

Topics

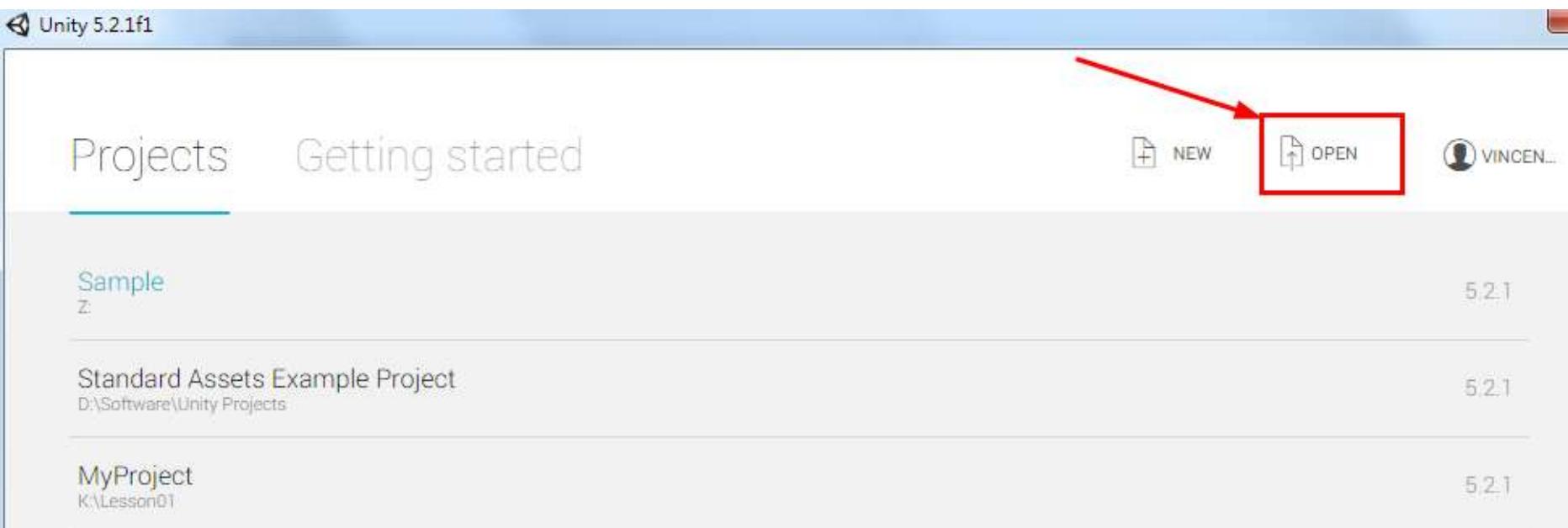
- Basic operation and viewport control
- Scene building
- Material
- Lighting
- Particle Systems

Basic operation and viewport control

Getting Start

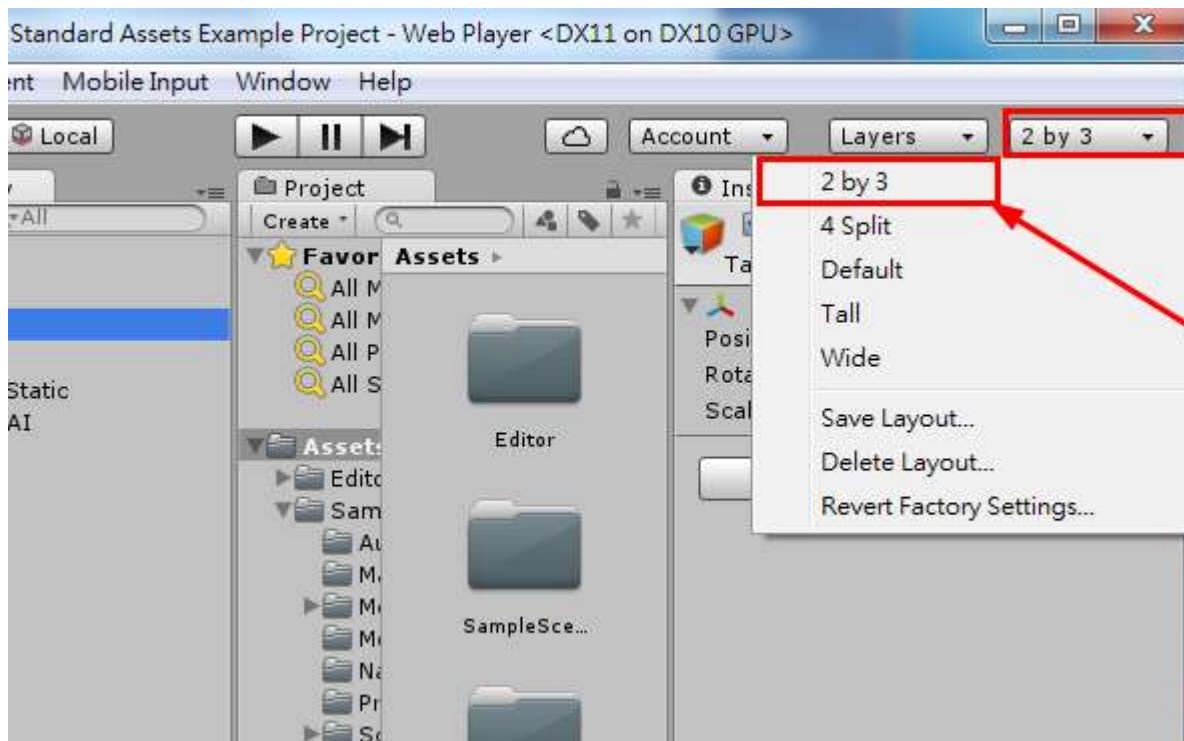
■ Start Unity and open a sample project

- You may install the “Standard Assets Example Project” for the Unity3D official website



Layout

- Change the layout to 2 by 3



Interface

■ Hierarchy panel

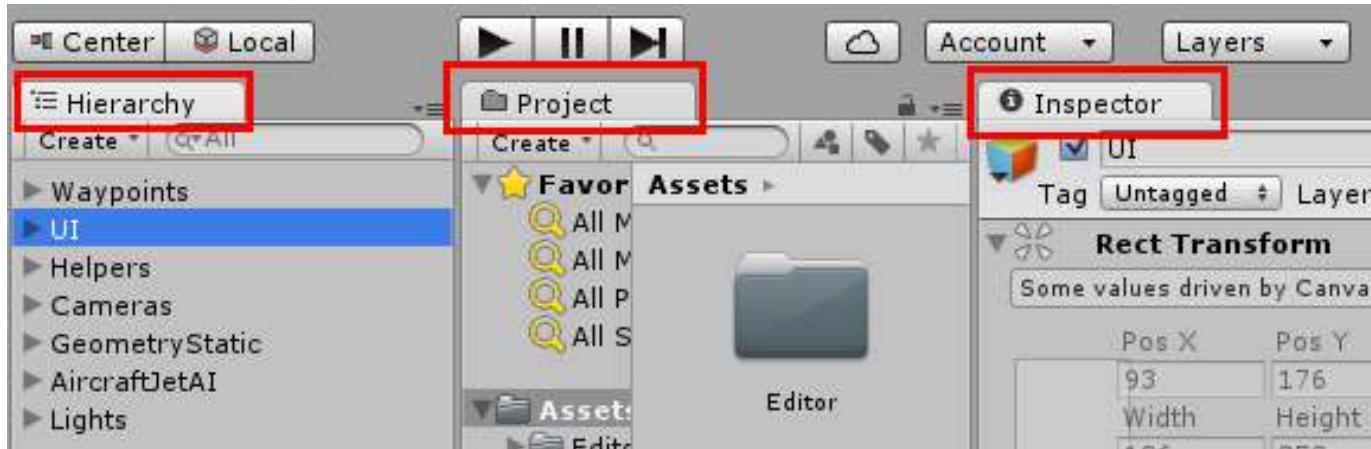
- Text list of game objects and sub-objects in the current scene

■ Project panel

- Contents of Project 'assets' folder (i.e. files in that folder), library of scripts, digital media files, and scenes

■ Inspector panel

- Settings for currently selected asset/object/setting



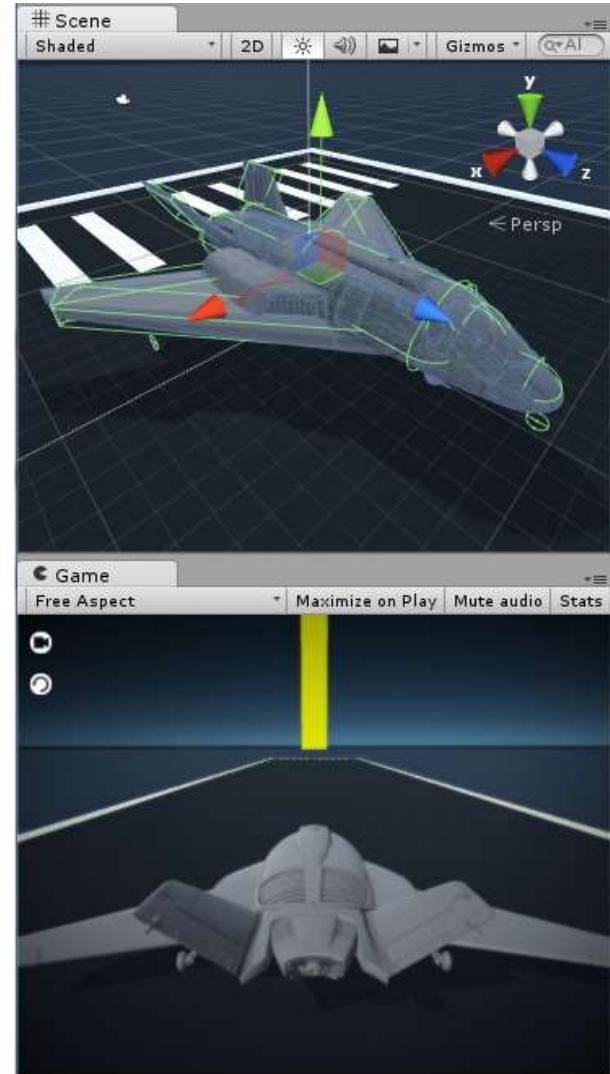
Interface

■ Scene panel

- Where the game is constructed

■ Game panel

- The preview window, active only in play mode
- Press “Ctrl+P” to run the game



Scenes

- A unity “scene” is essentially a “level” or “screen”
- Typical game
 - Welcome / main menu screen
 - Buttons: play game / see high scores / read instructions / change input settings
 - Level 1 / Level complete / Level 2 etc...
 - Game Over / Enter details for new High Score ...
- All the above would be separate “scenes” in unity
- Some scenes may be entirely based around the Unity GUI scripts / components – i.e. be text / buttons on screen

Project Assets

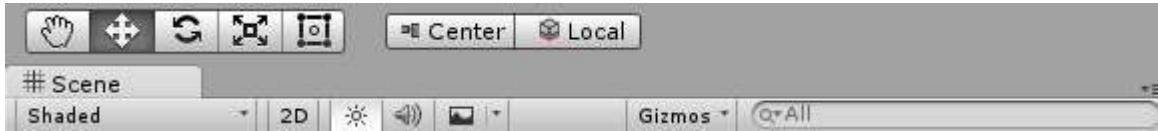
- The Assets folder for each Unity project contains:
 - Scenes
 - Media assets
(images, sounds files, 3D models)
 - Script files
 - “packages”
(collections of unity assets, ready to import)
- The contents of the Unity “Project” panel reflect the contents of the “Assets” folder

Game Objects – in current ‘scene’

- Everthing in a scene is either a Game Object
 - or a component INSIDE a Game Object
- Every Game Object has at least 1 COMPONENT
 - Its TRANSFORM – an object’s position, scale, rotation
 - Other components depend on object type (audio, mesh, material, script etc.)
- Game objects can be in a HIERARHCY – so an object can be a sub-object of another object
 - E.g. an “arm” object can be a sub-object of a “body” object etc.

