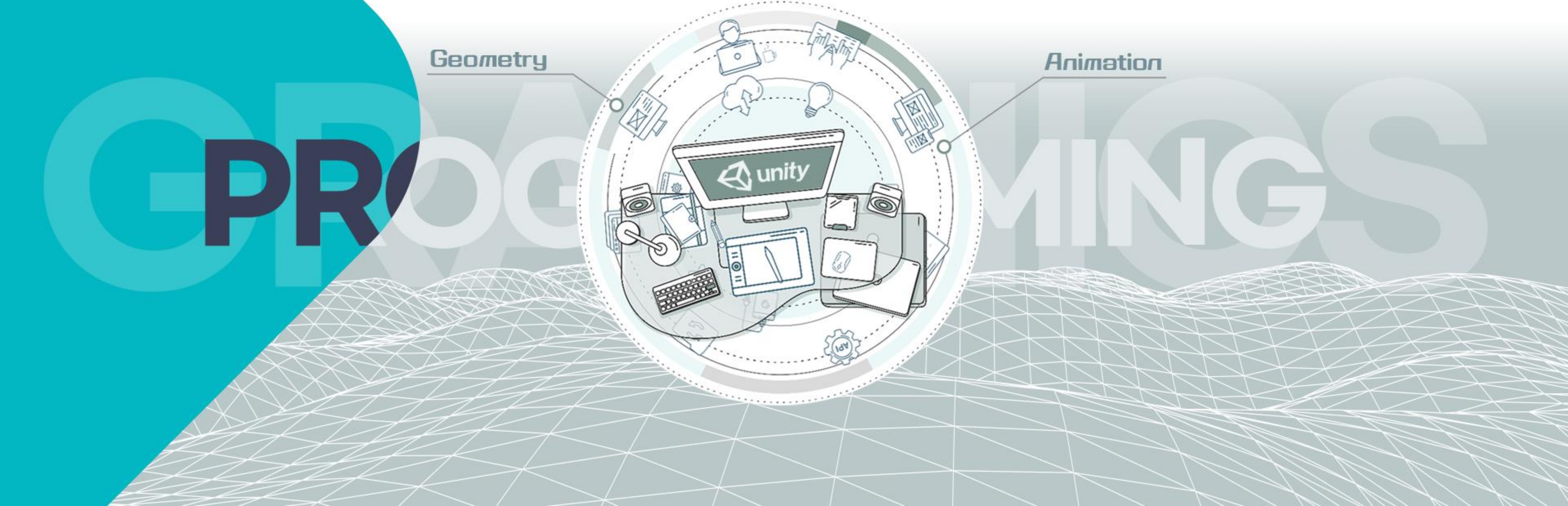
유니티(Unity)를 활용한

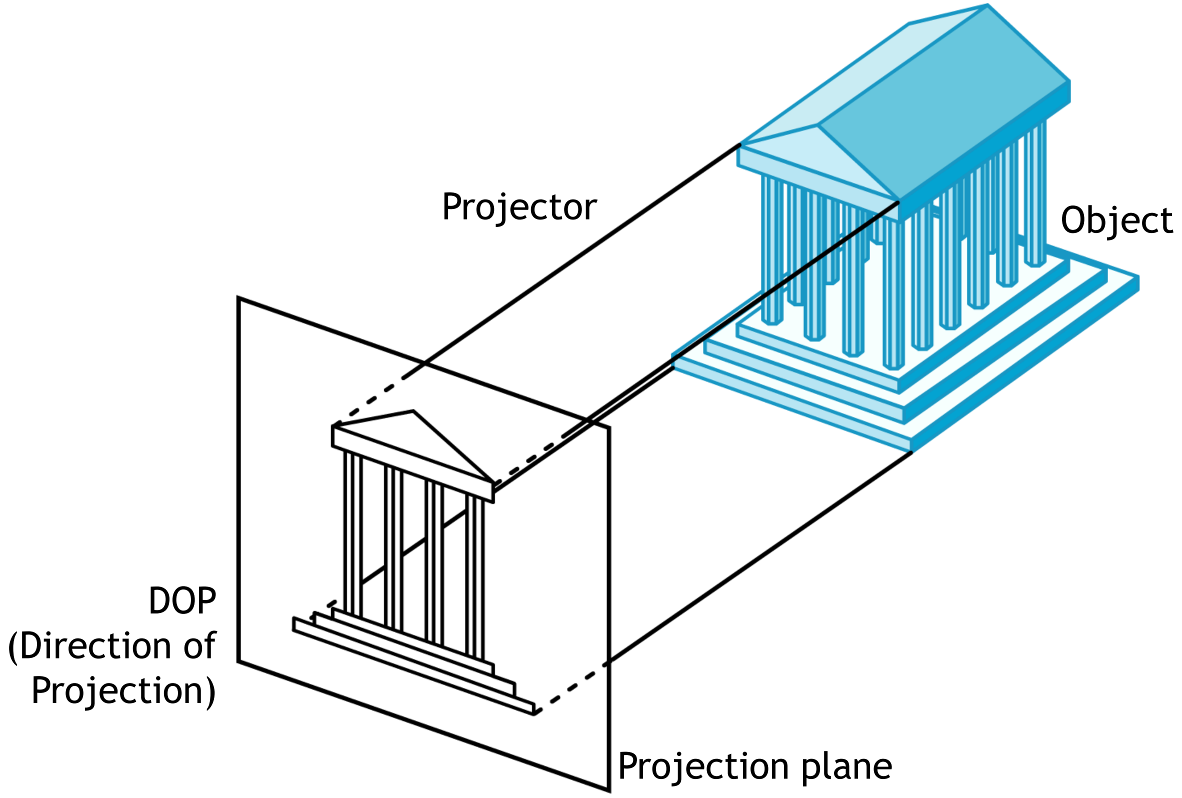
그래픽스 프로그래밍

10 Camera



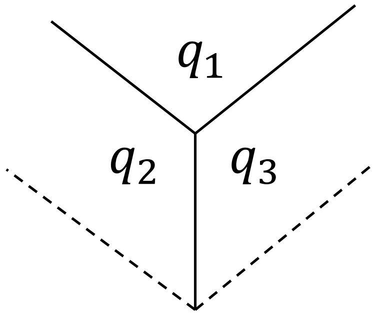
Orthographic Projection

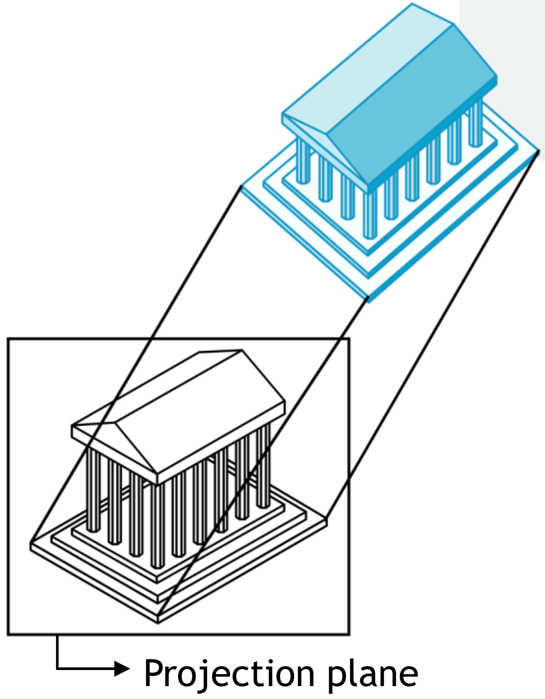
- In the orthographic projection, projectors are orthogonal to projection plane.



Axonometric Projections

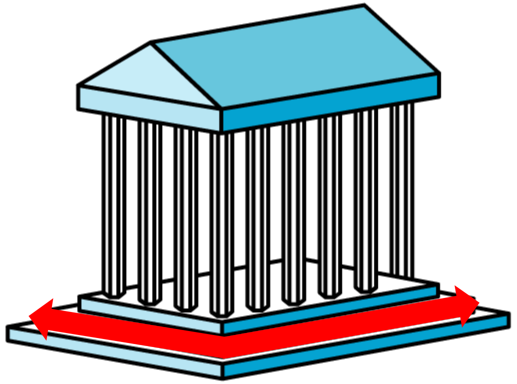
- Axonometric projections allow projection plane to move relative to object.
  - classify by how many angles of a corner of a projected cube are the same



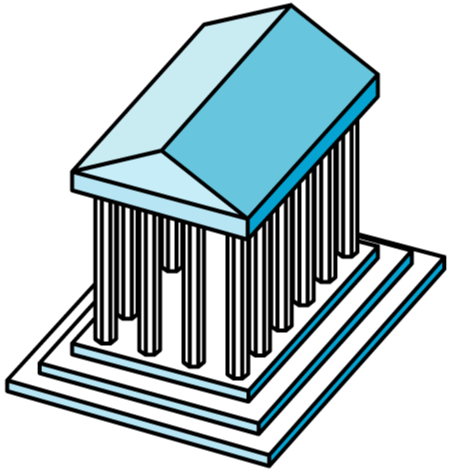


trimetric	dimetric	isometric
none	two	three

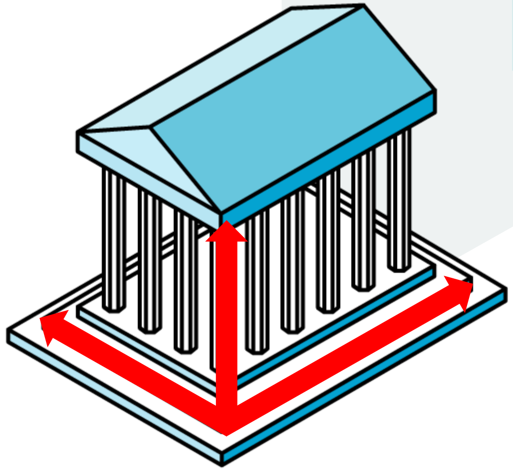
Types of Axonometric Projections



Dimetric



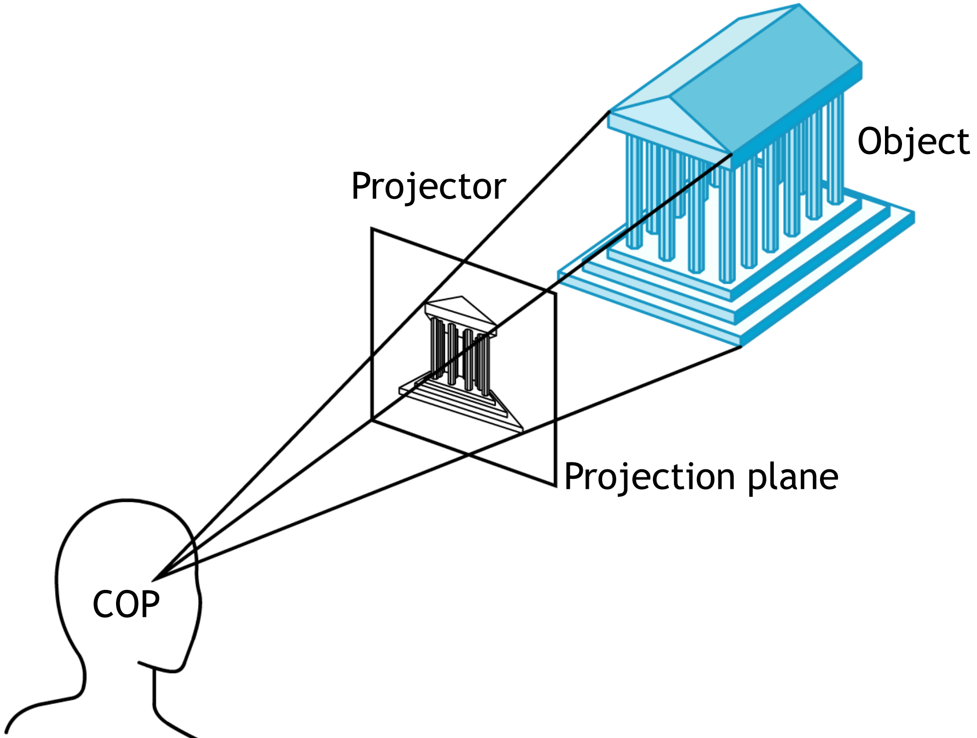
Trimetric



Isometric

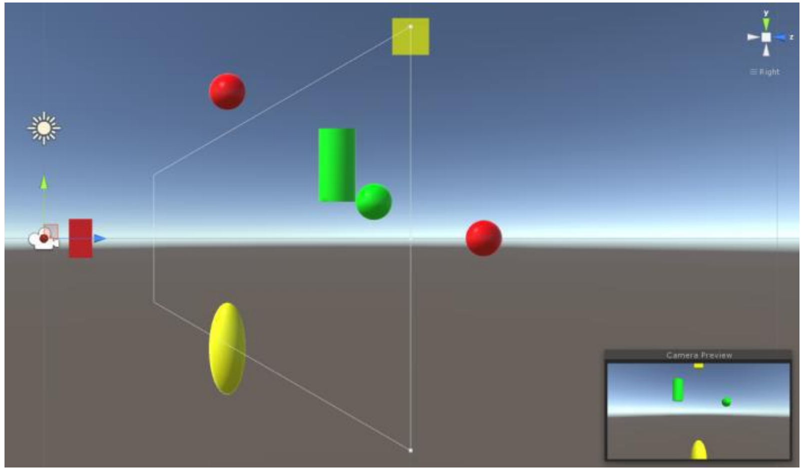
Perspective Projection

- Parallel lines (not parallel to the projection plan) on the object converge at a single point in the projection (the vanishing point)
- Drawing simple perspectives by hand uses these vanishing point(s)

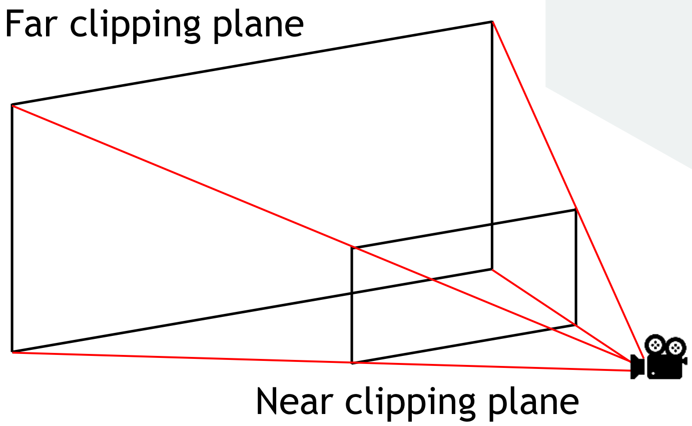


View Frustum

- View Frustum is the shape of the region that can be seen and rendered by a camera.

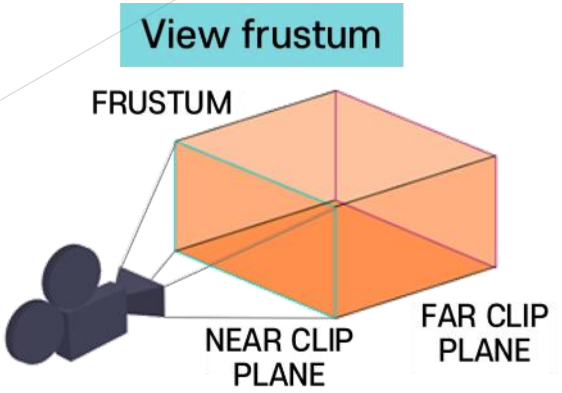


이미지 출처 :https://docs.unity3d.com/Manual/UnderstandingFrustum.html



View Frustum

- Orthographic projection projects the rectilinear box viewing volume onto the screen.
- The size of the object does not change with distance.
- Points are projected onto the z=0 plane towards the z- axis.

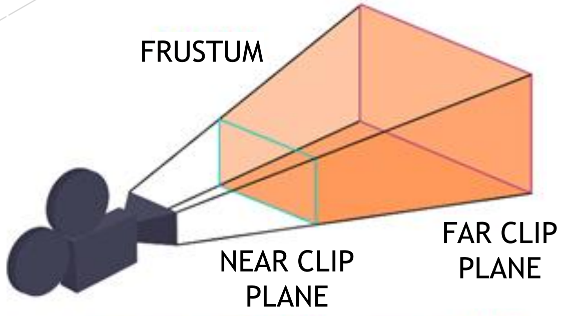


이미지 출처 :http://latedreamer.blogspot.com/2017/08/unity-tutorial-tanks-part-2.html

View Frustum

- Perspective projection projects the frustum (i.e., truncated pyramid) viewing space onto the screen.
- Near objects appear larger and object far away appear smaller.

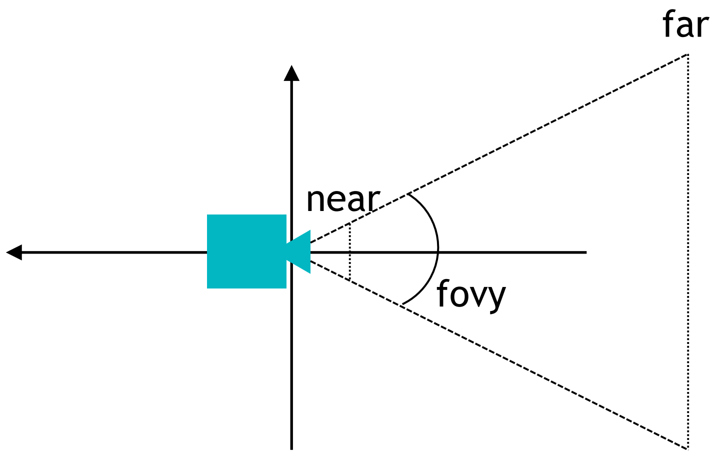
View frustum

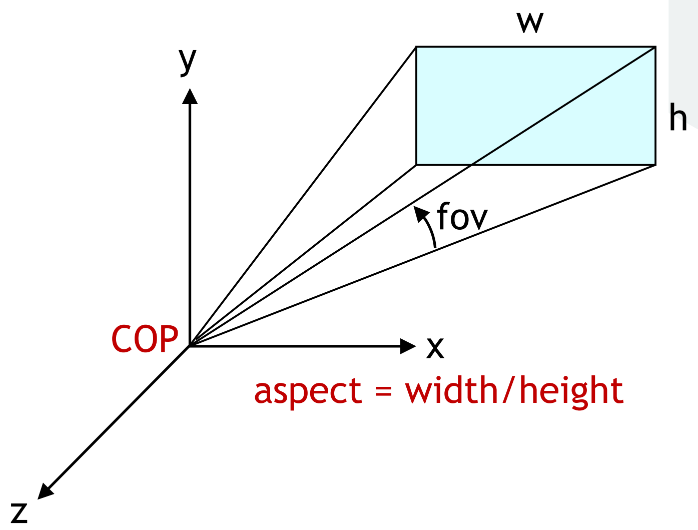


이미지 출처 :http://latedreamer.blogspot.com/2017/08/unity-tutorial-tanks-part-2.html

View Frustum

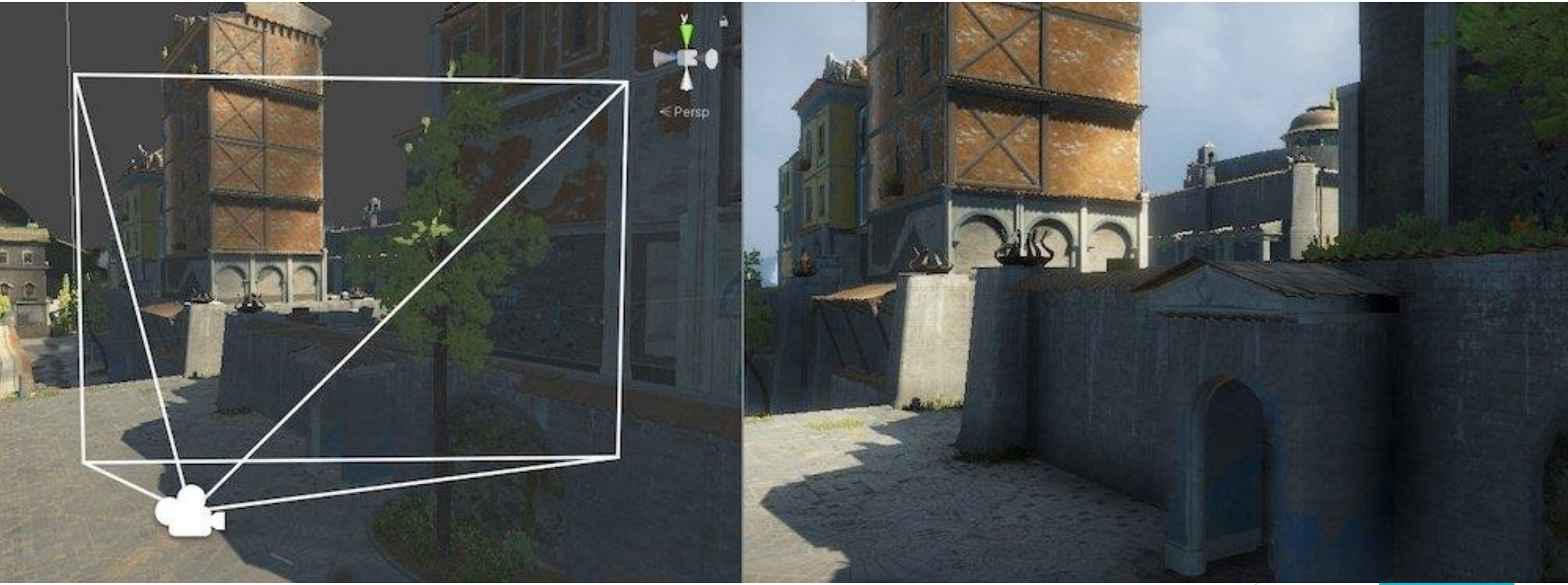
- Perspective projection uses the y-direction viewing angle (FOV) and the aspect ratio (the value of the width of the nearest clipping plane divided by the height).





