# **Object Control for Blender Users**



# 1. About Object Control for Blender Users

Due to Asset Store policy, the name has been changed from "Blender-Like Object Control" to "Object Control for Blender Users". Changing the names of already used classes can affect users' projects, so programmatically I still use the BlenderLikeObjectControl or BLOC.

Object Control for Blender Users(OCBU) is a plugin that helps you move objects in Unity's scene view (viewport) similar to how you would in Blender.

Currently, when moving or rotating an object in Unity, you have to click and drag the gizmo in the scene view every time. This is a very inefficient and time-consuming task, especially in the process of creating large-scale scene environments, especially when placing natural objects.

Anyone who has used the Blender probably knows this. In Blender you don't need gizmos to move objects. All you have to do is press the movement-related shortcut keys and move the mouse pointer around to make adjustments.

OCBU provides a system similar to Blender's object control system. And going one step further, It also provides functions that allow you to handle objects more easily in Unity.

Are Unity's default transform tools still available ?

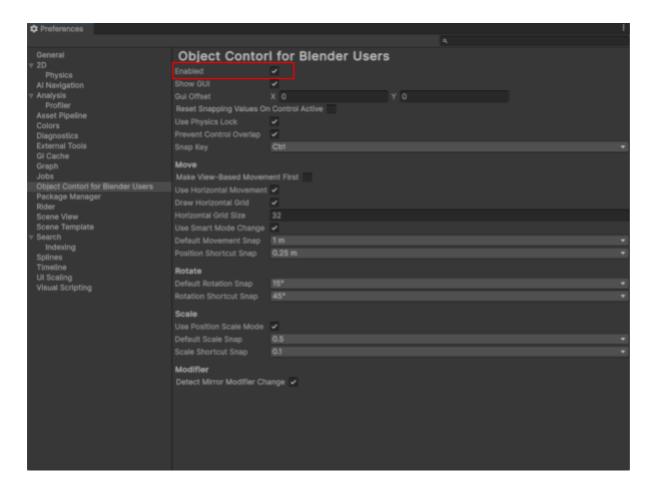
Of course!

You can still use Unity's native transform tools using scene view gizmos while still using OCBU.

## 2. How to use

First, import the package downloaded from the Asset Store.

And after the import is complete, go to **Edit > Preferences > Object Control for Blender Users** tab and change **Enabled to true**. If enabled is false, all functions related to the OCBU will stop immediately.



After that, it's very easy. If you have experience with Blender, you can move objects in the scene view the same way you move objects in Blender.

For a description of each field in Preferences, please refer to <u>9.Preferences</u>.

## 3. What OCBU Supports

OCBU supports the following moves:

- Movement
  - Horizontal Movement
  - View-Based Movement
  - Axis-Based Movement
- Rotation
  - View-Based Rotation
  - o Trackball Rotation
  - Axis-Based Rotation
- Scale
  - o Uniform Scale
  - Axis-Based Scale
  - o Scale Position Mode

And each action can be used with the snap function by pressing the **[Ctrl]** key. Detailed operation methods are explained later.

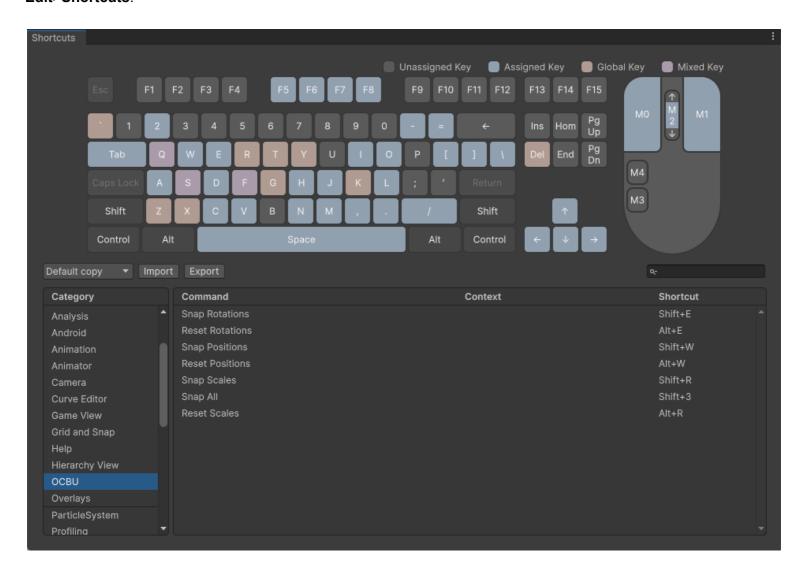
Movement includes a Horizontal Movement function that is not present in Blender. Based on my experience creating game environments, I found it much more convenient to have this function, so I added this to OCBU. Therefore, you can disable this function in Preferences or change the order in which it is activated through the navigation button.

And just like in Blender, you can reset position, rotation, and scale by pressing the **[Alt]** key plus the Move, Rotate, and Scale keys.

Additionally, shortcut keys are provided for snapping to an object's current location. Just press the **[Shift]**, Move, Rotate, and Scale keys.

- Movement
  - Reset Positions [Alt]
  - o Snap Positions [Shift]
- Rotation
  - o Reset Rotations [Alt]
  - Snap Rotations [Shift]
- Scale
  - o Reset Scales [Alt]
  - o Snap Scales [Shift]
- Snap All [Shift + 3]

Since initialization and snap functions are registered in Unity's Shortcuts, you can change the shortcuts in **Edit>Shortcuts**.