Interface

■ Control bar

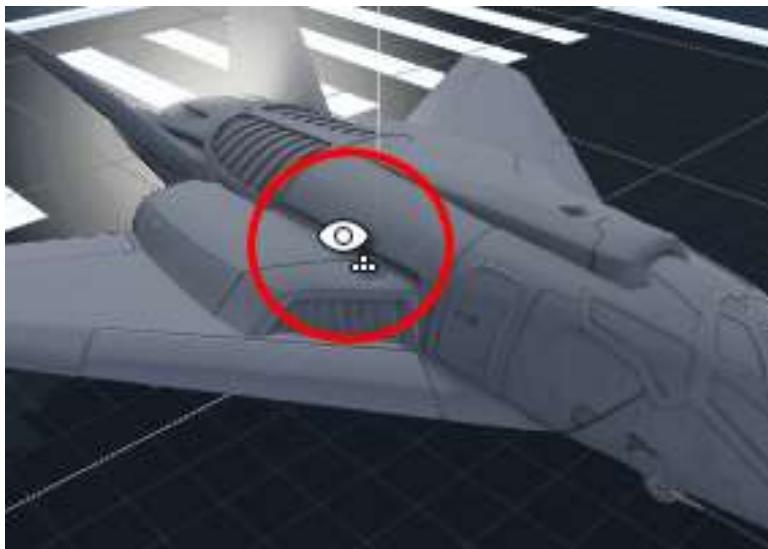


- Draw mode (default is 'Shaded')
- Display mode – switching between 2D and 3D view
- Toggle scene lighting
- Toggle audition mode—previews audio sources in the current scene
- Toggle overlays—shows and hides GUI elements and Skyboxes and toggles the 3D grid
- Gizmos—use this pop-out menu to show or hide Gizmos, the 2D icons of cameras, lights, and other components shown in the scene

Viewport Control

■ Flythrough Scene navigation

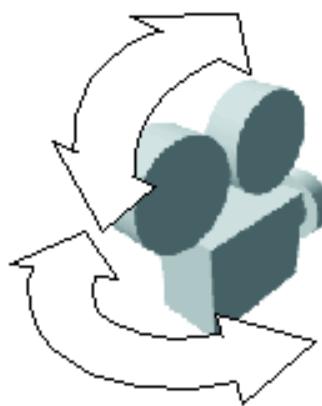
- Hold down the right mouse button
- Drag to look around in first-person style
- Use W, A, S and D to move
- Q and E to descend and ascend
- Hold down “Shift” to move faster



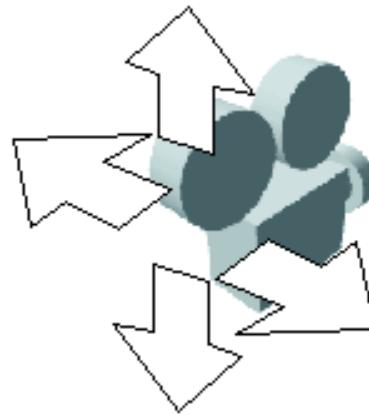
Viewport Control

- By using the following short-cut

- Orbit (Alt + LMB) / MMB
- Move (Alt + MMB)
- Zoom (Alt + RMB) / Scroll-wheel
- Expand view (Shift + Spacebar)
- Focusing (select game object and press “F”)



Orbit



Move



Zoom

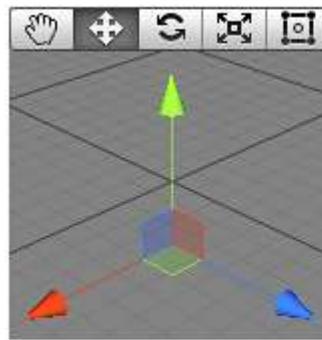
Positioning Game Objects

- By using the hand tool

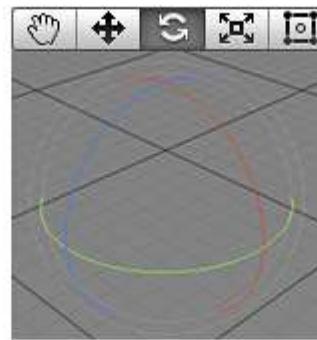


- By Shortcuts

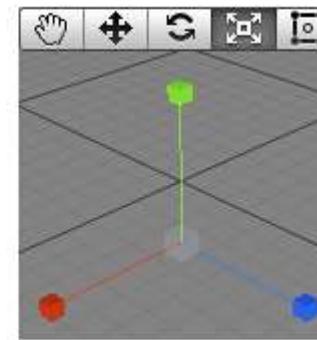
- Move tool (W)
- Rotate tool (E)
- Scale tool (R)



Translate (W)



Rotate (E)

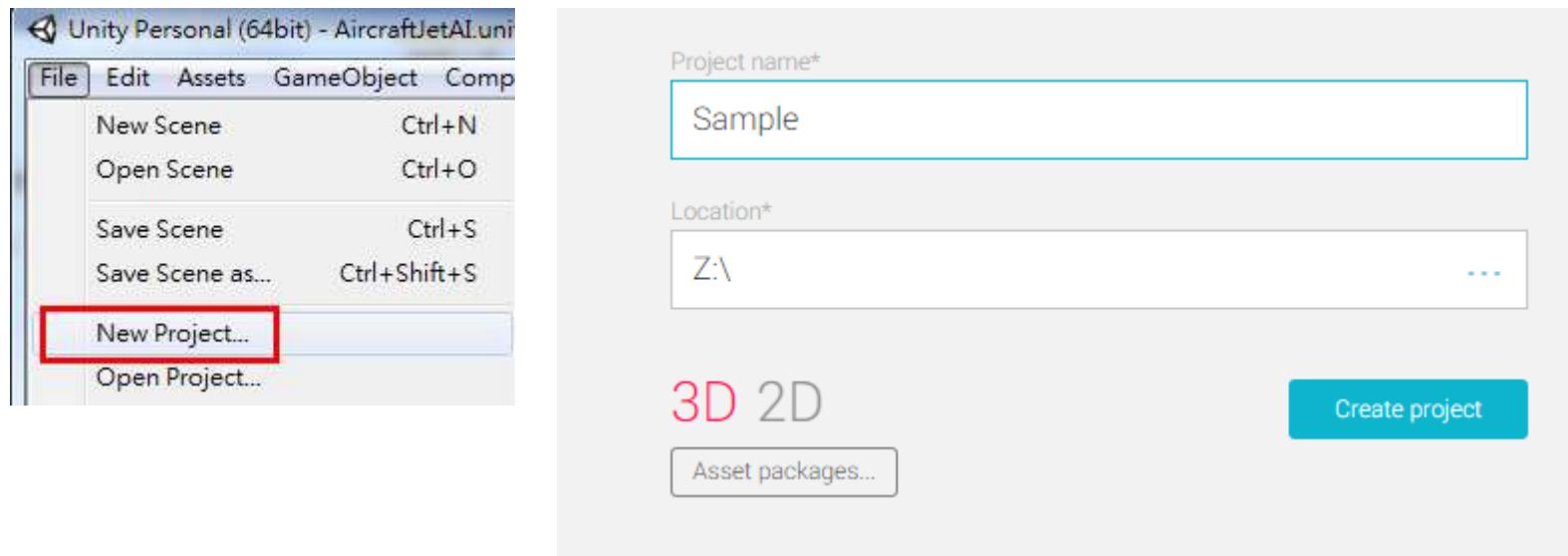


Scale (R)

Scene Building

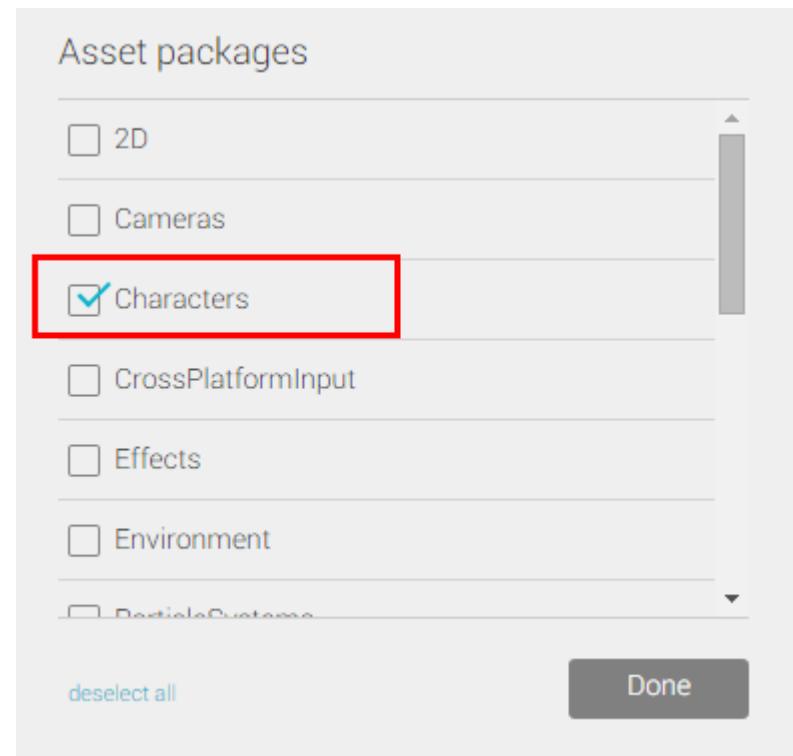
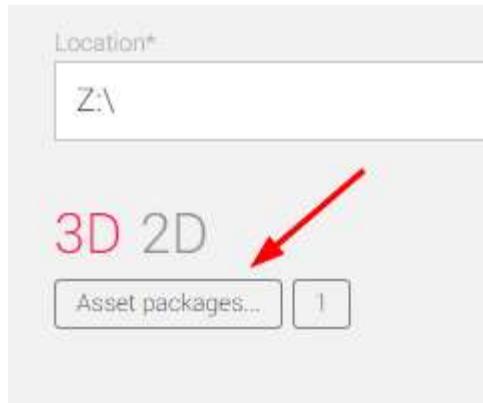
Create Project

- Each game should start with a new project
- File → New Project
- Select the Location, the project name will be the folder of your project
- Choose between 3D/2D



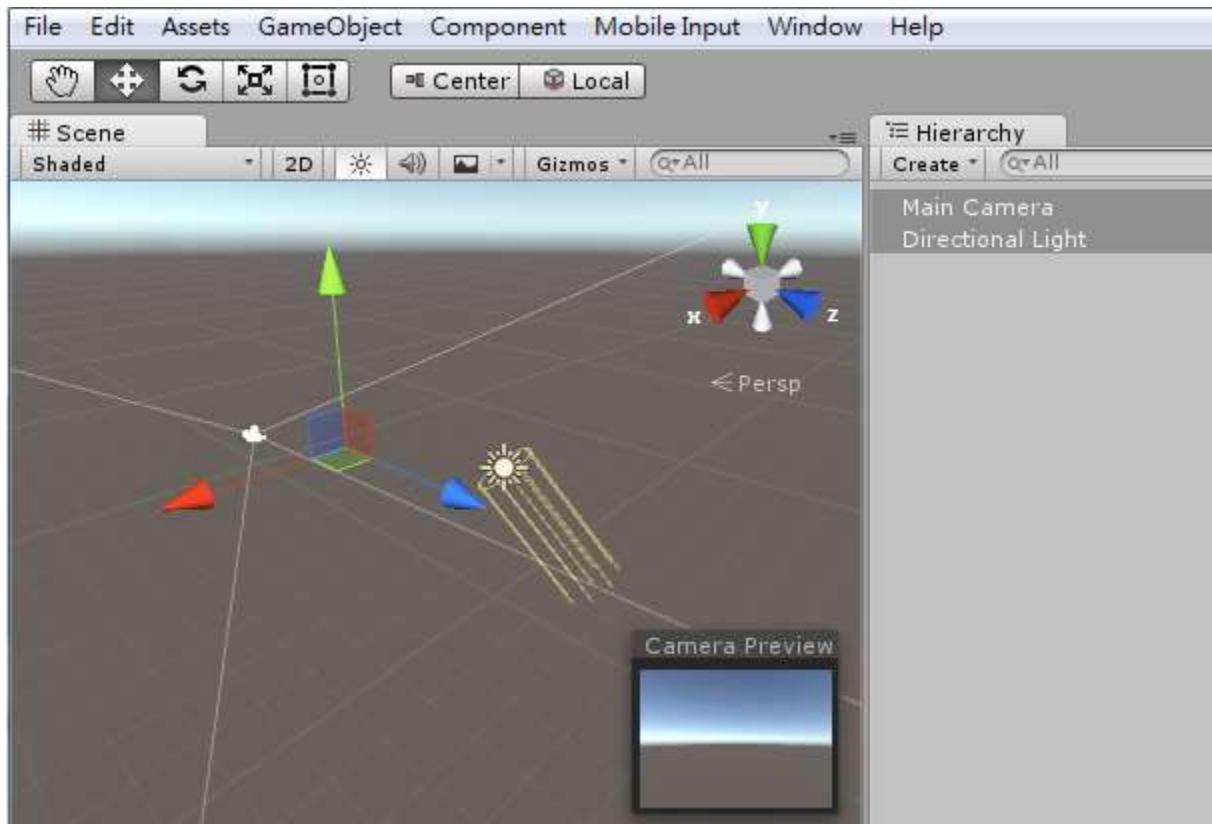
Create Project

- You may include the downloaded/installed asset package
- Download the Standard Assets packages from the Unity3D official website
- Select the “Character” asset



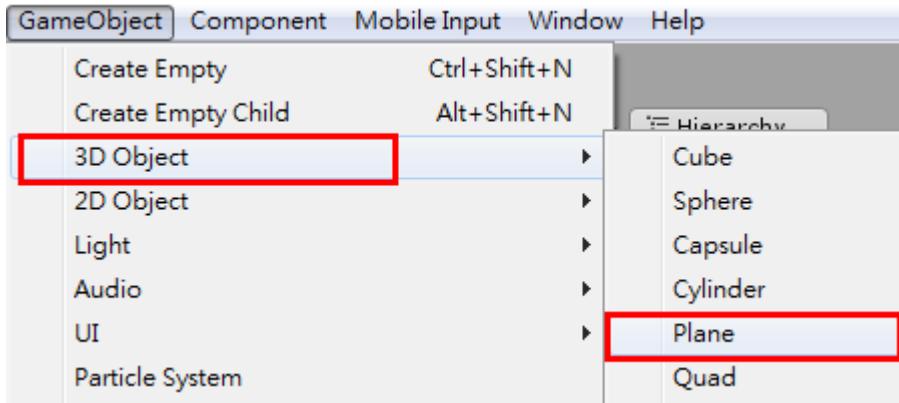
Scene

- A new scene is created with a default camera and lighting, each game will have a number of scenes

