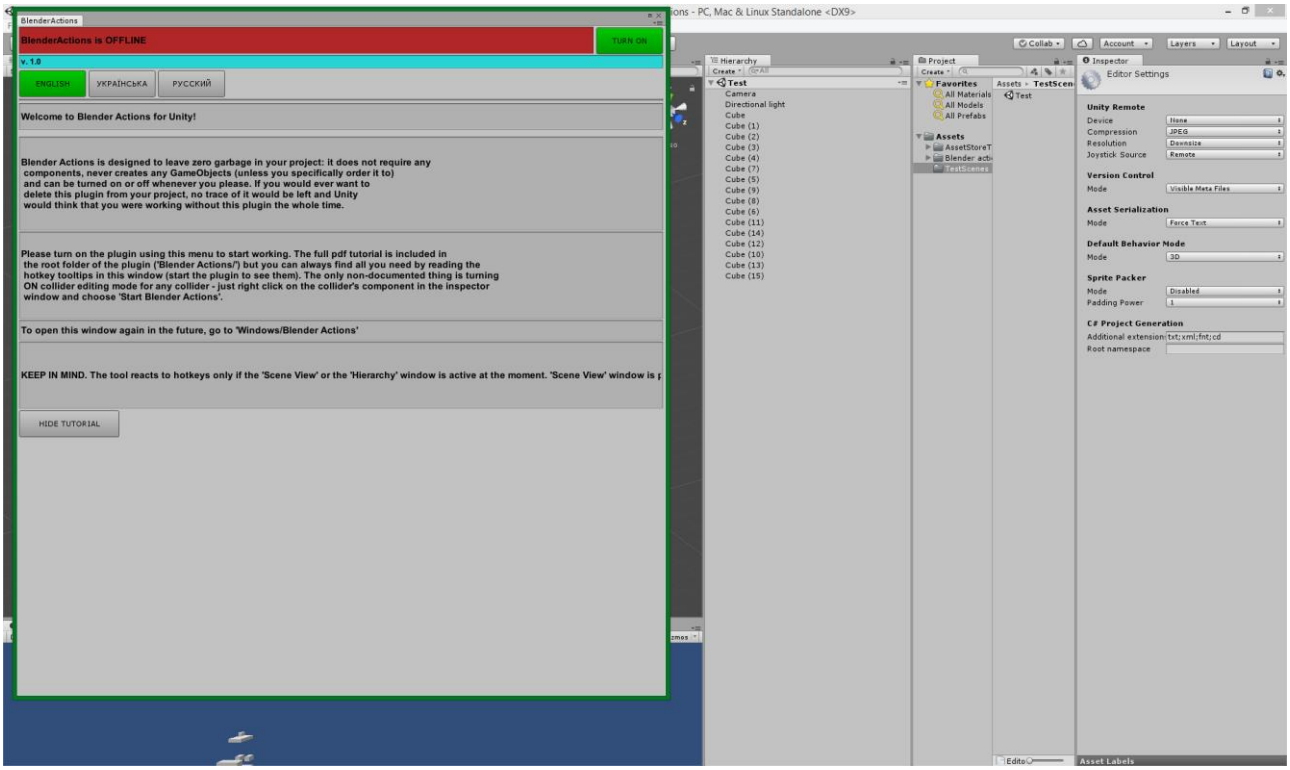
Installation and setup

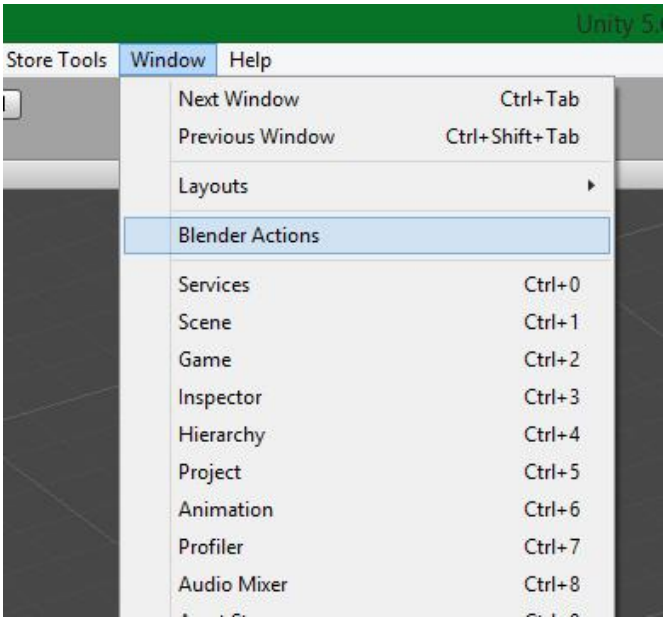
This section covers the basics of injecting the plugin into your Unity project and activating it via the button in the options window. If you have already done this and just want to learn how to make your first steps – jump to the next section “Using transformation actions”.

Blender Actions is a hotkey-based gizmo plugin for Unity. Under the hood, the plugin is a set of hotkey-driven commands that execute certain actions on models in the Unity Editor and/or control the Editor’s Scene View Camera. Because of its straightforward nature Blender Actions can be turned “ON” or “OFF” at any point in time and even completely deleted from project without leaving any debris or causing harm.

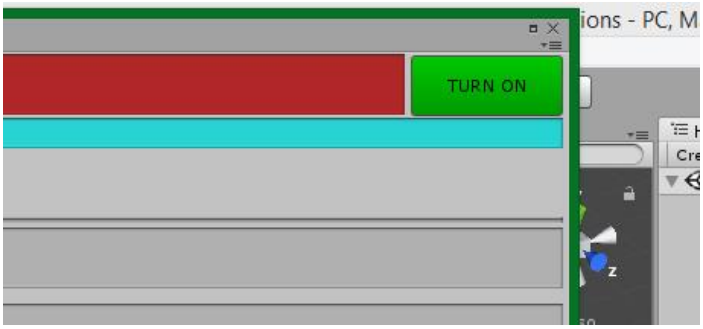
To add the plugin into your project, simply copy the “Blender Actions” folder with the source code files anywhere inside the “Assets” folder (does not need to be in the root of the “Assets” folder). Once you open your project in Unity, you will be greeted by the tool’s options window.



In the unlikely event of the window not showing, please open it using a top Unity Editor menu: Window->Blender Actions:



By default the plugin is turned “OFF”, to activate it please push the green “TURN ON” button in the top right corner of the window:



Please feel free to turn the plugin “ON” or “OFF” whenever you please – it will not damage anything in your project.