I prefer to change the movement, rotation, and scale to **G**, **R**, **S** as in Blender, and also change the initialization and snap functions to **Alt + G / R / S**. However, these shortcut keys are set to Unity's default values of **W**, **E**, **R** when the asset is imported.

#### What is the difference between snapping using [Ctrl] and snapping using [Shift] ?

If you've only read this document, you might be a little confused that these two things exist separately.

Snapping using [Ctrl] is the same as snapping when using Unity's gizmo.

Snapping with [Shift] is a simple shortcut. This simply snaps the position, rotation, and scale to close values.

See **7.**Snapping for a detailed description of snapping.

#### Tips |

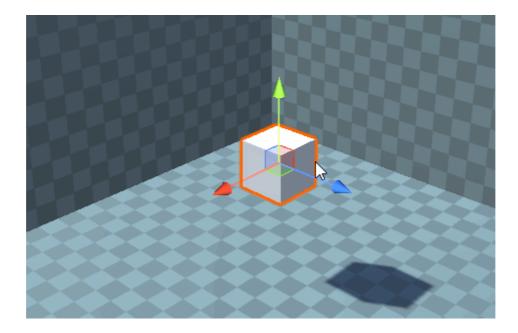
OCBU supports Undo in all operations. Don't be afraid to Undo!

OCBU also supports editing Unity Splines, and support will be added to Probuilder in the future.

## 4. Movement

#### 4-1. Horizontal Movement

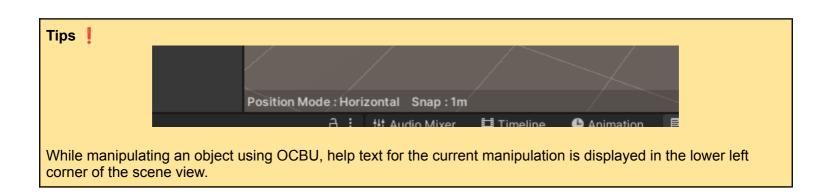
When you press the movement shortcut key, the Horizontal Movement function is activated first. When Horizontal Movement is activated, you can move the object while keeping the Y value constant.



If **ShowGUI** is **true** in Preferences, a grid is created based on the y value of the object. The grid at this time represents the movement interval when snapping. Additionally, the size of the entire grid can be adjusted in Preferences.

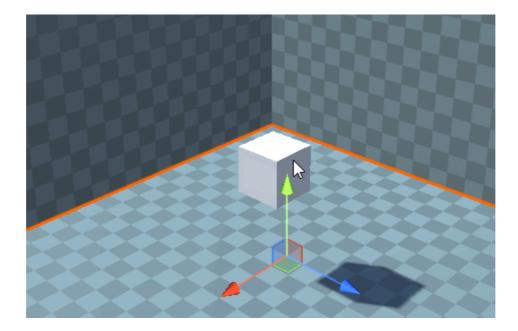
For a description of each field in Preferences, please refer to <u>9.Preferences</u>.

The position where this grid is created depends on the PivotMode of Tools. If the PivotMode is Local, this position is based on the position of the main object (usually the last selected object) among the currently selected objects. If the PivotMode is Center, the position is based on the center of the bounds of all selected objects.



### 4-2. View-Based Movement

If you press the movement shortcut key again while Horizontal Movement is activated, View-Based Movement will be activated.



View-Based Movement is the same function as when moving an object after pressing **[G] in Blender**. It moves objects based on the viewpoint from which objects are currently viewed in the scene view (viewport).

When View-Based Movement is activated, if you press the movement shortcut key again, the Horizontal Movement is activated.

If you want this View-Based Movement to work before the Horizontal Movement, set **Make View-Based Movement First** in Preferences to **true**.

For a description of each field in Preferences, please refer to <u>9.Preferences</u>.

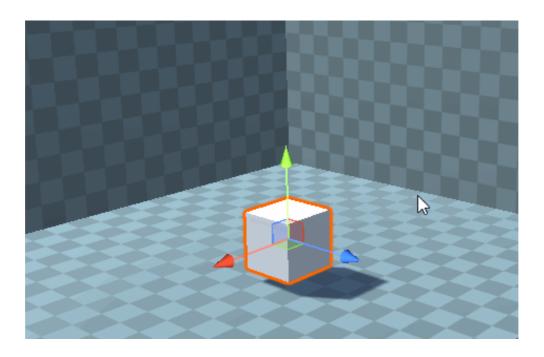
#### 4-3. Axis-Based Movement

Axis-Based Movement is a function that moves objects along each of the X, Y, and Z axes. This function can be used in two ways: using the mouse wheel and using keyboard input.

This is exactly the same as Blender's axis locking feature.

#### With Mouse wheel

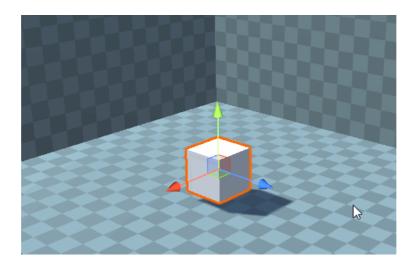
This function can be activated by pressing the mouse wheel while the OCBU's movement control is activated.



When you press the mouse wheel while using the movement controls, gizmos are drawn along each axis of the main object or world. You can also select which axis to select by moving the mouse pointer.

### • With Keyboard

When OCBU's movement control is activated and you press the **[X]**, **[Y]**, or **[Z]** keys on the keyboard, movement is locked to the axis corresponding to each key. Then, you will only be able to move along that axis, like when you select an axis with the mouse wheel.