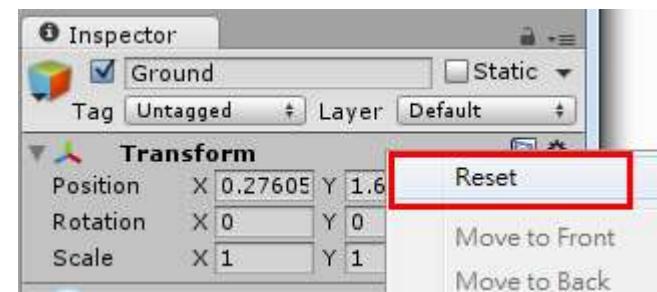


Scene Building

- Every object in your game is a GameObject
- Create a plane and name it “ground”

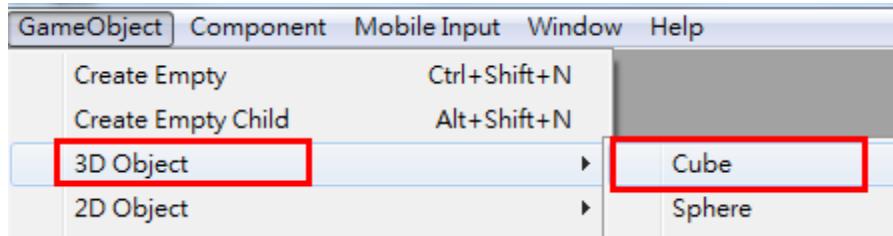


- From the Inspector panel, right-click Transform and “Reset” the ground position

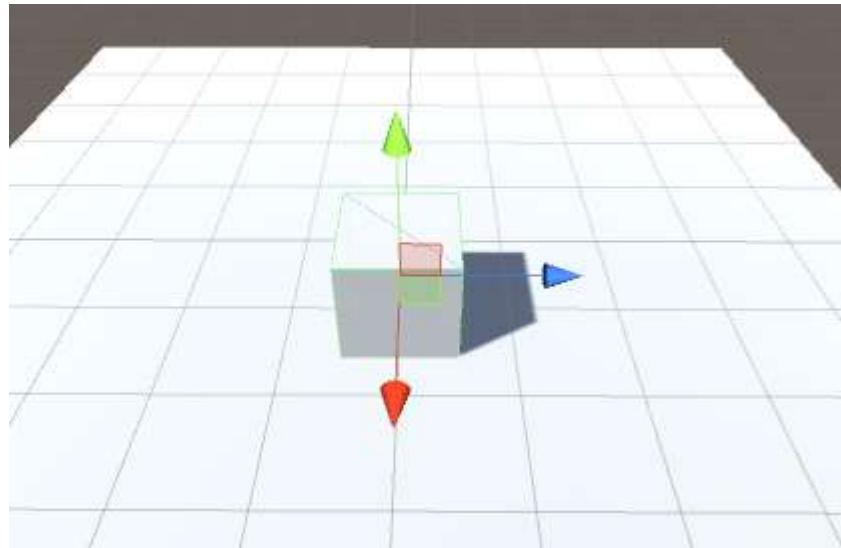


Scene Building

- Create a cube

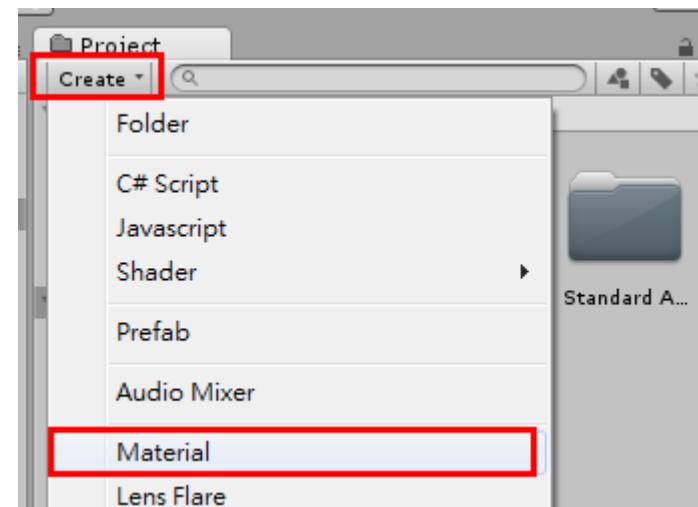
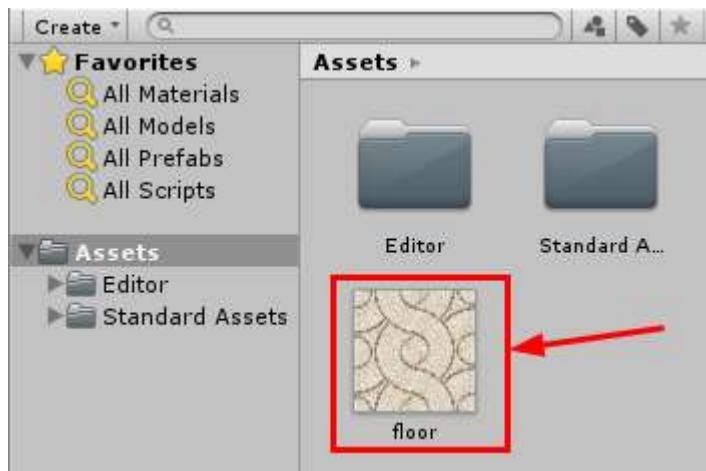


- Hold "Shift+Ctrl" to snap it to the ground surface



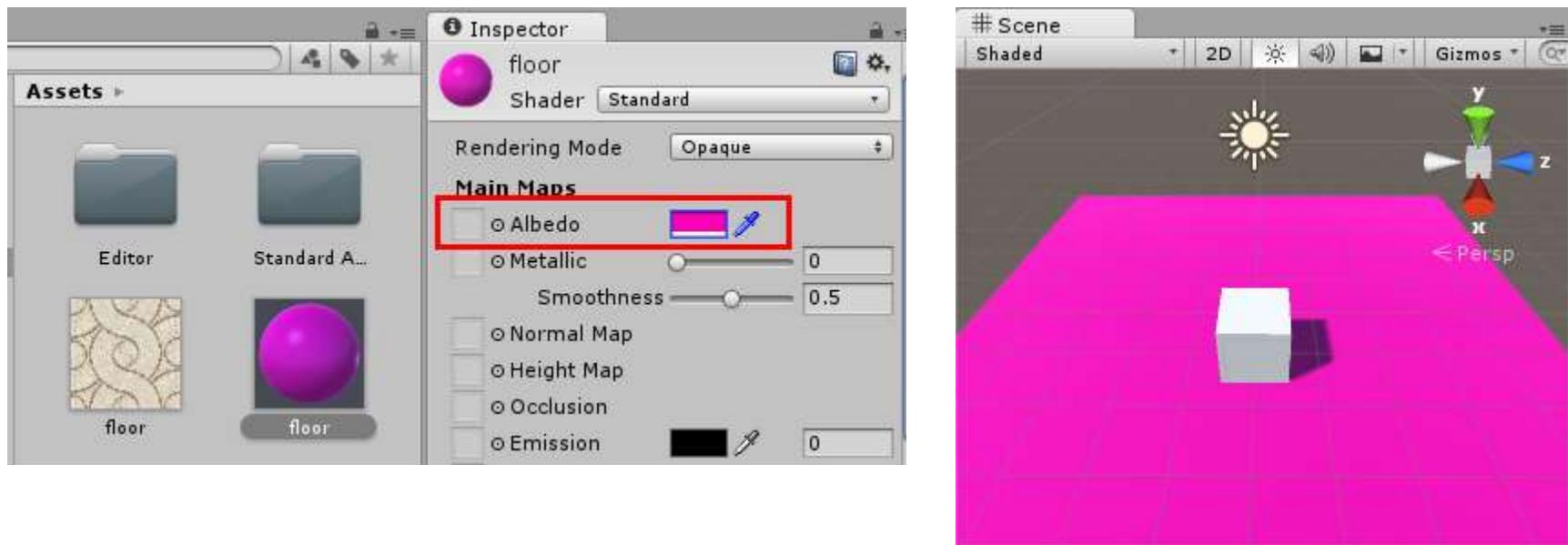
Assign Material

- Import the texture file to the Assets folder of the Project Panel by dragging from file manager
- Create a new material and name it floor



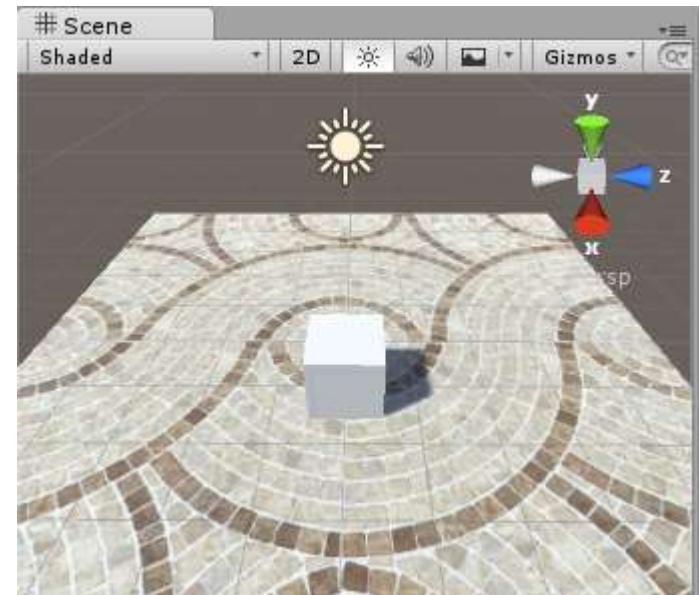
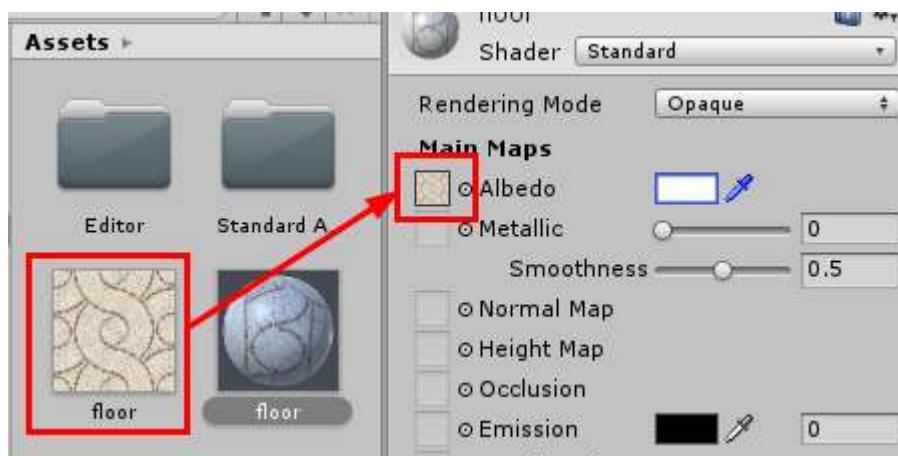
Assign Material

- Select the material and change the Albedo color
- Albedo is the diffuse color channel
- Drag the material to the ground to change its appearance



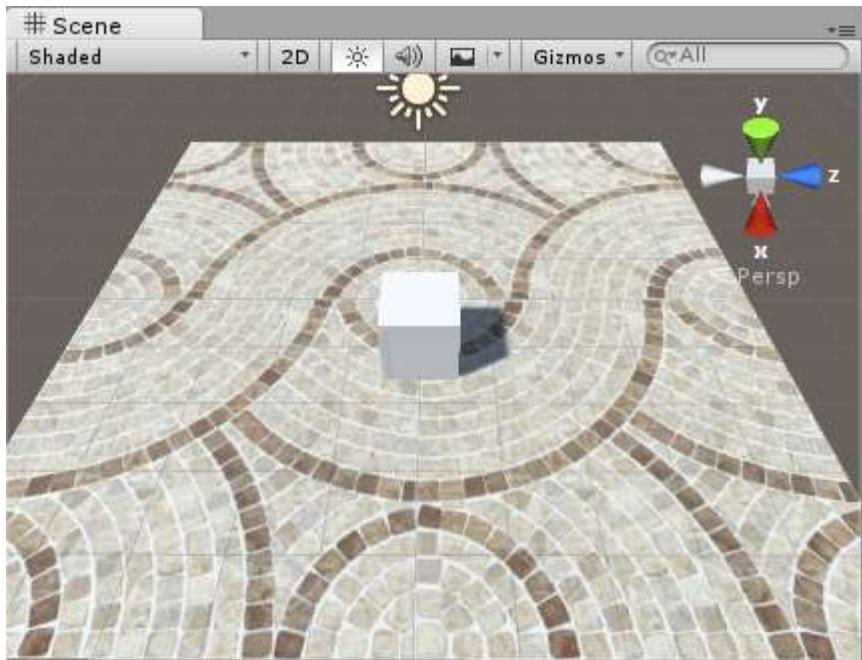
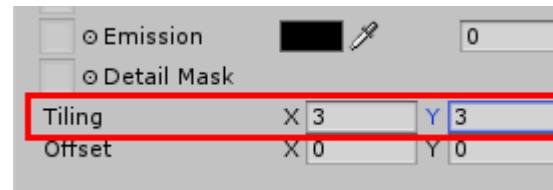
Assign Material

- You may assign the floor texture to the Albedo channel
- Remember to change the color to white, otherwise the texture will be tinted