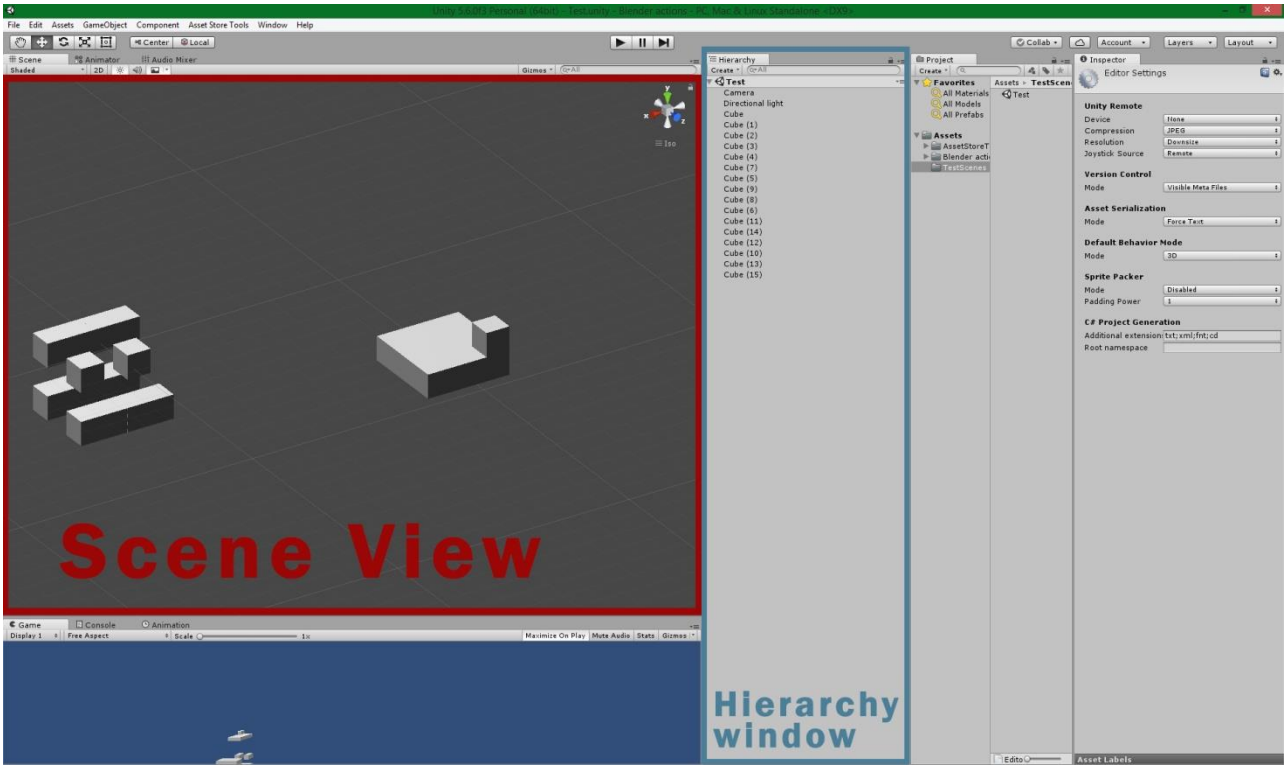
The quick tutorial text you see in the options window is not obligatory, but it is recommended as it contains some useful information.

Once you turn on the plugin, the options window will be populated by various settings and hotkeys. Feel free to adjust them to your personal needs.

Now that the plugin is activated, you can close the options window and start using it in Unity Editor.

Important notice about hotkeys

Before we start, there is one important thing to remember: the plugin will only react to your hotkeys while either Scene View window or Hierarchy window is active.



Unfortunately in Unity there is no clear way to tell which window is active right now. As a rule of thumb, the active window is the one you last clicked on. Just keep in mind, if the plugin does not react to your hotkey commands, this might be due to the fact that neither of the two windows is selected. To fix this, just click either onto the Scene View window, or onto the Hierarchy window. If you don't want to lose your selection inside the game world, you can always click with the "Right Mouse Button" instead of the left mouse button on the Scene View window.

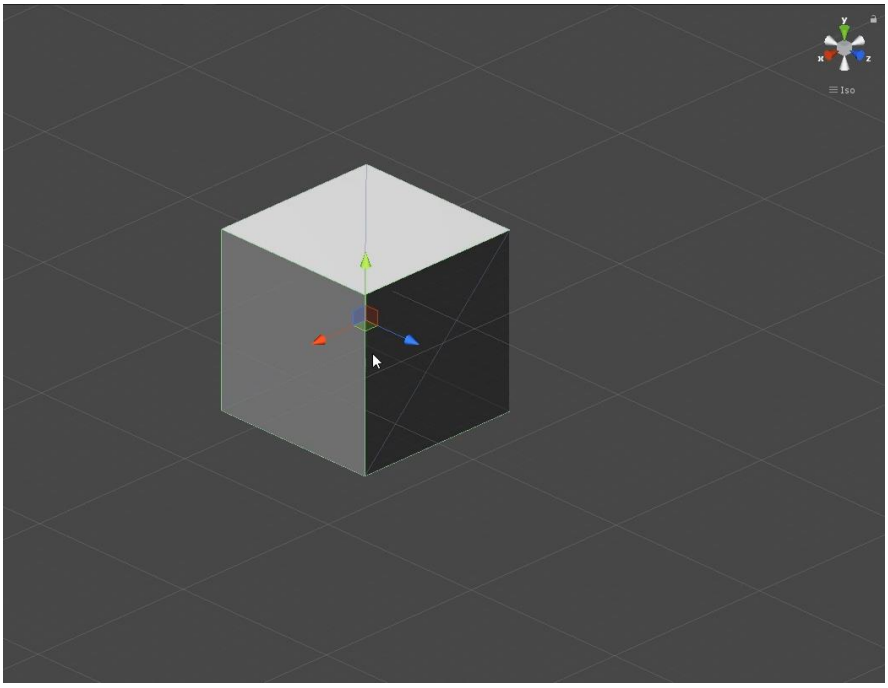
Another thing to keep in mind is that hotkeys are not functioning while the "Right Mouse Button" is held down. This ensures that you can fly around with FPS camera controls in the editor viewport without triggering any of the Blender Actions hotkeys in the process.

Using transformation actions

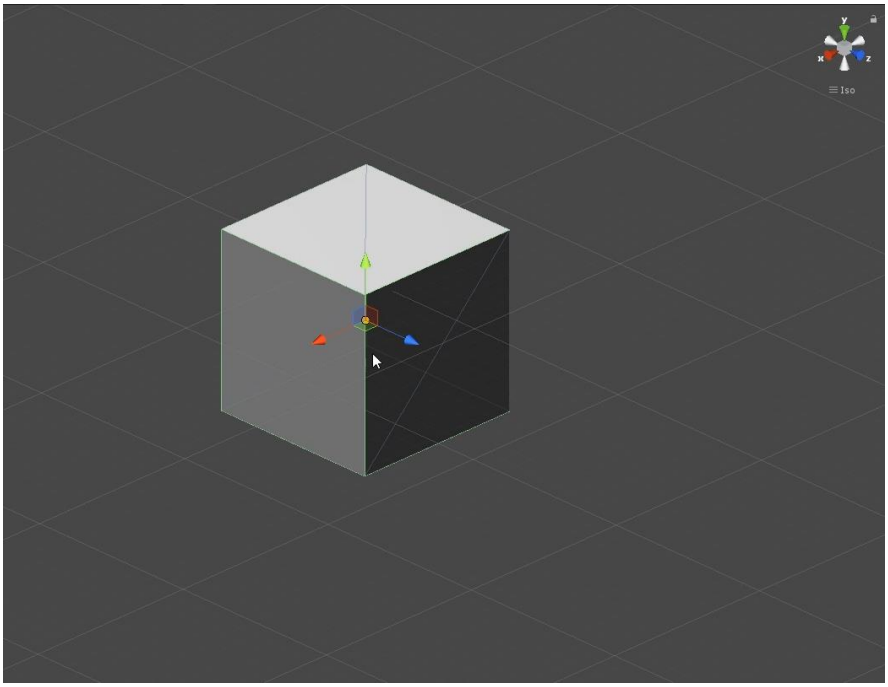
There are three transformation actions: Translation (repositioning/moving the object), Rotation and Scaling. Each of them works in the following way:

1. You select an object or a group of objects you want to work on in your scene.
2. Activate the needed action by pressing and releasing a certain hotkey
3. Use your mouse to change the objects (reposition/rotate/scale them)
4. You either confirm the result by pressing “Left Mouse Button” or cancel it by pressing “Right Mouse Button”.