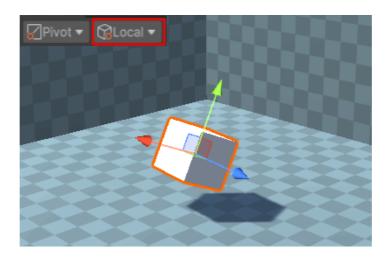


After selecting an axis, you can move objects using the mouse or enter the distance objects will move using the keyboard. Objects cannot be moved with the mouse while entering numbers with the keyboard.

If showGUI in Preferences is true, you can see the number currently being entered in the text displayed at the bottom left of the scene view.

After entering numbers, press the Enter (or Return or Space) button to apply the position.

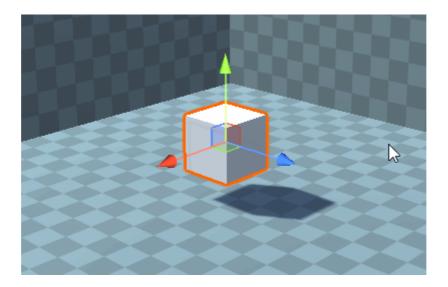
If **Tools.pivotRotation** is Global, each axis is based on the world. Conversely, if Tools.pivorRotation is Local, it is based on the rotation of each object. Even if multiple objects are selected, each object moves according to its local axis.



## 5. Rotation

### 5-1. View-Based Rotation

When you press the rotation hotkey, the View-Based Rotation control will be activated first. In this control, objects are rotated around the direction of the scene view facing the object.

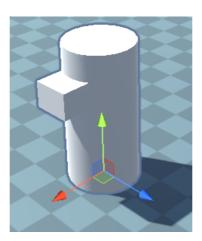


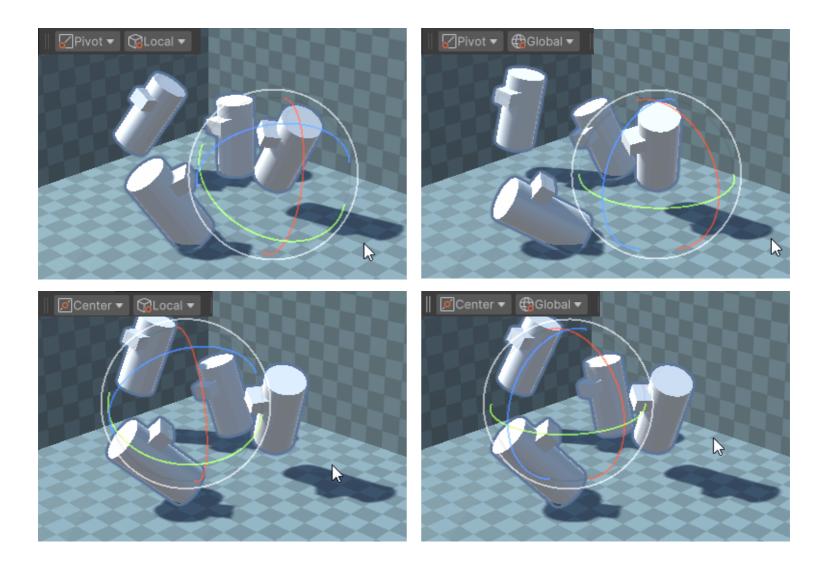
This functionality is influenced by Tools.pivotMode and Tools.pivotRotation.

**Tools.pivotMode** determines whether the center of rotation will be the center of all selected objects or the pivot position. And **Tools.pivotRotation** determines whether each object rotates all around a point or on its own axis.

You can easily see the difference in the GIF image below.

Keep in mind that the objects used in the examples below have its pivot at the bottom.





You may find the use of PivotMode and RotationPivot a bit confusing since they are not completely the same as Blender's pivot system.

To put it simply, PivotMode's Pivot and Center are to find the **center point** for rotation, and PivotRotation's Local and Global are to change whether to rotate alone or together.

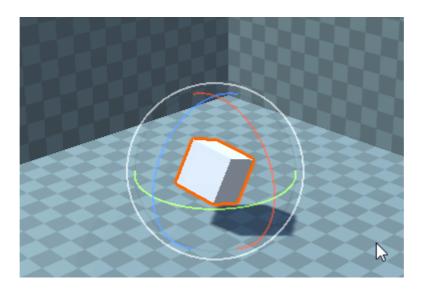
### Whose pivot is it ?

The pivot when multiple objects are selected is the position of **Selection.activeTransform**.

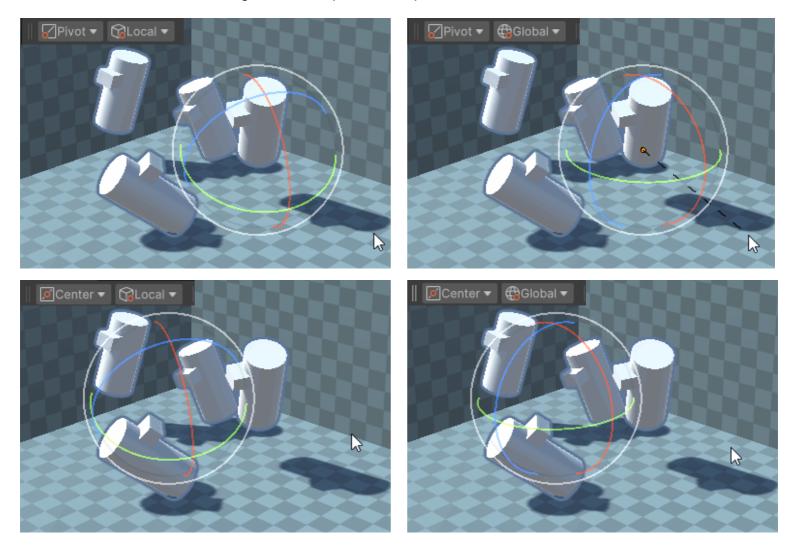
This is usually the last object selected, but when compilation occurs it changes to be the first object selected in the Hierarchy window.

