Snapping Tool and Mesh Merge Documentation

This document is an overview of the **SnappingTools** editor extension and the **MeshMerge** component found in some of our packs.

Snapping Tool

Overview:

The Snapping Tool works similarly to Unity's own built in snap functions, with a few notable exceptions:

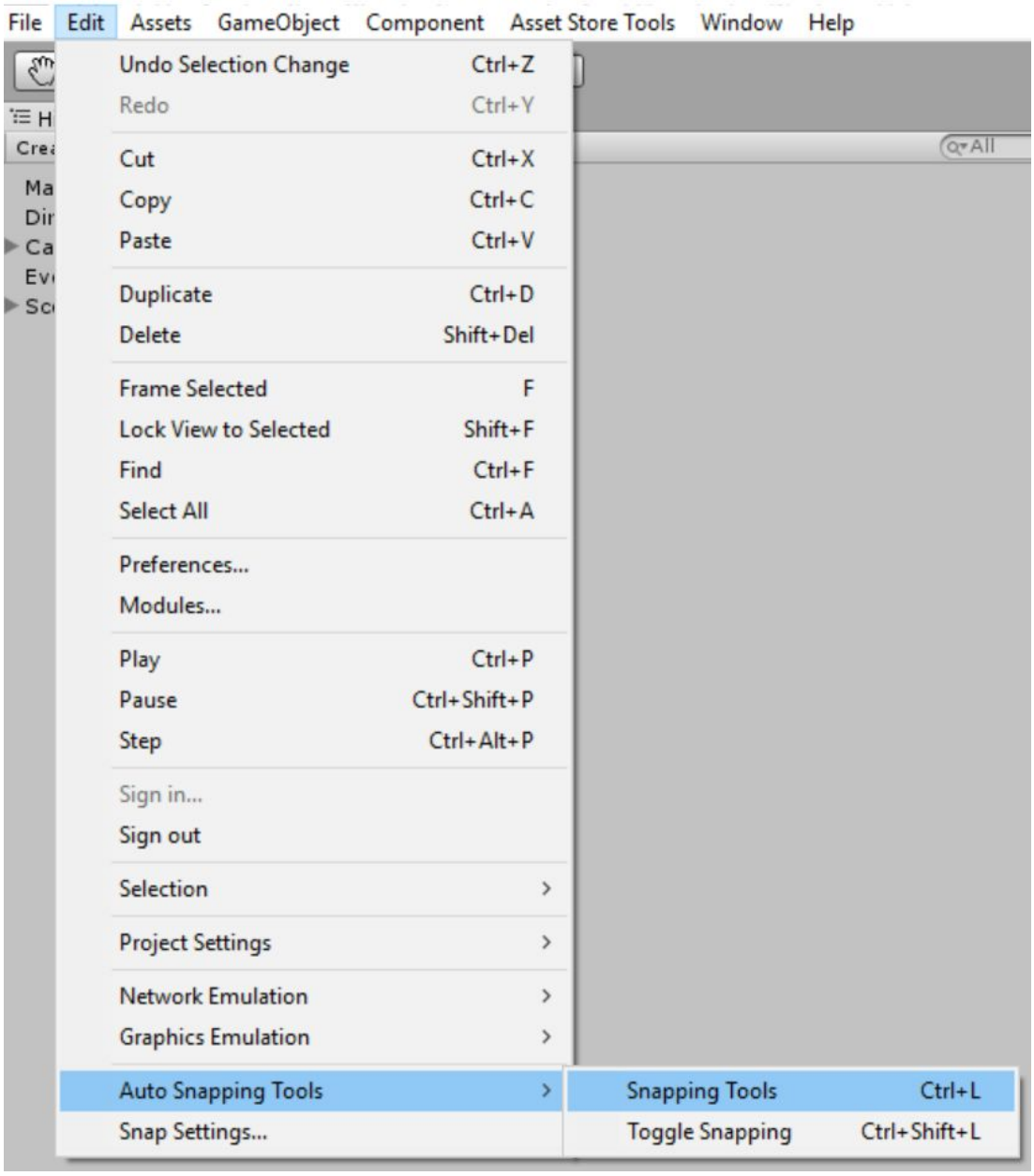
- The Snapping Tool is toggled on or off instead of requiring the ctrl (or cmd on mac) key to be held down.
- The Snapping Tool will snap any GameObject's transforms directly to the nearest position/scale/rotation based on whatever increments you specify, this makes it easy to drag around terrain tiles and keep them all lining up with the grid.
- The Snapping Tool provides an option to restrict transforms on the x/y/z axis.