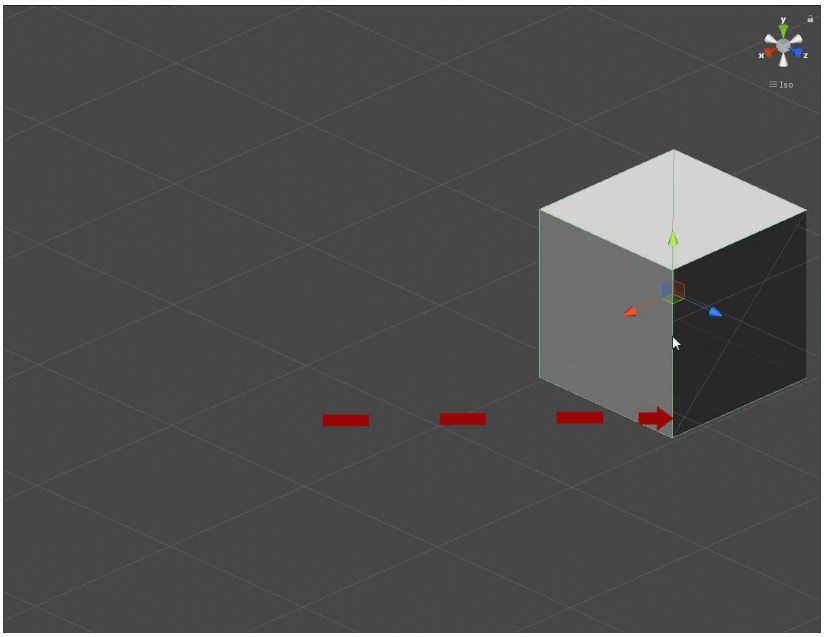
Here is an example: we start with a cube, positioned in the center of the screen:



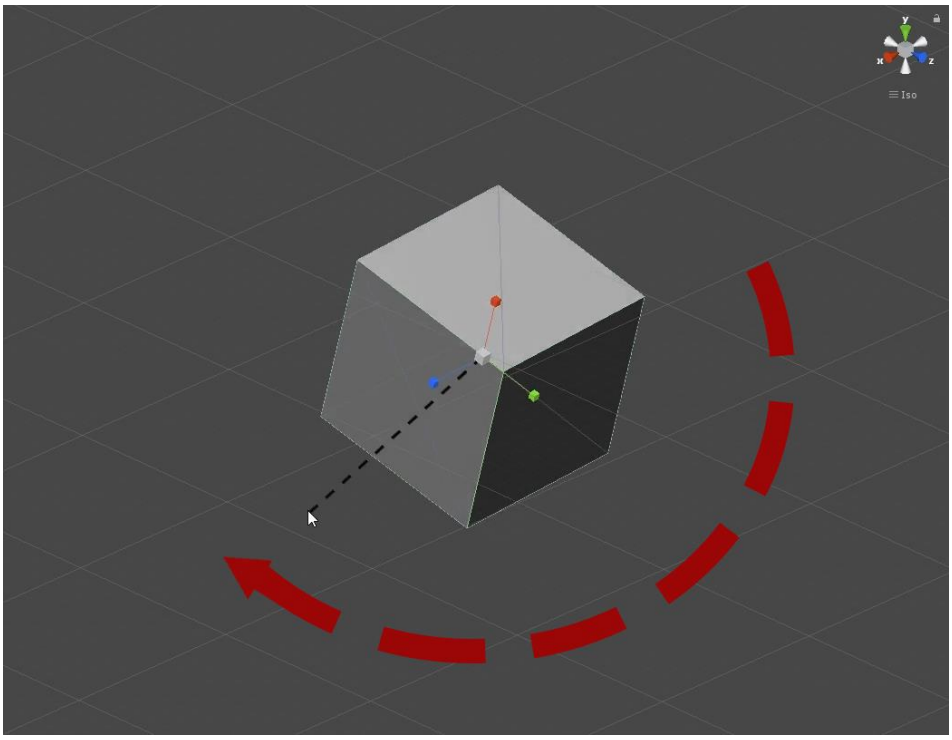
With the object selected, we press the “G” key to activate the Translation action (keys can be rebinded in the options at any time).



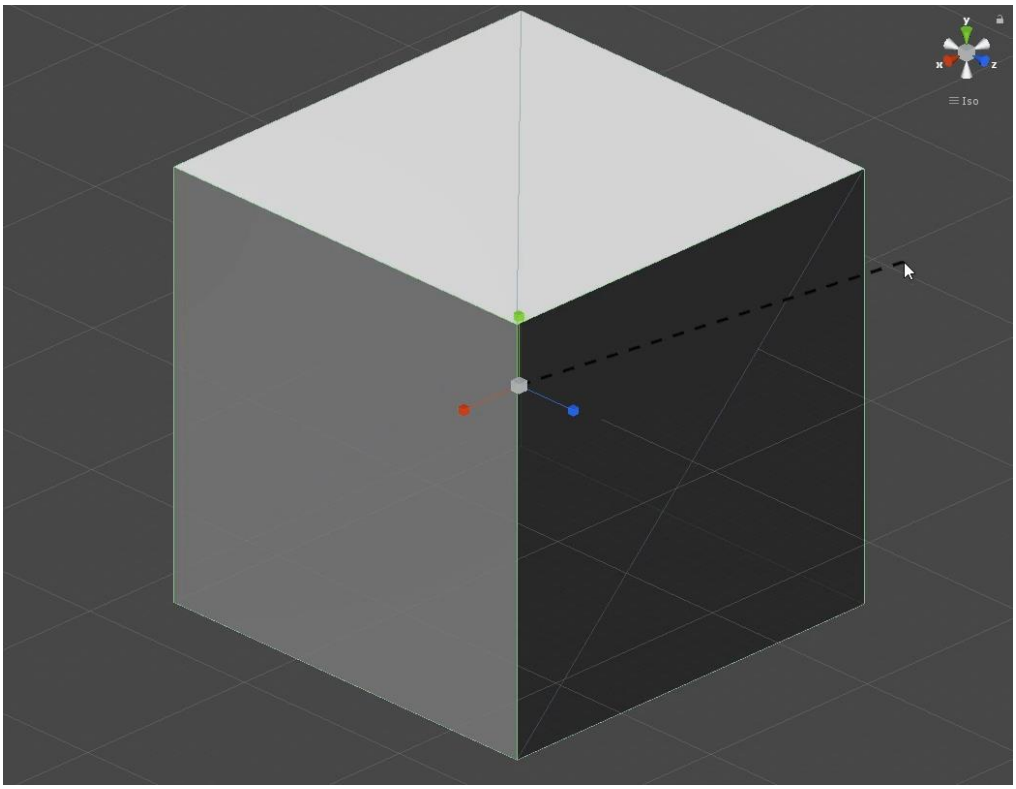
The yellow dot appeared at the pivot point of the object which indicates that the action has been activated. Now when we move the mouse – the object follows the mouse in the 2d camera view plane. After we have repositioned our object, we press “Left Mouse Button” to apply the result of the Translation.



Rotation action follows the same principle: push “R” key to start rotating the object, the dotted line, pointing towards the center will appear alongside the yellow circle, indicating that the rotation action is active. Now you can rotate the mouse around your object and the object will follow the mouse. Once you are happy with your result push “Left Mouse Button” to apply or “Right Mouse Button” to cancel and revert to the initial object rotation:



Finally to utilize the Scaling action, you push the “S” key. The yellow dot at the object’s pivot will appear alongside the dotted line, pointing towards the object’s pivot – these indicate that the scaling action is activated. Now you can move your mouse away or towards the object’s center to change the object’s scale:

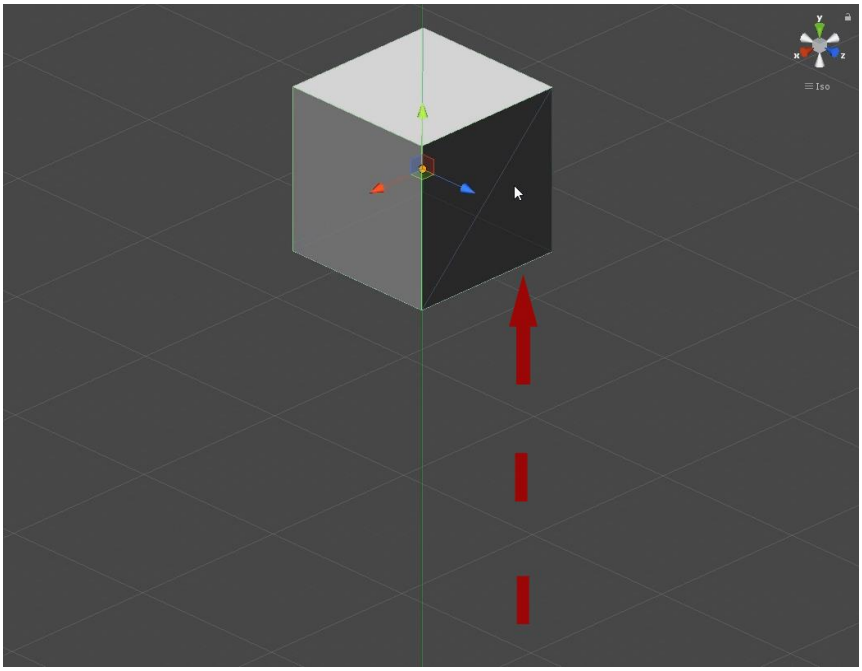


Mind, that with the Scaling tool, the closer the mouse is to the center of the object when the Scaling Action is being activated, the faster the object will scale with mouse movement. If you want very slow, precise scaling – start the scaling action with your mouse further away from your object’s pivot.

Also, keep in mind that absolutely every single action of the plugin can be reversed by pushing “Ctrl+Z” and re-applied afterwards by pushing “Ctrl+Y”.

Applying constrains when transforming objects

Now moving objects in the camera’s view plane is not a very useful thing, in most cases you would need to move the object alongside an axis or two axis and that is where Blender Actions shine: to specify the movement constraints for the translation tool you just need to hit a specific hotkey while the translation action is active. For instance, say you want to move the object up a little bit alongside the world Y axis: we first hit “G” key to start the translation action and then hit the “Y” key while translation action is active. The green vertical line will appear, indicating that a Y axis constraint is in place and the object will still try to follow the mouse, but will only move alongside the word Y axis:



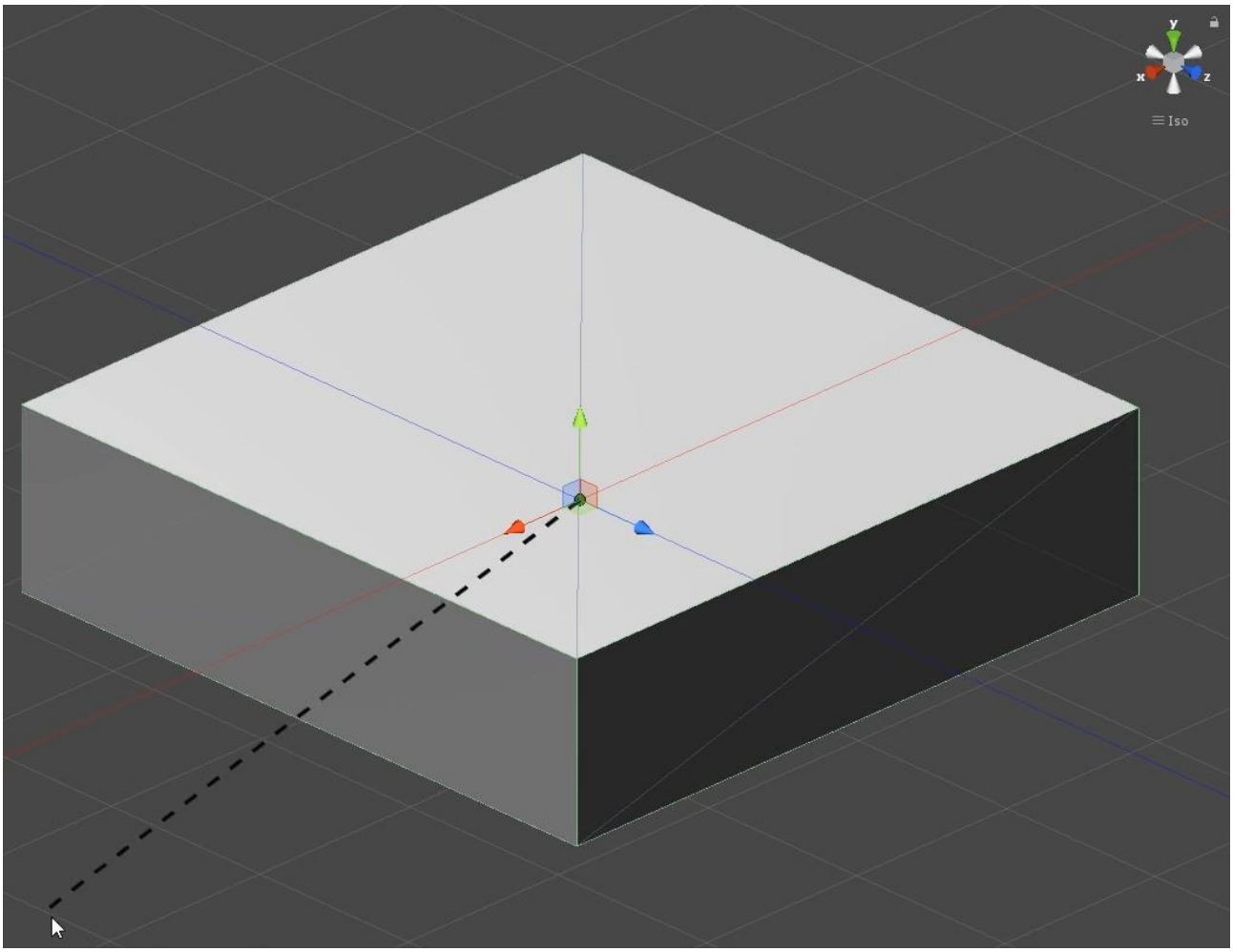
You can set the following constrains:

1. Only move alongside the world X axis (press “X”).
2. Only move alongside the world Y axis (press “Y”).
3. Only move alongside the world Z axis (press “Z”).
4. Only move alongside the world XY plane (hold “Shift” and press “Z”).
5. Only move alongside the world XZ plane (hold “Shift” and press “Y”).
6. Only move alongside the world YZ plane (hold “Shift” and press “X”).

Local space (the space of the active selected object):

7. Only move alongside the local X axis (press “X” twice).
8. Only move alongside the local Y axis (press “Y” twice).
9. Only move alongside the local Z axis (press “Z” twice).
10. Only move alongside the local “XY” plane (hold “Shift” and press “Z” twice).
11. Only move alongside the local “XZ” plane (hold “Shift” and press “Y” twice).
12. Only move alongside the local “YZ” plane (hold “Shift” and press “X” twice).

Constrains can be set and changed at any moment during an active Translation Action so don’t be afraid to push the wrong button. Constrains can be set during Rotation and Scaling Actions as well as the Translation example above. Say you want to make your cube “fat” and scale it alongside X and Z axes, just push “S” to start the Scaling Action and then hold “Shift” and press “Y”:



Note that during the Scaling Action the circle at the pivot point of the object has the green color. That indicates that you are working in the object’s local space. Unfortunately you cannot set world space constrains during the scaling action due to internal Unity limitations. To scale the object in world space, you can parent it to a new empty GameObject and scale the parent instead.