Field of View

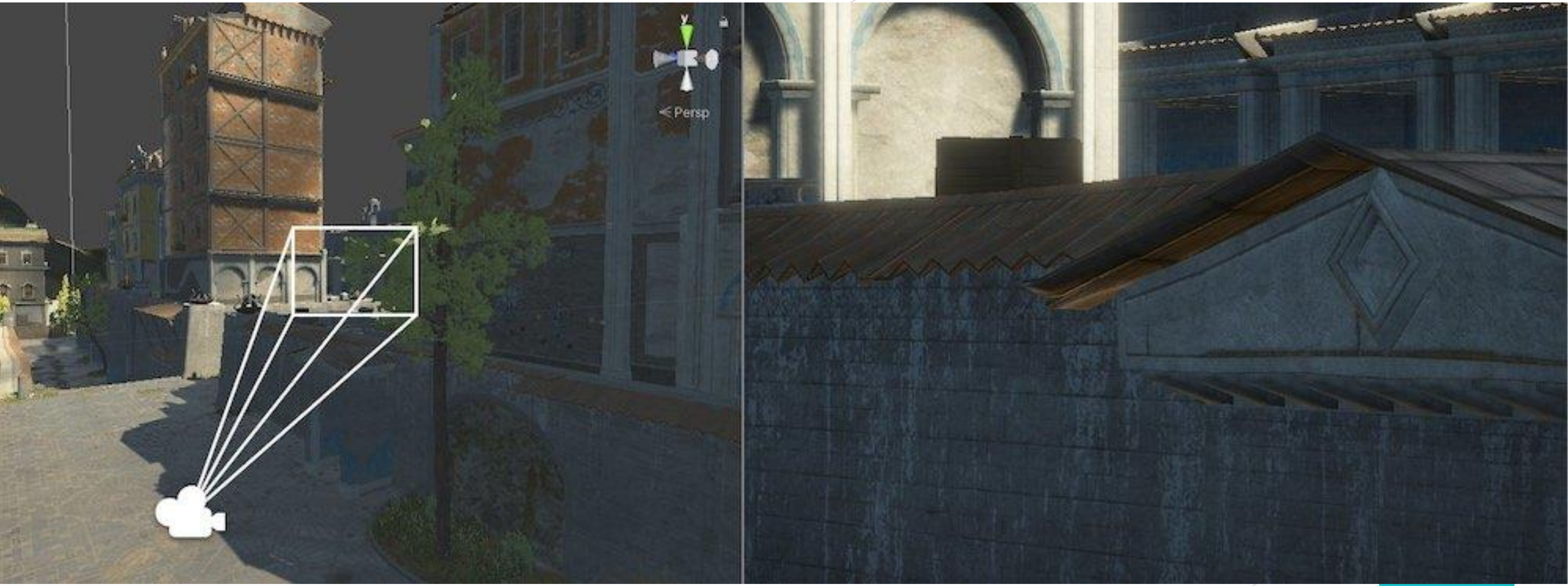
- A wide field of view shows more of the scene.



이미지 출처 :https://gamedevbeginner.com/how-to-zoom-a-camera-in-unity-3-methods-with-examples/

Field of View

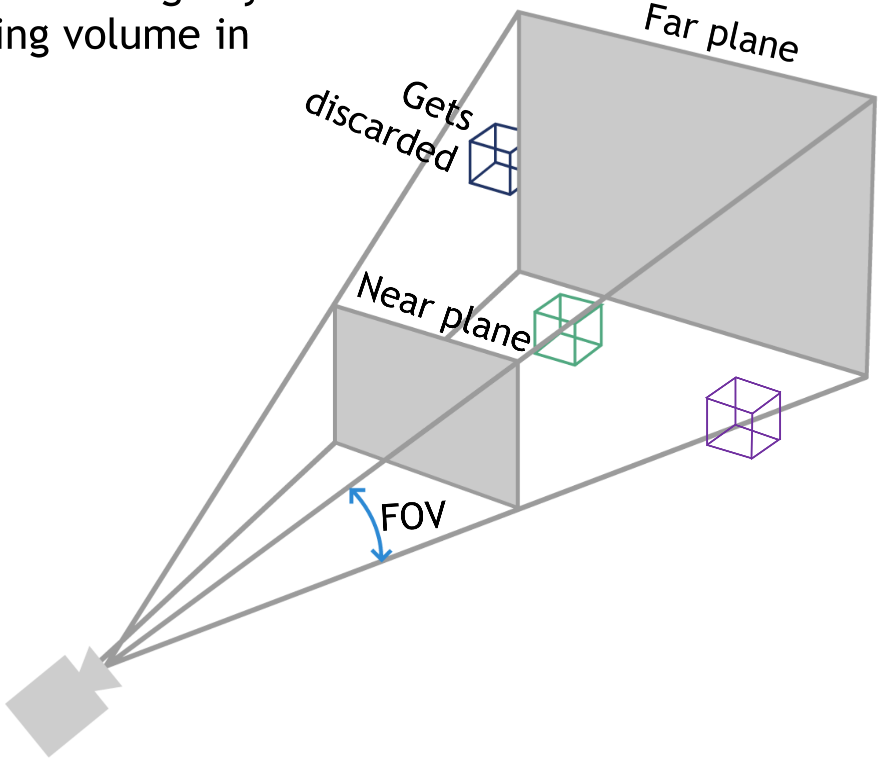
- A narrow field of view shows less of the camera image, zooming it in scene.



이미지 출처 :https://gamedevbeginner.com/how-to-zoom-a-camera-in-unity-3-methods-with-examples/

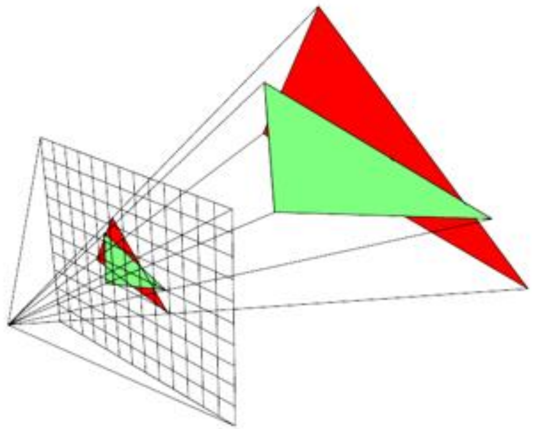
Frustum Culling

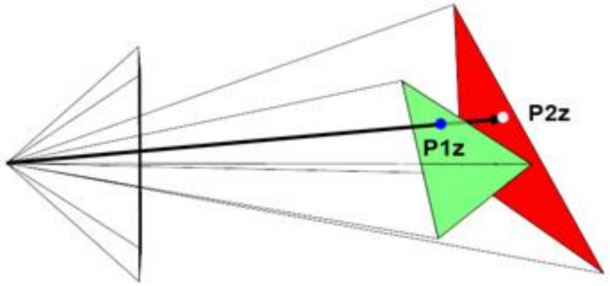
- Objects projected outside the window are clipped without appearing as an image by placing a pyramid like clipping volume in front of the camera.



Projection

- Projection determines which point on the 2D screen is a point in the 3D space that constitutes an object when the observer composes the composition.

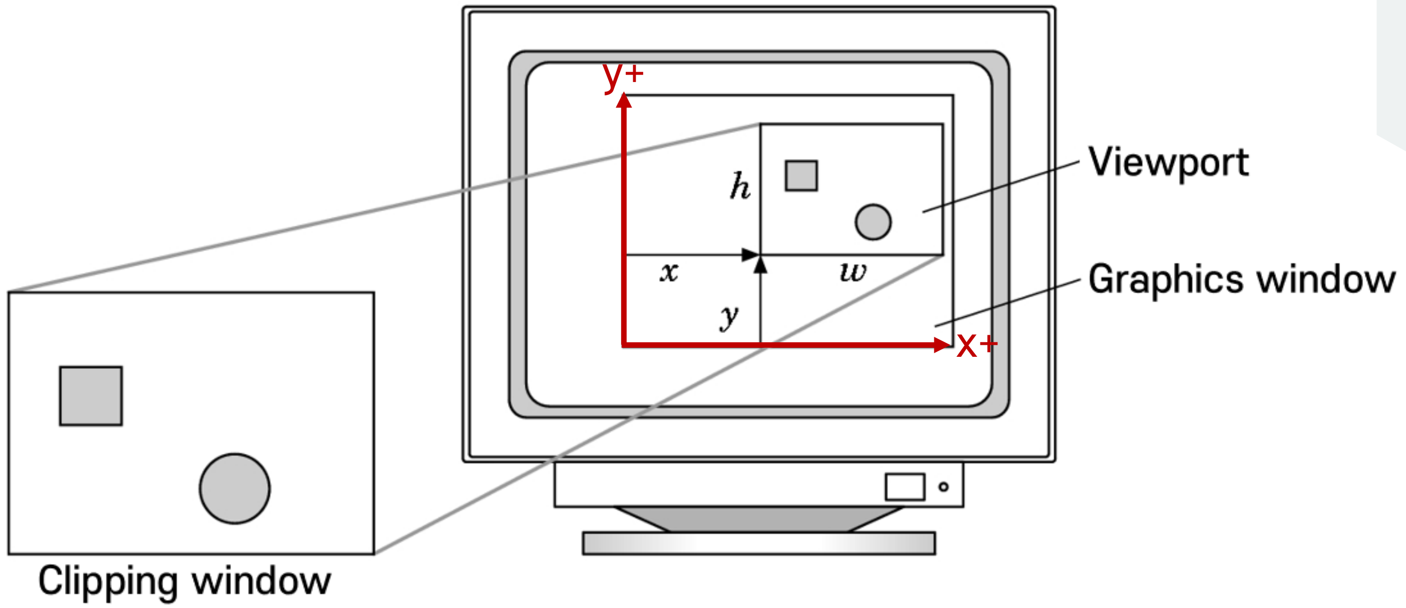




이미지 출처 :www.scratchapixel.com

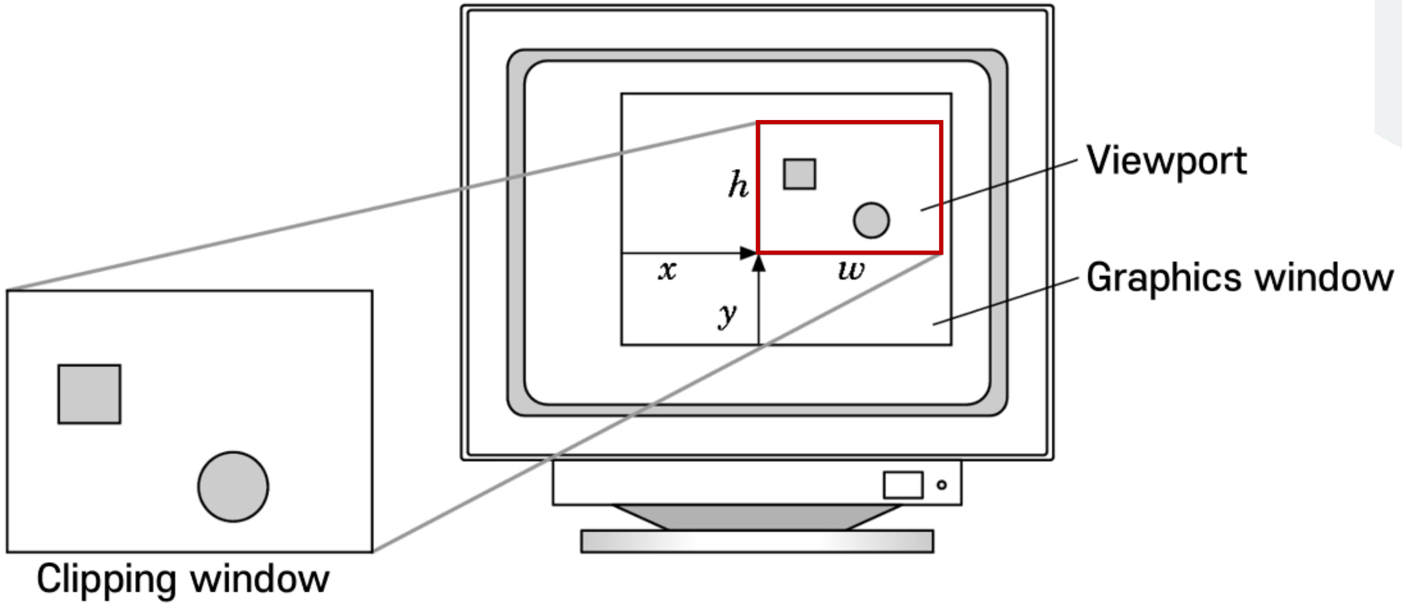
Viewport

- The space set inside the window. Drawing is restricted to inside the viewport.



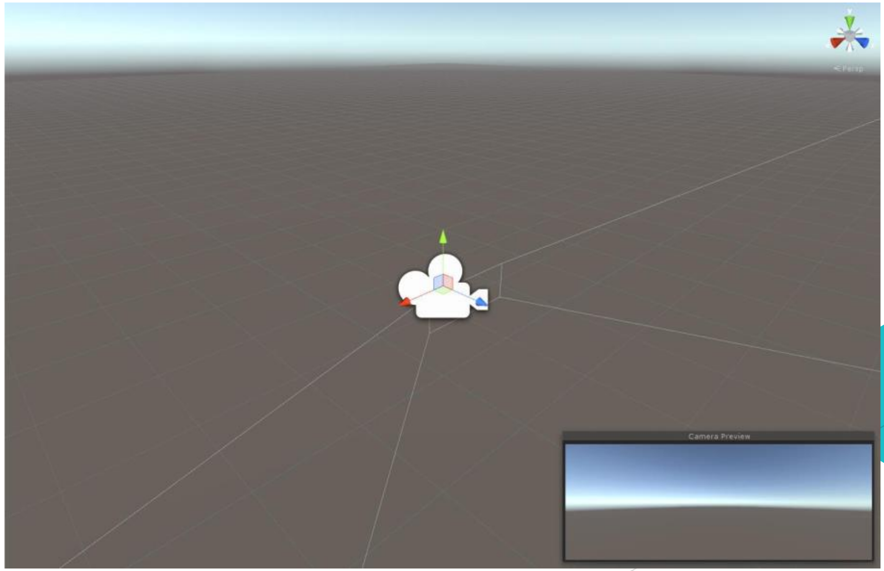
Viewport

- The space set inside the window. Drawing is restricted to inside the viewport.



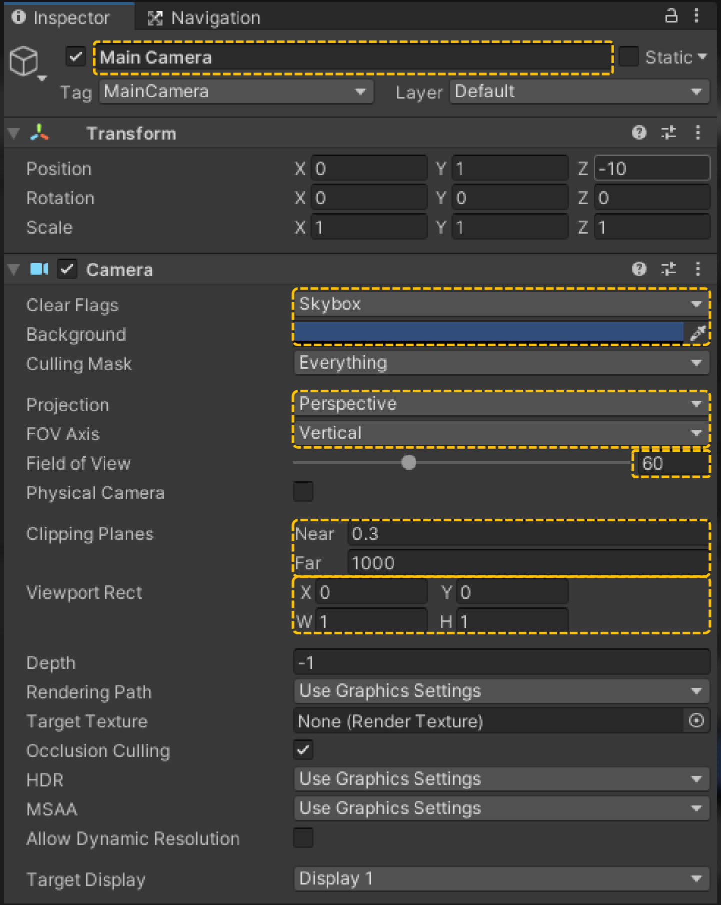
Camera in Unity

- Camera
  - A Unity scene represents GameObjects in a three-dimensional space. Since the viewer's screen is two-dimensional, Unity needs to capture a view and "flatten" it for display. It does this using cameras.
  - In Unity, you create a camera by adding a Camera component to a GameObject.



이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

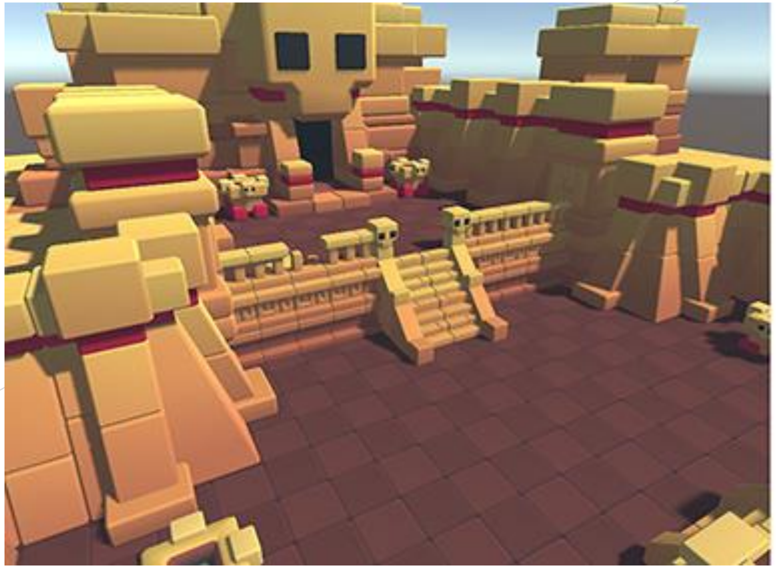
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3 / Far 1000
Viewport Rect	X 0 Y 0 / W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

이미지 출처 : Unity

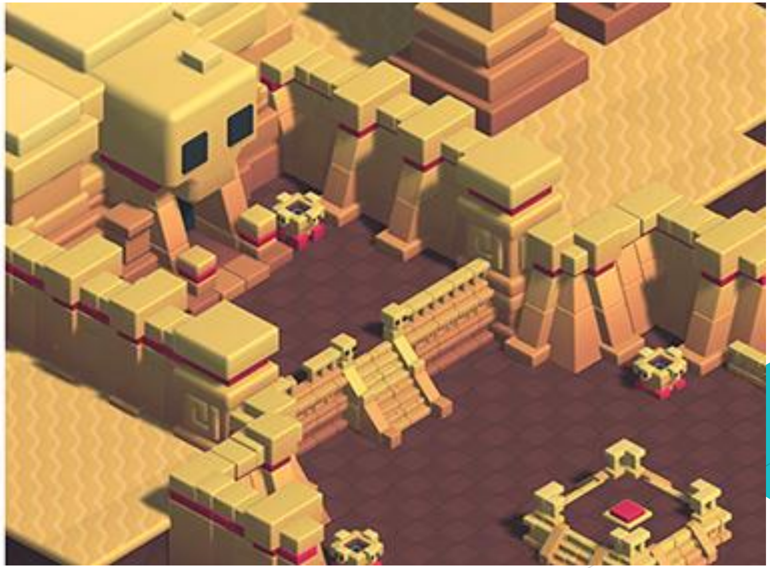
Viewer's Perspective (Camera Input)

- Before rendering the environment on the screen we consider the camera input such as (field of view, Projection mode [Orthographic or Perspective]).

Perspective camera

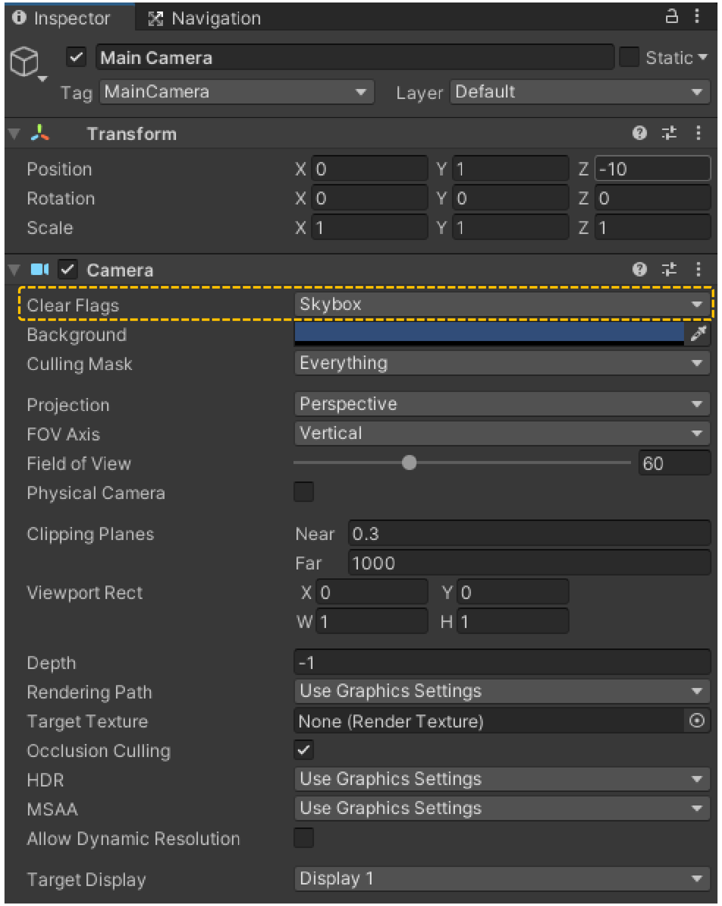


Orthographic camera



이미지 출처 :https://docs.unity3d.com/Manual/CamerasOverview.html

Camera Components


Inspector | Navigation

Main Camera — Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

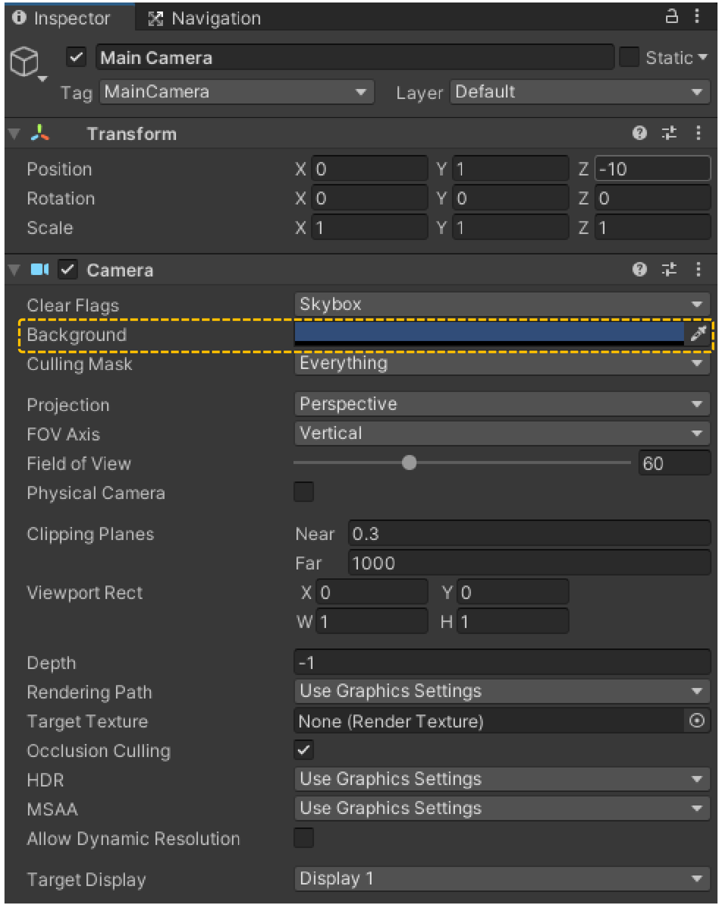
Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Clear Flags

Determines which parts of the screen will be cleared.
This is handy when using multiple Cameras to draw different game elements. Skybox is the default setting.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

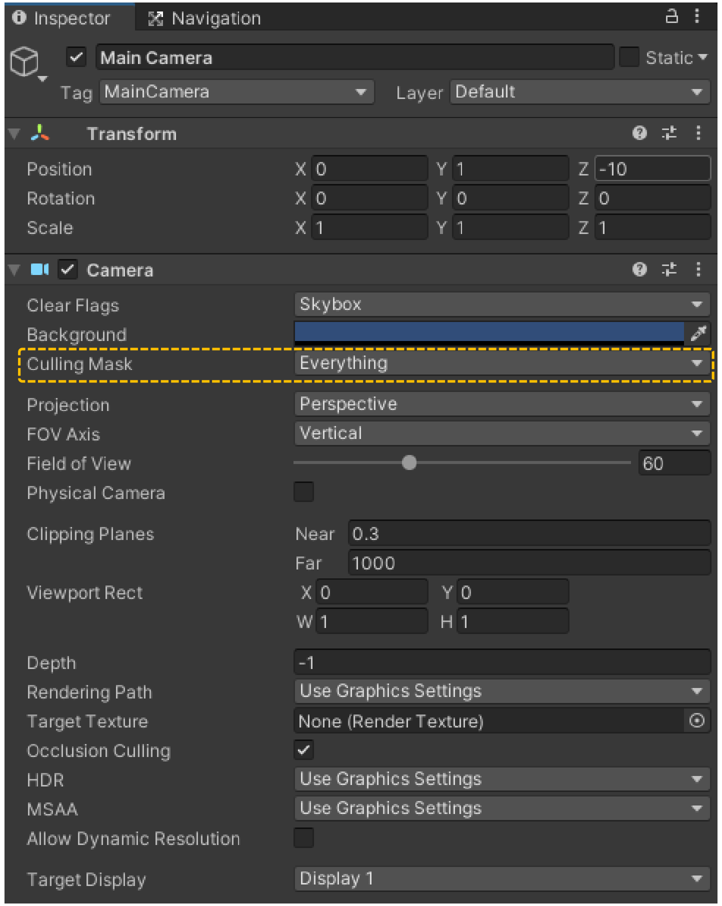
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Background

The color applied to the remaining screen after all elements in view have been drawn and there is no skybox.