Usage:

The Snapping Tool is accessed from the Unity Menu under

[Edit>Auto Snapping Tools>Snapping Tools].

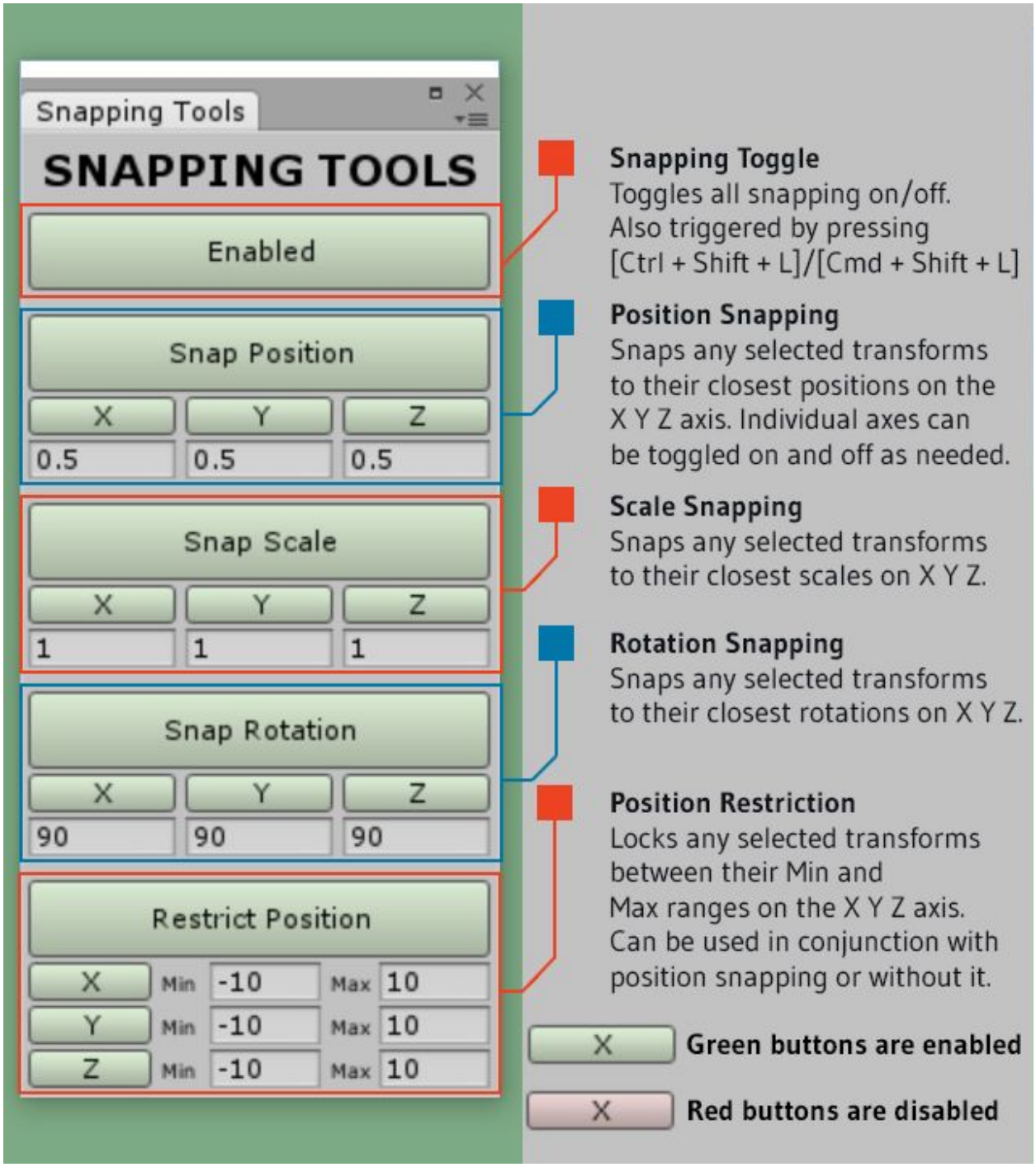
It can also be brought up with Ctrl + L, (or Cmd + L if you are on a Mac).



When opened pressing Ctrl + L / Cmd + L again will close the window.

Note: For rotation snapping, the PivotMode and RotationMode are automatically set to Pivot and Local until rotation snapping is disabled.