Assign Material

- Set the “Tiling” for repeating the texture mapping



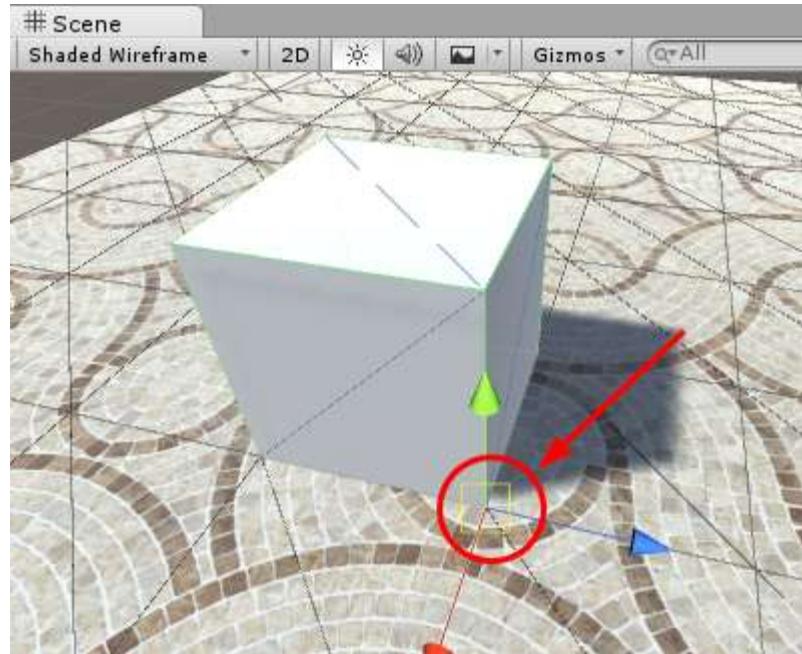
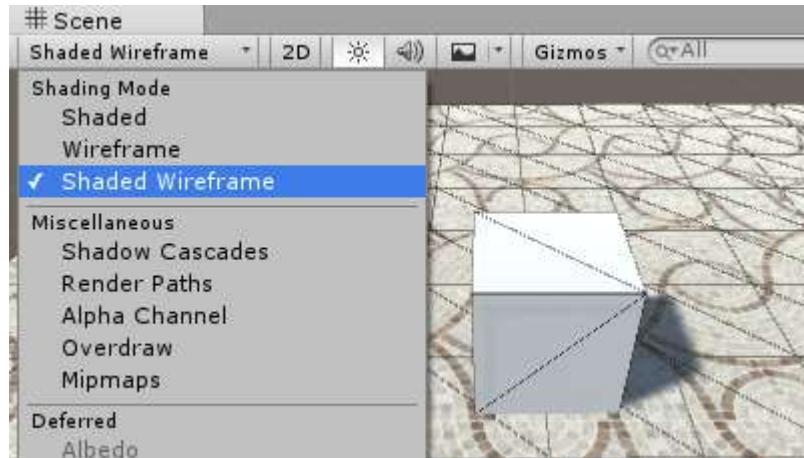
$x=1, y=1$



$x=3, y=3$

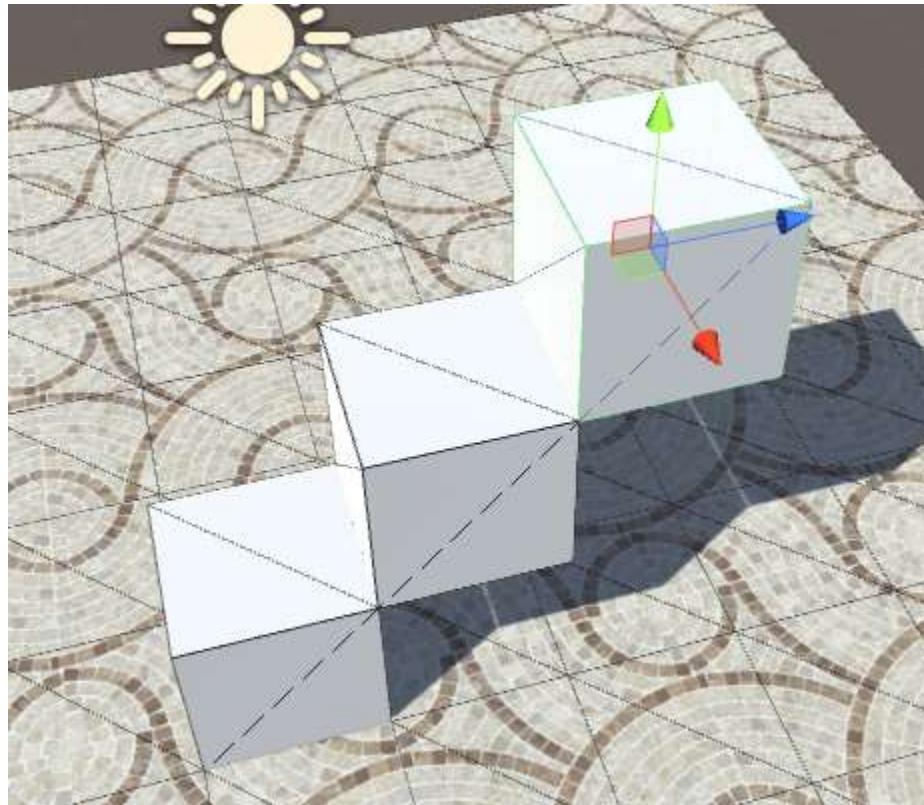
Vertices Snapping

- Change the draw mode to “Shaded Wireframe” for each viewing of wireframe and vertices
- Hold “V” to snap and move object vertex to vertex



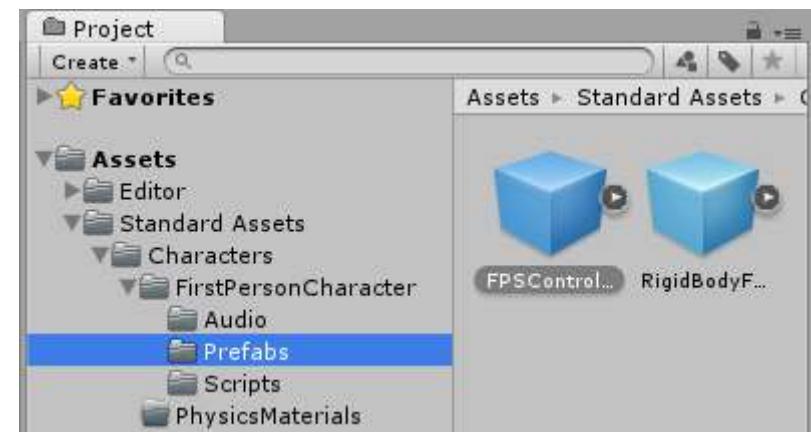
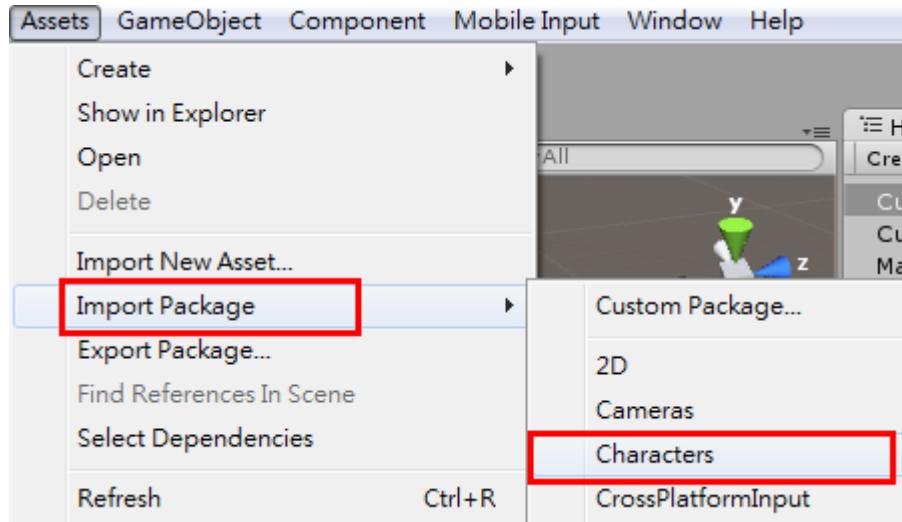
Vertices Snapping

- Select the cube, press “Ctrl+D” for duplication
- Try to use the vertices snapping skill to build a stair



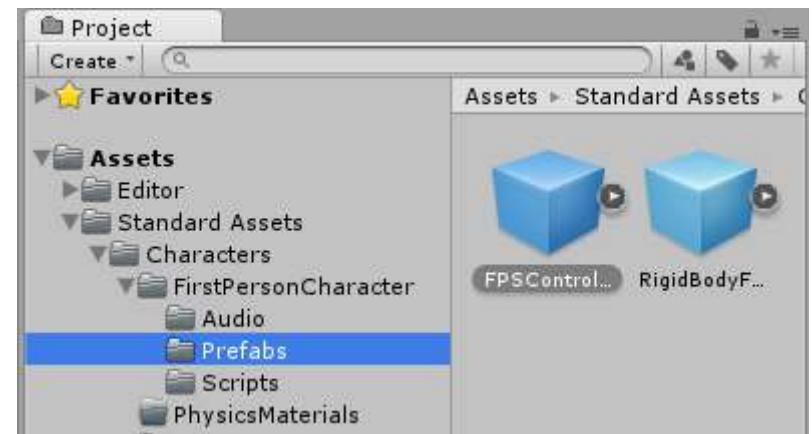
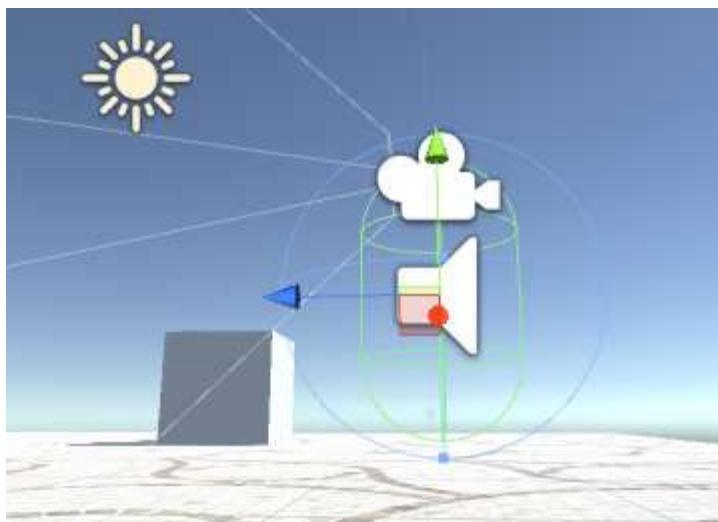
First Person Controller

- Make sure character asset is imported
- You may import at any time from the Assets menu
- It will then be placed at the following path
 - Standard Assets → Characters → FirstPersonCharacter → Prefabs



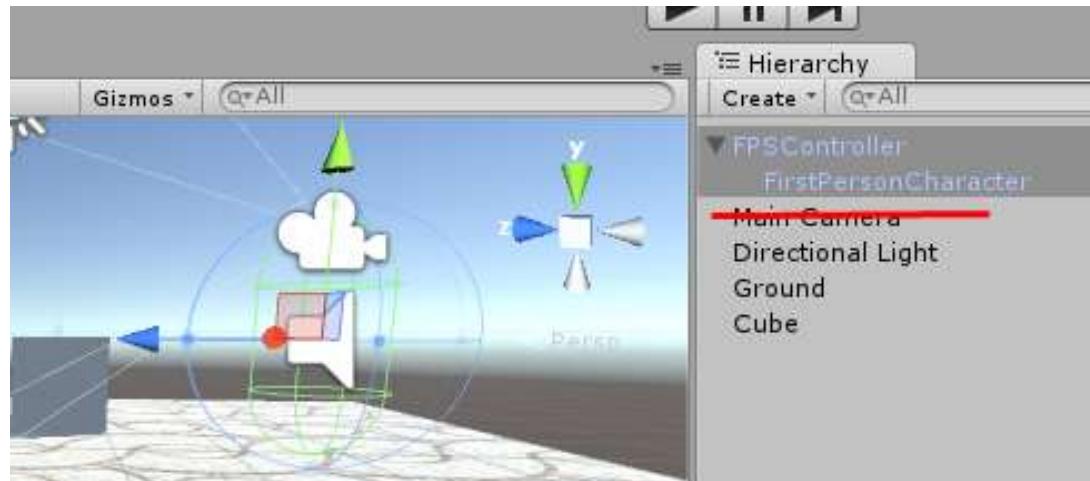
First Person Controller

- Drag the FPSController to the scene, make sure it is placed above the floor
- You may hold down “Ctrl” and pull the green arrow to move it up by 1 unit



First Person Controller

- The default camera will be override by the camera of the first person controller
- Remove the main camera so that sound can be received properly
- Press “Ctrl+P” to run the scene
- Use W/A/S/D and SpaceBar to control the character movement



Save

■ Final save your scene

- File → Save Scene as
- Save your scene under the “Assets” folder
- Each scene may refer to a part or a level of your game
- If you want to backup your game, try to duplicate the whole project folder rather than saving another scene