## 5-2. Trackball Rotation

In View-Based Rotation, pressing the rotation hotkey once more changes the rotation mode to Trackball Rotation. In this control, the object rotates horizontally and vertically according to changes in the position of the mouse pointer.



This functionality is also affected by Tools.pivotMode and Tools.pivotRotation. The logic for the pivot system is the same. Please refer to the GIF image below for specific examples.

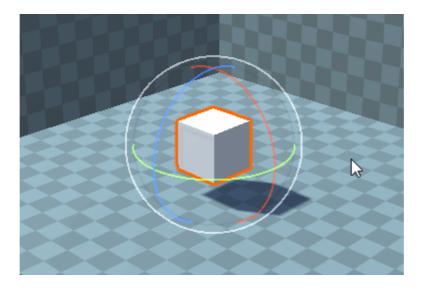


#### 5-3. Axis-Based Rotation

This is a function that rotates objects along a specified axis. The method to activate this function is the same as for the Axis-Based Movement, and it is the same for the axis-based scale control.

#### With Mouse Wheel

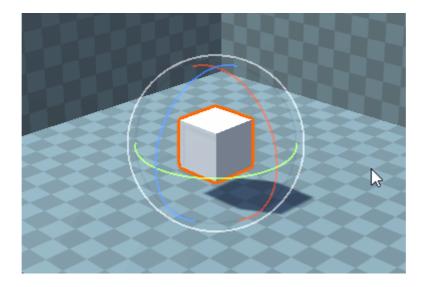
This function can be activated by pressing the mouse wheel while the OCBU's rotation control is activated.



When you press the mouse wheel while using the rotation controls, gizmos are drawn along each axis of the main object or world. You can also select which axis to select by moving the mouse pointer.

#### With Keyboard

When OCBU's rotation control is activated and you press the [X], [Y], or [Z] keys on the keyboard, rotation is locked to the axis corresponding to each key. Then, you will only be able to rotate along that axis, like when you select an axis with the mouse wheel.



After selecting an axis, you can rotate objects using the mouse or enter the angle you want objects to rotate by using the keyboard. You cannot adjust the angle with the mouse while entering numbers with the keyboard.

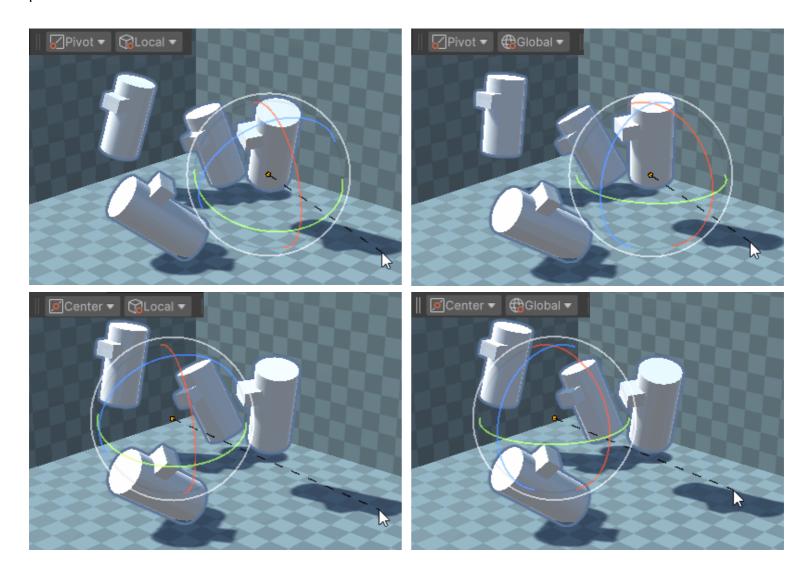
If showGUI in Preferences is true, you can see the number currently being entered in the text displayed at the bottom left of the scene view.



After entering numbers, press the Enter (or Return or Space) button to apply the rotation.

Like other rotation functions, this function is also affected by **Tools.pivotMode** and **Tools.pivotRotation**.

In the example below, the object is rotated around the X axis. Look at the difference between pivotMode and pivotRotation.

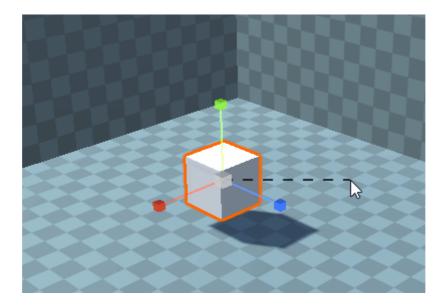


## 6.Scale

### 6-1. Uniform Scale

When you press the scale shortcut key, the Uniform Scale control is activated first. In this control, scales of objects increase or decrease depending on the distance between the object and the mouse pointer.