IMPORTANT! Most transformation actions of the plugin will malfunction if you are working with a Child Object, which sits inside a Parent and the Parent has non-uniform scale values (say 1, 0.5, 1). To avoid this, you can do the following:

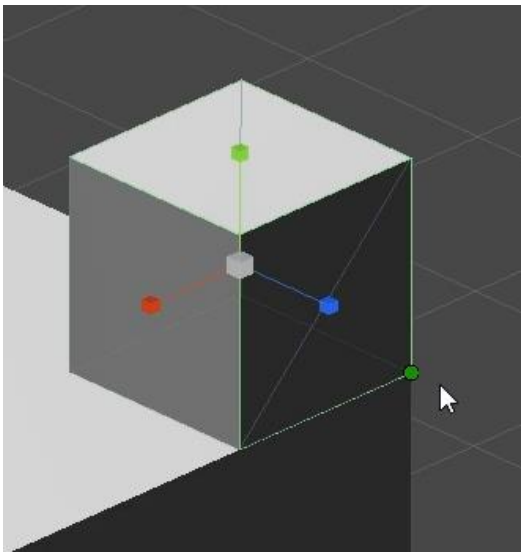
a) just transform the Parent instead (making sure that the Parent has no other parent in it's hierarchy chain with non-uniform scaling);
b) change the parent's scale so that it's uniform (every value X, Y and Z are equal to each other, e. g.: (3.07, 3.07, 3.07);
c) use the default Unity gizmo – don't forget that it is always an option!

Improving precision when transforming objects

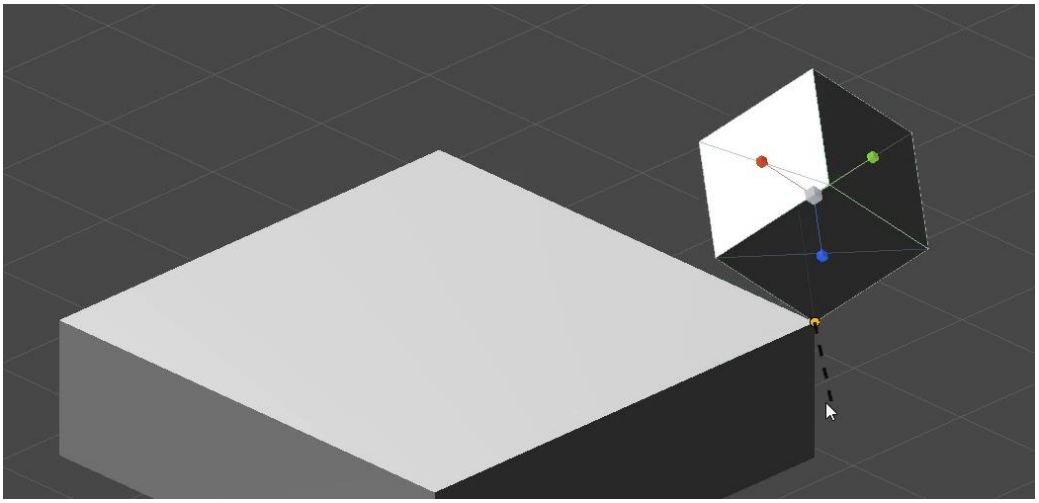
Sometimes the Translation Actions change the object too fast and you just can't find that sweet spot where the object looks perfect, to improve your precision in such cases you can always hold the "Shift" key, which will slow down the speed of the gizmo 10 times allowing you to move that pile of trash just a tit-bit to the left with ease.

The 3D cursor

When rotating and scaling objects, it is sometimes useful to set a custom pivot. To do that, you can simply hold down “Ctrl”+”Shift” and hover a mouse over any vertex in the scene – the green circle will jump to that vertex, indicating that the 3D cursor has been repositioned.



Now that the cursor has been set, you need to activate it for your translation actions. To do that just press “Alt+`” (that is “Alt” plus [Backquote], where [Backquote] is located to the left of the “1” key on your keyboard). This activates the 3D cursor mode for your rotation and scale, now that the mode has been activated, start a rotation or a scaling action and you will see, that the dotted line now goes not to the pivot point of your model but to the vertex you picked, while holding “Ctrl” + “Shift”.

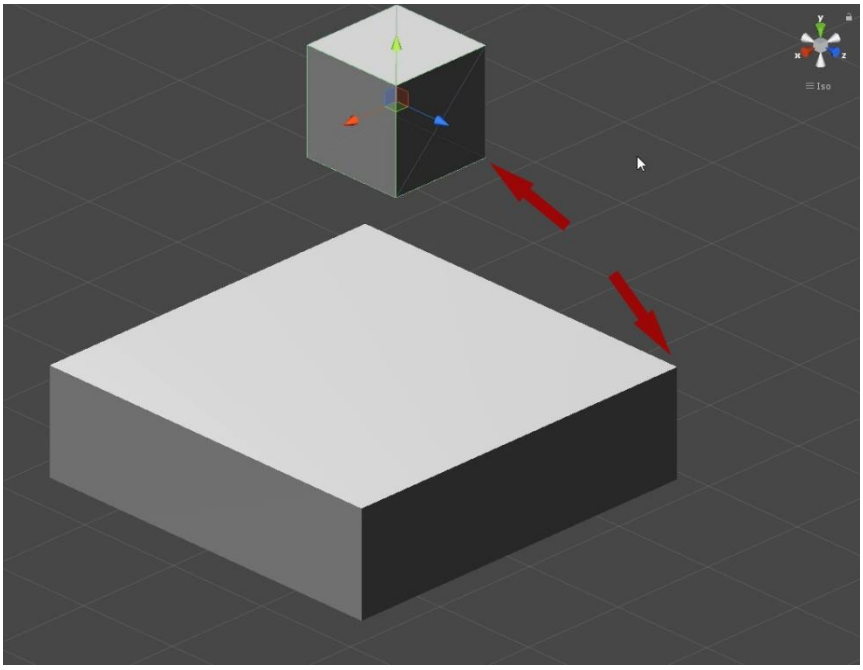


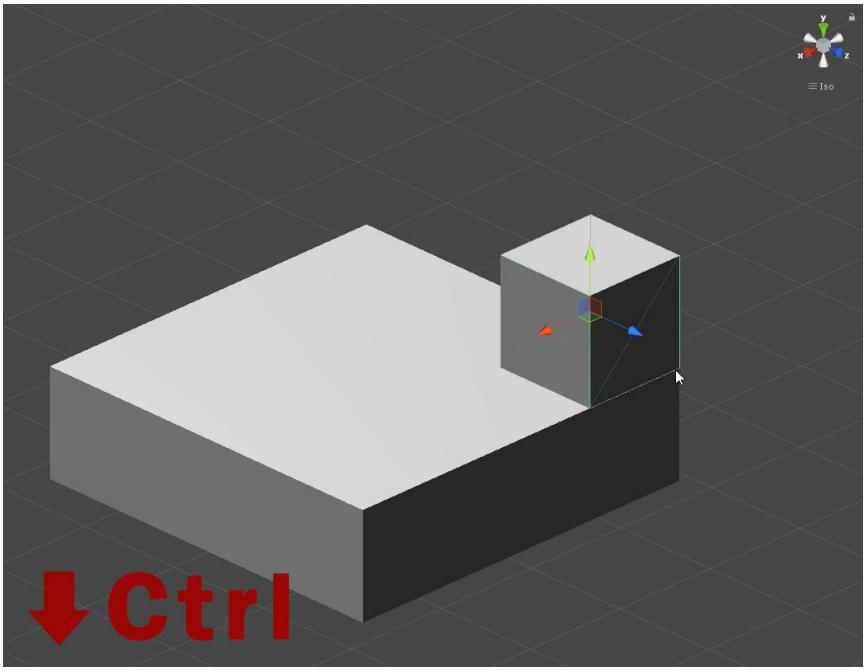
Now when you try to rotate or scale the mode, the 3D cursor will be used as a pivot, instead of the model's pivot. When you would want to cancel the 3D cursor mode and use the model pivot's instead – just press the [Backquote] key without holding "Alt".

The 3D cursor can also be repositioned automatically when you are translating (moving) objects and using the Vertex Snap (vertex snap is explained in the paragraph below). By default this mode is ON and every time you reposition an object using the Vertex Snap, the 3D cursor will jump at the vertex you snapped your object to. This is often the desired behavior as you would usually rotate/scale your model after moving it and you can specify the vertex snap vertex in a smart way so that you rotate your model around the pivot of your choice. In any case, there is a hotkey in the settings menu ("Q" by default) that allows you to toggle this mode ON or OFF whenever you want.