Export 3D Mesh

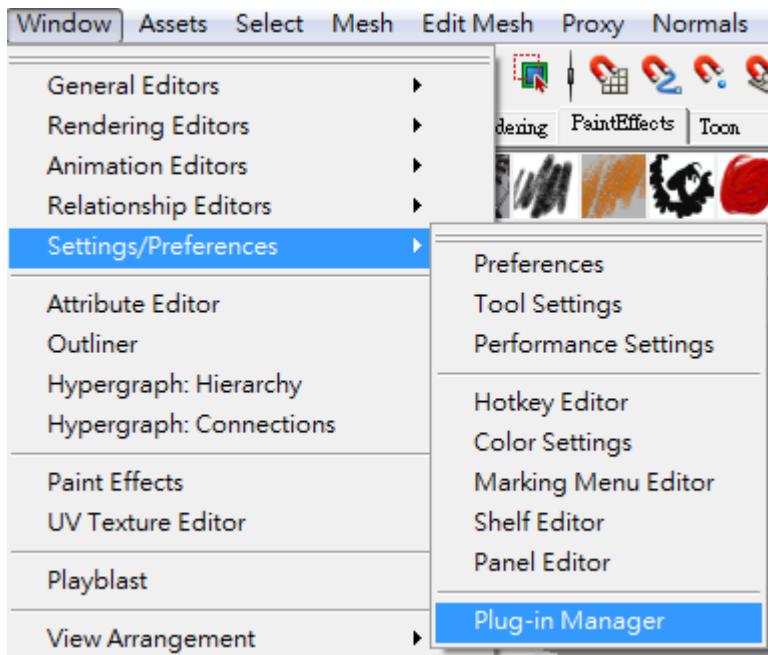
- Unity can import a number of 3D model, animation and bones including:

- Maya .mb & .ma
- 3Ds Max .max
- Autodesk .fbx
- Collada
-

3D Package Support	Meshes	Textures	Anims	Bones
Maya .mb & .ma ¹	✓	✓	✓	✓
3D Studio Max .max ¹	✓	✓	✓	✓
Cheetah 3D .jas ¹	✓	✓	✓	✓
Cinema 4D .c4d ^{1,3}	✓	✓	✓	✓
Blender .blend ¹	✓	✓	✓	✓
modo .lko ²	✓	✓	✓	
Autodesk FBX	✓	✓	✓	✓
COLLADA	✓	✓	✓	✓
Carrara ¹	✓	✓	✓	✓
Lightwave ¹	✓	✓	✓	✓
XSI 5.x ¹	✓	✓	✓	✓

Export 3D Mesh from Maya

- It will be better to export .fbx rather than .mb
- Remember to turn on the .fbx plugin before you can export .fbx from Maya
 - Window → Settings/Preferences → Plug-in Manager

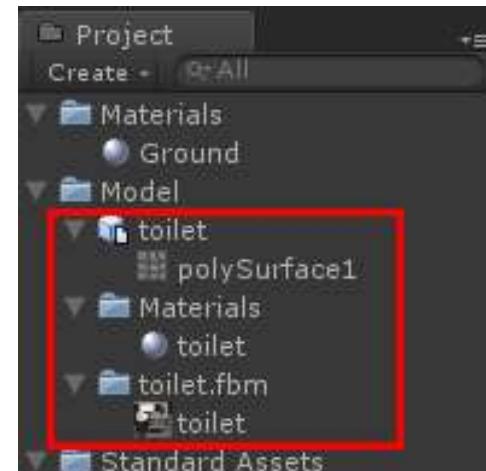
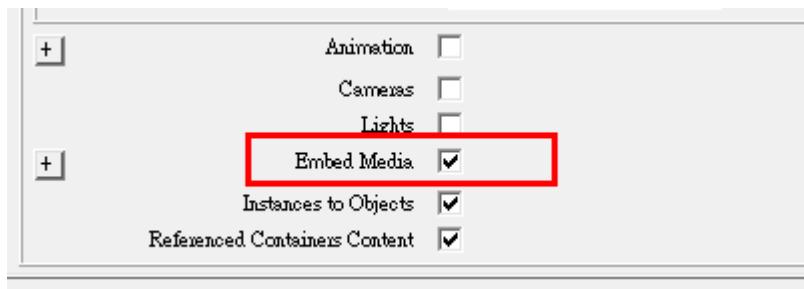


A screenshot of the Maya Plug-in Manager window. The title bar says 'Plug-in Manager'. Below it are 'Filter' and 'Help' buttons. The main pane shows a list of plugins with their status: 'Loaded' or 'Not Loaded' and 'Auto load' checked or unchecked. One plugin, 'fbxmaya.mll', is highlighted with a red border. Its status is 'Loaded' and 'Auto load' is checked. Other plugins listed include animImportExport.mll, ogfxShader.mll, cleanPerFaceAssignment.mll, cleacoat.mll, compositingIntexp.mll, ddsFloatReader.mll, decomposeMatrix.mll, DirectConnect.mll, fitTranslator.mll, and Fnx.mll.

Plugin	Loaded	Auto load
animImportExport.mll	<input type="checkbox"/>	<input type="checkbox"/>
ogfxShader.mll	<input type="checkbox"/>	<input type="checkbox"/>
cleanPerFaceAssignment.mll	<input type="checkbox"/>	<input type="checkbox"/>
cleacoat.mll	<input type="checkbox"/>	<input type="checkbox"/>
compositingIntexp.mll	<input type="checkbox"/>	<input type="checkbox"/>
ddsFloatReader.mll	<input type="checkbox"/>	<input type="checkbox"/>
decomposeMatrix.mll	<input type="checkbox"/>	<input type="checkbox"/>
DirectConnect.mll	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
fbxmaya.mll	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
fitTranslator.mll	<input type="checkbox"/>	<input type="checkbox"/>
Fnx.mll	<input type="checkbox"/>	<input type="checkbox"/>

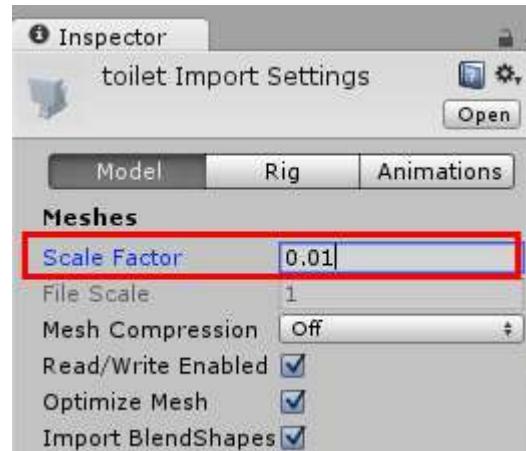
Export 3D Mesh from Maya

- Create a “model” folder inside the “Assets”
- Export the .fbx file directly to the “model” folder
 - File → Export Selection
- Remember to click the “Embed media” option
- Go back to Unity you will find that the mesh is exported together with the materials and texture



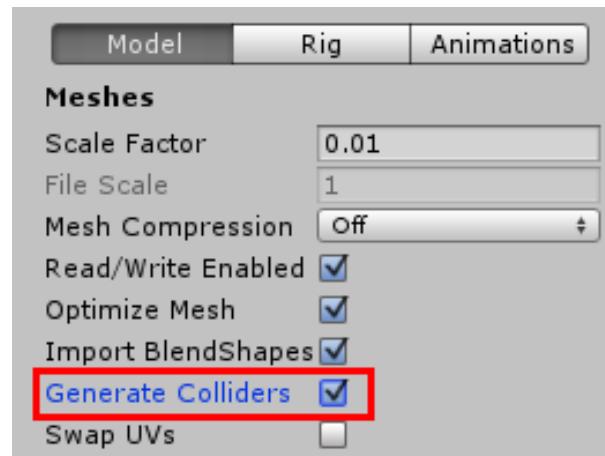
Export 3D Mesh from Maya

- Drag the mesh to the scene, the size seems to be very small
- Unity will automatically resize the scale of the imported mesh to 0.01
- Select the imported mesh, from the inspector panel change the scale factor to 0.1 1 and press “Apply”



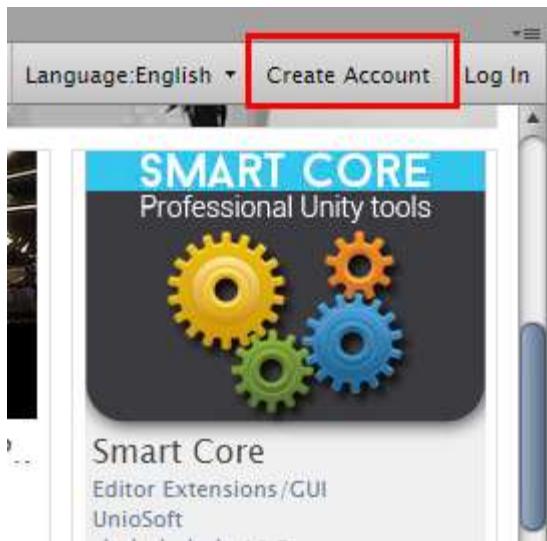
Export 3D Mesh from Maya

- You cannot step on the imported mesh as it is not included in the collision calculation
- Select the mesh from the project panel and click the “Generate Colliders” from the inspector panel
- Remember to press the “Apply” button



Assets Store

- You may download more free models from Unity Assets Store
- Press “Ctrl+9” to turn on the Assets store window
- Create Account before you can access the store



[Create account](#) [Sign in](#)

Create a Unity Account

You need to create a Unity Account to shop in the Online and Asset Stores, participate in the Unity Community and manage your license portfolio.

Assets Store

- After searching for the model, press Import to download and import to your current project

Free Furniture Pack 1

Category: 3D Models/Props/Furniture
Publisher: Ka Design
Rating: ★★★★☆ (1.93)
Price: Free

Import

Free Furniture Pack 1 is a sample of a series of beautiful modern furniture assets. This sample package contains two furniture elements.

Features:

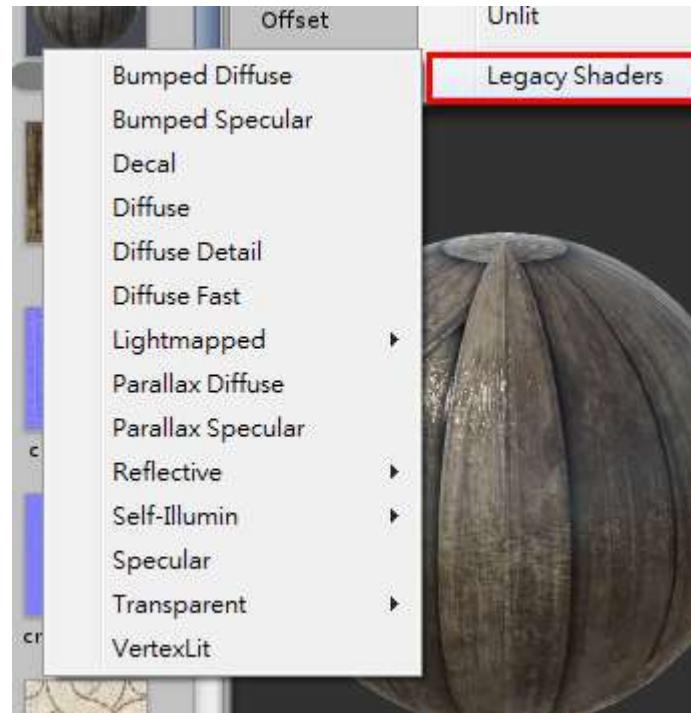
- Table model - 584 Tris
- Chair model - 3.9K Tris



Materials

Materials

- In Unity 5.0, Physically Based Standard Shader has replaced the built-in shaders in previous version
- They have been grouped under the Legacy Shaders

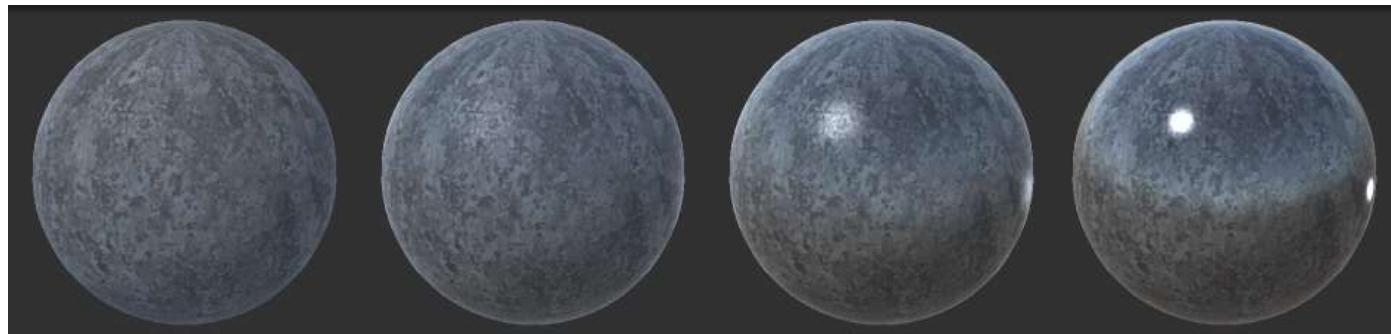


Materials

- Physically Based Standard (PBS) combine different shader properties into a single shader
- E.g. Diffuse, Specular, Bump, Transparent...
- Energy conservation
 - Never reflect more light than they receive
 - More specular → Less diffuse
 - Smoother surface → stronger and smaller highlight

Standard (Metallic)