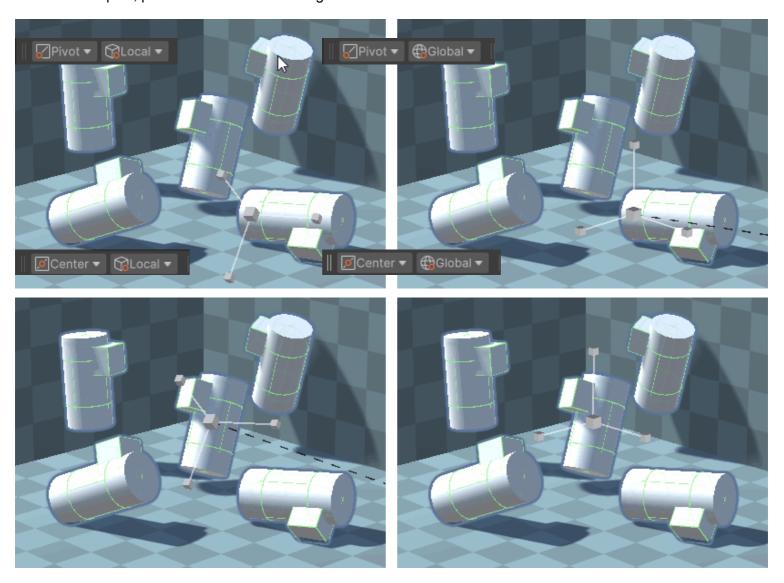
As the name suggests, the change value is applied equally to each axis, so objects maintain their shape and only change their own size.



Even when using the Uniform Scale control, you can adjust the scales of objects not only with the mouse but also **with keyboard input**. The size entered with the keyboard is multiplied by the scales of the objects.

This function is also influenced by both **Tools.pivotMode** and **Tools.pivotRotation**.

When Tools.pivotRotation is set to Global, it manipulates the scales and positions of objects to either converge or diverge from a single point. Meanwhile, Tools.pivotMode is responsible for determining this central point. For detailed examples, please refer to the GIF images below.



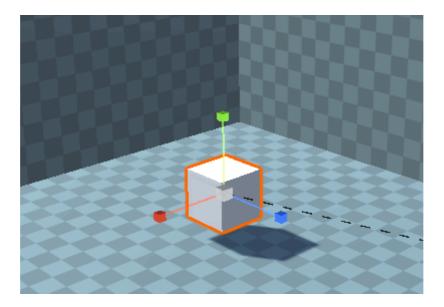
The way the pivot system works is the same as it did for rotation. The pivotMode determines the position of the center point, and the pivotRotation determines whether they move independently or together.

### 6-2. Axis-Based Scale

This is a function that scales objects along a specified axis. The method to activate this function is the same as for the Axis-Based Movement.

#### • With Mouse Wheel

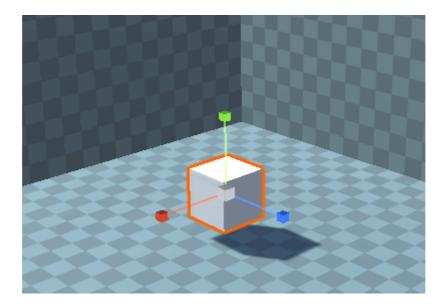
This function can be activated by pressing the mouse wheel while the OCBU's scale control is activated.



When you press the mouse wheel while using the scale controls, gizmos are drawn along each axis of the main object or world. You can also select which axis to select by moving the mouse pointer.

### With Keyboard

When OCBU's scale control is activated and you press the **[X]**, **[Y]**, or **[Z]** keys on the keyboard, rotation is locked to the axis corresponding to each key. Then, you will only be able to rotate along that axis, like when you select an axis with the mouse wheel.



After selecting an axis, you can scale objects using the mouse or enter the angle you want the object to scale by using the keyboard. You cannot adjust the scale with the mouse while entering numbers with the keyboard.

If showGUI in Preferences is true, you can see the number currently being entered in the text displayed at the bottom left of the scene view.

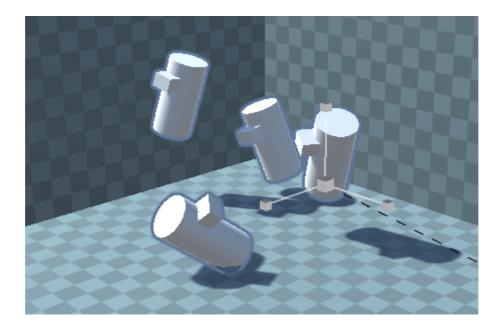


After entering numbers, press the Enter (or Return or Space) button to apply the scale.

### 6-3. Position Scale Mode

Position Scale Mode is a special mode that updates only the positions of objects without affecting their scale.

You can turn this feature on by pressing the scale shortcut again when the OCBU's scale control is activated.



You can check whether this feature is enabled in the GUI at the bottom left of the scene view.



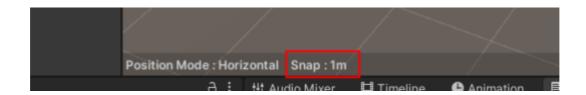
In the Uniform Scale control, objects can be gathered at one point or spread out, and in the Axis-Based Scale control, objects can be moved closer or farther away in the direction of the selected axis.

# 7. Snapping

## 7-1. Snap during control

Snapping is a function that changes the value to fit a specific unit when controlling objects. This function can be used by pressing the **[Ctrl]** key when controlling objects. You may be familiar with this as it can also be used to control objects using Unity's default gizmos in scene view.

When you use the snapping feature of OCBU, snapping-related information is displayed in the bottom left corner of the scene view.