이미지 출처 : Unity

Camera Components


Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

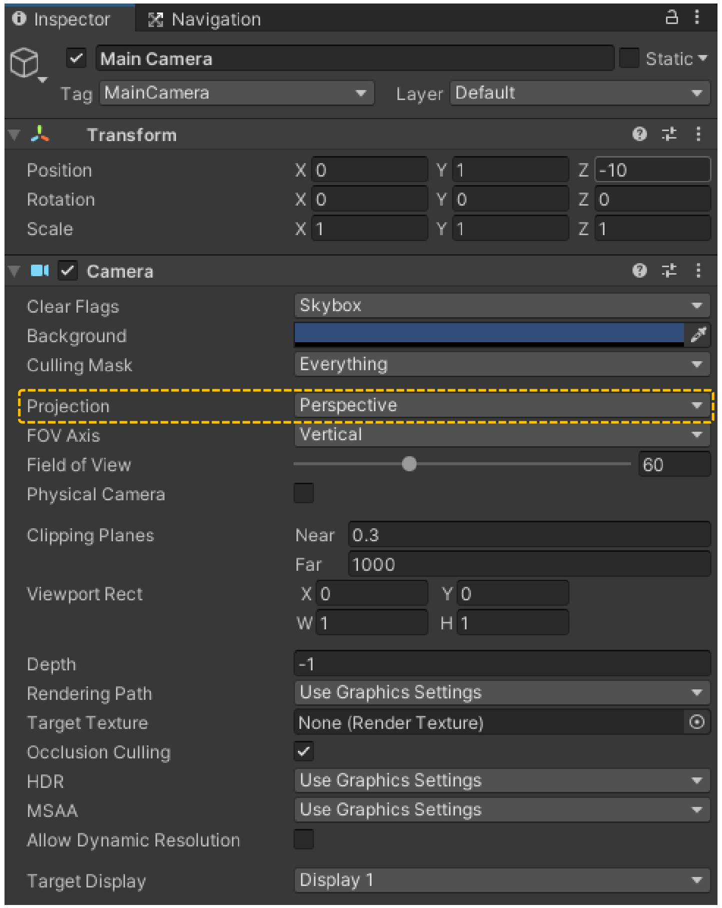
Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

Culling Mask

Includes or omits layers of objects to be rendered by the Camera. Assigns layers to your objects in the Inspector.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera — Static

Tag: MainCamera — Layer: Default

Transform

	X	Y	Z
Position	0	1	-10
Rotation	0	0	0
Scale	1	1	1

Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0, Y 0
	W 1, H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Projection

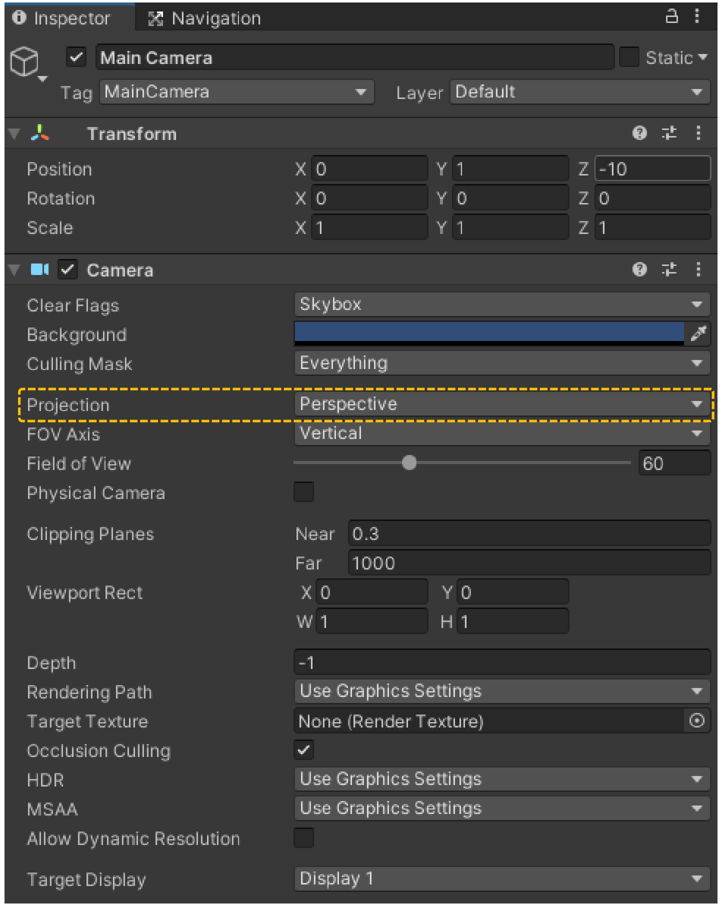
Toggles the camera's capability to simulate perspective.

Perspective

Orthographic

이미지 출처 : Unity

Camera Components


Inspector | Navigation

✓ Main Camera	Static
Tag MainCamera	Layer Default

Transform

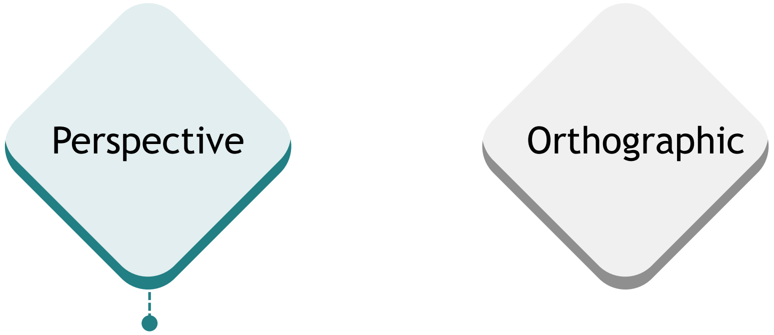
	X	Y	Z
Position	0	1	-10
Rotation	0	0	0
Scale	1	1	1

✓ Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Projection

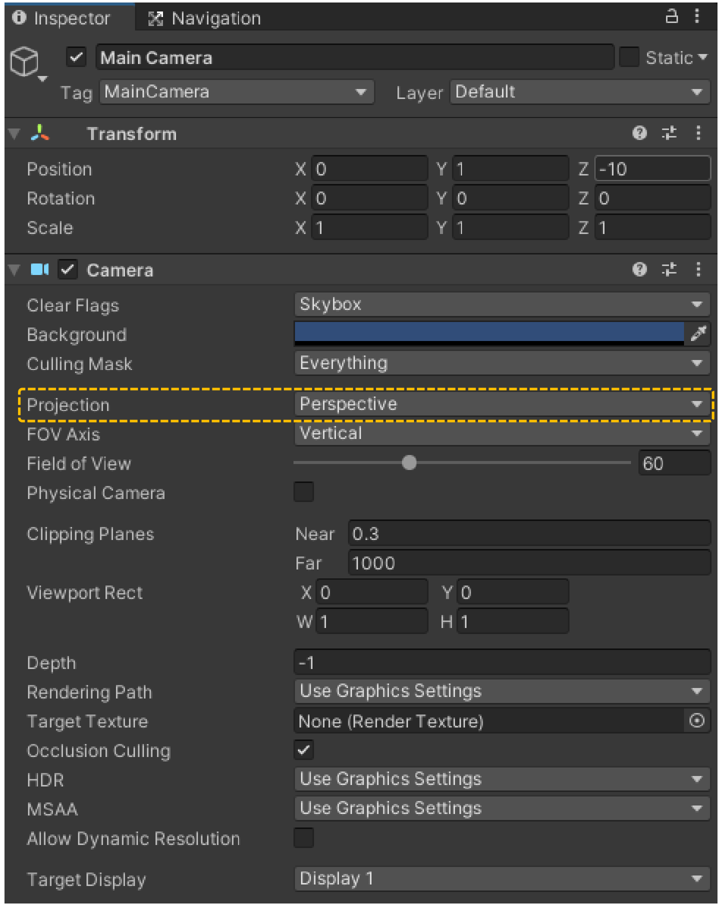
Toggles the camera's capability to simulate perspective.



Camera will render objects with perspective intact.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Projection

Toggles the camera’s capability to simulate perspective.

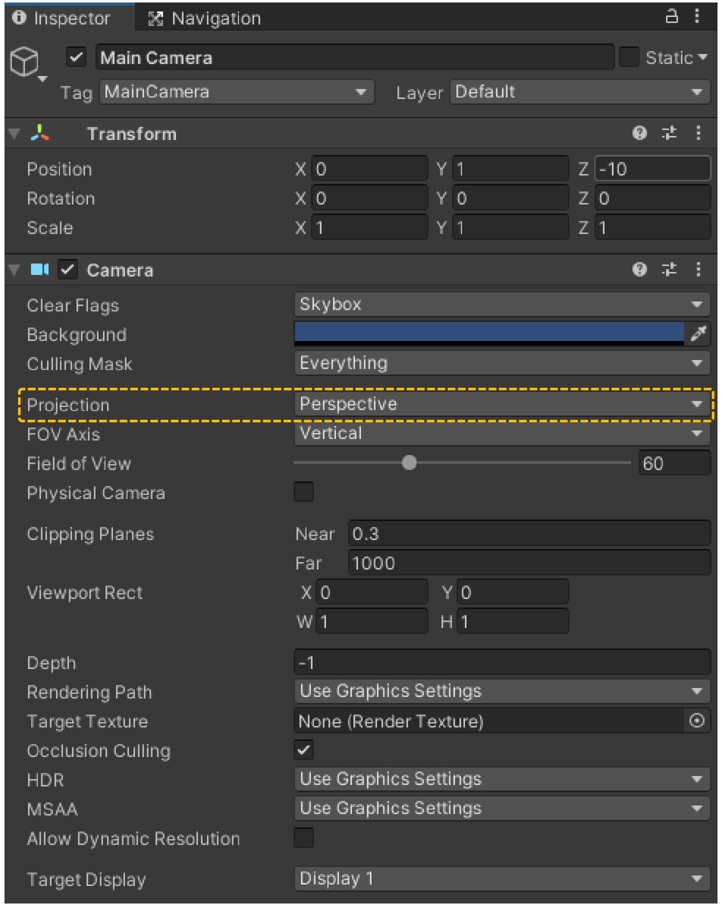
Perspective

Orthographic

Camera will render objects uniformly, with no sense of perspective. NOTE : Deferred rendering is not supported in Orthographic mode. Forward rendering is always used.

이미지 출처 :Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera ✓

Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Projection

Toggles the camera’s capability to simulate perspective.

Perspective

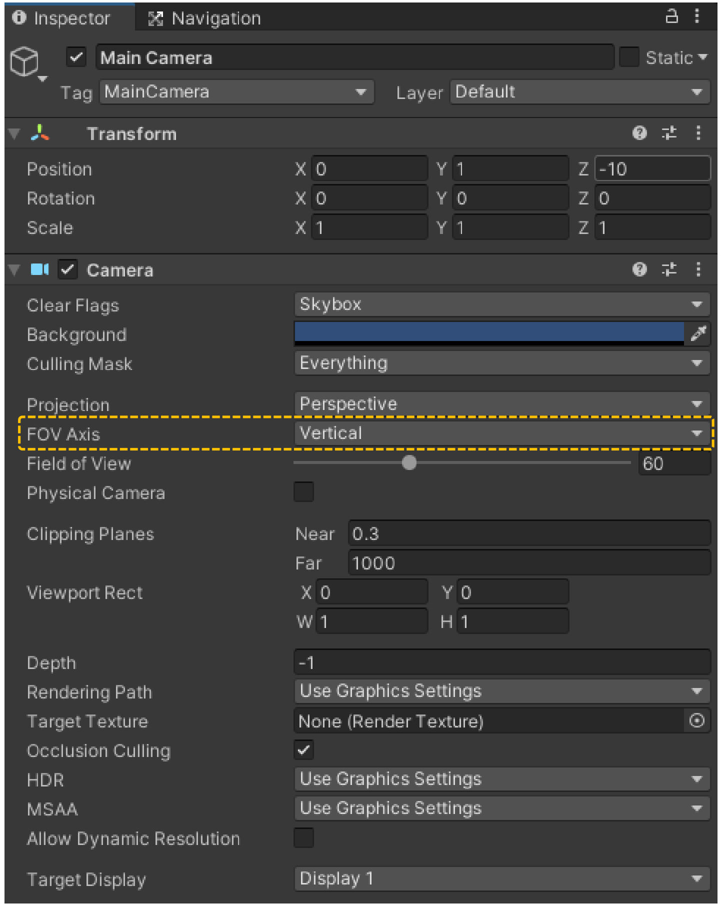
Orthographic

Size
(when Orthographic is selected)

The viewport size of t
he Camera when set to
Orthographic.

이미지 출처 : Unity

Camera Components


Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

FOV Axis (when Perspective is selected)

Field of view axis.

Horizontal

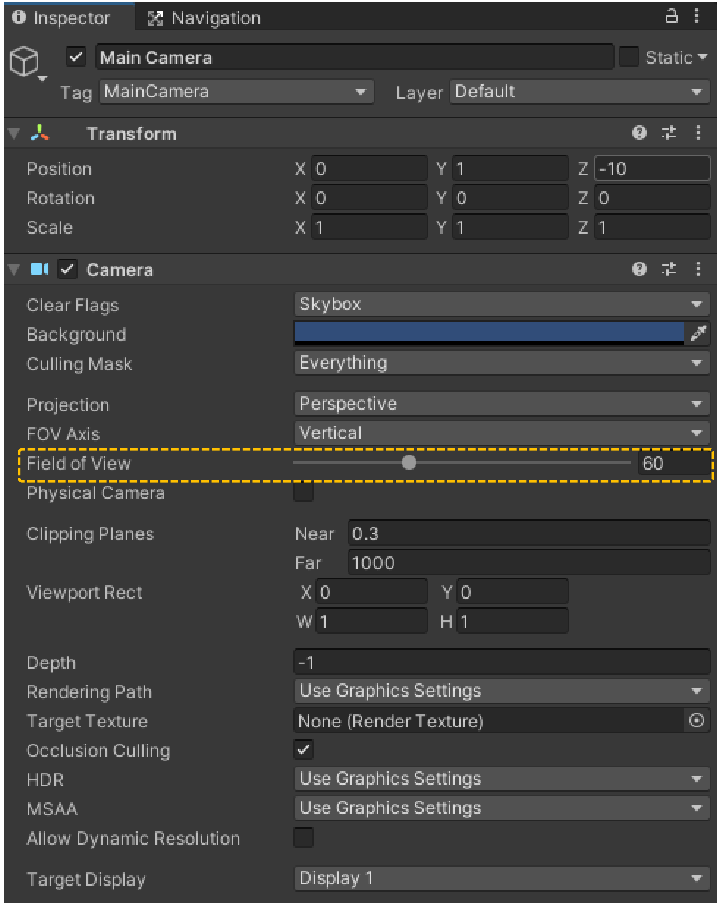
The Camera uses a horizontal field of view axis.

Vertical

The Camera uses a vertical field of view axis.

이미지 출처 : Unity

Camera Components



Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

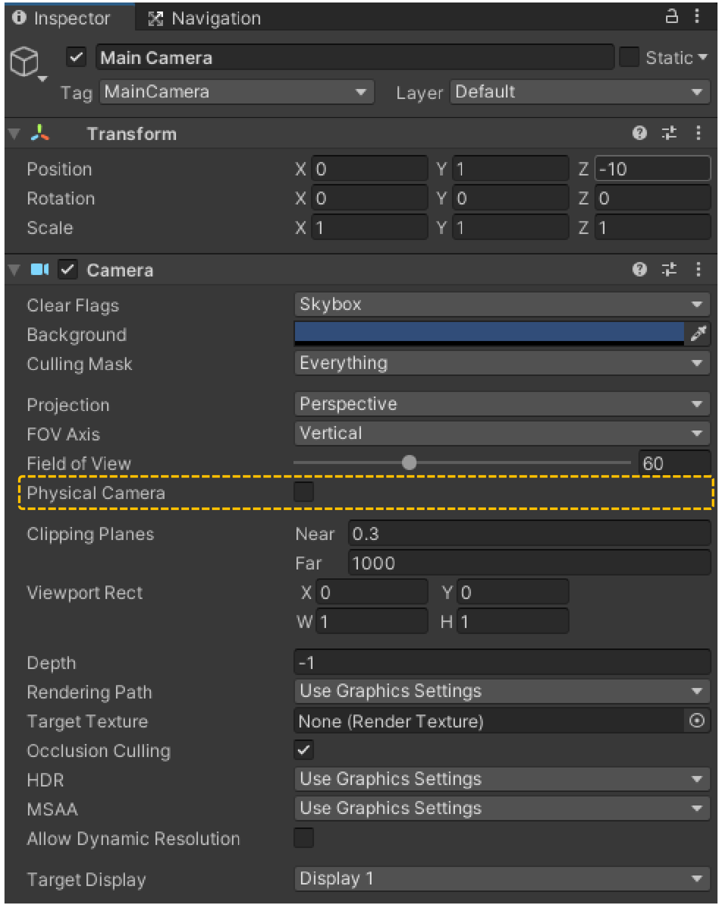
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Field of view (when Perspective is selected)

The Camera’s view angle, measured in degrees along the axis specified in the FOV Axis drop-down.

이미지 출처 : Unity

Camera Components



Main Camera
Tag MainCamera
Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

이미지 출처 : Unity

Camera Components

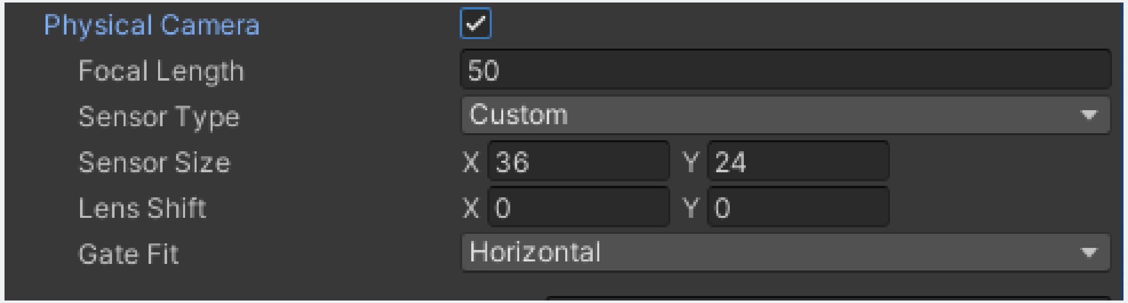
Property	Function
Physical Camera	Tick this box to enable the Physical Camera properties for this camera. When the Physical Camera properties are enabled, Unity calculates the Field of View using the properties that simulate real-world camera attributes : Focal Length, Sensor Size, and Lens Shift. Physical Camera properties are not visible in the Inspector until you tick this box.
Focal Length	Set the distance, in millimeters, between the camera sensor and the camera lens. Lower values result in a wider Field of View, and vice versa. When you change this value, Unity automatically updates the Field of View property accordingly.
Sensor Type	Specify the real-world camera format you want the camera to simulate. Choose the desired format from the list. When you choose a camera format, Unity sets the Sensor Size > X and Y properties to the correct values automatically. If you change the Sensor Size values manually, Unity automatically sets this property to Custom.

Camera Components

Property	Function
Sensor Size	Set the size, in millimeters, of the camera sensor. Unity sets the X and Y values automatically when you choose the Sensor Type. You can enter custom values if needed.
Lens Shift	Shift the lens horizontally or vertically from center. Values are multiples of the sensor size; for example, a shift of 0.5 along the X axis offsets the sensor by half its horizontal size. You can use lens shifts to correct distortion that occurs when the camera is at an angle to the subject (for example, converging parallel lines). Shift the lens along either axis to make the camera frustum oblique.
Gate Fit	Options for changing the size of the resolution gate (size/aspect ratio of the game view) relative to the film gate (size/aspect ratio of the Physical Camera sensor). For further information about resolution gate and film gate.

Using Physical Cameras

- The camera component's Physical Camera properties simulate real-world camera formats on a Unity camera.
- Unity provides the same settings as those in most 3D modeling application's physical camera settings.
  The two main properties that control what the camera sees are Focal Length and Sensor Size.