- Increase the smoothness will increase the specular reflection



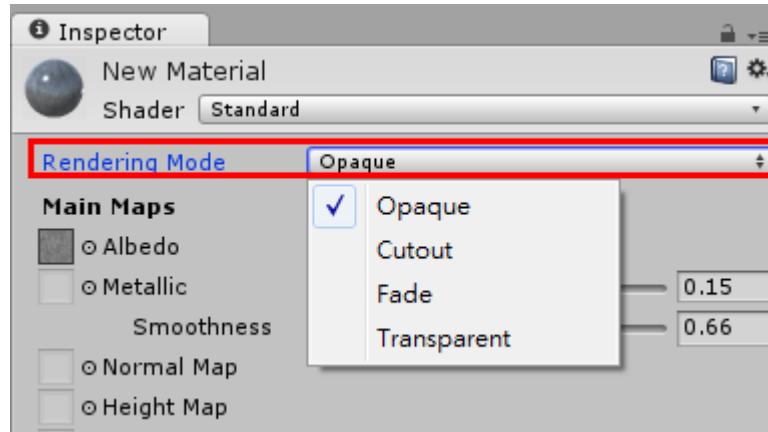
- Increase the Metallic value will lower the visibility of the diffuse color, more environment reflection



A range of metallic values from 0 to 1 (with smoothness at a constant 0.8 for all samples)

Rendering Mode

- Choose whether the object uses transparency
- 4 types of rendering model
 - Opaque
 - Cutout
 - Transparent
 - Fade



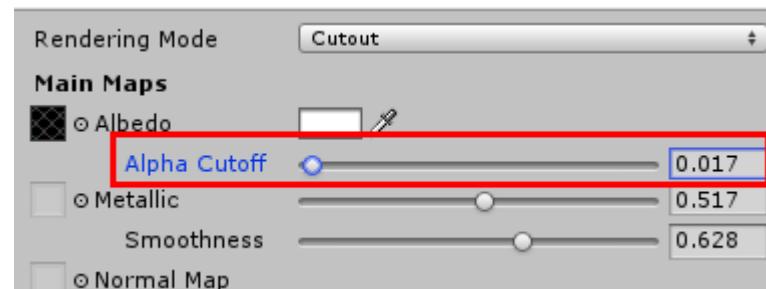
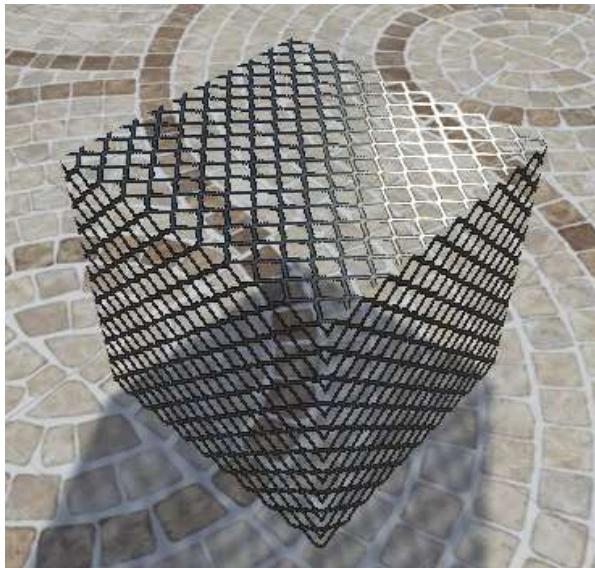
Rendering Mode

■ Opaque

- No transparent areas

■ Cutout

- No semi-transparent areas, either 100% opaque or invisible
- Adjust the Alpha Cutoff value to refine the edge



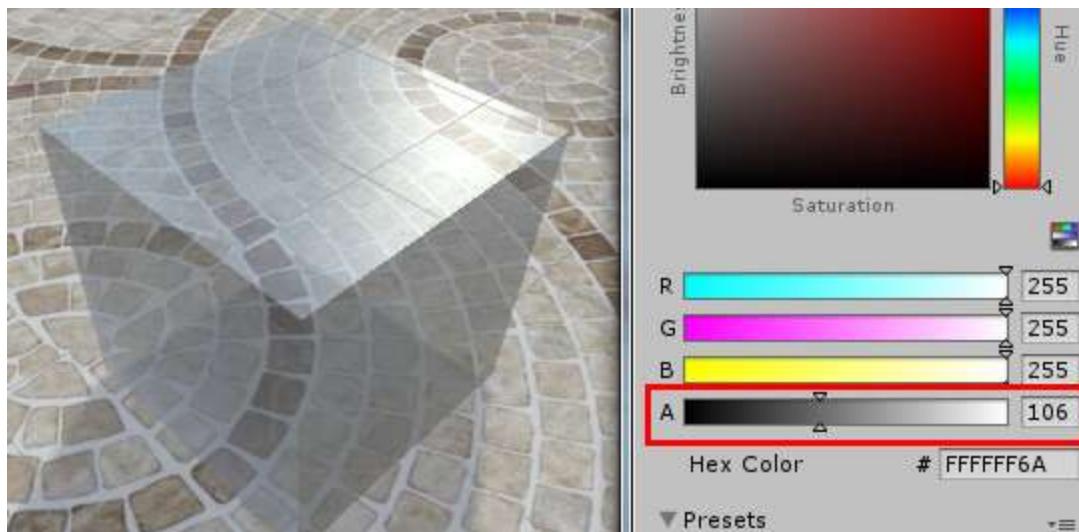
Rendering Mode

■ Fade

- Use for fading an object out entirely including the specular highlight or reflections

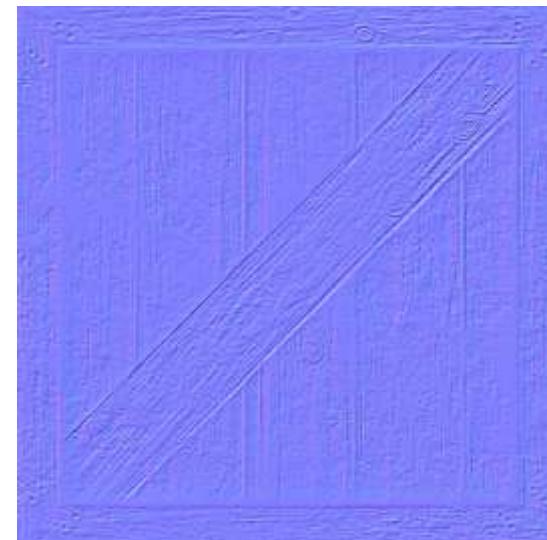
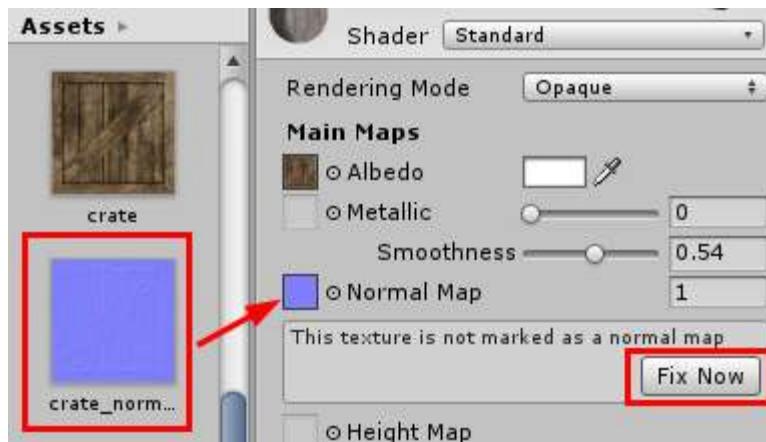
■ Transparent

- Realistic transparent take on the alpha channel of texture
- Reflection and highlight will remain visible



Normal Map

- Allow you to add surface detail such as bumps, grooves, and scratches to a model
- Require normal map texture generated from 3D modeling software
- Press “Fix Now” for the first time assigning the normal map



Normal Map

Without normal map

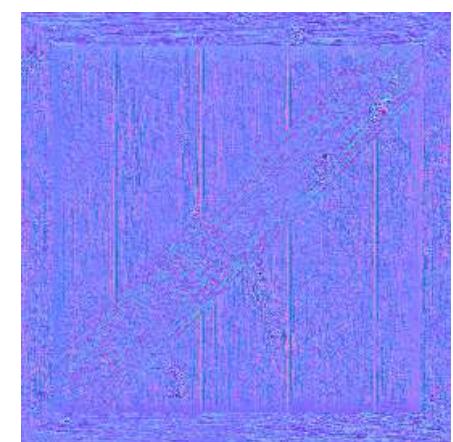
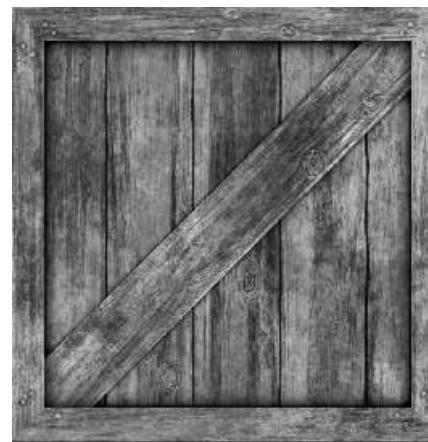
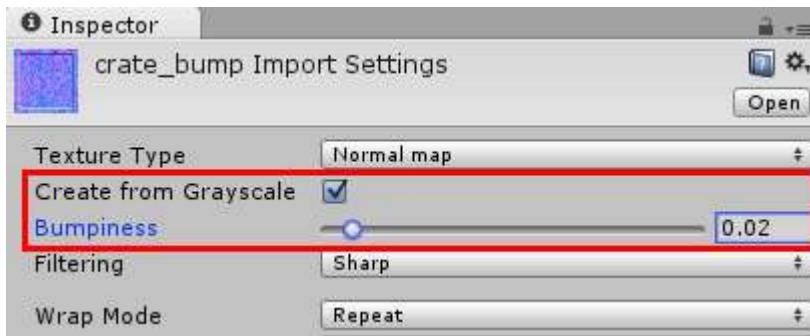


With normal map



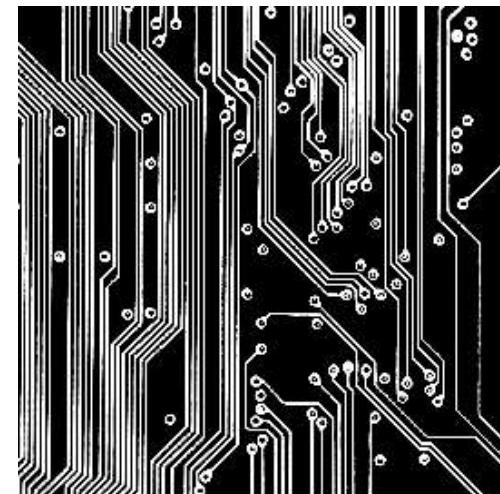
Normal Map

- Normal map can also be simulated by using gray scale image
- Select the texture and enable “Create from Grayscale”
- Setup the Bumpiness



Emission

- Use flat color to control over emission
- Or using texture map to define the emission areas



Lighting

Lighting

- Unity supports 4 types of lighting
 - Directional Lights
 - Point lights
 - Spot lights
 - Area lights

Lighting

■ Directional Light

- Does not have any identifiable source position
- All objects in the scene are illuminated
- Use for simulating sun light

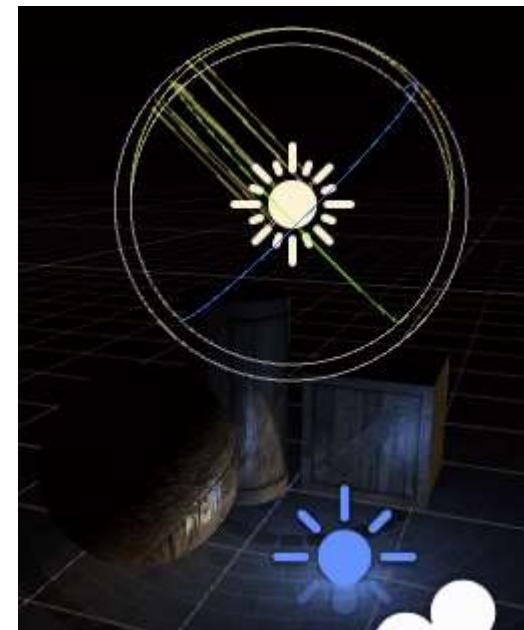
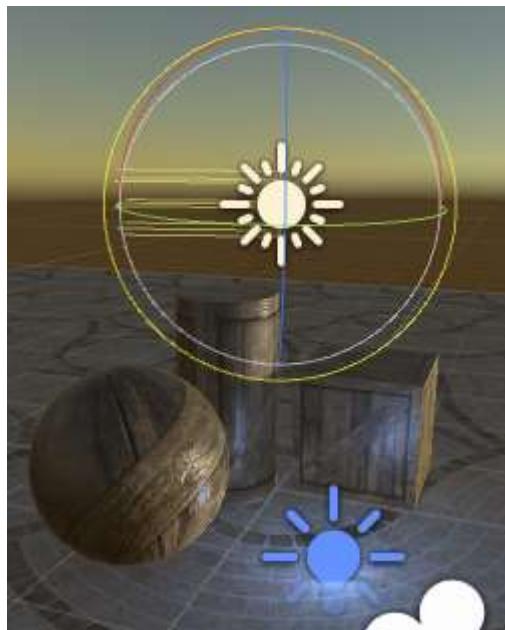
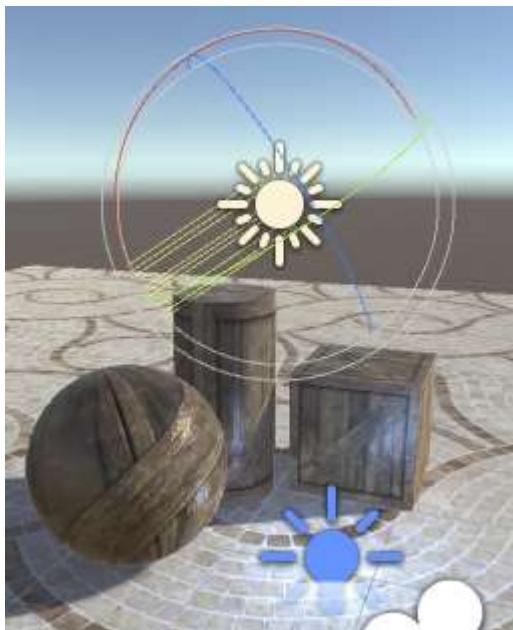