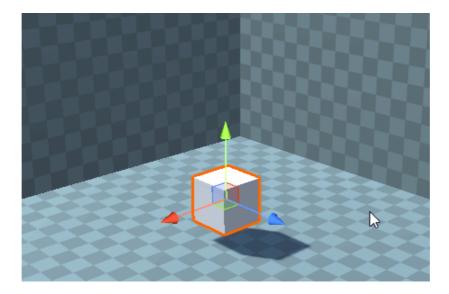


Snapping is available for all operations such as movement, rotation, and scale. When each control of OCBU is activated, press the **[Ctrl]** key.

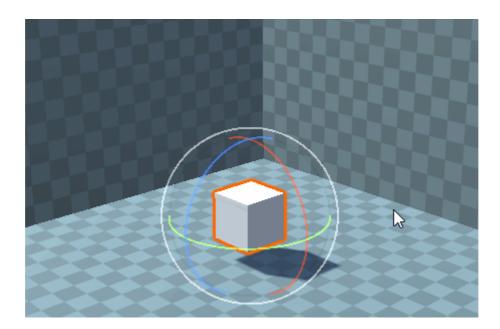
You can also change the unit to be snapped by holding down the [Ctrl] key and manipulating the mouse wheel.

In movement control, snapping is possible in increments of **0.25m**, **0.5m**, **1m**, **5m**, **and 10m**. The default snap unit is 1m. You can change the default snap unit in OCBU's Preferences.

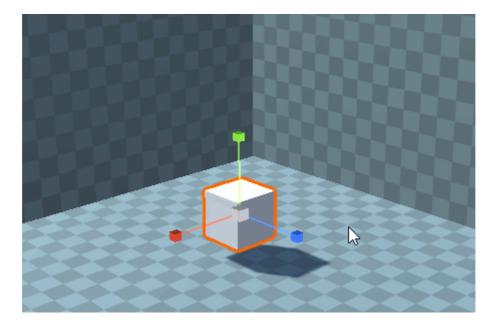
And in the Horizontal Movement control, when the snap unit changes, the size of the grid changes accordingly.



In rotation control, snapping is possible in increments of 1°, 5°, 10°, 15°, 30°, 45°, and 90°. The default snap unit is 15°. You can change the default snap unit for rotation in OCBU's Preferences.



In scale control, snapping increments of **0.25**, **0.5**, **1**, **5**, **and 10** is possible. The default snap unit is 0.5. You can change the default snap unit for the scale in OCBU's Preferences.



### Can I maintain snap units ?

Snap units are reset when an operation is canceled or changed to a new operation state. However, if you change **Reset Snapping Values On Control Active** to false in the OCBU's Preferences, the last used snapping unit will remain.

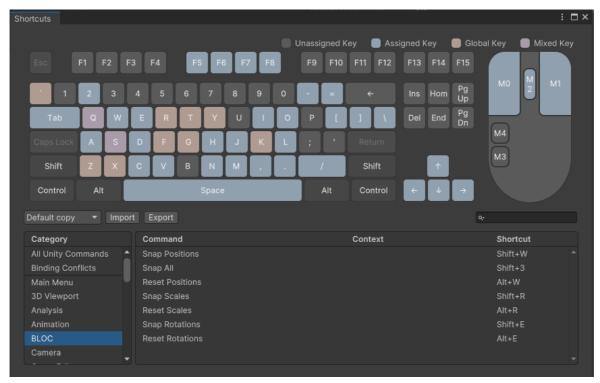
For a description of each field in Preferences, please refer to <u>9.Preferences</u>

### 7-2. Snap with shortcut

You can use the snapping function even when you are not controlling objects.

You may want to place an object in a position of 1m, or keep it rotated at an angle of 90 degrees. In that case, you can use the snapping function using the shortcut prepared in advance.

These are shortcuts that the blender does not have, but I found their usefulness during my personal work, so I added them to this asset.



This snapping shortcut is registered in Unity's Shortcuts. You can change these shortcuts if you want. I use the same as the blender, switching to **G**, **R**, **S**. What you do is up to you.

And you can set the unit of each snap function in Preferences. Please see the <u>9.Preferences</u> for a detailed description.

#### Why doesn't snap shortcuts work after switching to G, R, S ?

This is probably because it conflicts with other shortcuts.

For example, because Unity's Splines package has a shortcut on [Shift+S], the scale snap registered on [Shift+S] is called before you change this shortcut.

If other shortcuts do not work, check if there is a shortcut crash.

## 8. Modifiers

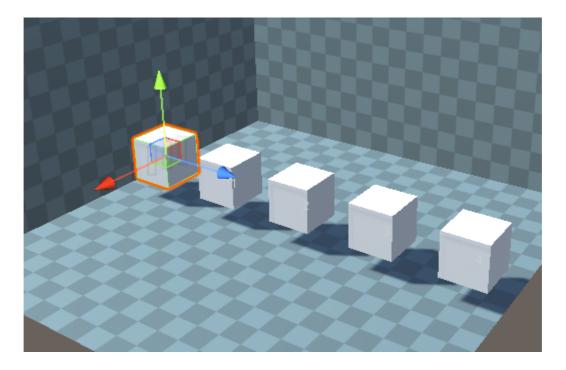
OCBU provides useful modifiers when placing objects. These modifiers aren't exactly the same as Blender's, but it's useful for things like simple placement of objects.

These modifiers are editor-only components. Therefore, please apply the modifiers before building your game, or set applyOnBuild to true. Although these modifiers will probably work in the build and runtime even if you don't apply them, please keep in mind that these modifiers were not designed for runtime.