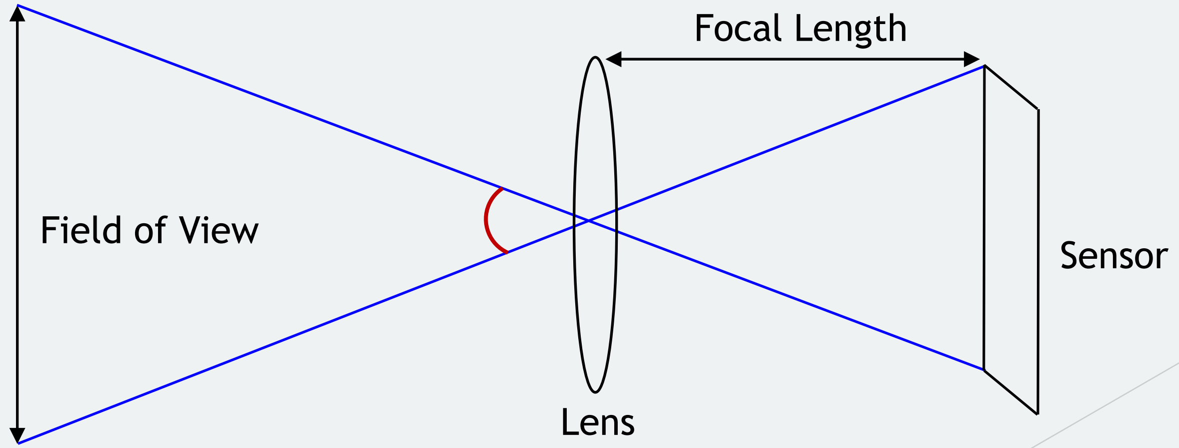


Physical Camera	✓
Focal Length	50
Sensor Type	Custom
Sensor Size	X 36 Y 24
Lens Shift	X 0 Y 0
Gate Fit	Horizontal

이미지 출처 :Unity

Using Physical Cameras

- Focal Length
  - The distance between the sensor and the camera lens. This determines the vertical field of view. When a Unity camera is in Physical Camera mode, changing the Focal Length also changes the field of view accordingly. Smaller focal lengths result in a larger field of view, and vice versa.



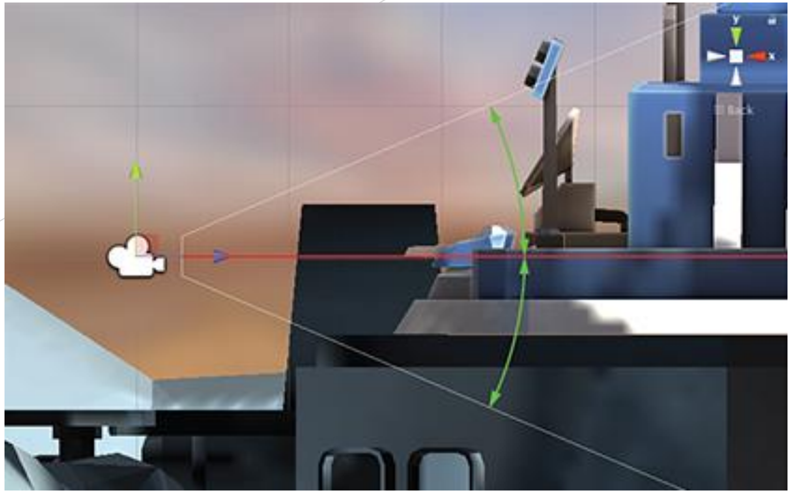
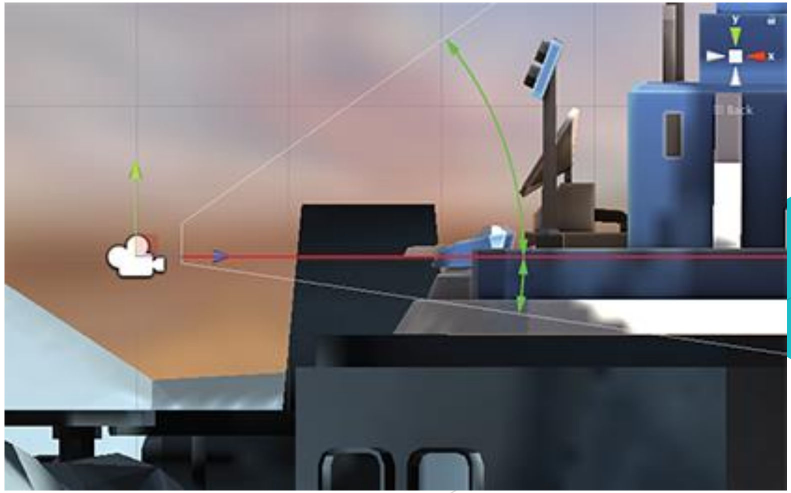
Using Physical Cameras

- Sensor Size
  - The width and height of the sensor that captures the image. These determine the physical camera's aspect ratio.
  - You can choose from several preset sensor sizes that correspond to real-world camera formats, or set a custom size. When the sensor aspect ratio is different to the rendered aspect ratio, as set in the Game view, you can control how Unity fits the camera image to the rendered image.
- Lens Shifts
  - Lens Shift offsets the camera's lens from its sensor horizontally and vertically. This allows you to change the focal center, and reposition a subject in the rendered frame, with little or no distortion.

Using Physical Cameras

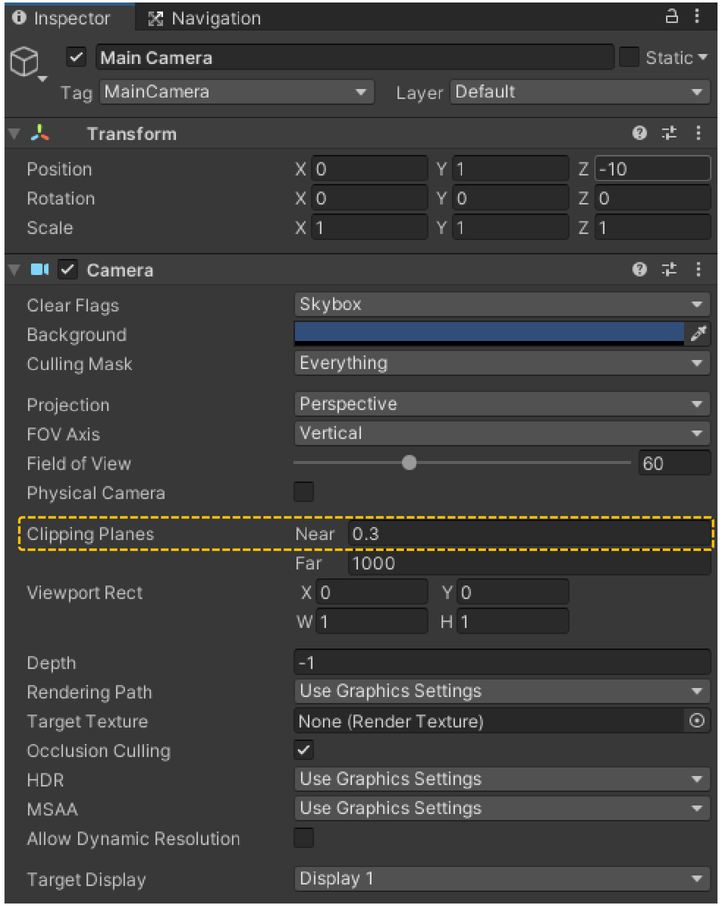
Lens Shifts

- One side effect of a lens shift is that it makes the camera's view frustum oblique. That means the angle between the camera's center line and its frustum is smaller on one side than on the other.
- The image show the camera frustum before (left) and after (right) a Y-axis lens shift.

이미지 출처 :https://docs.unity3d.com/Manual/PhysicalCameras.html

Camera Components


Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform
Position X 0 Y 1 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Camera
Clear Flags: Skybox
Background
Culling Mask: Everything
Projection: Perspective
FOV Axis: Vertical
Field of View: 60
Physical Camera
Clipping Planes: Near 0.3, Far 1000
Viewport Rect: X 0 Y 0 W 1 H 1
Depth: -1
Rendering Path: Use Graphics Settings
Target Texture: None (Render Texture)
Occlusion Culling: ✓
HDR: Use Graphics Settings
MSAA: Use Graphics Settings
Allow Dynamic Resolution
Target Display: Display 1

Clipping Planes

Distances from the camera to start and stop rendering.

Near

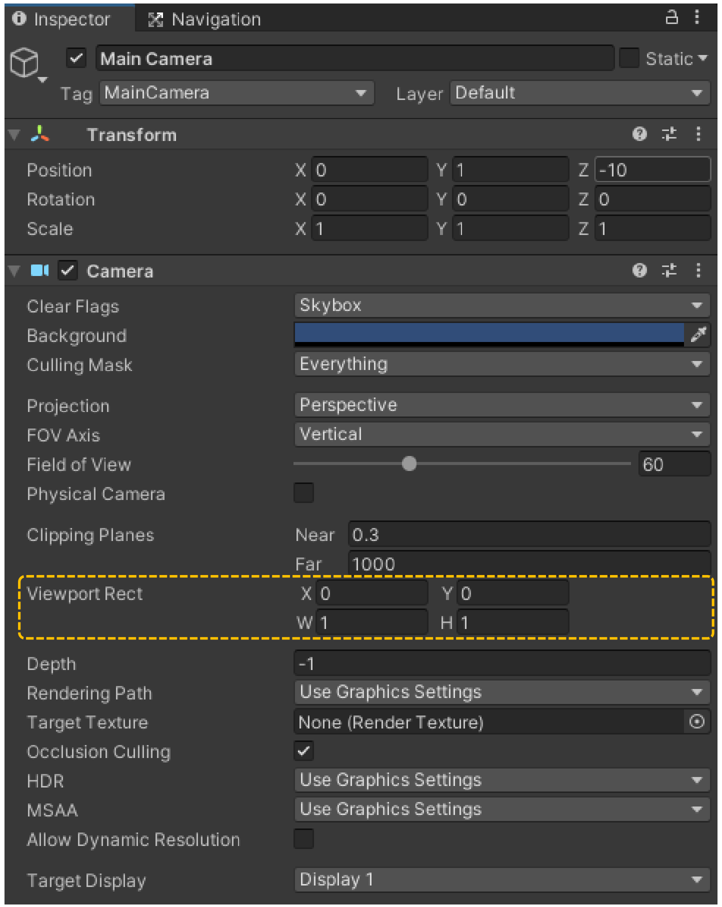
The closest point relative to the camera that drawing will occur.

Far

The furthest point relative to the camera that drawing will occur.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Viewport Rect

Four values that indicate where on the screen this camera view will be drawn. Measured in Viewport Coordinates (values 0~1).

X

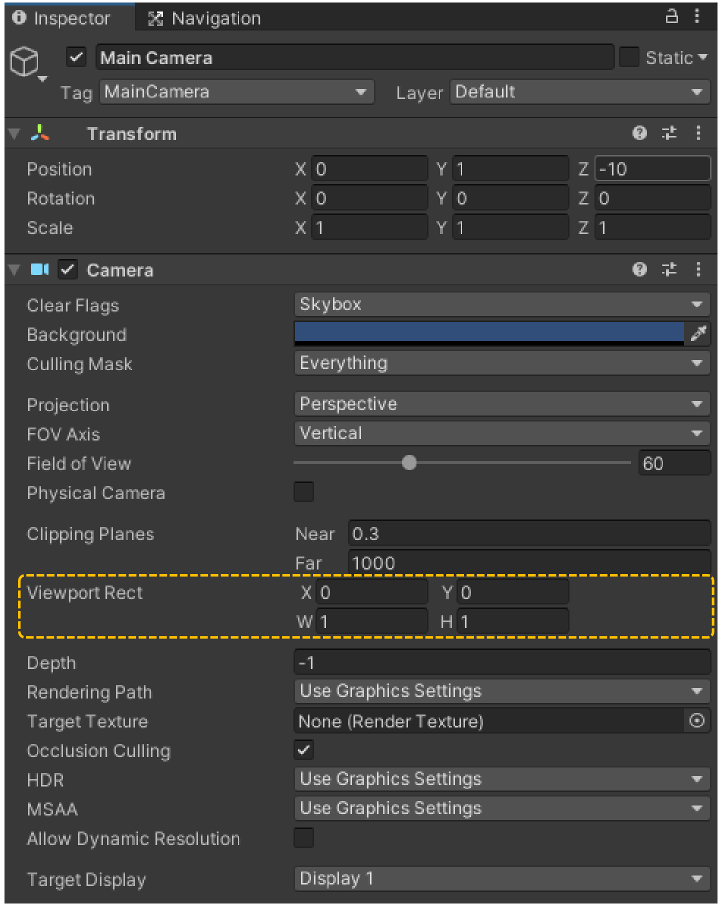
The beginning horizontal position that the camera view will be drawn.

Y

The beginning vertical position that the camera view will be drawn.

이미지 출처 :Unity

Camera Components



Inspector | Navigation

✓ Main Camera | Static

Tag: MainCamera | Layer: Default

Transform

	X	Y	Z
Position	0	1	-10
Rotation	0	0	0
Scale	1	1	1

✓ Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Viewport Rect

Four values that indicate where on the screen this camera view will be drawn. Measured in Viewport Coordinates (values 0~1).

W (Width)

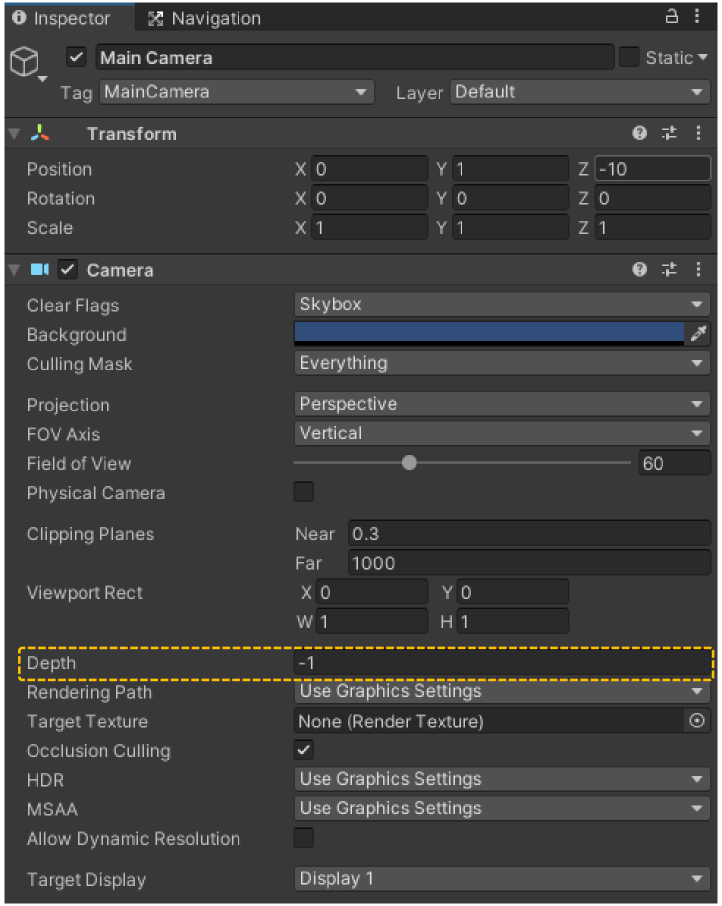
Width of the camera output on the screen.

H (Height)

Height of the camera output on the screen.

이미지 출처 :Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag: MainCamera | Layer: Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

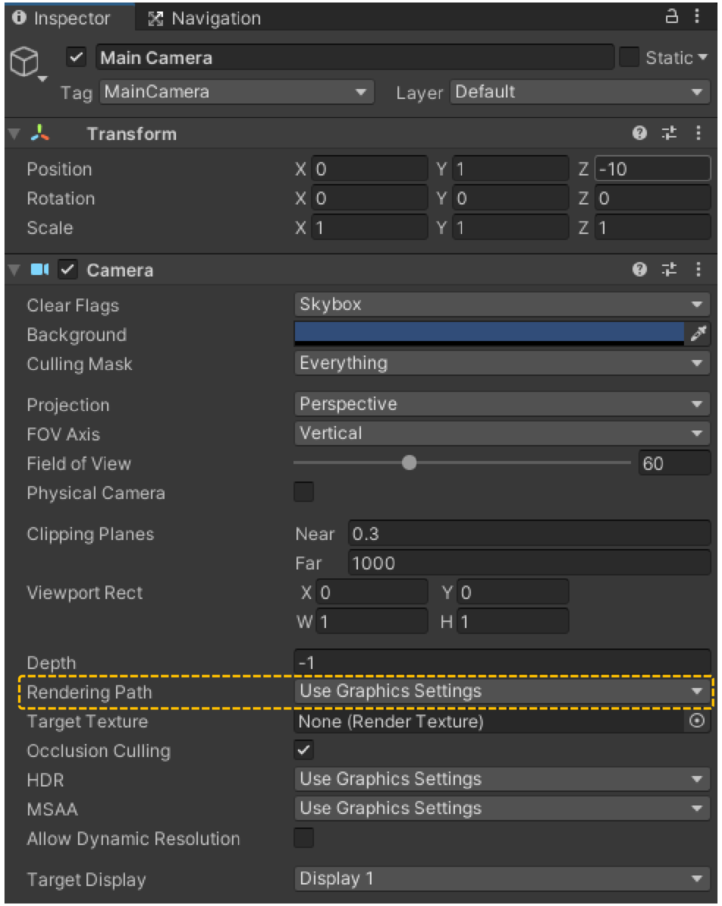
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Depth

The camera's position in the draw order. Cameras with a larger value will be drawn on top of cameras with a smaller value.

이미지 출처 : Unity

Camera Components


Inspector Navigation

Main Camera Static

Tag MainCamera Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

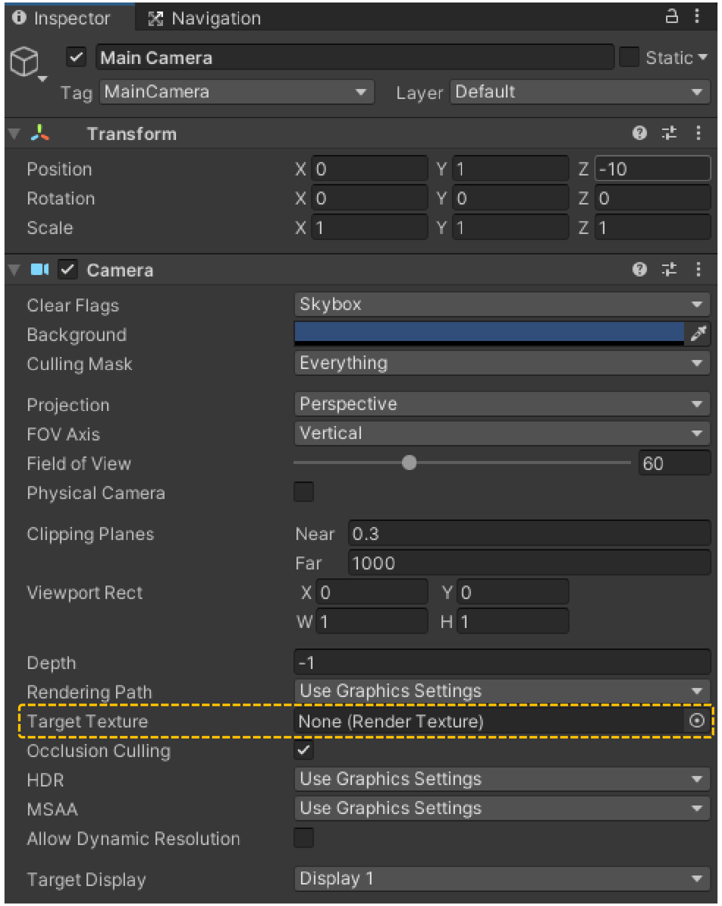
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

이미지 출처 : Unity

Camera Components

Property	Function
Rendering Path	Options for defining what rendering methods will be used by the camera.
Forward	Forward is the traditional rendering path.
Deferred Lighting	Deferred Shading is the rendering path with the most lighting and shadow fidelity, and is best suited if you have many real time lights. It requires a certain level of hardware support.
Legacy Vertex Lit	Legacy Vertex Lit is the rendering path with the lowest lighting fidelity and no support for real time shadows. It is a subset of Forward rendering path.
Legacy Deferred	Legacy Deferred (light prepass) is similar to Deferred Shading, just using a different technique with different trade-offs.

Camera Components



Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

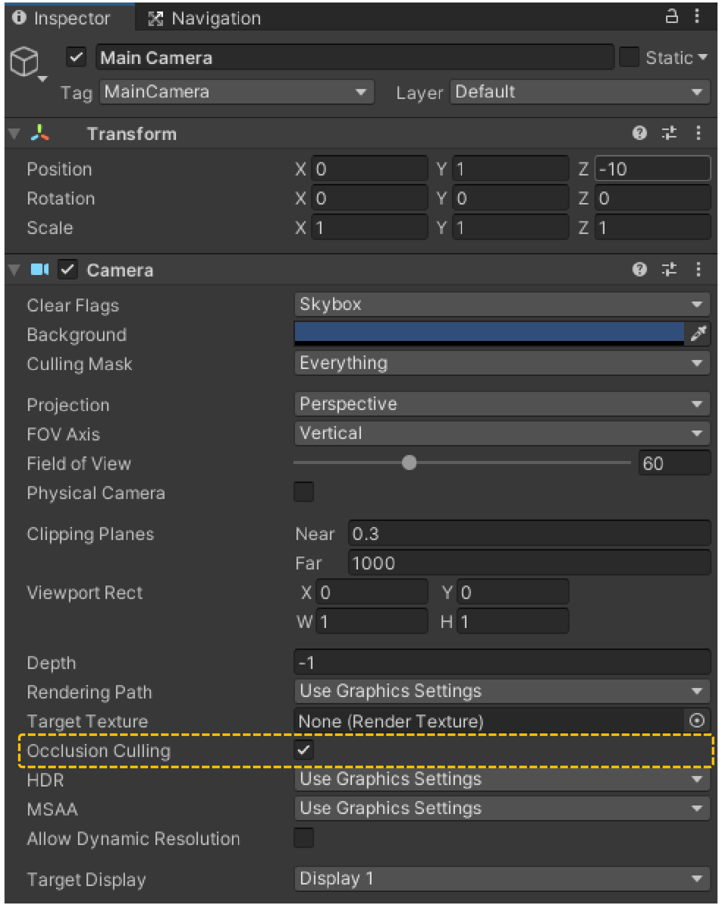
Clear Flags	Skybox
Background	
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

Target Texture

Reference to a Render Texture that will contain the output of the Camera view. Setting this reference will disable this Camera's capability to render to the screen.