Effect of a Directional Light in the scene

Lighting

■ Directional Light

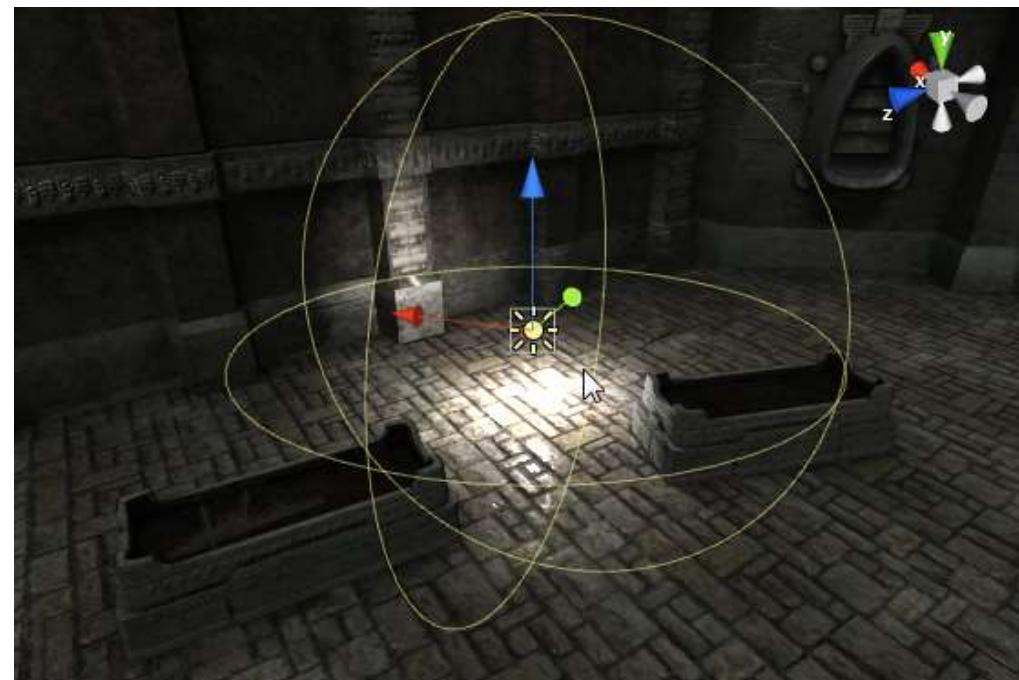
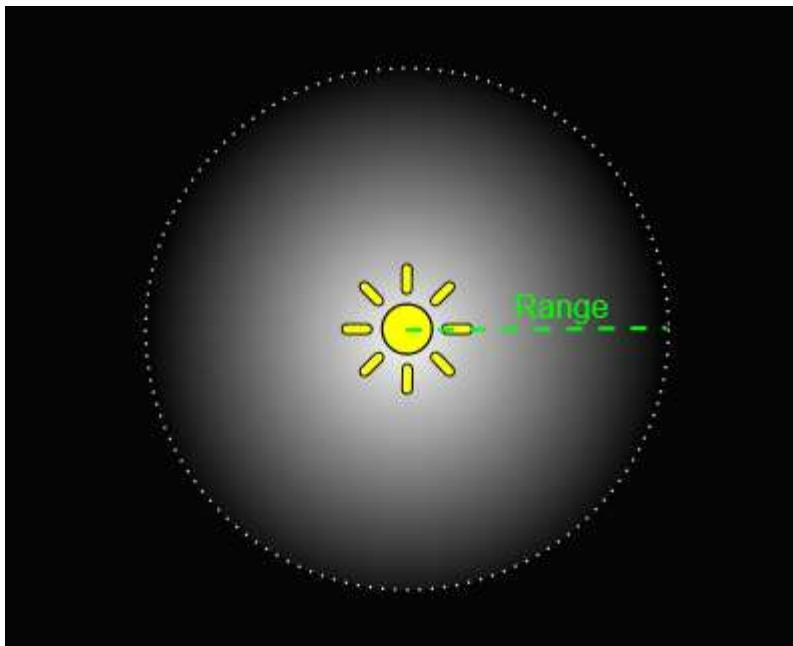
- Default lighting in each new scene
- Changing lighting direction will affect the dynamic sky box



Lighting

■ Point light

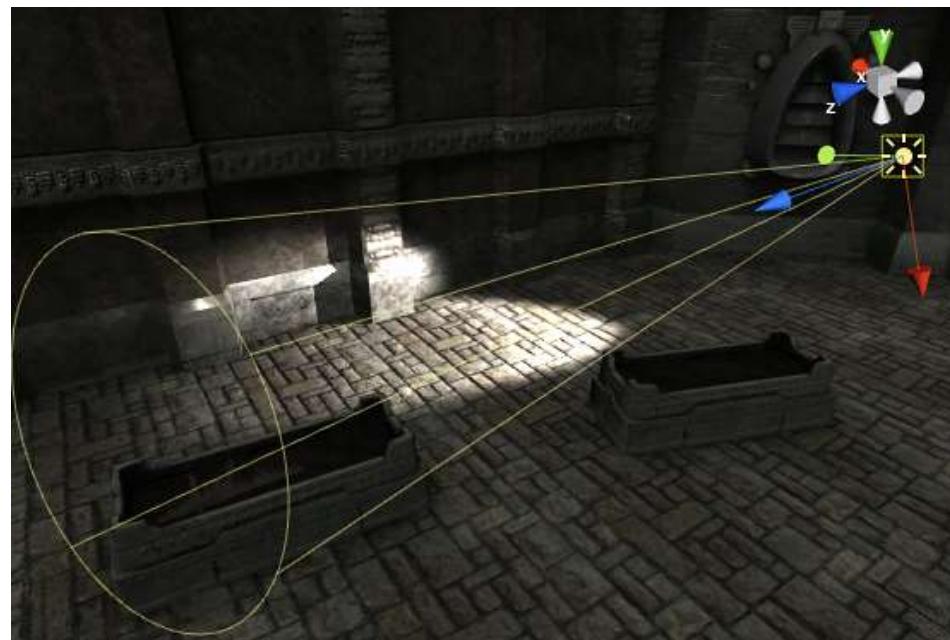
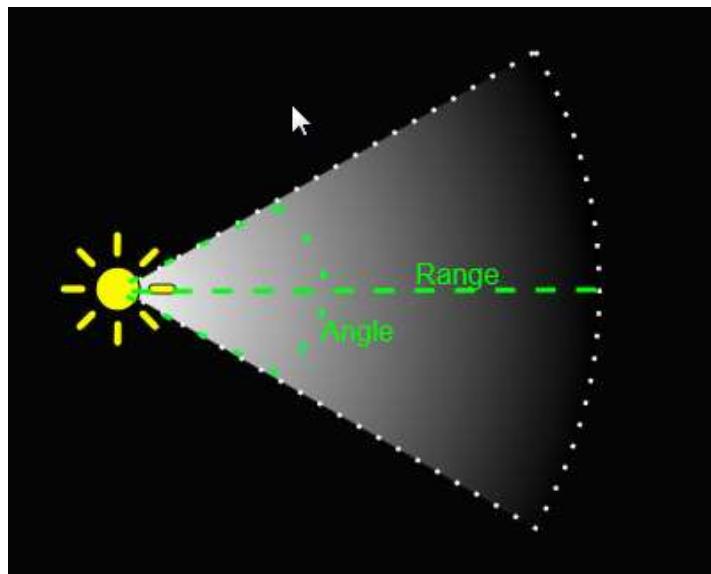
- Light is located at a point in space and sends light out in all directions equally



Lighting

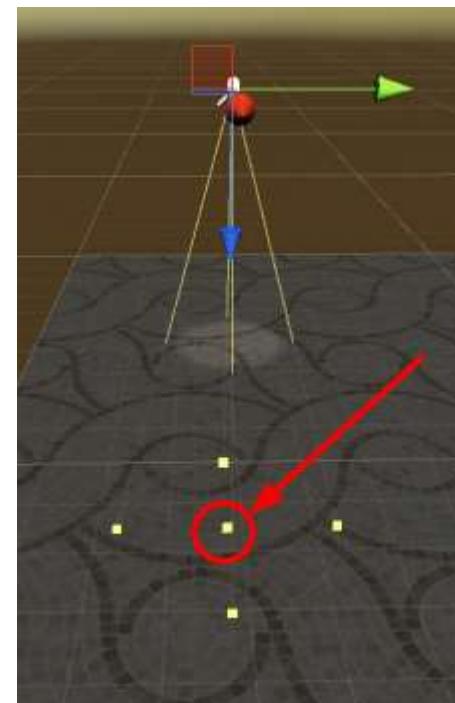
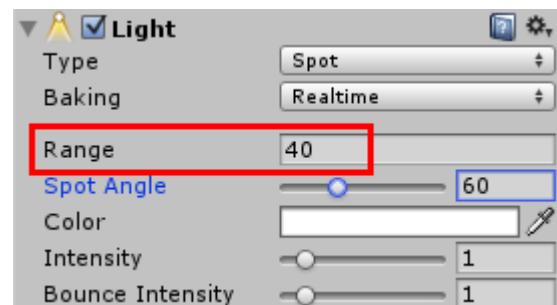
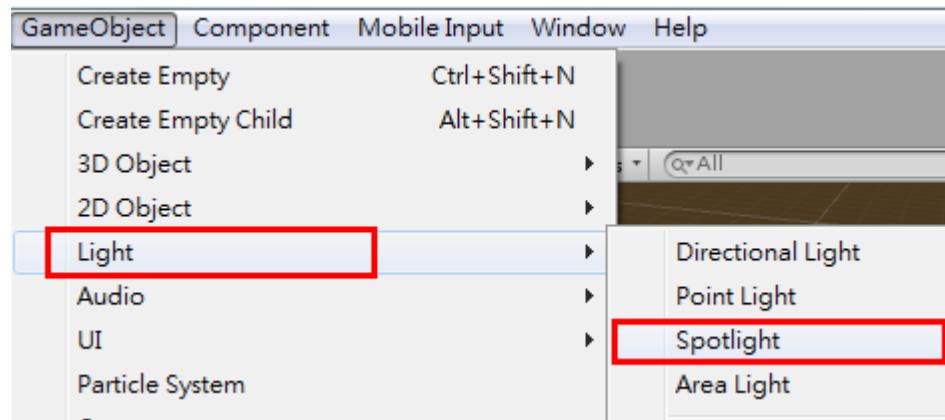
■ Spot Light

- Used for artificial light sources such as flashlights, car headlights and searchlights



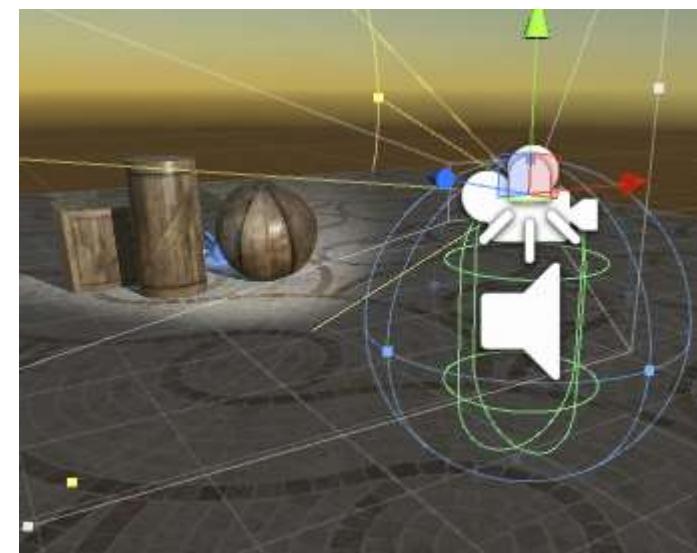
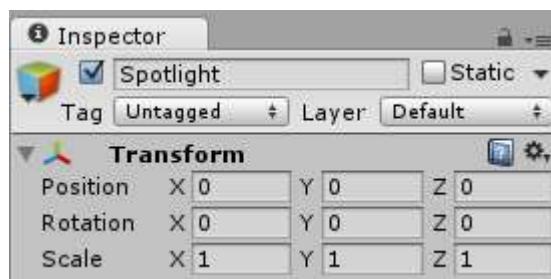
Lighting Example