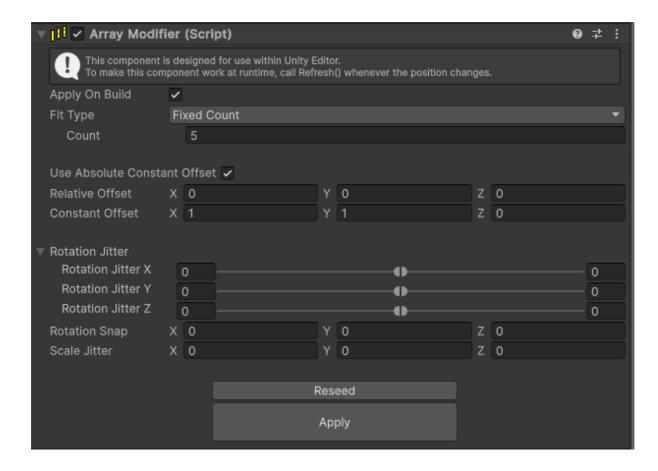
## 8-1. Array Modifier

The Array Modifier repeats and lines up objects. Copied objects are listed based on the local axis of the original object.

To use this component, add the ArrayModifier component to the object you want to clone.



This component is similar to what is used in Blender, but there are some differences. Let's go into more detail by explaining the inspector's fields below.



**Apply On Build**: Apply this Modifier during the build process. For more information, please refer to the description of the Apply button below.

Fit Type : Determine the criteria for adjusting the number of objects.

- **Fixed Count**: Directly specify the number of object arrays. the Count field will appear in the inspector, and objects will be created equal to the number of values entered here.
- **Fixed Length**: The number of objects is calculated to fit the specified length. the Length field will appear in the Inspector, and objects will be created to match the length of the value entered.

**Use Absolute Constant Offset**: Use Constant Offset as the unscaled length. Default value is false because Blender's Constant Offset is also scale relative.

**Relative Offset**: Offset relative to the object's rendering bounds. For example, if this value is (1, 0, 0), a copy will be created immediately to the right of the object. If this value is (2, 0, 0), a copy is created after skipping once the width of the object. This is the same as the Relative Offset in Blender's Array Modifier.

**Constant Offset**: Offset based on distance. If this value is (1, 0, 0), a copy is created at an interval of 1m to the right. This is the same as Blender's Array Modifier's Constant Offset.

**Rotation Jitter**: Adds a random rotation to the copied instance. This is useful when you want to eliminate repetitiveness in a series of instances.

**Rotation Snap**: Apply snapping functionality to rotation. If this value is non-zero, snapping is applied in the entered units. Assuming you're placing square pillars, the rotation of each pillar will have some randomness, but you'll want it to snap to an angle of 90 degrees. In that case, just set the y value of Rotation Snap to 90.

**Scale Jitter**: Adds randomness to the size of objects.

**[Reseed]**: Reinitialize the randomness for the Rotation Jitter and the Scale Jitter. You can also enter values directly into ArrayModifier.seed via script.

**[Apply]**: Delete the current modifier and realize the copied instances in the hierarchy window. Before Apply, the hideFlags of copied instances is set to HideAndDontSave, but after Apply it changes to None.

If you build without applying modifiers, all modifiers will execute the task of creating each array in OnEnable.

#### Why is there no Fit Curve feature ?

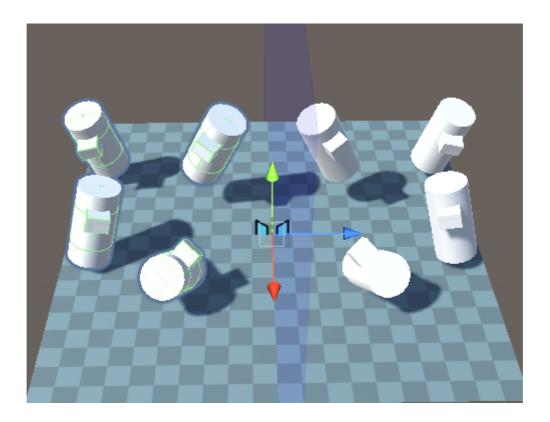
The counterpart to Blender's Curve is Unity's Splines. And Unity's Splines package already has an even better component called **<Spline Instantiate>**.

Therefore, if you want to place an object along a spline, use the Spline Instantiate component.

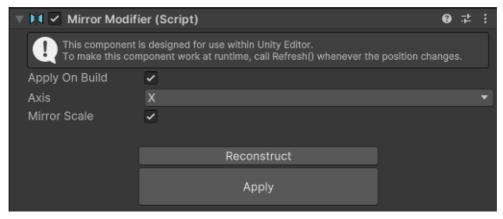
## 8-2. Mirror Modifier

Mirror Modifier is a component that duplicates child objects and then mirrors them on each axis.

To use this component, add a MirrorModifier component to an empty object, and then add the object you want to mirror below it.



This component is similar to what is used in Blender, but there are some differences. Let's go into more detail by explaining the inspector's fields below.



**Apply On Build**: Apply this Modifier during the build process. For more information, please refer to the description of the Apply button below.

**Axis**: This is the axis along which the objects will be mirrored. Because this field is an Enum field using Flag, it can be symmetrical along multiple axes at once.

**Mirror Scale**: Executes mirror by inverting the scale of duplicated objects. If this value is false, only the positions are mirrored without scale change.

**[Reconstruct]**: Delete all mirrored objects and then reconstruct them.

**[Apply]**: Delete the current modifier and realize the copied instances in the hierarchy window. Before Apply, the hideFlags of copied instances is set to HideAndDontSave, but after Apply it changes to None.