이미지 출처 : Unity

Camera Components


Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera ✓

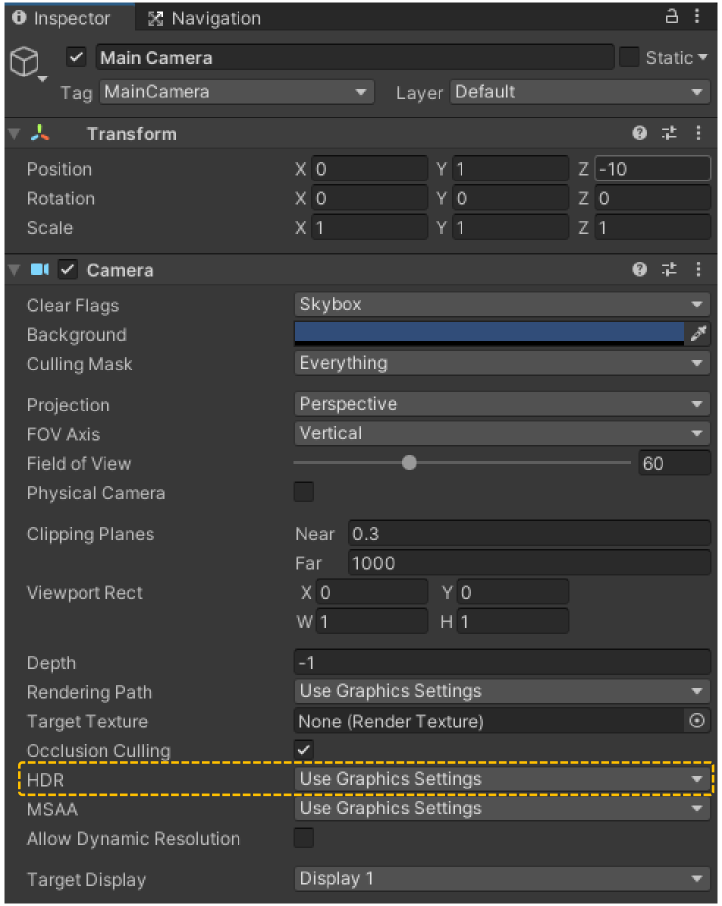
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Occlusion Culling

Enables Occlusion Culling for this camera.
Occlusion Culling means that objects that are hidden behind other objects are not rendered, for example if they are behind walls.

이미지 출처 :Unity

Camera Components


Inspector | Navigation

Main Camera | Static
Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

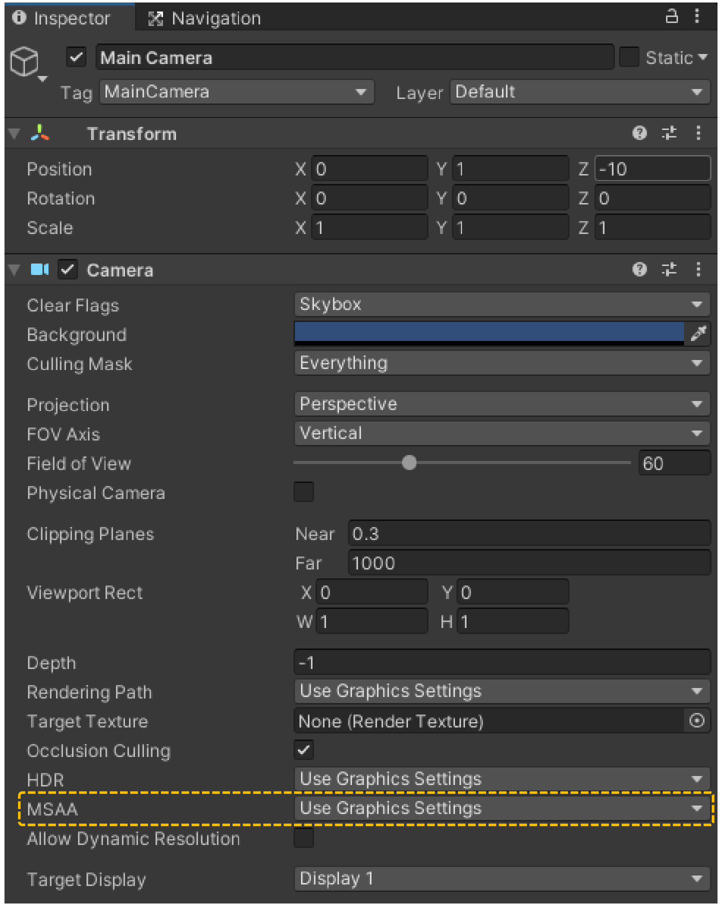
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Allow HDR

Enables High Dynamic Range rendering for this camera.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag: MainCamera | Layer: Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

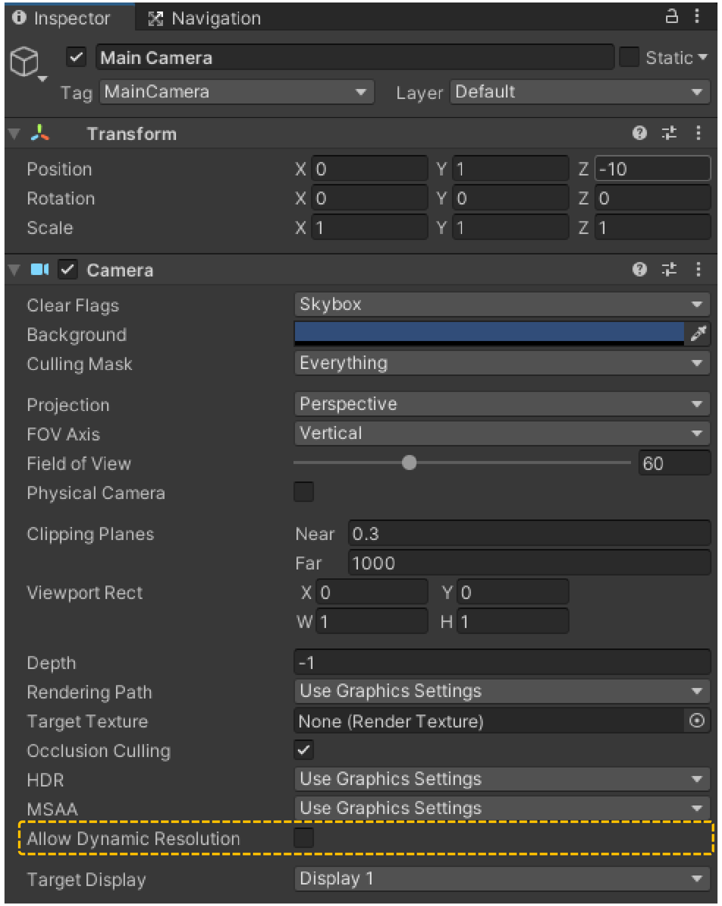
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Allow MSAA

Enables multi sample antialiasing
for this camera.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

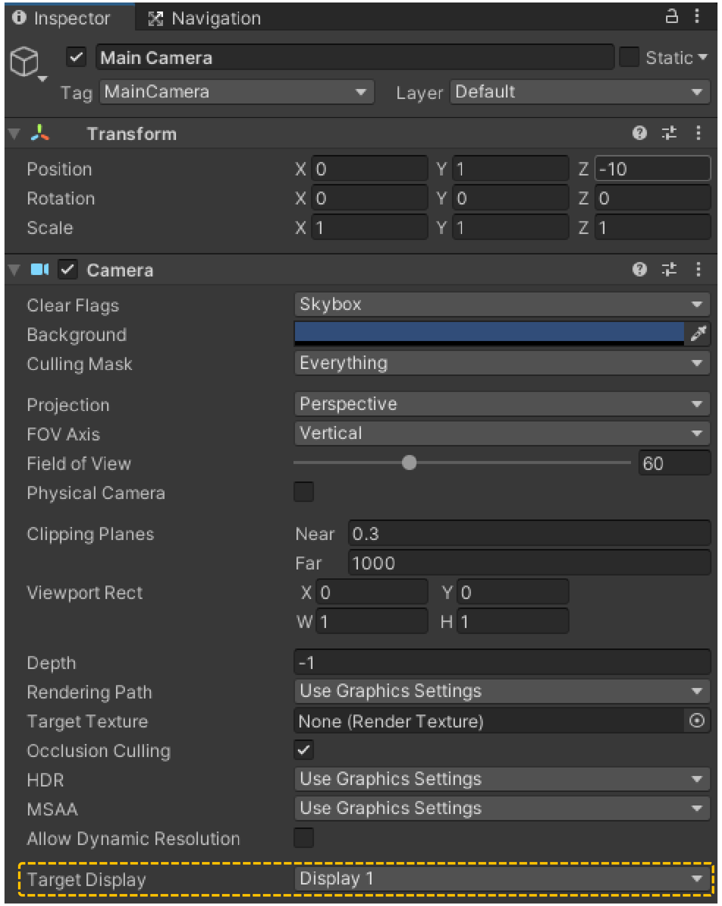
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Allow Dynamic Resolution

Enables Dynamic Resolution rendering for this camera.

이미지 출처 : Unity

Camera Components



Inspector | Navigation

Main Camera — Static

Tag: MainCamera — Layer: Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	☐
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Target Display

Defines which external device to render to.
Between 1 and 8.

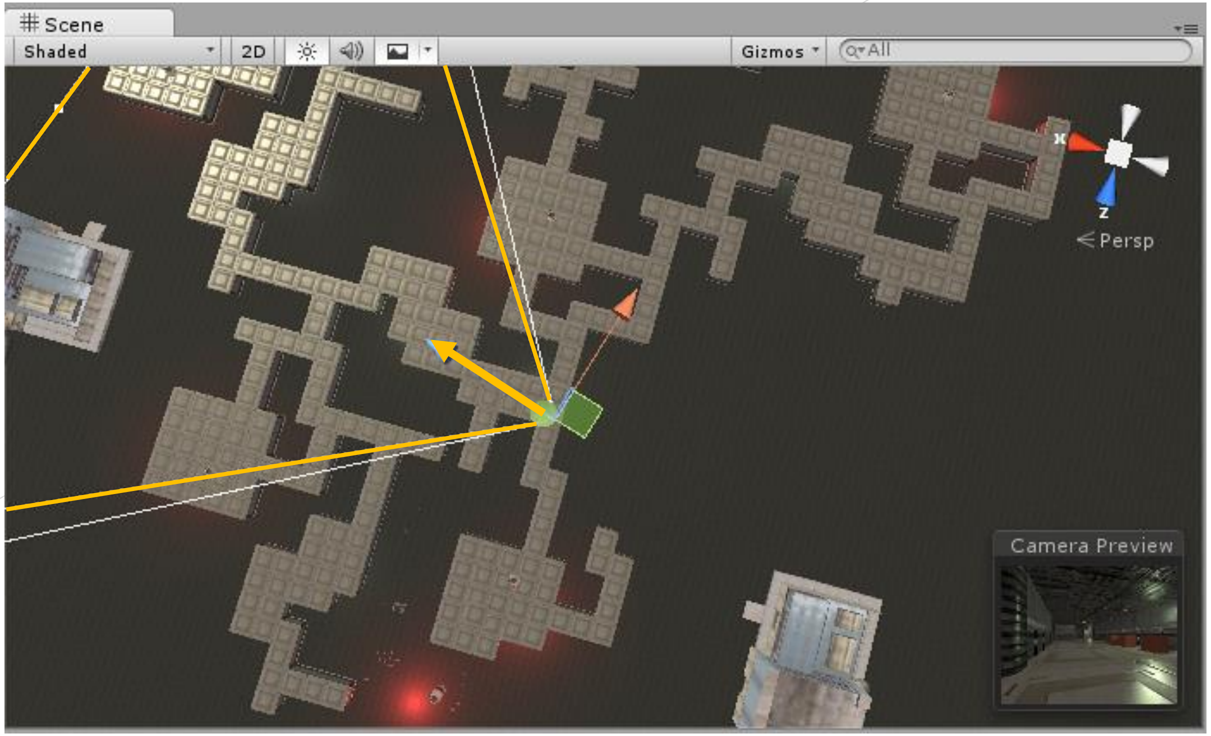
이미지 출처 : Unity

Occlusion Culling

- Occlusion Culling is a feature that disables rendering of objects when they are not currently seen by the camera because they are obscured (occluded) by other objects.
- This does not happen automatically in 3D computer graphics since most of the time objects farthest away from the camera are drawn first and closer objects are drawn over the top of them
(this is called “overdraw”)
- Occlusion Culling is different from Frustum Culling.
- Frustum Culling only disables the renderers for objects that are outside the camera’s viewing area but does not disable anything hidden from view by overdraw.
- Note that when you use Occlusion Culling you will still benefit from Frustum Culling.

Occlusion Culling

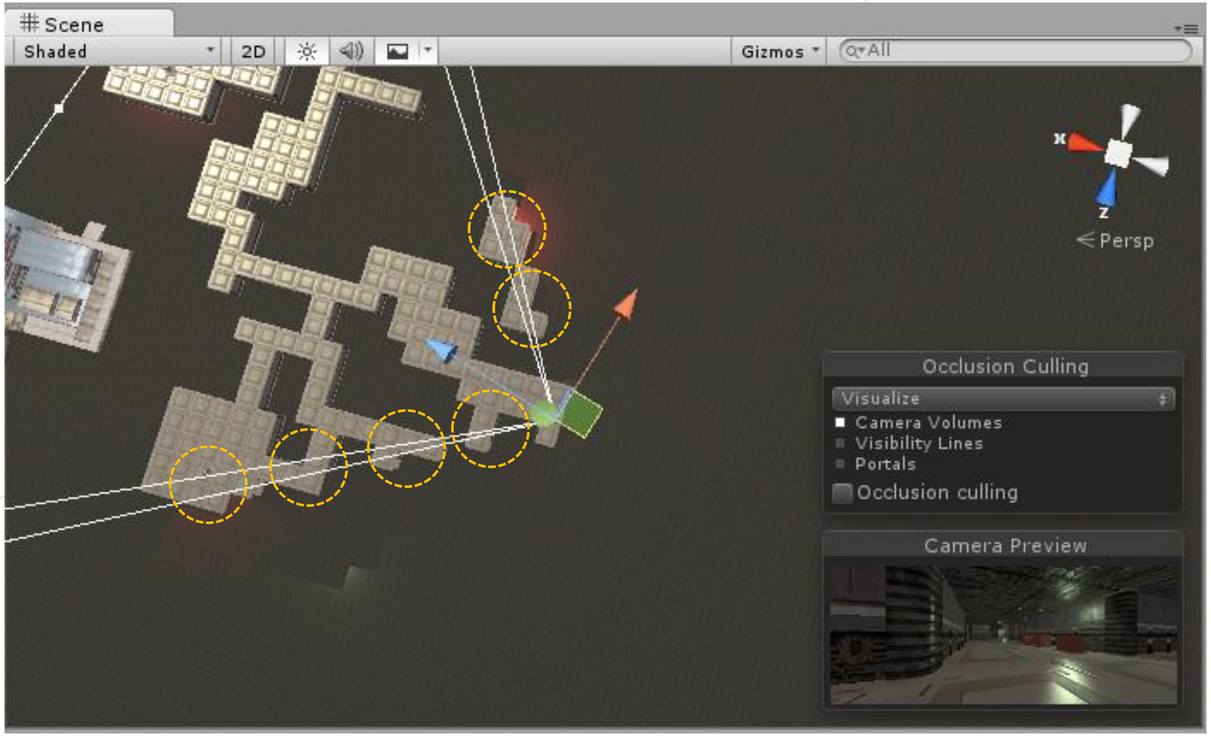
- A maze-like indoor level. This normal scene view shows all visible Game Objects.



이미지 출처 :https://docs.unity.com/Manual/OcclusionCulling.html

Occlusion Culling

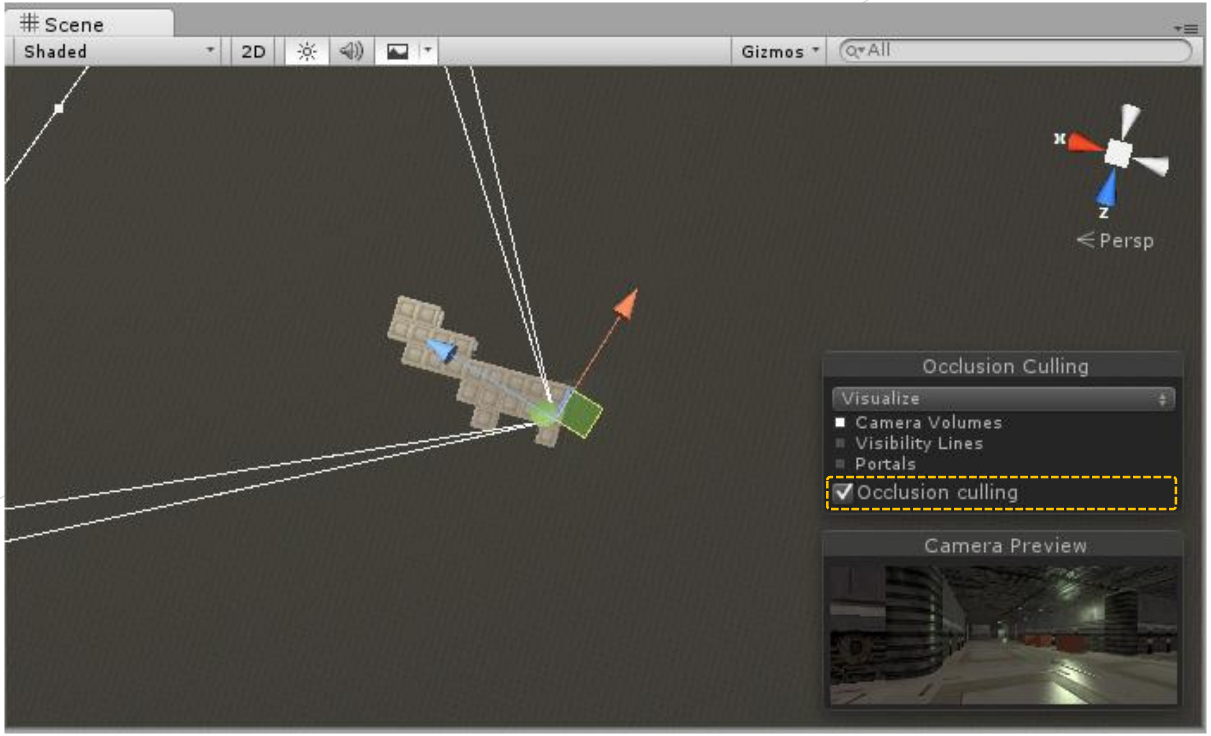
- Regular frustum culling renders all Renderers within the Camera’s view.



이미지 출처 :https://docs.unity.com/Manual/OcclusionCulling.html

Occlusion Culling

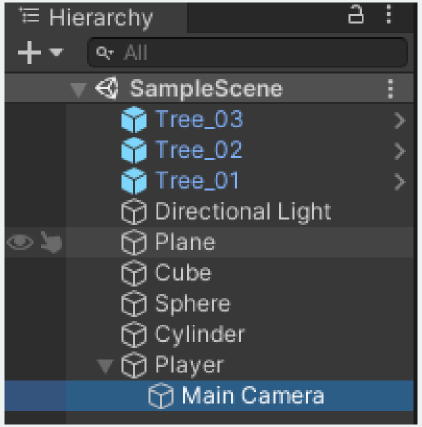
- Occlusion culling removes Renderers that are entirely obscured by nearer Renderer.



이미지 출처 :https://docs.unity.com/Manual/OcclusionCulling.html

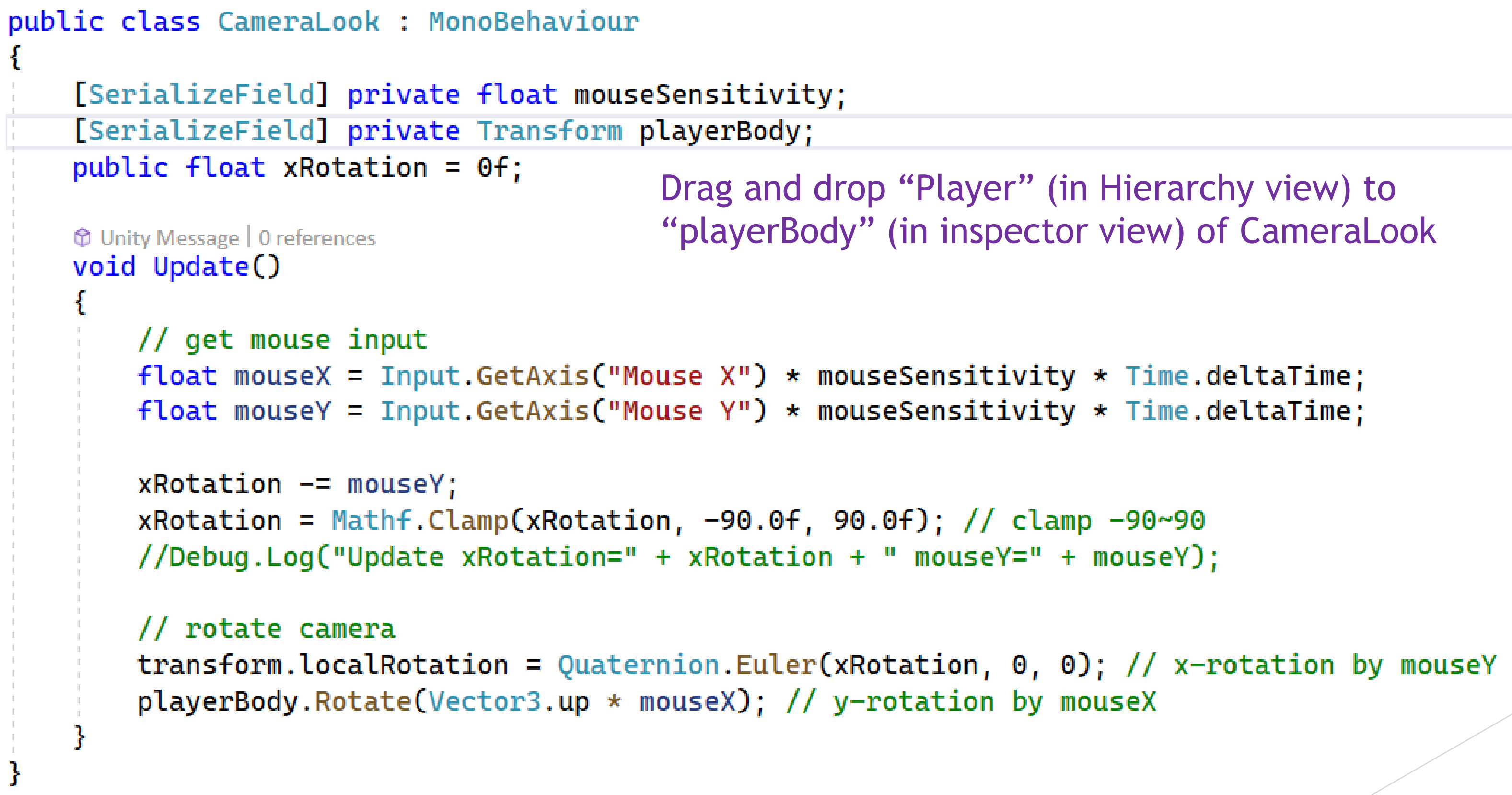
First Person Camera in Unity

- To create Unity 1st person camera
  - ➢ Create a scene ➡ Plane, Cube, sphere, Cylinder, Tree, etc
  - ➢ Create a capsule position (0, 1, 0) and rename it to “Player”
  - ➢ Make a Main Camera as a child of Player and set its position (0, 0.9, 0)
  - ➢ Attach CameraLook script to a Main Camera
  - ➢ Attach Player to CameraLook script


Hierarchy
All
SampleScene
Tree_03
Tree_02
Tree_01
Directional Light
Plane
Cube
Sphere
Cylinder
Player
Main Camera