Once open you will be presented with the Snapping Tools Editor Window:



Each type of snapping can be used independently of one another so you can use whichever type of snapping on whichever axes you choose. All snapping will then be applied based on which types you have selected, when you select any object with a Transform component.

Note: Remember to turn snapping off when you no longer need it as it will continue to Snap until disabled.

Closing the Snapping Tools window will also have the same effect as disabling all snapping.

Mesh Merge

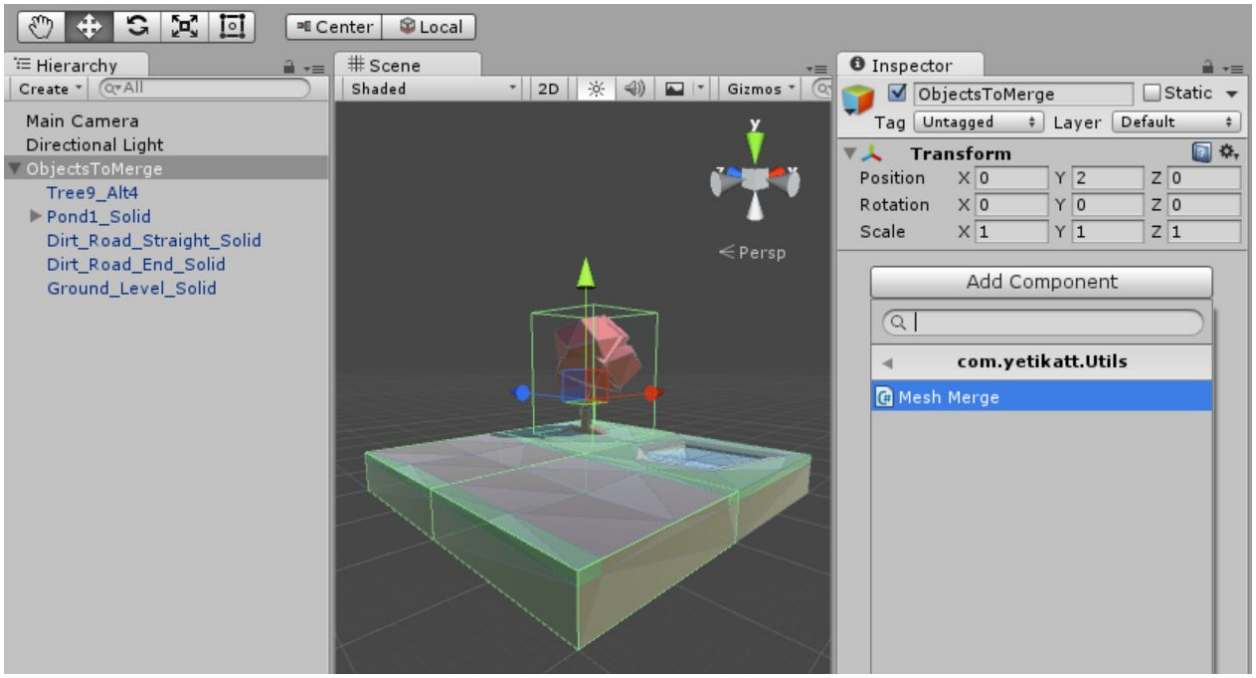
Overview:

The Mesh Merge component lets you combine the Meshes of GameObjects into one single Mesh. This reduces the number of batches that need to be made which is essential for ensuring decent performance on mobile platforms.

It also enables you to save the combined meshes into your project library to reuse, and also save the object as a prefab.

Usage:

To use the MeshMerge component, you must add it onto a GameObject whose children you want to merge.



With the GameObject selected, go to [Add Component>Scripts>com.yetikatt.Utils>Mesh Merge]

Note: If you don't see com.yetikatt.Utils or Mesh Merge, make sure you have downloaded and imported the asset from the Unity Asset Store.

Once this component has been added, it will add a MeshRenderer component and MeshFilter component to the GameObject, as these are required for the merged mesh.