Vertex Snap

Vertex Snap allows you to quickly transform an object so that its vertex touches a vertex of another non-selected object. To perform a vertex snap all you need to do is start any transformation action, hover your mouse over a non-selected object's vertex and hold down "Control" key. The selected object will snap so that its vertex touches (or is as close as possible) to the non-selected object's vertex. Make sure to apply the result of your action by pressing "Left Mouse Button" WHILE holding the "Ctrl" key. If you release the "Ctrl" key before applying – the snapping will be canceled.



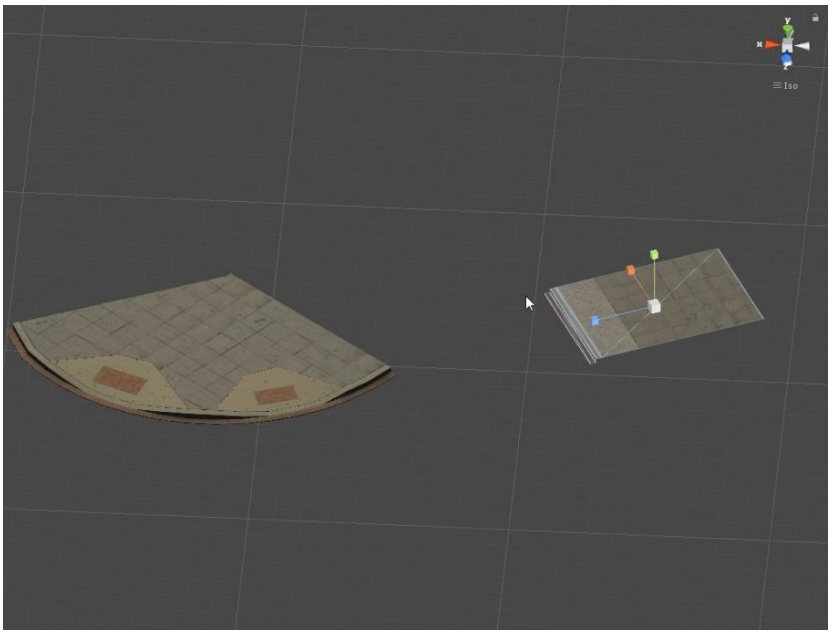


Keep in mind, that vertex snap tries to find THE FIRST of the two vertices amongst the NON-selected models so that that vertex is the closest to your cursor in certain SMALL radius around the mouse cursor, while THE SECOND vertex is being searched amongst the SELECTED models (models that are actively being transformed) and there is no radius in this case. Therefore, it is recommended to ensure that you have your cursor positioned in-between the two desirable vertices before snapping. Sometimes the result of the snapping might not satisfy you, because the algorithm used the wrong vertices for snapping – you can fix this by repositioning/transforming your model first, before snapping, so that it's target vertex is closer to the other target vertex you want to snap it to.

Vertex Snapping works with rotation and scaling as well as with translation. It can also be used in conjunction with constrains. You can make one building be exactly as tall as the other one by starting the scaling action with the "S" key, setting the world Y axis constraint by pressing "Y" key and then hovering over the target building's topmost vertex and holding the "Ctrl" key:

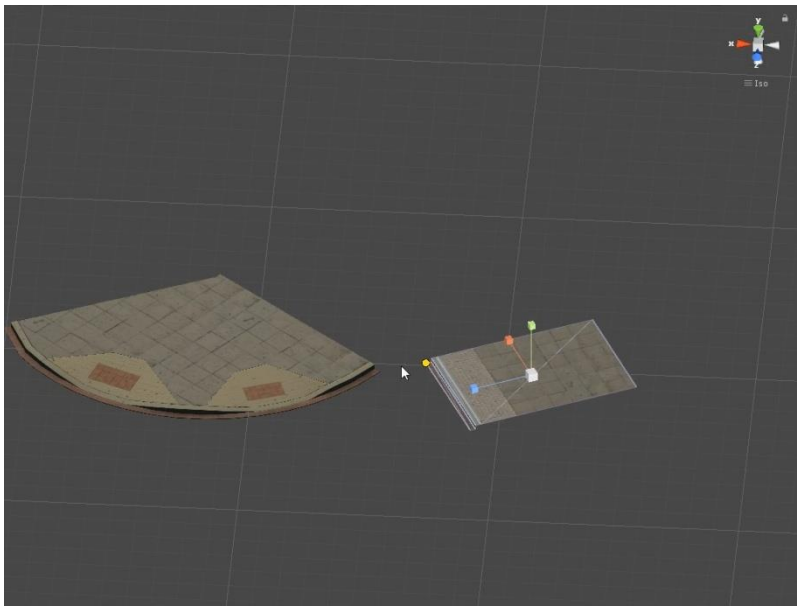
Vertex Snap also works with the 3D cursor, as mentioned in the previous section, by default the 3D cursor will reposition to the last vertex snap location during your

translation action. Let's see an example when this might be useful. Say you have two pieces of modular pavement that looks like this:



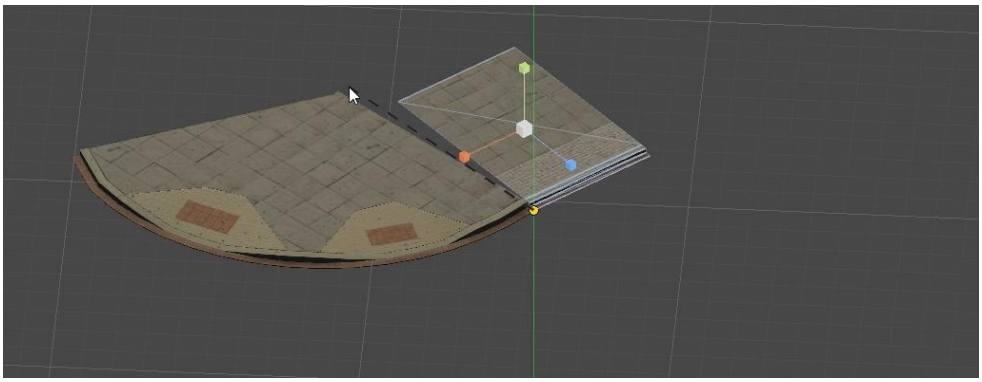
You want to snap them together. Here is how you do it:

1. Make sure your mouse is positioned near the corner of the right pavement piece, just like the screenshot above shows. We want to snap our model using this very corner, that is why we position our mouse near it to help the vertex snapping algorithm.
2. With the right piece selected, activate the Translation Action with the "G" key.
3. Reposition the model towards the corner of the left pavement piece:

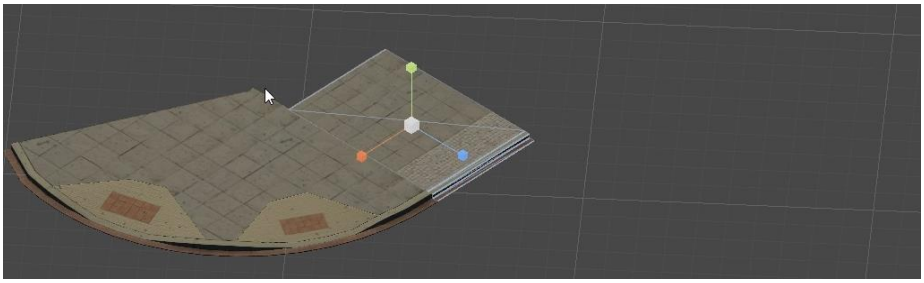


4. Hold down "Ctrl" key to activate the vertex snapping and watch the both corners perfectly snap together.

5. Apply the transformation by pressing “Left Mouse Button” while holding down the “Ctrl” key.
6. With this action we repositioned the 3D cursor directly on top of our vertex snapping corner, which is exactly what we need, because now we will be rotating our model around that corner. Keep in mind, the 3D cursor is not automatically repositioned, when you use vertex snap during rotation or scaling actions – only during translation actions.
7. To activate the 3D cursor mode for rotation action press “Alt + [Backquote]”.
8. Activate the rotation action by pressing “R”.
9. Set the world Y axis rotation constraint by pressing “Y”.
10. Rotate the model around the corner vertex and the world Y axis by making rotational moves with your mouse, in the end you want the two models to look this way (note the position of the mouse cursor – it is important):