이미지 출처 :Unity

First Person Camera in Unity



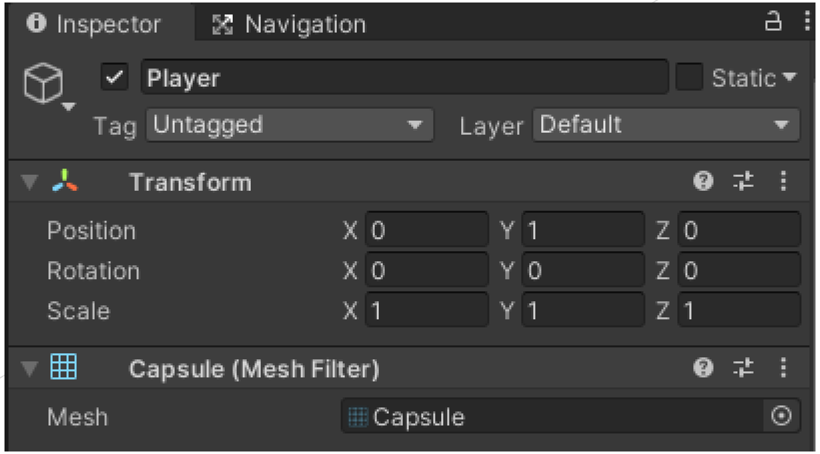
```
public class CameraLook : MonoBehaviour
{
    [SerializeField] private float mouseSensitivity;
    [SerializeField] private Transform playerBody;
    public float xRotation = 0f;

    Unity Message | 0 references
    void Update()
    {
        // get mouse input
        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

        xRotation -= mouseY;
        xRotation = Mathf.Clamp(xRotation, -90.0f, 90.0f); // clamp -90~90
        //Debug.Log("Update xRotation=" + xRotation + " mouseY=" + mouseY);

        // rotate camera
        transform.localRotation = Quaternion.Euler(xRotation, 0, 0); // x-rotation by mouseY
        playerBody.Rotate(Vector3.up * mouseX); // y-rotation by mouseX
    }
}
```

Drag and drop “Player” (in Hierarchy view) to “playerBody” (in inspector view) of CameraLook

이미지 출처 :Unity

First Person Camera in Unity

- To create Unity 1st person camera
  - Add a Rigidbody component in Player
  - Add PlayerMove script to Player



Inspector | Navigation

☑ Player ☐ Static

Tag Untagged | Layer Default

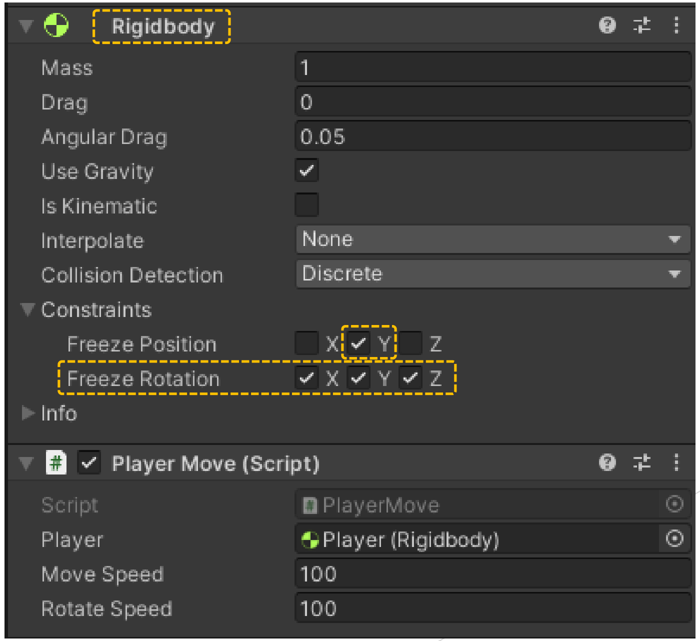
Transform

	X	Y	Z
Position	0	1	0
Rotation	0	0	0
Scale	1	1	1

Capsule (Mesh Filter)

Mesh: Capsule

이미지 출처 :Unity



Rigidbody

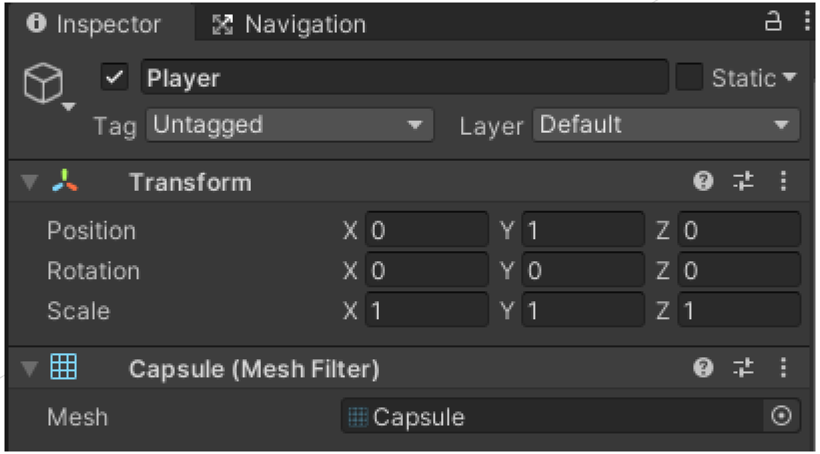
Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	✓
Is Kinematic	☐
Interpolate	None
Collision Detection	Discrete
Constraints	
Freeze Position	☐ X ✓ Y ☐ Z
Freeze Rotation	✓ X ✓ Y ✓ Z
Info	

☑ Player Move (Script)

Script	PlayerMove
Player	Player (Rigidbody)
Move Speed	100
Rotate Speed	100

First Person Camera in Unity

- To create Unity 1st person camera
  - Add a Rigidbody component in Player
  - Add PlayerMove script to Player


Inspector | Navigation

Player | Static

Tag Untagged | Layer Default

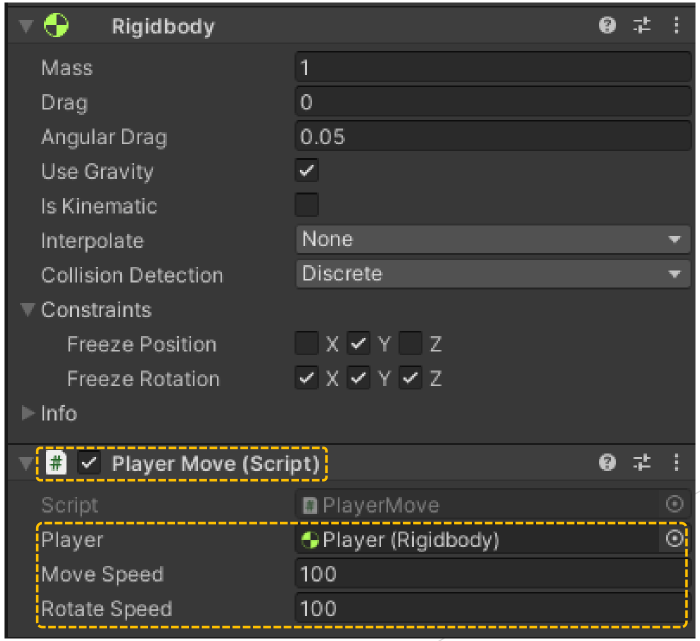
Transform

Position	X 0	Y 1	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Capsule (Mesh Filter)

Mesh: Capsule

이미지 출처 :Unity

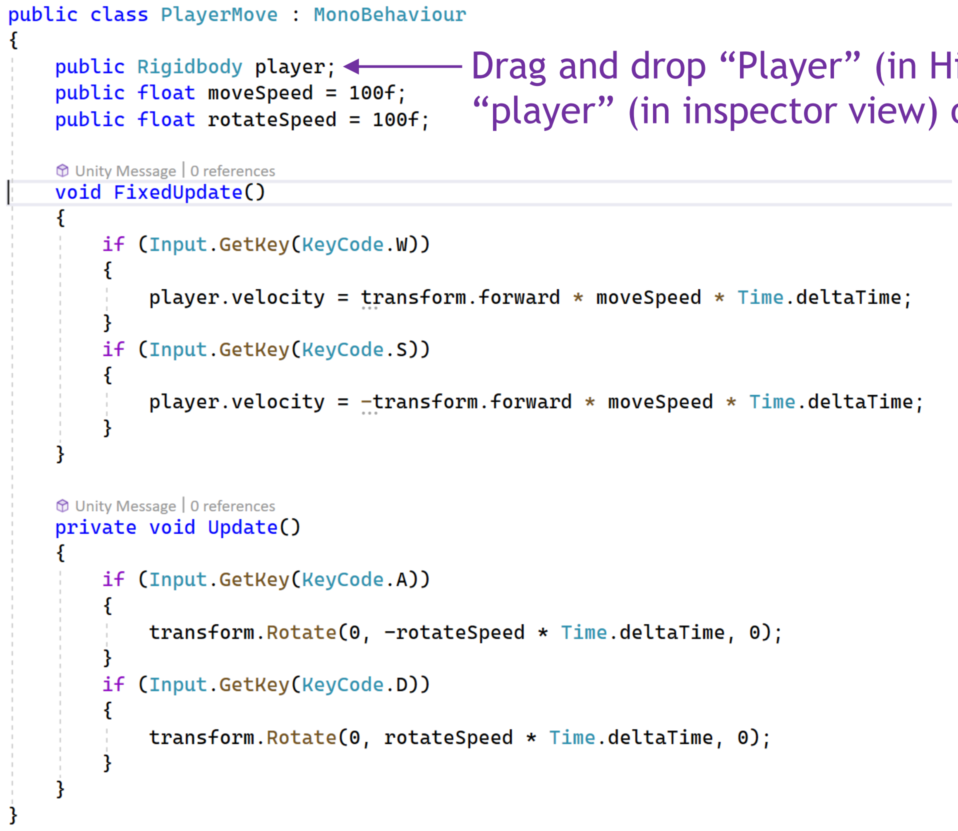

Rigidbody

Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	✓
Is Kinematic	
Interpolate	None
Collision Detection	Discrete
Constraints	
Freeze Position	X ✓Y Z
Freeze Rotation	✓X ✓Y ✓Z
Info	

Player Move (Script)

Script	PlayerMove
Player	Player (Rigidbody)
Move Speed	100
Rotate Speed	100

First Person Camera in Unity



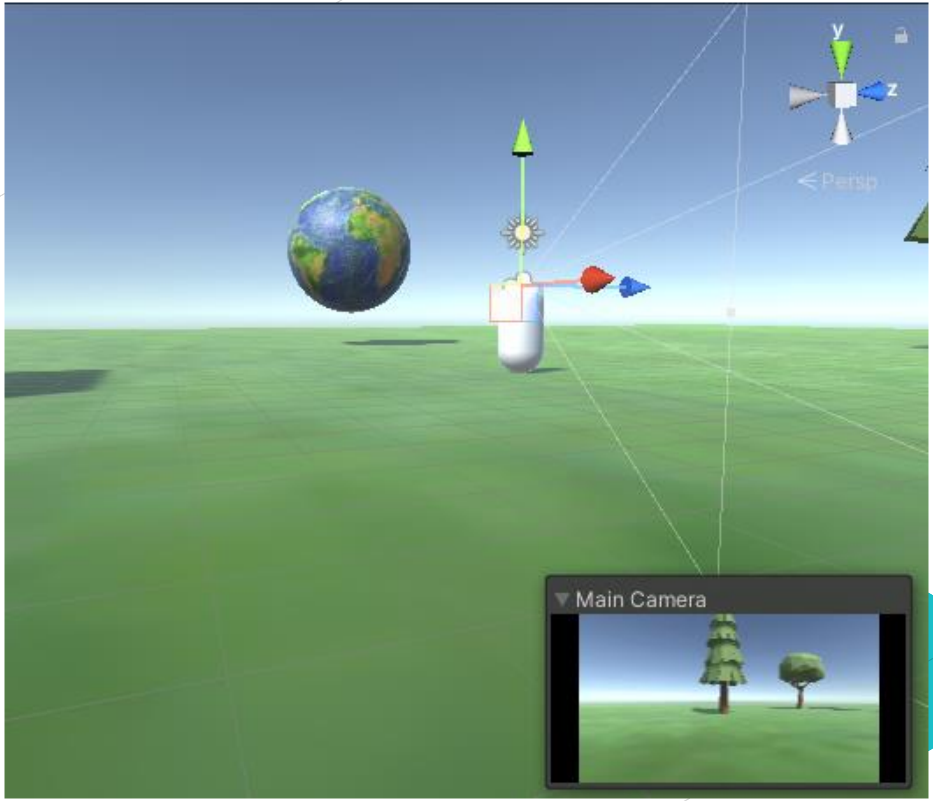
```
public class PlayerMove : MonoBehaviour
{
    public Rigidbody player;
    public float moveSpeed = 100f;
    public float rotateSpeed = 100f;

    Unity Message | 0 references
    void FixedUpdate()
    {
        if (Input.GetKey(KeyCode.W))
        {
            player.velocity = transform.forward * moveSpeed * Time.deltaTime;
        }
        if (Input.GetKey(KeyCode.S))
        {
            player.velocity = -transform.forward * moveSpeed * Time.deltaTime;
        }
    }

    Unity Message | 0 references
    private void Update()
    {
        if (Input.GetKey(KeyCode.A))
        {
            transform.Rotate(0, -rotateSpeed * Time.deltaTime, 0);
        }
        if (Input.GetKey(KeyCode.D))
        {
            transform.Rotate(0, rotateSpeed * Time.deltaTime, 0);
        }
    }
}
```

Drag and drop “Player” (in Hierarchy view) to “player” (in inspector view) of PlayerMove

이미지 출처 :Unity

First Person Camera in Unity

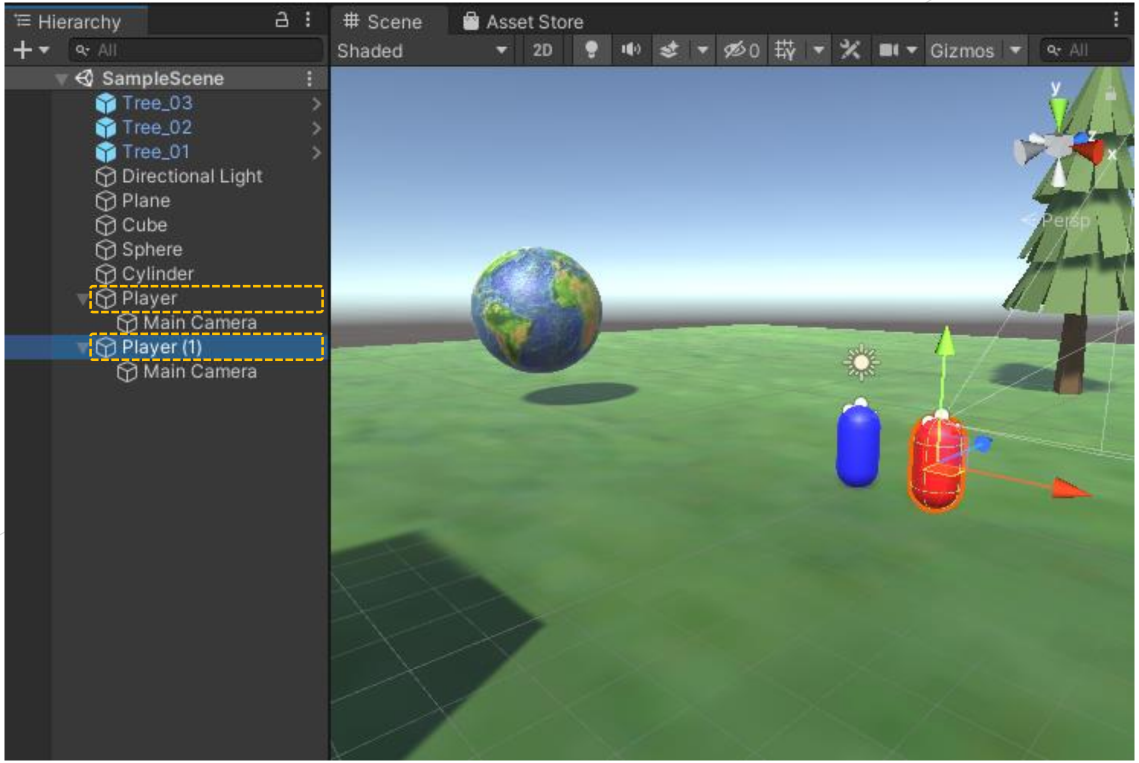
- Mouse x/y move to look around
- WS-key to move forward/backward
- AD-key to left/right panning



이미지 출처 :Unity

Split Screen for Multiplayer in Unity

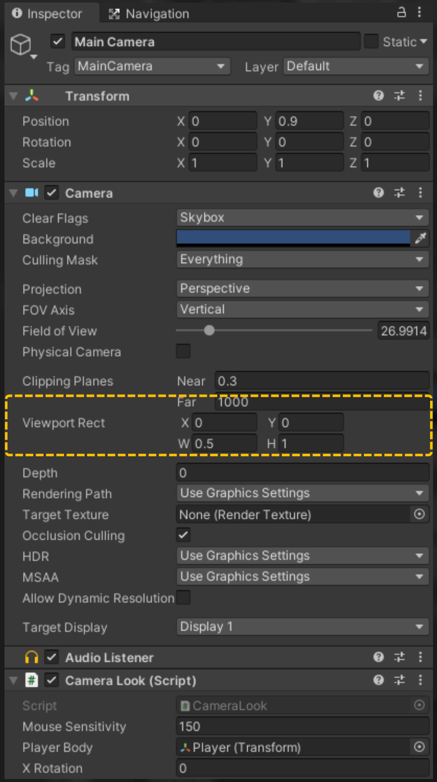
- Copy and Paste Player to make Player (1) Pos X=2



SampleScene
- Tree_03
- Tree_02
- Tree_01
- Directional Light
- Plane
- Cube
- Sphere
- Cylinder
- Player
  - Main Camera
- Player (1)
  - Main Camera

이미지 출처 : Unity

Split Screen for Multiplayer in Unity



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 0.9	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

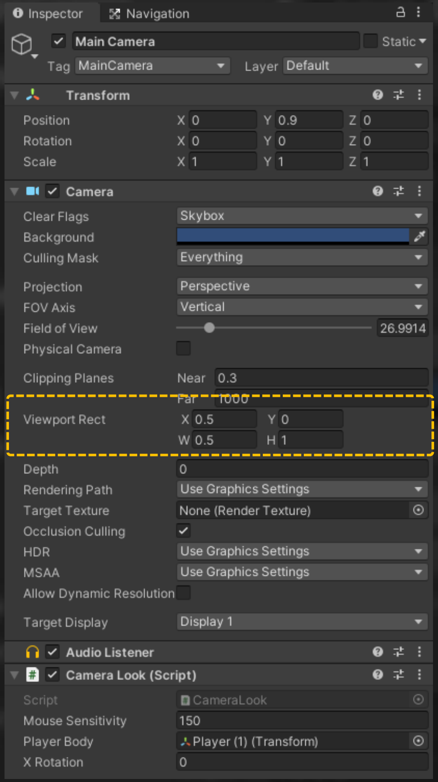
Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	26.9914
Physical Camera	☐
Clipping Planes	Near 0.3 / Far 1000
Viewport Rect	X 0 Y 0 / W 0.5 H 1
Depth	0
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Audio Listener

Camera Look (Script)

Script	CameraLook
Mouse Sensitivity	150
Player Body	Player (Transform)
X Rotation	0

Player’s MainCamera



Inspector | Navigation

Main Camera | Static

Tag MainCamera | Layer Default

Transform

Position	X 0	Y 0.9	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

Clear Flags	Skybox
Background	🟦
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	26.9914
Physical Camera	☐
Clipping Planes	Near 0.3 / Far 1000
Viewport Rect	X 0.5 Y 0 / W 0.5 H 1
Depth	0
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	☐
Target Display	Display 1

Audio Listener

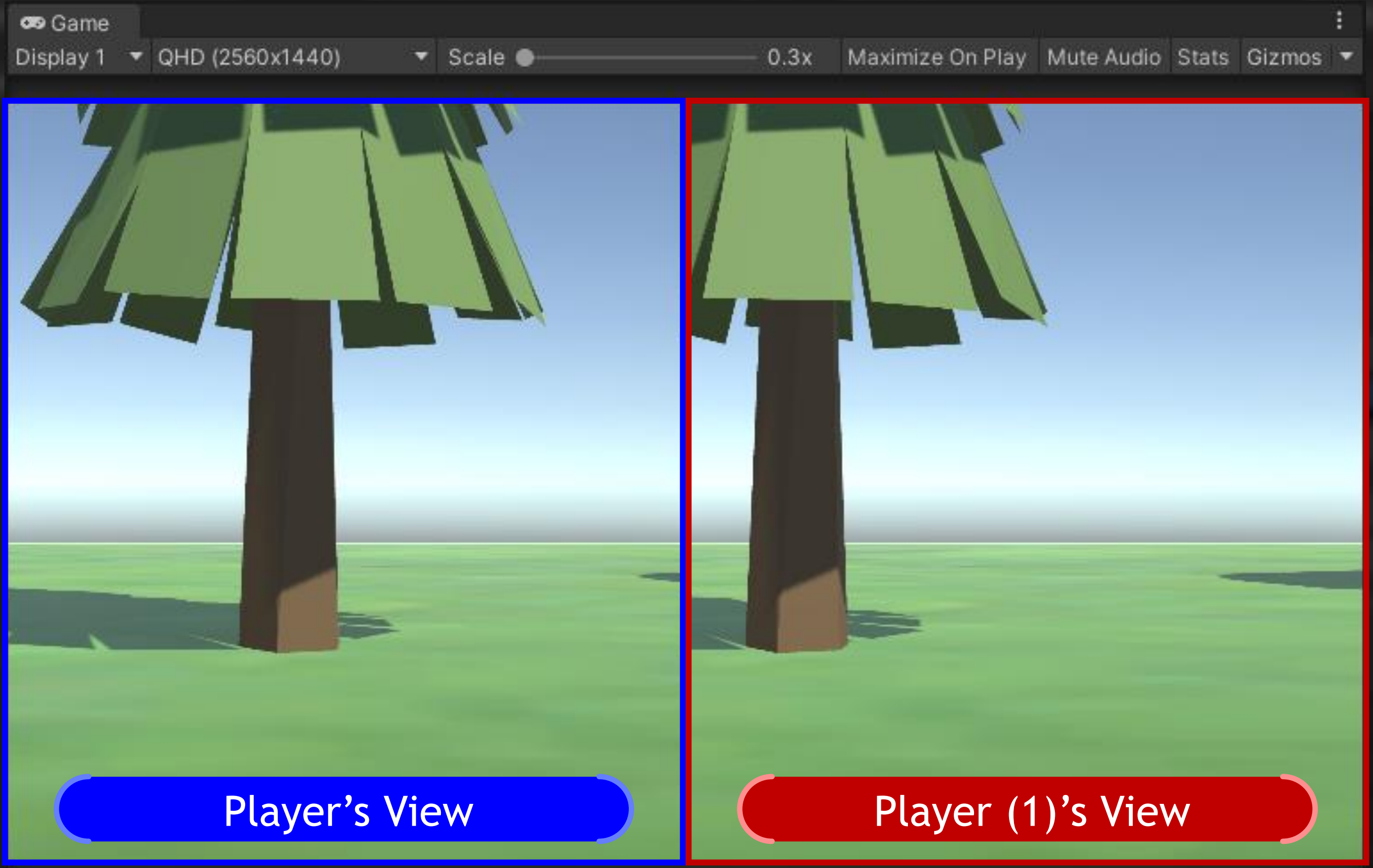
Camera Look (Script)