The GameObject should then have the MeshMerge component:

[**Save as new:**] If this is selected the Meshes in the children of this GameObject will get combined and saved as a new GameObject. This is useful if you want to make multiple different merged meshes in scene but don't particularly care about saving the meshes, or storing them as prefabs. The GameObject that is created will be added as the First child of this GameObject

[**Parent Transform mode:**] This lets you choose if the origin of the GameObject should move when merging the children's meshes, there are 3 modes to choose from:

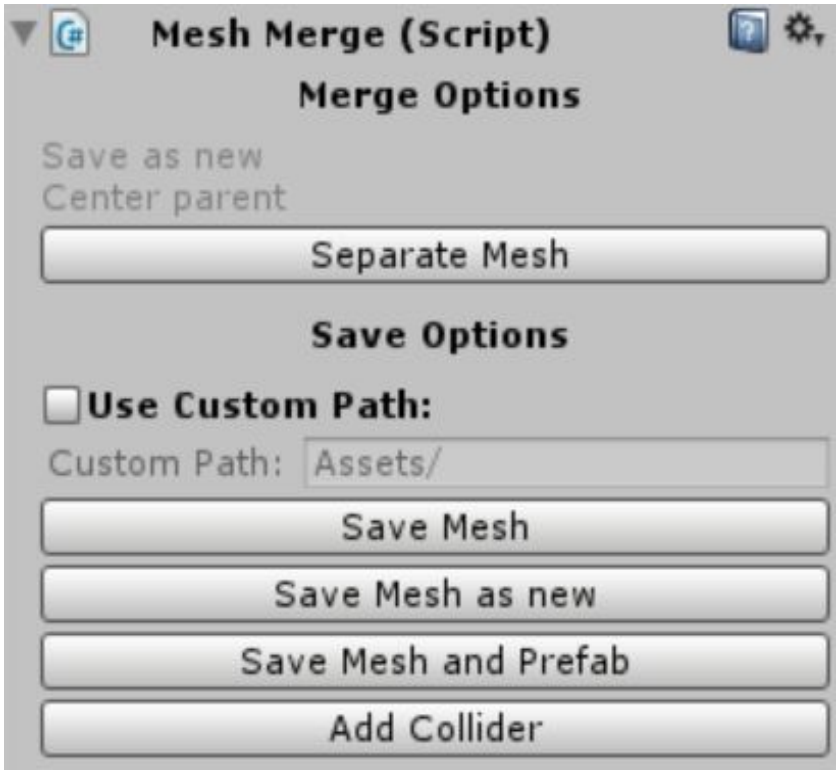
Don't Center: this is the default mode and will Merge the meshes with the transform remaining where it is.

Center: this will move the transform to the absolute middle of all the children GameObject that were merged.

Grid Align Center: this will move the transform to the closest point to the absolute middle that is still aligned with the Grid, this is useful if you are using grid snapping and would like to keep your terrain or other objects aligned with the grid.
(**Note:** Make sure the X Y Z values shown underneath the Grid Align Center option match the grid units you are snapping to.)

Once you are happy with your setting clicking on the **Combine Meshes** button will merge all of the meshes and store the combined mesh in the MeshFilter (or in a separate GameObject if you chose Save as new)

This will disable all the children objects and bring up a new set of options in the Mesh Merge component's inspector.



[**Separate Mesh**] This reactivates all the disabled GameObjects and destroys the Merged mesh stored in the MeshFilter. The Mesh Merge options revert back to the initial selection.

[**Use Custom Path:**] If selected this will allow you to save any prefabs in a custom location that differs from the Meshes saved location. The custom path is relative to the root of the Assets folder, (so for example, to save prefabs in a folder called CustomPrefab that is located within the Assets folder you would put "Assets/CustomPrefab/"

[**Save Mesh**] This brings up a file browser and saves the existing Mesh attached to the GameObject into the Project. Modifying the mesh in the Project window would then directly affect the Mesh attached to the GameObject.

[**Save Mesh as New**] This brings up a file browser and saves the Mesh as a new separate instance into the Project. Modifying the mesh in the Project window would have no affect on the Mesh attached to the GameObject.

[**Save Mesh and Prefab**] This saves a mesh like the previous two options, and also saves a Prefab, which will have the Mesh and any Materials that were merged stored with it in the Project. The location of this Prefab will either be with the Mesh or wherever the custom path was specified.

[**Add Collider**]
This will create a MeshCollider using the Merge Mesh. If a Collider is added to the Mesh this will also be included in the saved prefab when Save Mesh and Prefab is used.

[**Remove Collider**]
After adding a MeshCollider you will be given the option to remove the mesh collider if you decide you no longer want it.

I hope you find the assets useful and if you have any feedback or suggestions please let me know at:
chris@yetikatt.com
And thank you for your interest in our assets :)