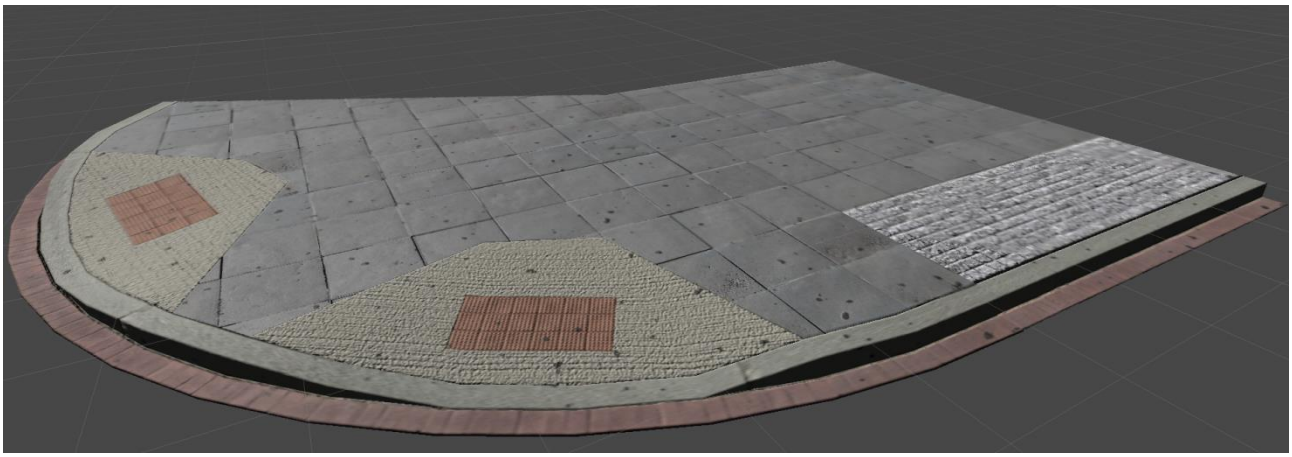


11. Once you get your models into this position while your rotation action is active, hold down the “Ctrl” key to activate the Vertex Snapping – the two pavement pieces will perfectly snap together:



12. The right pavement piece is smaller than the left one, to fix that we will use vertex snap with scaling – don’t move your mouse anywhere, just activate the scaling action with the “S” key, hold down the “Ctrl” key to activate vertex snapping (note that scaling action will still use the 3D cursor and the 3D cursor will still be in the same position as before) and, as the right pavement piece finally perfectly aligns with the left one, confirm the result by pressing “Left

Mouse Button” while holding down “Ctrl” key. Now your pavement pieces are perfectly snapped together:



Rotating/scaling an object by a specific amount of degrees/units

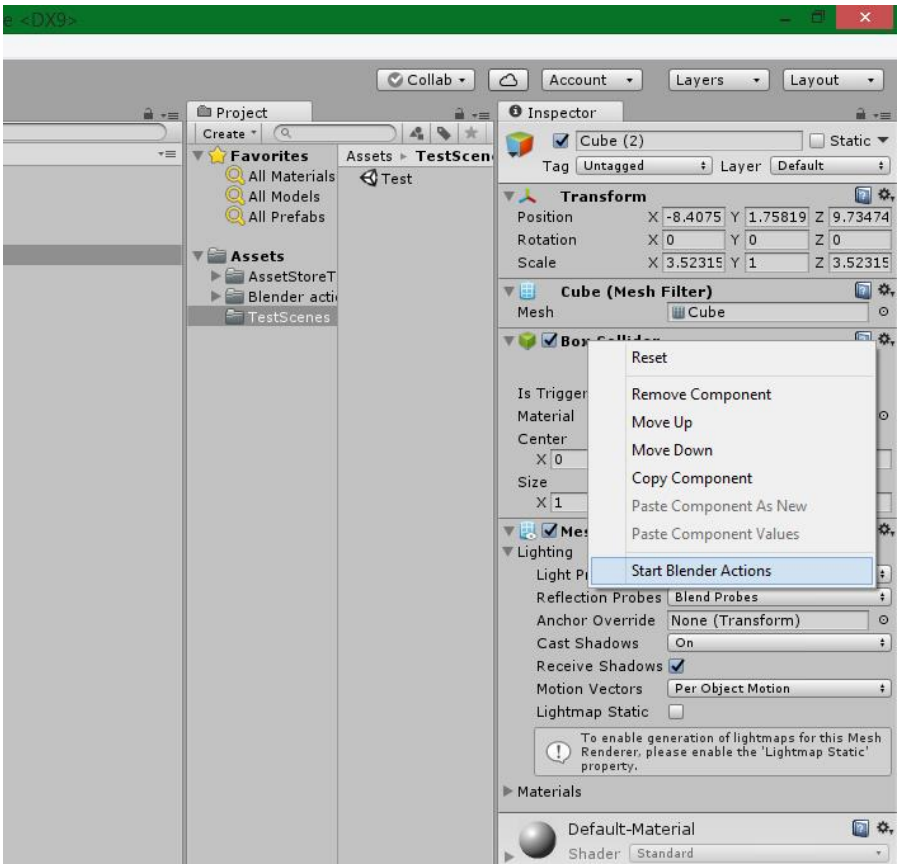
Blender Actions provides a way to rotate/scale an object by a specific amount of degrees/units just like it works in Blender. To do this you simply type in the number you want to use, while rotation action or scaling action is active. Here is an example: you want to rotate your house 45 degrees around the world Y axis. To do so, select the house, start the Rotation Action by pressing “R”, set the world Y axis constraint by pressing “Y” and start typing in “45” (just press the “4” key, followed by the “5” key on your keyboard).

Congratulations! You have mastered the Translation Actions! Now that you wield the power of the Hotkey-Based Gizmo, you can hide the default Unity gizmo UI so that it no longer annoys you with its useless presence! Just press “Ctrl” + “Space” to hide it (press it again whenever you want to unhide the gizmo).

Collider Editign Mode

The plugin includes a special mode that allows you to edit colliders. When activated, your Translation, Rotation and Scaling Actions will edit a collider's properties instead of the Transform component of the GameObject (apart from cases when you are rotating the collider). There are three ways to enable the Collider Edit Mode:

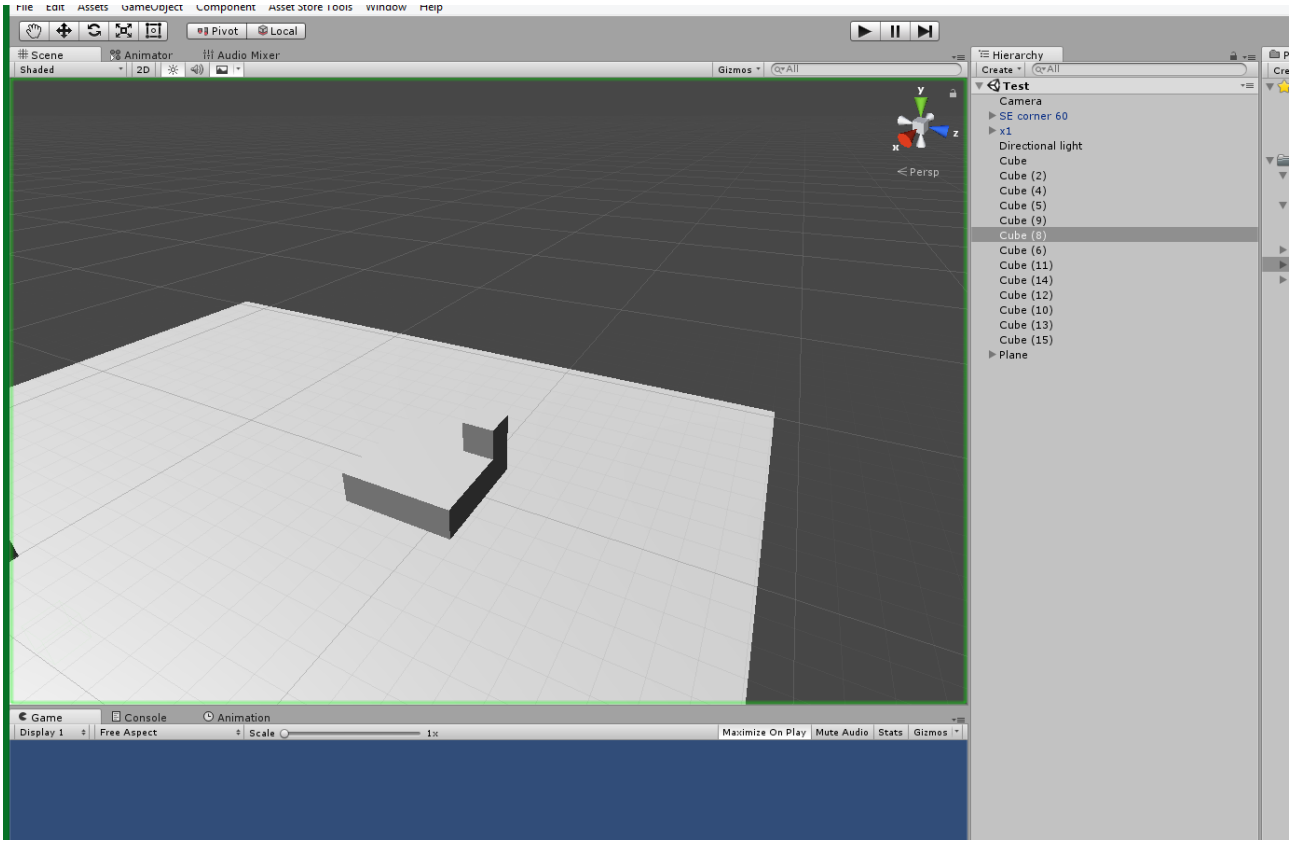
1) Right click on your collider component's name in the Inspector and choose "Start Blender Actions" in the menu.



2) Make sure you have selected A SINGLE GameObject that has NO CHILDREN and only ONE COMPONENT attached (besides the Transform component): a supported collider component (BoxCollider/SphereCollider/CapsuleCollider). Now if you start any Transform Action (move/rotate/scale), the collider editing mode would activate automatically.

3) Create a new GameObject and attach a supported collider to it with a hotkey action – the Collider Editing Mode would activate automatically (read more about auxiliary hotkey actions below).

To check if you are working in the Collider Editing Mode, make sure that your editor's Scene View window has a green outline around it like so:



To turn off the Collider Editing Mode – push "Escape".

Auxiliary hotkey actions

The plugin has a number of hotkeys dedicated to quality of life improvements when building levels in Unity Editor.

Reset actions

These include Reset translation / Reset rotation and Reset Scale, they allow you to quickly reset the relevant part of the transform of the selected GameObject to it's default state (zero for translation and rotation, one for scales).

Rotate 90 degrees around Y axis

Allows you to quickly rotate a model 90 degrees around the world Y axis (up axis). Handy shortcut, try it out!

Hide actions

These include hide selected, hide unselected and unhide hidden. One of the most powerful auxiliary features - allows you to temporary hide certain objects. To do this, objects' MeshRenderer's are set to 'disabled' in order to ensure that lights are still ON after you hide the objects.

The most useful action is "hide unselected" as it allows you to hide the entire scene and only keep a few objects you want to manipulate. Very useful when working with indoor objects. Help you ensure that buildings don't get in the way. Also quite useful when working in forests, etc.

WARNING! Don't save the scene with objects still being hidden! The list of hidden objects is lost when you turn off Unity so that if you then load the scene you would be forced to use the "Unhide all" fail-safe hotkey to enable all MeshRenderers back.

Reset all selected prefabs / apply all selected prefabs.

This basically allows you to mass-apply changes to a number of prefabs in your scene, or mass-revert them to their saved states.

Duplicate selection and start Translation Action

This action does exactly the same as the Unity's "Control+D" action (duplication), but also immediately starts translation action after the duplication for the newly cloned object, allowing you to immediately place the clone wherever you want.

Camera controls

These allow you to position the editor camera in certain fixed states with hotkeys. You can set top view/front view/ back view and the inverse as well as set the orthographic/perspective mode super quickly with a dedicated hotkey.

Toggle show/hide Unity default gizmo

As you familiarize yourself with Blender Actions, you might decide at some point that you no longer need the default gizmo – this action allows you to hide it. It is useful, because the gizmo can often get in the way and block your view. At the same time, every once in the blue moon you might want to actually see it again (for instance to get a good understanding how the object's local axes are set up) and then you can use this hotkey to bring it back.

Deselect All

Handy action to ensure that none of the objects is selected in the scene. This helps prevent misclicks and applying transformation on objects you forgot to deselect.

Create an empty GameObject + rename

This creates an empty GameObject as a child of the currently selected object or in the root of the scene and starts renaming operation immediately, allowing you to instantly type in its name.

Parent selected to a new empty GameObject + rename

Allows you to quickly put a bunch of GameObject under another, newly created empty GameObject as children. The new GameObjec is created in the same scope as all the topmost selected GameObject. Just like the previous action, this action immediately starts renaming operation so that you can type in the new object's name.

Toggle enable/disable selected GameObjects

Allows you to quickly turn on or off selected GameObjects. Nothing fancy, just enable/disable entire GameObject.

Create Collider and start Translation

Allows you to quickly create a GameObjects, attach a supported collider to it, parent the GameObject to selected GameObject and start Translation action.

Limitations

The plugin was designed to allow for maximum freedom when binding keys. You can bind any mouse or keyboard key you want and use various combinations with “Control”, “Shift” and “Alt”, you can also bind up to two keys for the same actions. However, it is impossible to use a few keys that open corresponding sections of the Unity Editor menu strip (“Alt”+”F” for instance, opens “File” dialogue and while it can be binded for Blender Actions it will always active the dialogue instead of launching your hotkey). At the same time “Control”, “Shift” and “Alt” keys cannot be used freely (without non-modal buttons) apart from special cases. Depending on user feedback the latter feature might be added in future releases.

Next, keep in mind, that right now, working in a crowded scene (500 000+ vertices) can result in sluggish Vertex Snapping behavior. The editor can sometimes freeze for a few seconds. If you experience this, release the Vertex Snap key and wait for the editor to catch up. It is recommended to use “Hide Unselected” action before engaging in Vertex Snapping for smooth performance. For this reason the maxim number of times Vertex Snap can trigger over a second has been limited to 4 (so it can feel sluggish as opposed to Blender). Future versions of the plugin will include a faster version of Vertex Snapping algorithm.

Also there is a glitch with scaling the model and using Vertex Snap – sometimes the model will start jumping around like mad. This can easily be fixed by scaling model in such a way that it is as close as possible to the state you want it to be after the Vertex Snap and only then using Vertex Snap.

Finally when you position your camera in such a way that the model you are working with stretches across several screens and you only see a fraction of it and, most importantly, the pivot of said model is behind the editor camera, the Translation Actions can invert in several cases (with some particular constrains set) this means that the model will follow the mouse in the opposite direction than normally, this bug has partially been fixed and will most likely be eradicated in the future.

Happy blending! ☺