Script	CameraLook
Mouse Sensitivity	150
Player Body	Player (1) (Transform)
X Rotation	0

Player (1)’s MainCamera

이미지 출처 :Unity

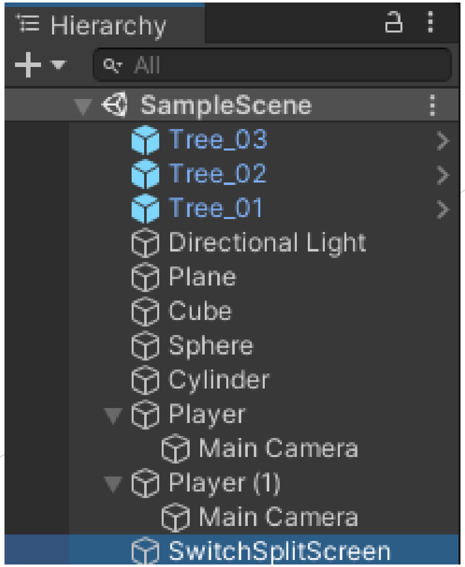
Split Screen for Multiplayer in Unity



이미지 출처 :Unity

Split Screen for Multiplayer in Unity

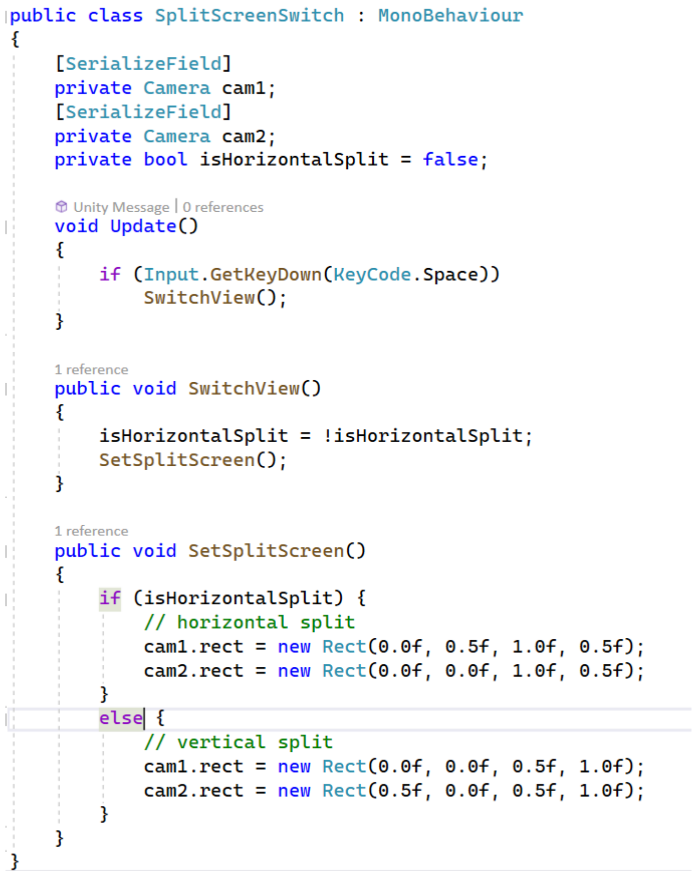
- Create Empty GameObject, and name it SwitchSplitScreen
- Then, add SplitScreenSwitch C# script component



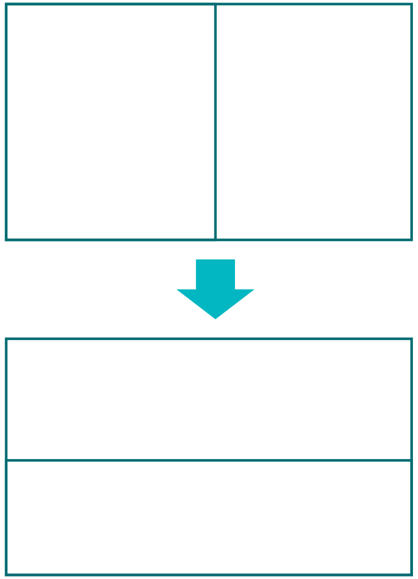
```
SampleScene
  Tree_03
  Tree_02
  Tree_01
  Directional Light
  Plane
  Cube
  Sphere
  Cylinder
  Player
    Main Camera
  Player (1)
    Main Camera
  SwitchSplitScreen
```

이미지 출처 :Unity

Split Screen for Multiplayer in Unity

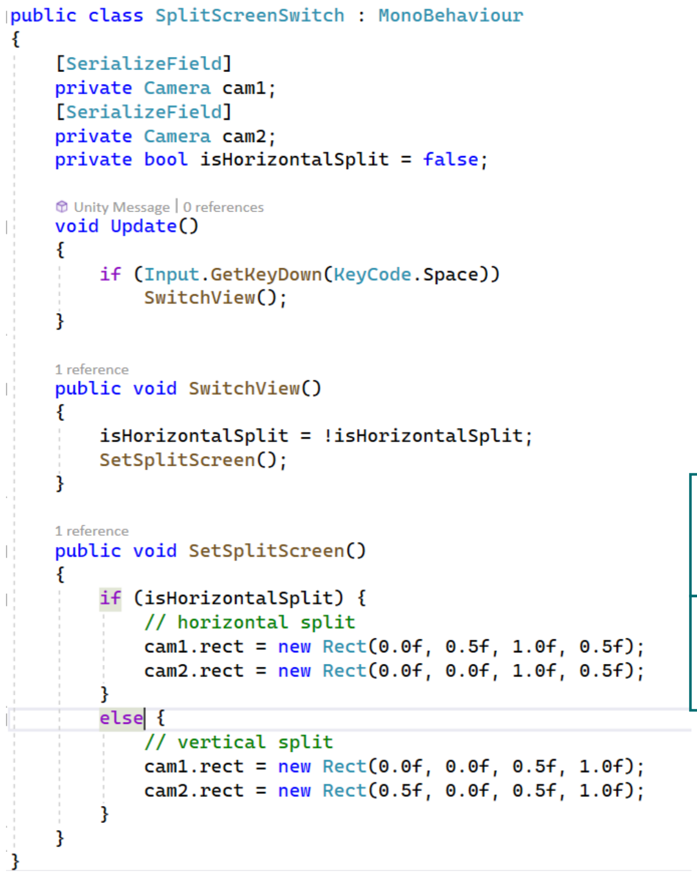


```
public class SplitScreenSwitch : MonoBehaviour
{
    [SerializeField]
    private Camera cam1;
    [SerializeField]
    private Camera cam2;
    private bool isHorizontalSplit = false;

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
            SwitchView();
    }

    1 reference
    public void SwitchView()
    {
        isHorizontalSplit = !isHorizontalSplit;
        SetSplitScreen();
    }

    1 reference
    public void SetSplitScreen()
    {
        if (isHorizontalSplit) {
            // horizontal split
            cam1.rect = new Rect(0.0f, 0.5f, 1.0f, 0.5f);
            cam2.rect = new Rect(0.0f, 0.0f, 1.0f, 0.5f);
        }
        else {
            // vertical split
            cam1.rect = new Rect(0.0f, 0.0f, 0.5f, 1.0f);
            cam2.rect = new Rect(0.5f, 0.0f, 0.5f, 1.0f);
        }
    }
}
```

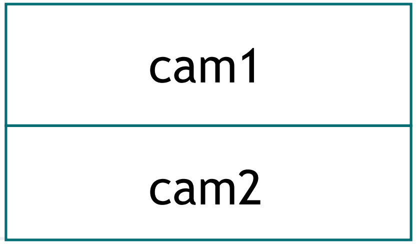


이미지 출처 :Unity

Split Screen for Multiplayer in Unity

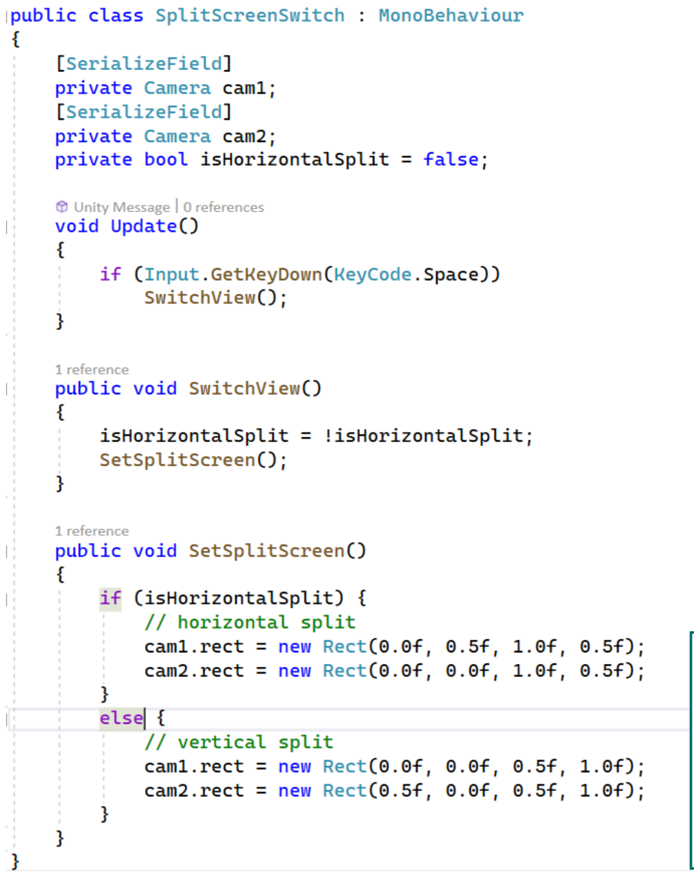


```
public class SplitScreenSwitch : MonoBehaviour
{
    [SerializeField]
    private Camera cam1;
    [SerializeField]
    private Camera cam2;
    private bool isHorizontalSplit = false;

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
            SwitchView();
    }

    1 reference
    public void SwitchView()
    {
        isHorizontalSplit = !isHorizontalSplit;
        SetSplitScreen();
    }

    1 reference
    public void SetSplitScreen()
    {
        if (isHorizontalSplit) {
            // horizontal split
            cam1.rect = new Rect(0.0f, 0.5f, 1.0f, 0.5f);
            cam2.rect = new Rect(0.0f, 0.0f, 1.0f, 0.5f);
        }
        else {
            // vertical split
            cam1.rect = new Rect(0.0f, 0.0f, 0.5f, 1.0f);
            cam2.rect = new Rect(0.5f, 0.0f, 0.5f, 1.0f);
        }
    }
}
```

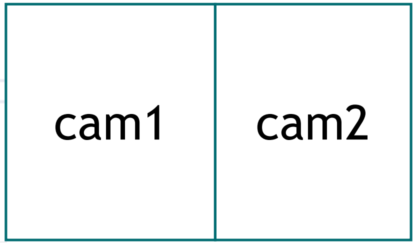


이미지 출처 :Unity

Split Screen for Multiplayer in Unity



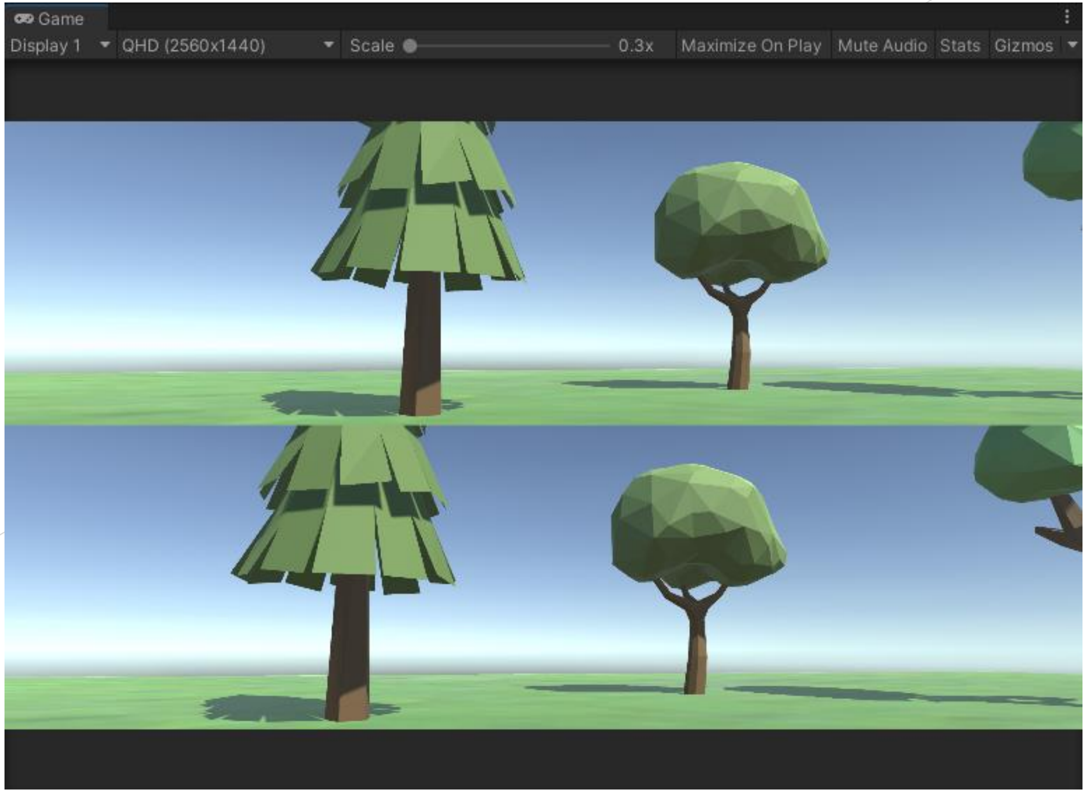
```
public class SplitScreenSwitch : MonoBehaviour
{
    [SerializeField]
    private Camera cam1;
    [SerializeField]
    private Camera cam2;
    private bool isHorizontalSplit = false;

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
            SwitchView();
    }

    1 reference
    public void SwitchView()
    {
        isHorizontalSplit = !isHorizontalSplit;
        SetSplitScreen();
    }

    1 reference
    public void SetSplitScreen()
    {
        if (isHorizontalSplit) {
            // horizontal split
            cam1.rect = new Rect(0.0f, 0.5f, 1.0f, 0.5f);
            cam2.rect = new Rect(0.0f, 0.0f, 1.0f, 0.5f);
        }
        else {
            // vertical split
            cam1.rect = new Rect(0.0f, 0.0f, 0.5f, 1.0f);
            cam2.rect = new Rect(0.5f, 0.0f, 0.5f, 1.0f);
        }
    }
}
```



이미지 출처 :Unity

Split Screen for Multiplayer in Unity

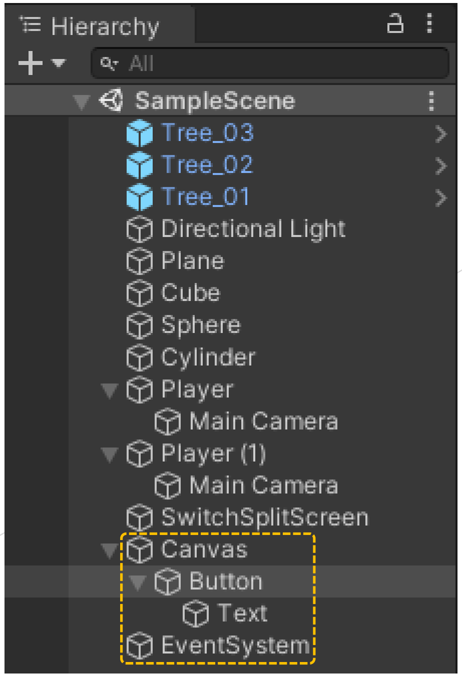
- Space-key to switch vertical/horizontal split screen.



이미지 출처 : Unity

Split Screen for Multiplayer in Unity

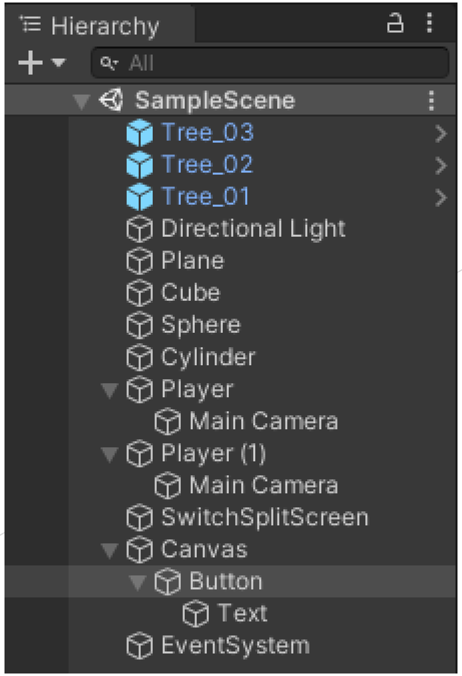
- Add UI Canvas, add Button under Canvas



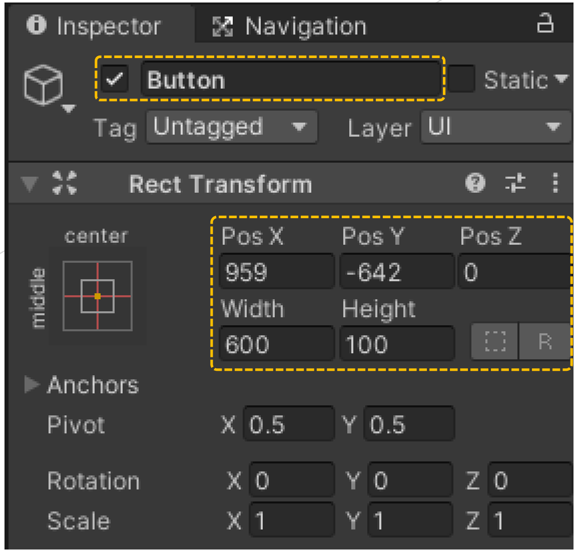
이미지 출처 :Unity

Split Screen for Multiplayer in Unity

- Add UI Canvas, add Button under Canvas



Hierarchy
- SampleScene
  - Tree_03
  - Tree_02
  - Tree_01
  - Directional Light
  - Plane
  - Cube
  - Sphere
  - Cylinder
  - Player
    - Main Camera
  - Player (1)
    - Main Camera
  - SwitchSplitScreen
  - Canvas
    - Button
      - Text
  - EventSystem



Inspector: Button, Tag Untagged, Layer UI

Rect Transform (center / middle)

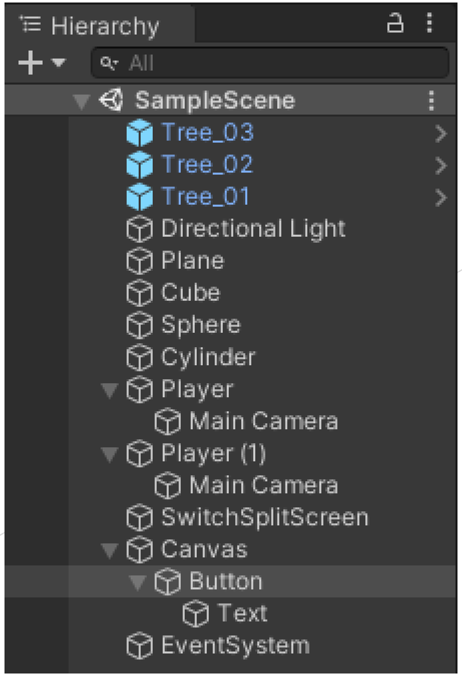
Pos X	Pos Y	Pos Z
959	-642	0
Width	Height	
600	100	

Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

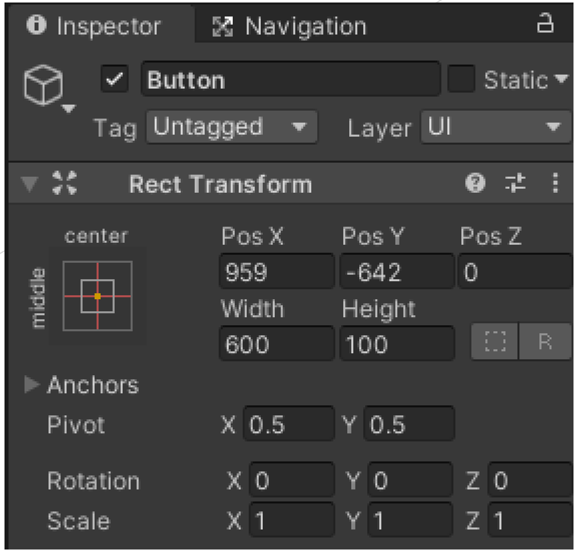
이미지 출처 :Unity

Split Screen for Multiplayer in Unity

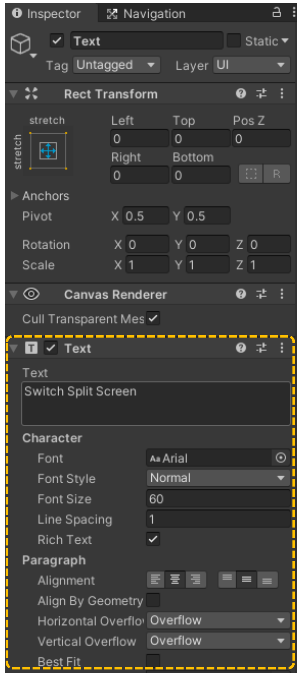
- Add UI Canvas, add Button under Canvas



```
SampleScene
    Tree_03
    Tree_02
    Tree_01
    Directional Light
    Plane
    Cube
    Sphere
    Cylinder
    Player
        Main Camera
    Player (1)
        Main Camera
    SwitchSplitScreen
    Canvas
        Button
            Text
    EventSystem
```



```
Button                 Static
Tag Untagged   Layer UI

Pos X 959    Pos Y -642    Pos Z 0
Width 600    Height 100
Pivot     X 0.5   Y 0.5
Rotation  X 0   Y 0   Z 0
Scale     X 1   Y 1   Z 1
```