- Create a spotlight and adjust the range by dragging the middle point downward



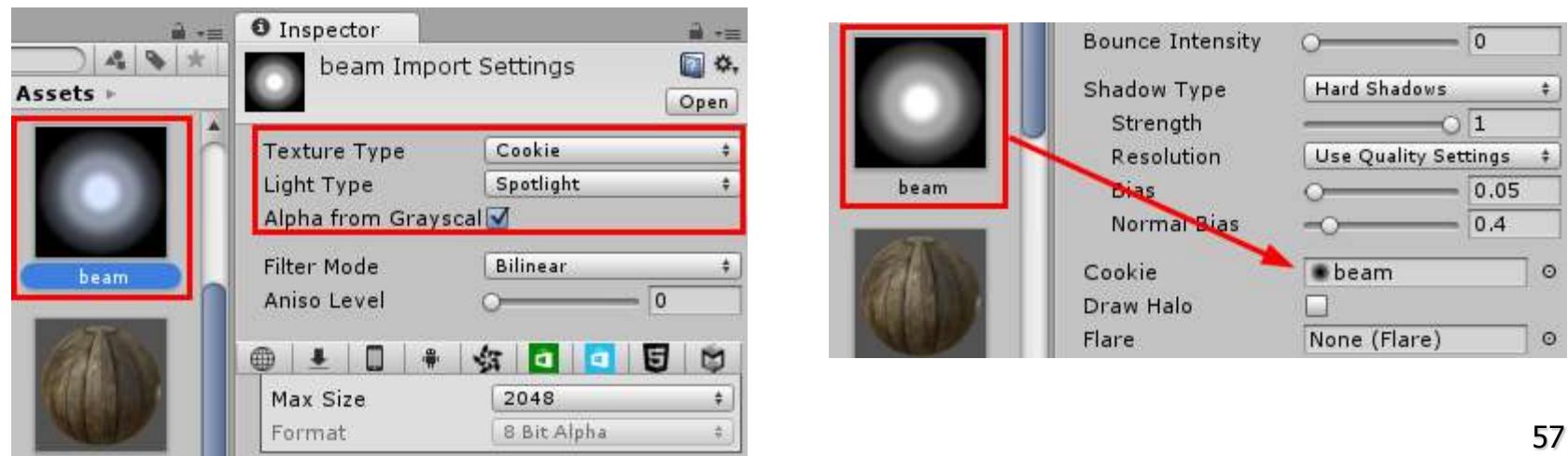
Lighting Example

- From the Hierarchy view, drag the spot light under the FirstPersonCharacter
- Reset the transform so that it will follow the FirstPersonCharacter direction and position



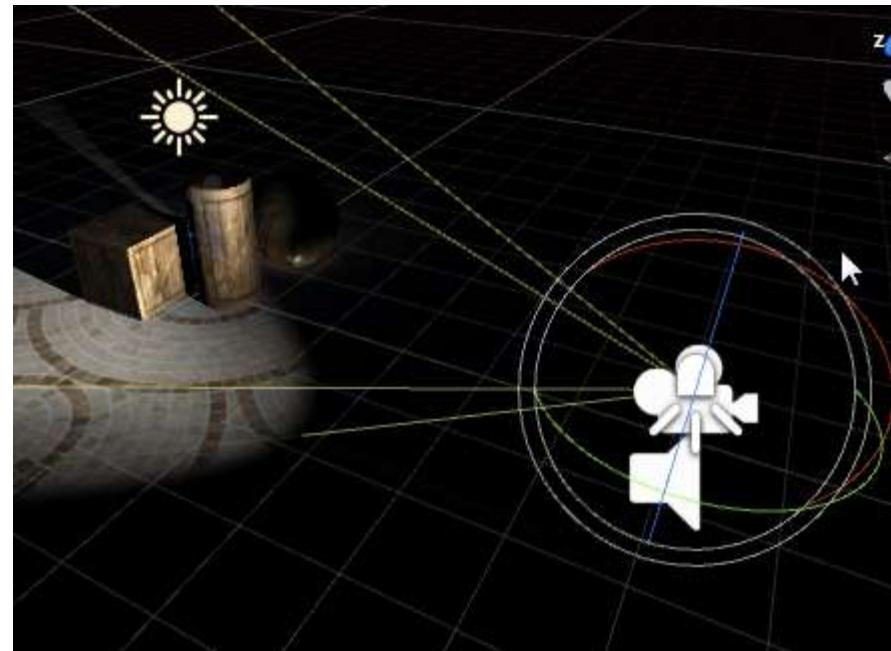
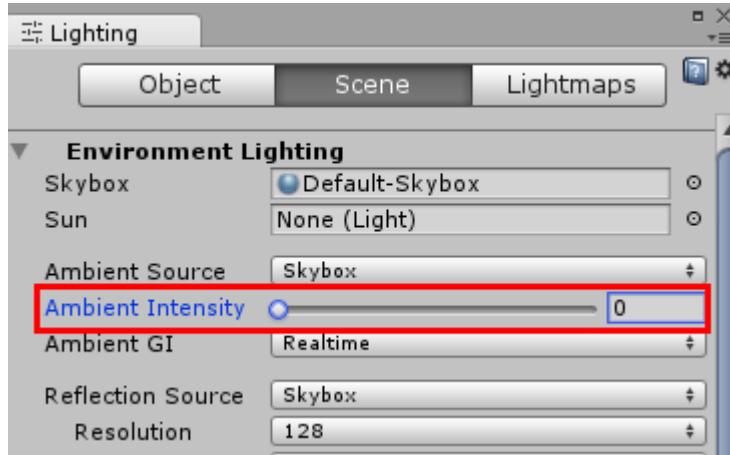
Lighting Example

- Import the beam texture and change the texture type to “Cookie”
- Enable Alpha from Grayscale
- Assign the texture to the cookie slot of the spot light



Lighting Example

- Now the beam texture acts as a mask to the light
- Turn on the lighting window, change the ambient intensity to 0 so that no more light can be seen other than the spot light



Particle Systems

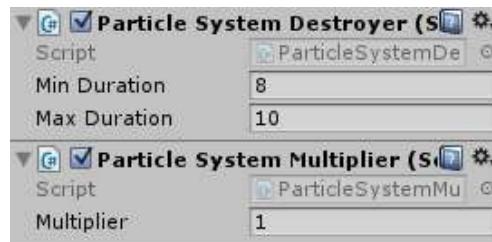
Particle System

- Particle System is used to simulate natural effect such as smoke, fire and rain
- Assets → Import Package → ParticleSystems



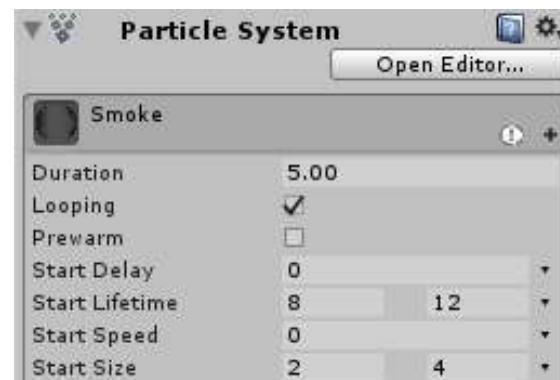
Particle System

- Particle Prefabs usually comes with 2 script
 - Particle System Destroyer
 - Use to remove the particle emitter after the max duration
 - Particle System Multiplier
 - Use to magnify the particle size



Basic Setting

- Duration
 - How long the does the particle emitter emit
- Prewarm
 - Let the particles to fully emitted at game start
- Lifetime
 - How long does each particles stay
- Speed
 - How fast the particles move
- Size
 - The size of each particles



Basic Setting

■ Max Particles

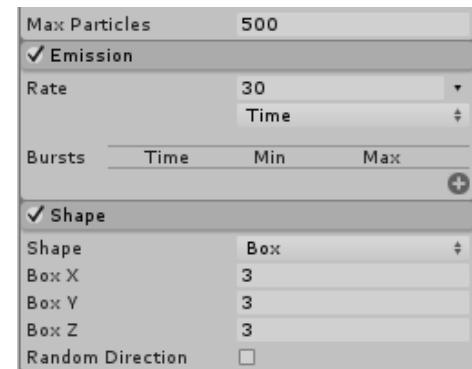
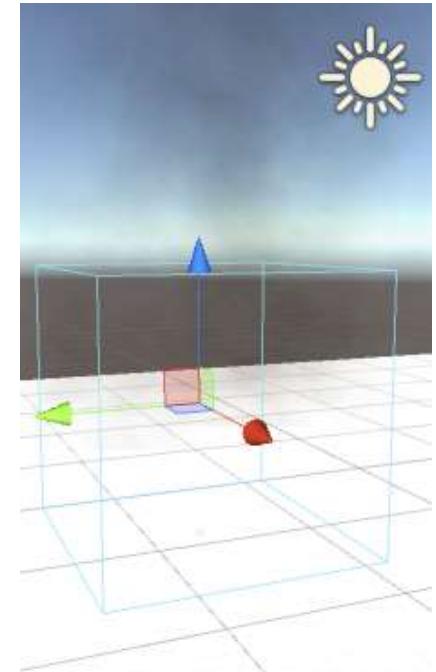
- Maximum number of particles allowed

■ Rate

- Number of particles emit per second

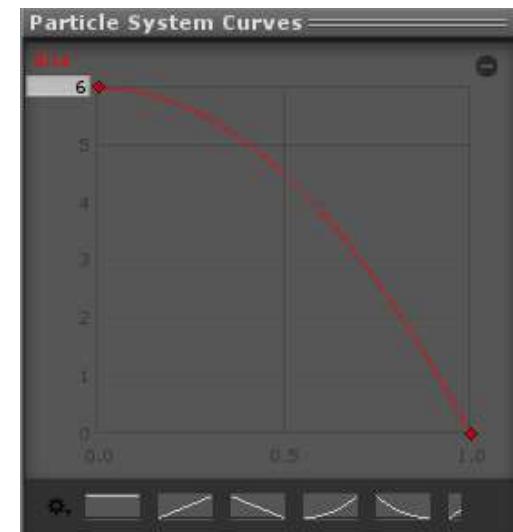
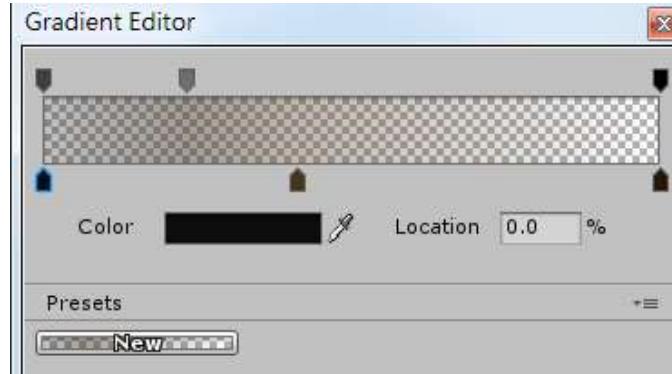
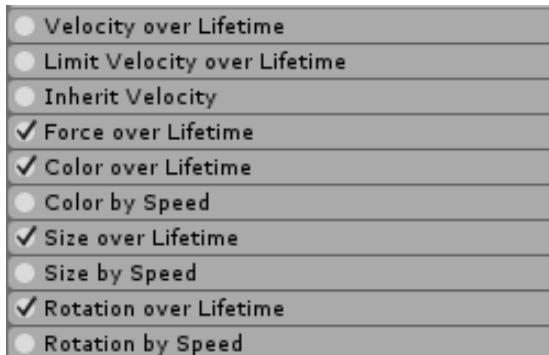
■ Shape

- Determine the shape of the emitter



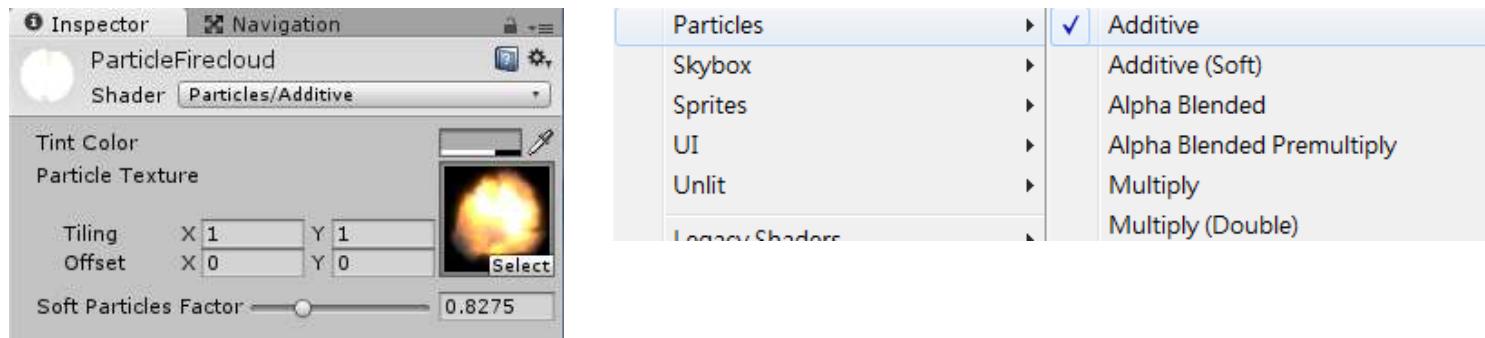
Lifetime and Speed

- Particles properties can be varied along their lifetime and speed
- E.g. Color, Size and Rotation



Material

■ Use Particle Shader for the material Setup



- 3 main types of shader
 - Additive (for lighting effect)
 - Alpha Blender (for transparent)
 - Multiply (for smoke or dirt)

Unity Reference

- <http://docs.unity3d.com/Documentation/Manual/index.html>