Mirrored objects are created with a root object corresponding to each axis.



If you build without applying modifiers, all modifiers will execute the task of creating mirrored object in OnEnable.

#### What is the difference between Refresh and Reconstruct ?

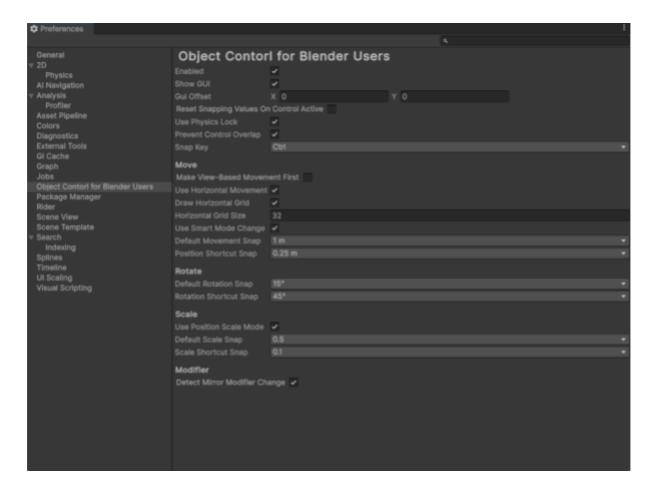
Reconstruct is a larger operation that includes Refresh.

MirrorModifier calls Refresh whenever there are changes to child objects. However, for optimization, Refresh does not track all changes. As a result, there may be cases where the symmetry is not updated.

Run Reconstruct if symmetry is not updated to changes to child objects.

### 9. Preferences

You can change the settings of OCBU in **Edit>Preferences>Object Control for Blender Users**.



#### • Common

**Enabled**: Enables the OCBU feature. If this value is false, all OCBU features are disabled.

**ShowGUI**: Activates the text GUI displayed at the bottom left of the scene view.

**GUI Offset**: An offset added to where the GUI is displayed. Since version 2023 of Unity, it is possible to create a toolbar at the bottom of the scene view, which causes the GUI to be obscured, so you can adjust this option to prevent the GUI from being obscured. Contrary to intuition, the Y value must be negative for the GUI to move upward.

**Reset Snapping Values On Control Active**: When functions such as movement, rotation, or scale are activated, reset the units for snapping. If this value is false, the last snapping value for each tool last used will be retained.

**Use Physics Lock**: Make objects kinematic when controls are active. This feature only works during play, and only happens for Rigidbodies with isKinematic set to false. And it also works automatically for child objects of the selected object.

**Prevent Control Overlap**: If a parent object and a child object are selected together, the control is activated only for the parent object.

**Snap Key**: Decide which modifier key to use when snapping. Windows users will not need this option since both Blender and Unity use the Ctrl key, but Mac users will find this option useful because Unity's snap modifier key is Command and Blender's snap modifier key is Ctrl.

#### Movement

**Make View-Based Movement First**: When movement control is activated, the View-Based Movement function is activated first.

**Use Horizontal Movement**: Enables the use of the Horizontal Movement in movement controls. If you want the exact same control method as Blender, change this value to false. For a detailed explanation of Horizontal Movement, please refer to 4-1. Horizontal Movement.

Draw Horizontal Grid: Draws a grid of planes in which objects move when the Horizontal Movement is activated.

**Horizontal Grid Size**: This is the number of cells on one side of the grid gizmo for the Horizontal Movement. Increasing this value increases the size of the grid displayed in the Scene View. The size of each cell depends on the snapping value, not this value.

**Use Smart Mode Change**: If the scene view's camera projection is set to Orthographic, and the camera's position is also aligned on the axis, View-Based Movement is automatically activated first even if Horizontal Movement is set first. This option is very useful for things like editing the avatar of a 3D model.

**Default Movement Snap**: This is the snapping default for movement controls. Once the snapping value is reset, this value will be used.

**Position Shortcut Snap**: This is the snap value when the position snap shortcut key is pressed. For detailed explanation, please refer to 7-2. Snap with shortcut.

#### Rotation

**Default Rotation Snap**: This is the snapping default for rotation controls. Once the snapping value is reset, this value will be used.

**Rotation Shortcut Snap**: This is the snap value when the rotation snap shortcut key is pressed. For detailed explanation, please refer to <u>7-2. Snap with shortcut</u>.

#### Scale

**Use Position Scale Mode**: Enable the Position Scale Mode by pressing the scale shortcut key once more. If you want the exact same control method as Blender, change this value to false. For a detailed explanation of Position Scale Mode, please refer to the 6-3. Position Scale Mode.

**Default Scale Snap**: This is the snapping default for scale controls. Once the snapping value is reset, this value will be used.

**Scale Shortcut Snap**: This is the snap value when the scale snap shortcut key is pressed. For detailed explanation, please refer to <u>7-2. Snap with shortcut</u>.

#### Modifier

**Detect Mirror Modifier Change**: If there is a change in MirrorModifier's child objects, it detects it and updates the MirrorModifier. This usually has little effect on editor performance.

However, editing symmetrical structures with thousands of sub-objects may slow down the performance. Set this value to false, and press the **[Reconstruct]** button in the MirrorModifier when you want to update your changes.

# 10. Contact

If you find a bug in this asset, or if you have any questions, please send me an email at <a href="mailto:olivecrow.report@gmail.com">olivecrow.report@gmail.com</a> . I'll get back to you as soon as possible.