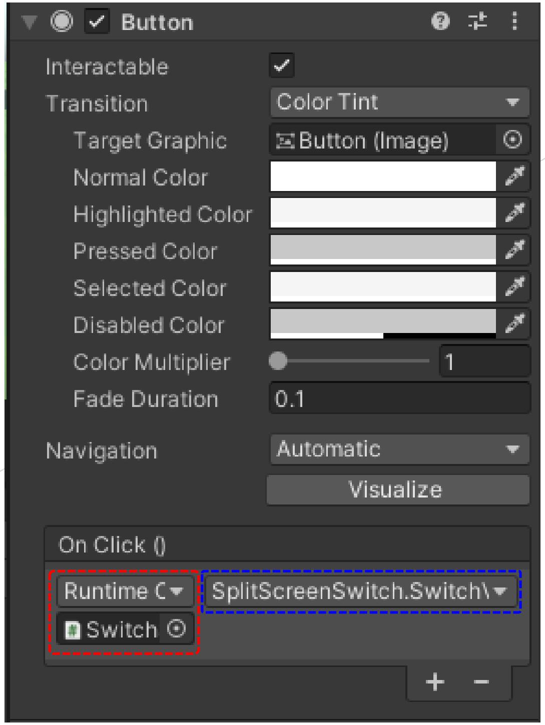


```
Text                   Static
Tag Untagged   Layer UI

Left 0   Top 0   Pos Z 0
Right 0  Bottom 0
Pivot     X 0.5   Y 0.5
Rotation  X 0   Y 0   Z 0
Scale     X 1   Y 1   Z 1

Cull Transparent Mes ✓

Text
Text: Switch Split Screen
Font: Arial
Font Style: Normal
Font Size: 60
Line Spacing: 1
Rich Text: ✓
Alignment
Align By Geometry
Horizontal Overflo: Overflow
Vertical Overflow: Overflow
Best Fit
```

이미지 출처 :Unity

Split Screen for Multiplayer in Unity

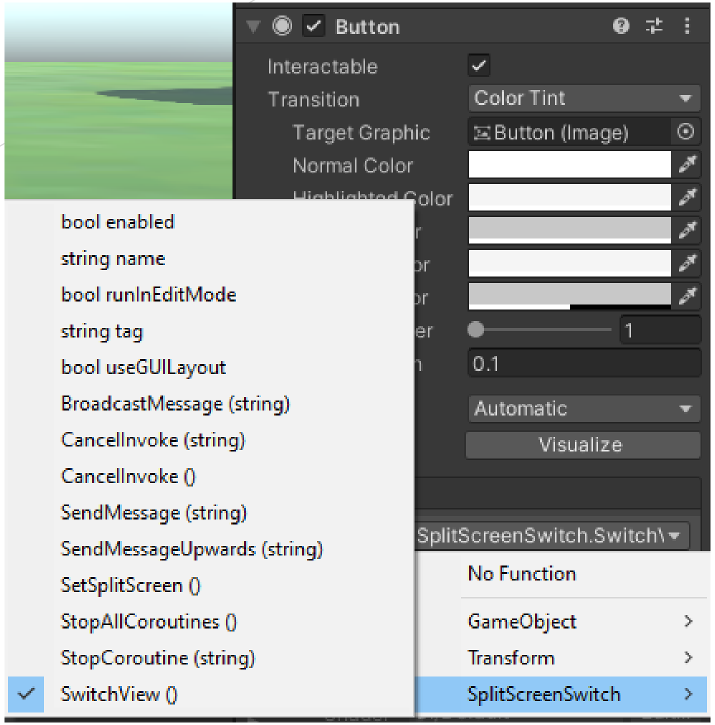
- Assign Button OnClick() event - + button, assign SwitchSplitScreen gameobject, select SplitScreenSwitch.SwitchView



Button

Interactable ✓
Transition: Color Tint
Target Graphic: Button (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic
Visualize

On Click ()
Runtime C | SplitScreenSwitch.SwitchV
Switch



bool enabled
string name
bool runInEditMode
string tag
bool useGUILayout
BroadcastMessage (string)
CancelInvoke (string)
CancelInvoke ()
SendMessage (string)
SendMessageUpwards (string)
SetSplitScreen ()
StopAllCoroutines ()
StopCoroutine (string)
✓ SwitchView ()

No Function
GameObject >
Transform >
SplitScreenSwitch >

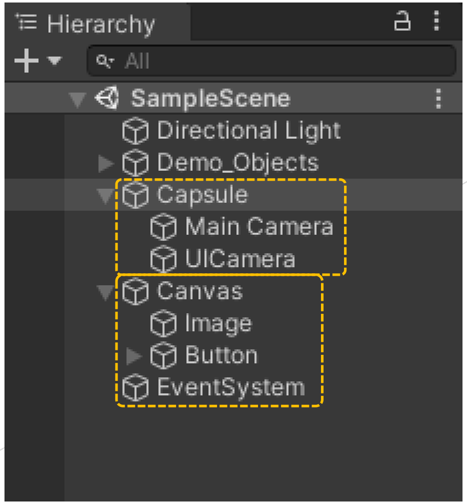
이미지 출처 :Unity

GUI Class

Property	Function
GUI.Label	Create a Label on the GUI Layer
GUI.Button	Create a Button on the GUI Layer
GUI.Box	Create a Box on the GUI Layer
GUI.Toggle	Create a ToggleBox on the GUI Layer
GUI.TextArea	Create a TextArea on the GUI Layer
GUI.TextField	Create a TextField on the GUI Layer
GUI.Toolbar	Create a Toolbar on the GUI Layer
GUI.VerticalSlider	Create a VerticalSlider
GUI.skin	define GUI Style
GUI.changed	if control is changed, it returns true

UI Camera Setting in Unity

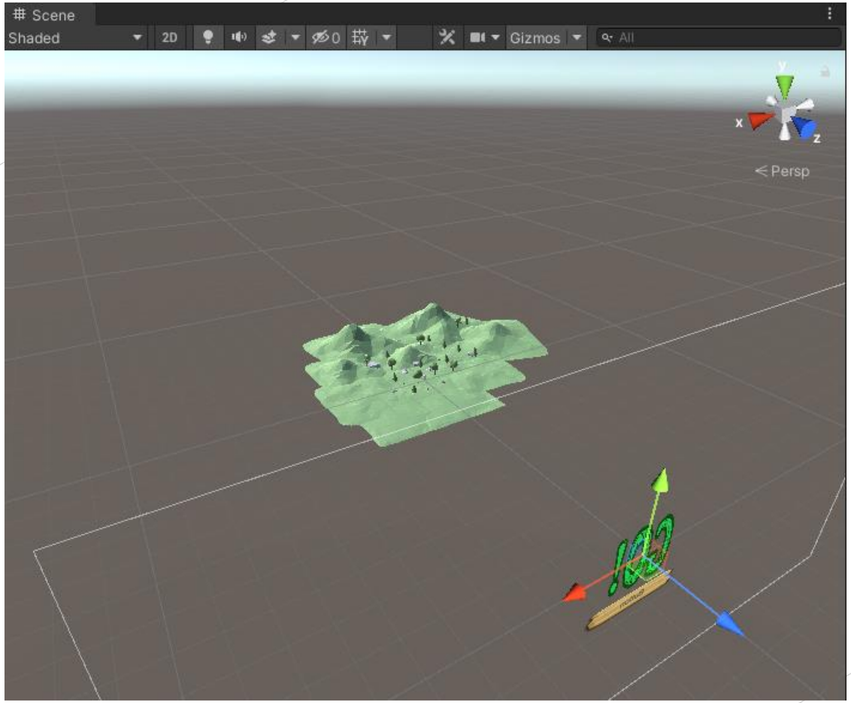
- Create a scene and UI canvas with image and button.
- Add a capsule (add PlayerMove C# script) contains Main Camera & UI Camera



Hierarchy

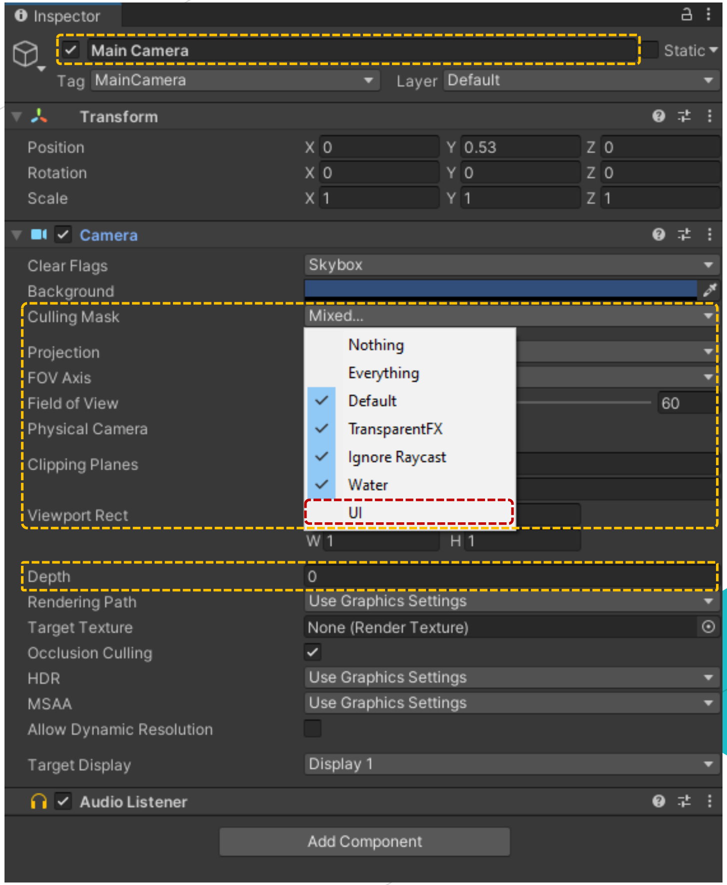
- SampleScene
  - Directional Light
  - Demo_Objects
  - Capsule
    - Main Camera
    - UICamera
  - Canvas
    - Image
    - Button
  - EventSystem

이미지 출처 : Unity



UI Camera Setting in Unity

- Main Camera
  - uncheck UI in Cull Mask, set Depth to 0



Inspector

Main Camera ✓ · Static

Tag: MainCamera · Layer: Default

Transform

	X	Y	Z
Position	0	0.53	0
Rotation	0	0	0
Scale	1	1	1

Camera ✓

Clear Flags	Skybox
Background	
Culling Mask	Mixed...
Projection	
FOV Axis	
Field of View	60
Physical Camera	
Clipping Planes	
Viewport Rect	W 1 H 1
Depth	0
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

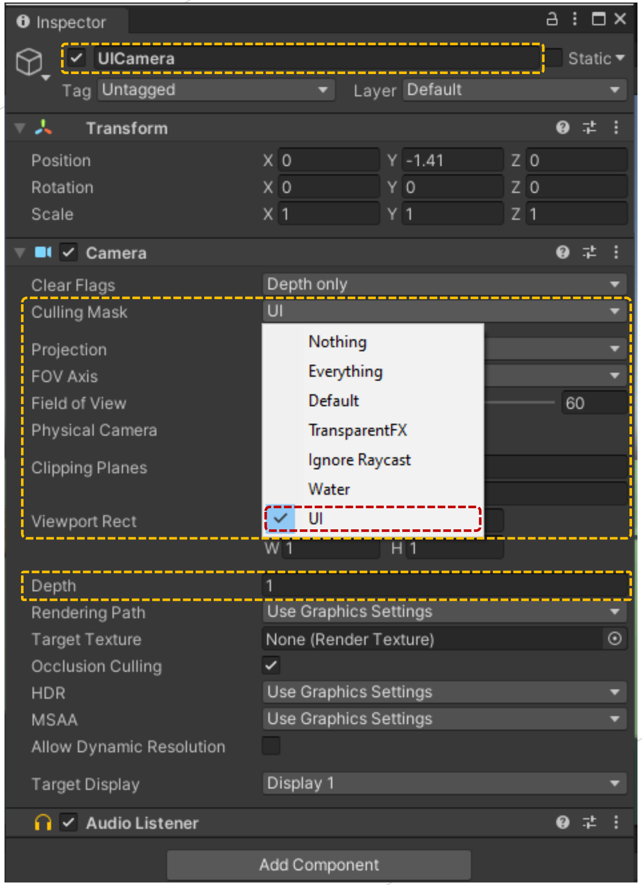
Culling Mask menu: Nothing, Everything, ✓ Default, ✓ TransparentFX, ✓ Ignore Raycast, ✓ Water, UI

Audio Listener ✓

이미지 출처 :Unity

UI Camera Setting in Unity

- UI Camera
  - ➢ select Depth only in Clear Flags, check only UI in Cull Mask, set Depth to 1


Inspector
✓ UICamera Static
Tag Untagged Layer Default

Transform
Position X 0 Y -1.41 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

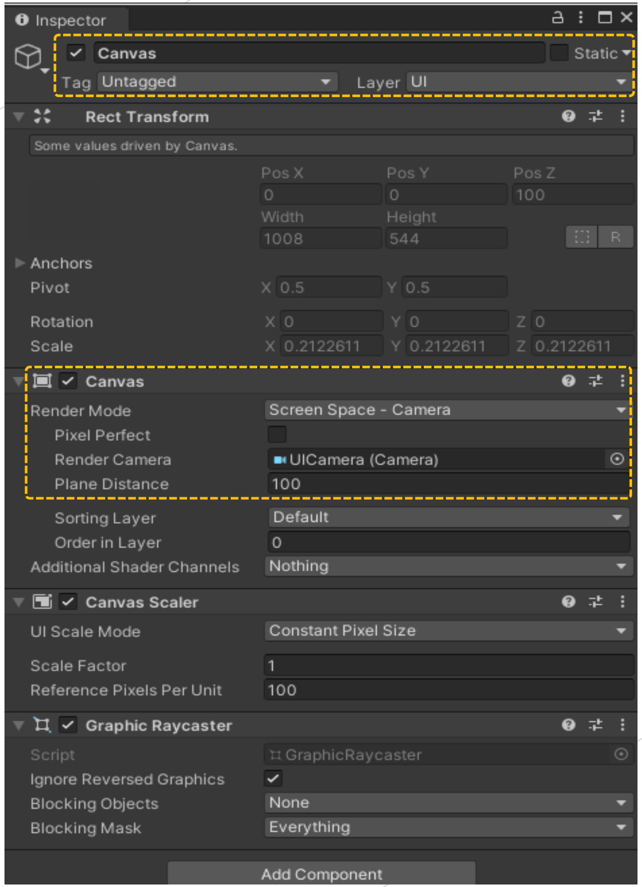
✓ Camera
Clear Flags: Depth only
Culling Mask: UI
Projection
FOV Axis
Field of View: 60
Physical Camera
Clipping Planes
Viewport Rect: W 1 H 1
Depth: 1
Rendering Path: Use Graphics Settings
Target Texture: None (Render Texture)
Occlusion Culling: ✓
HDR: Use Graphics Settings
MSAA: Use Graphics Settings
Allow Dynamic Resolution
Target Display: Display 1

✓ Audio Listener
Add Component

이미지 출처 :Unity

UI Camera Setting in Unity

- Canvas
  - set Layer to UI, set Render Mode to Screen
- Space
  - Camera, set Render Camera to UICamera



Inspector

Canvas — Static
Tag Untagged — Layer UI

Rect Transform
Some values driven by Canvas.

	Pos X	Pos Y	Pos Z
	0	0	100
	Width	Height	
	1008	544	
Anchors			
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 0.2122611	Y 0.2122611	Z 0.2122611

Canvas

Render Mode	Screen Space - Camera
Pixel Perfect	
Render Camera	UICamera (Camera)
Plane Distance	100
Sorting Layer	Default
Order in Layer	0
Additional Shader Channels	Nothing

Canvas Scaler

UI Scale Mode	Constant Pixel Size
Scale Factor	1
Reference Pixels Per Unit	100

Graphic Raycaster

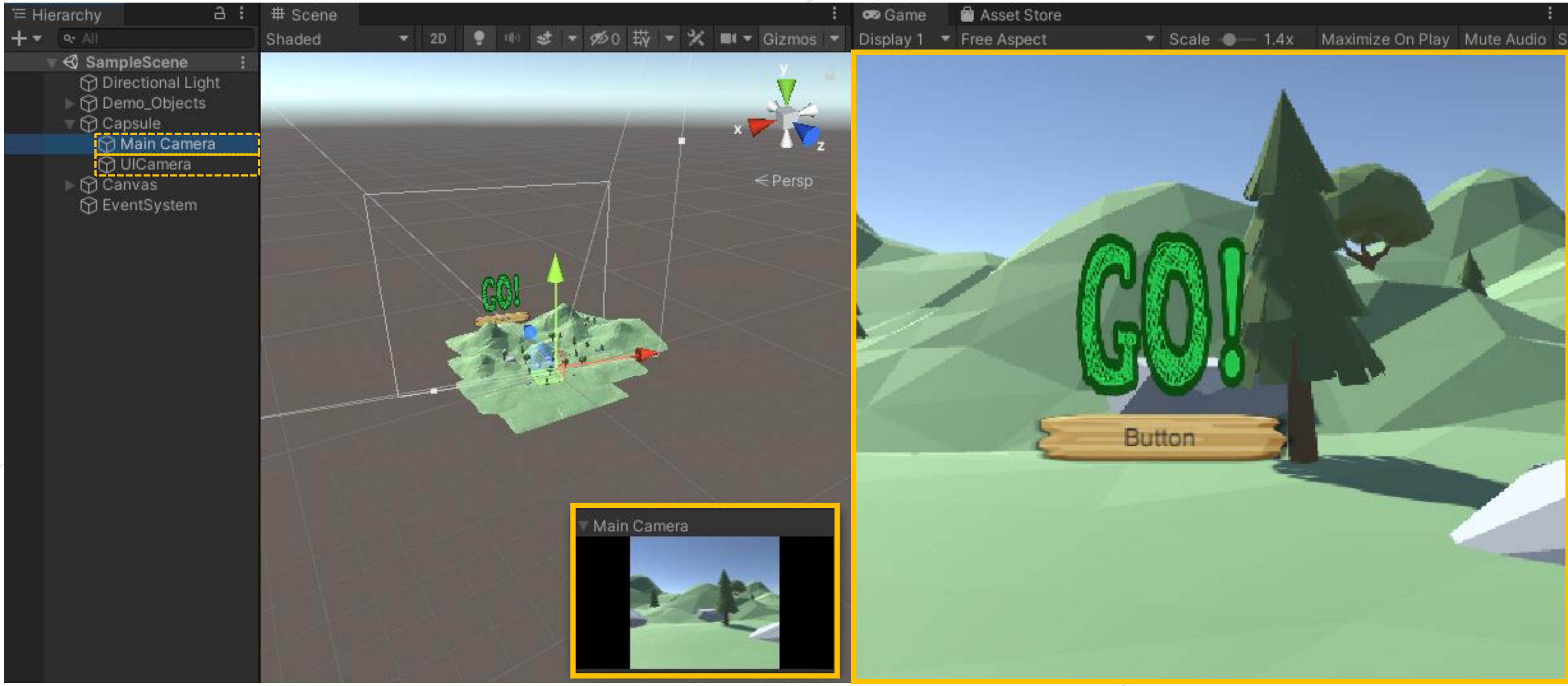
Script	GraphicRaycaster
Ignore Reversed Graphics	✓
Blocking Objects	None
Blocking Mask	Everything

Add Component

이미지 출처 :Unity

UI Camera Setting in Unity

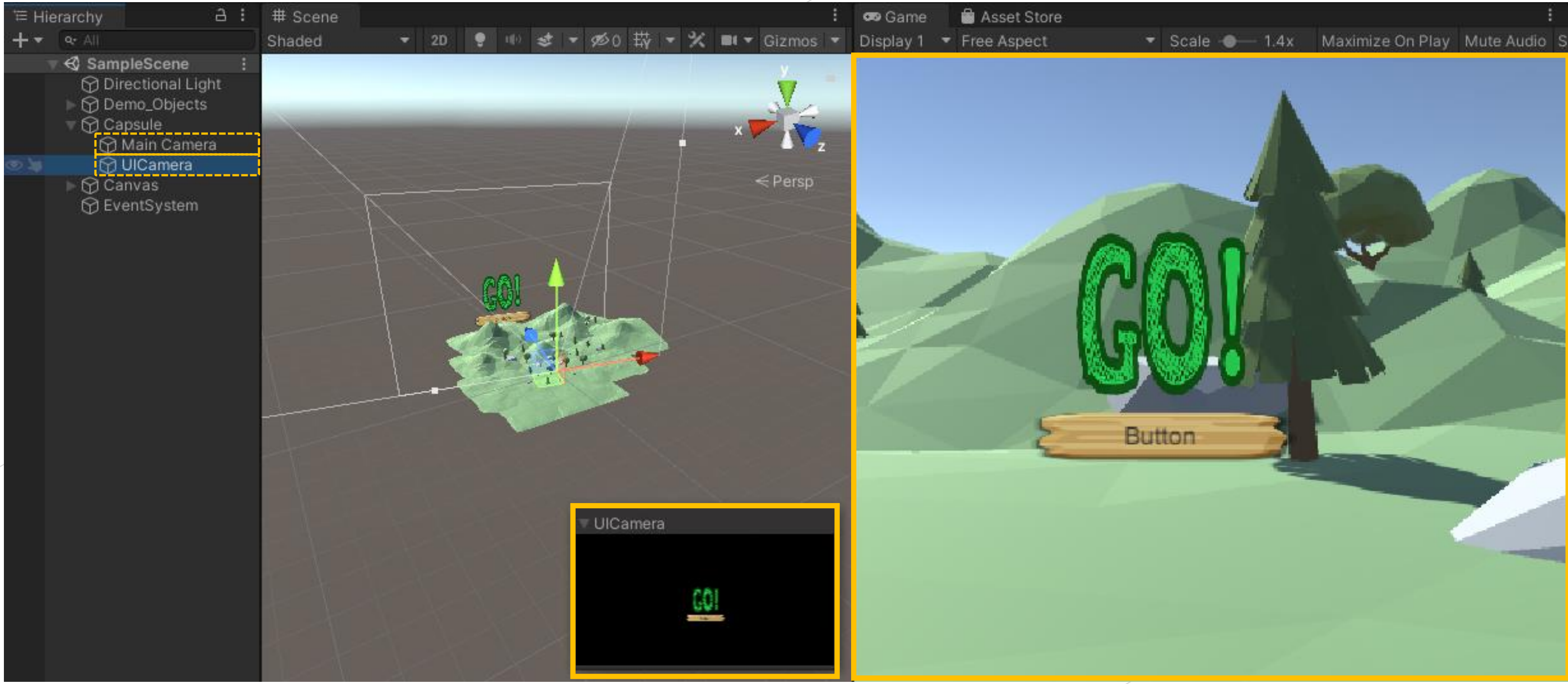
- Main Camera render Scene



이미지 출처 :Unity

UI Camera Setting in Unity

- UICamera render UI only



이미지 출처 :Unity

Reference

- How to Make a Split Screen for Multiplayer in Unity
    - https://www.youtube.com/watch?v=DBHLgrR60F0
- UI Canvas - Unity Official Tutorials
    - https://www.youtube.com/watch?v=OD-p1eMsyrU&t=166s
- Rendering 3D Objects in your UI Using Multiple Cameras
    - https://www.youtube.com/watch?v=tACRIWcCzK8