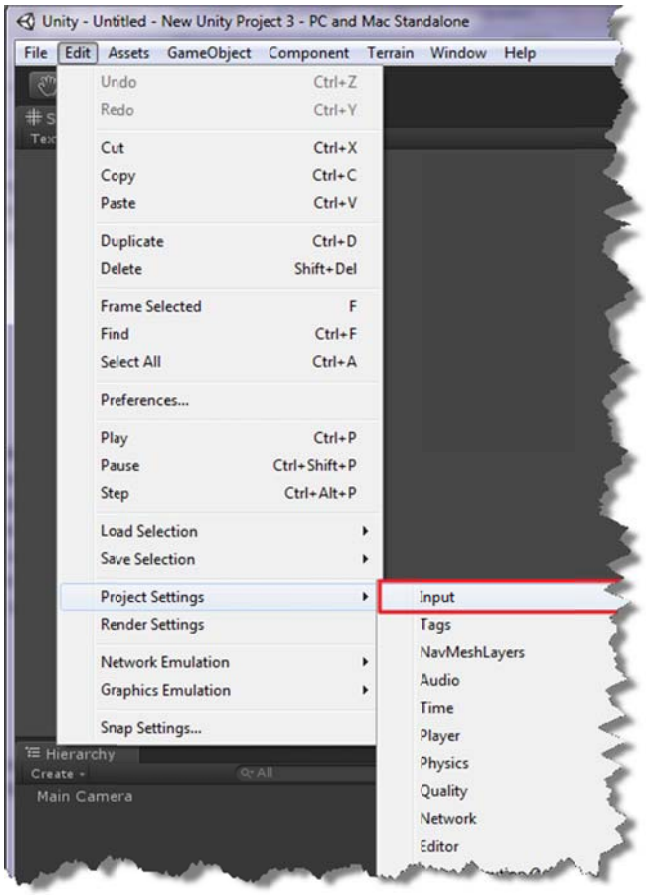


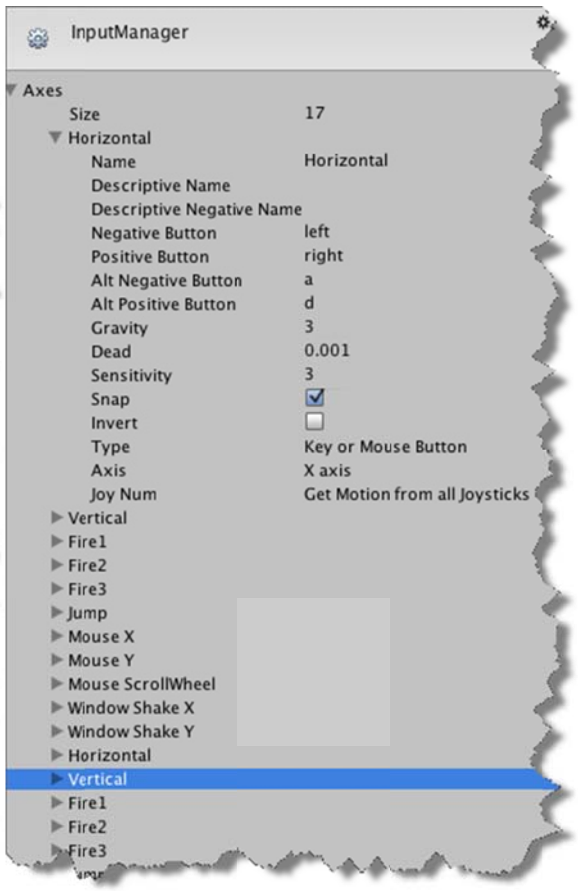
Others we didn't mention
For those wondering what else is possible there are virtual reality headsets that are starting to be made much more accurately and affordably, as well as 3d glasses. Probably most intriguing though would be the CAVE systems which would be as close to a Star Trek type holodeck that is available now, as it is a real 10' x 10' x 10' (typically) room that you can interact with virtual elements, some even have moving floors to allow a truly immersive experience.

Connecting the dots
Once we have chosen how we would like to interact with our models we can set up our project to understand the inputs that we want. We will do this via the Edit -> Project Settings -> Input . This opens up the Input manager where we can start to assign different values to the program to tell it what keys or buttons are assigned to what actions and responses.

<u>Below is from the Unity 3D Documentation and breaks down what the Input settings are:</u>

The Input Manager is where you define all the different input axes and game actions for your project.





InputManager

Axes
Size 17
Horizontal
Name Horizontal
Descriptive Name
Descriptive Negative Name
Negative Button left
Positive Button right
Alt Negative Button a
Alt Positive Button d
Gravity 3
Dead 0.001
Sensitivity 3
Snap ✓
Invert ☐
Type Key or Mouse Button
Axis X axis
Joy Num Get Motion from all Joysticks
Vertical
Fire1
Fire2
Fire3
Jump
Mouse X
Mouse Y
Mouse ScrollWheel
Window Shake X
Window Shake Y
Horizontal
Vertical
Fire1
Fire2
Fire3

Axes	Contains all the defined input axes for the current project: Size is the number of different input axes in this project, Element 0, 1, ... are the particular axes to modify.
Name	The string that refers to the axis in the game launcher and through scripting.
Descriptive Name	A detailed definition of the Positive Button function that is displayed in the game launcher.
Descriptive Negative Name	A detailed definition of the Negative Button function that is displayed in the game launcher.
Negative Button	The button that will send a negative value to the axis.
Positive Button	The button that will send a positive value to the axis.
Alt Negative Button	The secondary button that will send a negative value to the axis.

Alt Positive Button	The secondary button that will send a positive value to the axis.
Gravity	How fast will the input recenter. Only used when the Type is key / mouse button.
Dead	Any positive or negative values that are less than this number will register as zero. Useful for joysticks.
Sensitivity	For keyboard input, a larger value will result in faster response time. A lower value will be more smooth. For Mouse delta the value will scale the actual mouse delta.
Snap	If enabled, the axis value will be immediately reset to zero after it receives opposite inputs. Only used when the Type is key / mouse button.
Invert	If enabled, the positive buttons will send negative values to the axis, and vice versa.
Type	Use Key / Mouse Button for any kind of buttons, Mouse Movement for mouse delta and scrollwheels, Joystick Axis for analog joystick axes and Window Movement for when the user shakes the window.
Axis	Axis of input from the device (joystick, mouse, gamepad, etc.)
Joy Num	Which joystick should be used. By default this is set to retrieve the input from all joysticks. This is only used for input axes and not buttons.

Properties

Details

All the axes that you set up in the Input Manager serve two purposes:

- They allow you to reference your inputs by axis name in scripting
- They allow the players of your game to customize the controls to their liking

All defined axes will be presented to the player in the game launcher, where they will see its name, detailed description, and default buttons. From here, they will have the option to change any of the buttons defined in the axes. Therefore, it is best to write your scripts making use of axes instead of individual buttons, as the player may want to customize the buttons for your game.

Button Names

To map a key to an axis, you have to enter the key's name in the Positive Button or Negative Button property in the Inspector.

The names of keys follow this convention:

- Normal keys: "a", "b", "c" ...
- Number keys: "1", "2", "3", ...
- Arrow keys: "up", "down", "left", "right"
- Keypad keys: "[1]", "[2]", "[3]", "[+]", "[equals]"
- Modifier keys: "right shift", "left shift", "right ctrl", "left ctrl", "right alt", "left alt", "right cmd", "left cmd"
- Mouse Buttons: "mouse 0", "mouse 1", "mouse 2", ...
- Joystick Buttons (from any joystick): "joystick button 0", "joystick button 1", "joystick button 2", ...
- Joystick Buttons (from a specific joystick): "joystick 1 button 0", "joystick 1 button 1", "joystick 2 button 0", ...
- Special keys: "backspace", "tab", "return", "escape", "space", "delete", "enter", "insert", "home", "end", "page up", "page down"
- Function keys: "f1", "f2", "f3", ...

The names used to identify the keys are the same in the scripting interface and the Inspector.

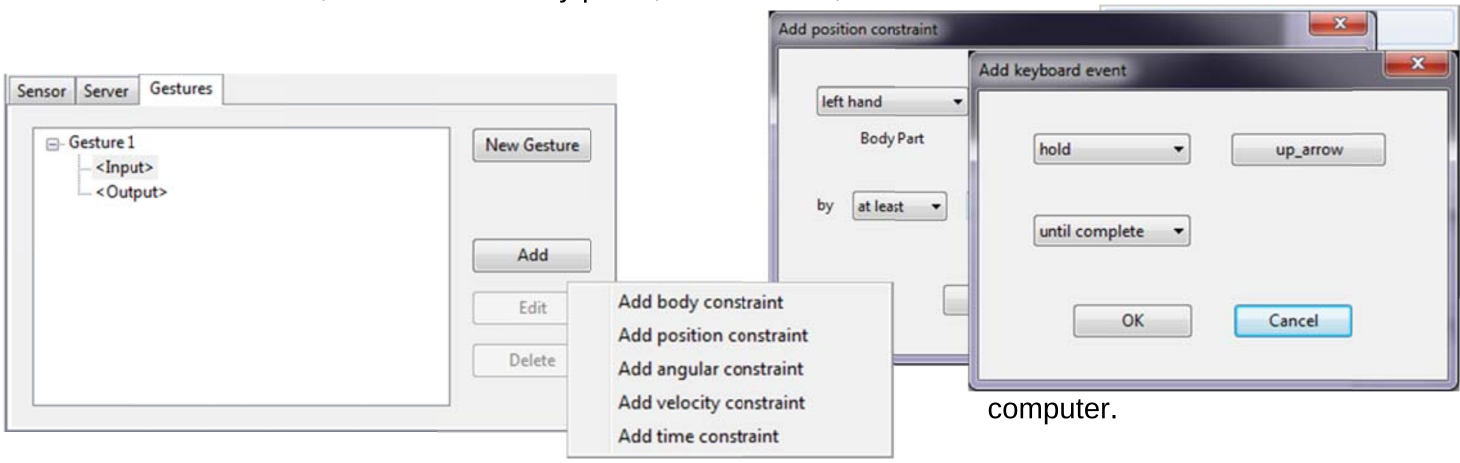
```
value = Input.GetKey ("a");
```

Beyond that

Those settings focus on the keyboard and game pads, but what if we want to go beyond that? Then we start to look at middleware. The kind that we will focus on here will be for the Microsoft Kinect, but there are many out there for differing purposes. If you want you can use some to make your game pad work for controlling Revit or Navisworks!

For working with Kinect we will look at two options. One is FAAST which is a program created at the University of Southern California. What it does is interpret the actions of a person using a Kinect and then tells the computer a command based on a predetermined set of “gestures” These gestures are created in the FAAST program and different scenarios and usage patterns can be saved and loaded as needed. Because this program interprets your actions and can redirect those to commands you can use the Kinect to do things that have not been specifically coded for it.

To set these gestures up you will need to have installed either the Microsoft or the OpenNI drivers for your Kinect, and then download and run (the program doesn't install) the FAAST tool. Once there you will go to the Gestures tab and we would start by saying new gesture. Once that is done you can then select the Input or Output and choose Add. This is where you can tell it what actions to look for, and then what key press, or command, etc that it should send to the computer.



The other option we have allows Unity to use the Kinect as a control method,

Glossary and Links:

Throughout this document there may be some terms that are not common to a typical AEC workflow, we have tried to break some of these down below:

Key to understanding Unity is the relationship between a GameObject and a Component.

GameObjects

A GameObject is the fundamental building block in Unity. Every object in your game is a GameObject. However, GameObjects don't do anything on their own. Think of a GameObject as an empty cooking pot, and Components as different ingredients that make up your recipe of gameplay

Components

Components are the building blocks of GameObjects. Without them, the GameObject won't do anything interesting.
A Component may represent visible entities, such as meshes, materials, terrain data or a particle system. Other Component types are more abstract, such as Cameras and Lights, which do not have a physical model representing them; instead, you will see an icon and some wireframe guidelines illustrating their key settings.
A Component is always attached to a GameObject.

Prefabs

A Prefab is a type of asset, a reusable Game Object stored in Project View. A Prefab is an Asset which has been defined as a template. It is to Unity what a template document is to a word processing application. When you place a Prefab into your Scene, Unity places a link to the Prefab into the Hierarchy Pane, not a complete copy. This is called instantiation. Each link you make is referred to as an instance of the Prefab.
Prefabs can be inserted into any number of scenes, multiple times per scene. When you add a Prefab to a scene, you create an instance of it. All Prefab instances are linked to the original Prefab and are essentially clones of it. No matter how many instances exist in your project, when you make any changes to the Prefab you will see the change applied to all instances.

Assets

All your imported Assets appear in the Project Pane and they can be almost anything: a simple material or texture, audio files, or even a complete, prefabricated GameOb--ject (known as a "Prefab").

Mapping:

Within the context of this class it can refer to one of two things, either taking a material or texture, and placing it on an elements surface for the purpose of rendering, or for the purpose of telling the system what should happen when a button is pressed.

Middleware:

This is a program that is a go-between for other programs, in the case of this class it will refer to programs that interpret the commands sent from a controlling device, and then translate those commands to input for a different program. IE pressing the 'X' button on a controller tells a program to run the door open animation.

Unity Software
http://unity3d.com/

3D Connexion
http://www.3dconnexion.com/

Microsoft Kinect
http://www.microsoft.com/en-us/kinectforwindows/
http://blogs.msdn.com/b/kinectforwindows/

Flexible Action and Articulated Skeleton Toolkit (FAAST) Kinect Toolkit
http://projects.ict.usc.edu/mxr/faast/

Cave System Explanation
http://en.wikipedia.org/wiki/Cave_automatic_virtual_environment