Dis|Assemble Tool by Oluss Studio
Ver. 1.1.1 – December 2017

Table of Contents

1. Introduction

In principle, this tool enables you to move your 3D object/s to a pre-determined location. By remembering this principle, you could actually achieve more than what has been demonstrated here in the scenes that come with this tool. This tool can also be combined with other tools to create more complex scenes, or even games and apps.

The codes work with the default Unity GUI system and will most probably work on any other kind of GUI systems available in the Asset Store. However, your prior knowledge of such GUI systems will be the most important factor on whether **Dis|Assemble Tool** can be integrated with them.

As mentioned on the Asset Store page and everywhere else, setting up this tool will require a certain degree of familiarity to Unity. This is due to the fact that **Dis|Assemble Tool** utilizes Unity built-in Tag system which can be learned for the following link. GUI customizations will also require coding skill. If you are totally alien to Unity's default GUI system, you can start from this link.

2. Getting Started

We strongly recommend you to explore the whole package before setting everything up. Below are a few steps to get yourself familiarized with the setup:

1. Open the *SampleScene.unity* in *Start* folder.
2. Pay close attention to the *TurbineMainController* object and the way the other objects are parented. This set-up is crucial to make everything works.
3. Screen through all objects under the *turbine* object and notice that some of them have the *receiver.js* script attached to them. See also that each of the objects with the script attached has different Tag on the inspector panel.
4. Open the codes and read through the comments.

3. Use Case

This list is not limited to what will be mentioned, but here is a list of scenario where you will find this tool useful:

1. Packaging design of a product.
2. Showcase of a material.
3. Anatomy presentation.
4. Digital assembling manual.
5. Engineering schematics, blue print or diagrams.

4. Setting It Up

We are going to explain how we set up the *SampleScene.unity* from scratch. By understanding the principles of these steps, you can then create your own scene. Below are the steps to set the scene:

1. Start by creating a new project and new scene in Unity.
2. Import **Dis|Assemble Tool** from Unity Asset Store.
3. Create an empty game object (Ctrl+Shift+N). Name it *turbineMainController*.
4. Drag the *turbine.fbx* from the Project tab and drop it to the Hierarchy tab over to the *TurbineMainController* so that *turbine* is automatically nested as a child.
   NOTE: Notice how there is no parent-child hierarchy in *turbine.fbx*. When you export your own model from Maya, 3DS Max, etc., make sure that they also have no hierarchy at all before you exported them.
5. At this point, you may want to adjust your camera and create some lights.
6. Duplicate *turbine* by selecting it from the Hierarchy panel and pressing Ctrl+D or via right click.
7. Rename the clone to *turbine_targets*. Do NOT rename any of the children. When you are importing your own object, they can have different name. However, the cloned one has to have the *_targets* suffix.
   NOTE: This tool is designed under the assumption that each model imported is uniquely named. There should be no two objects under one parent that share one identical name. E.g.: Never have two or more objects named 'Cube0001' when you export your model from any 3D modelling software.
8. From the Inspector panel, do the following to 7 children under *turbine_targets* :
   a. Change the position Z value of *blades* to 8.
   b. Change the position X value of *gear1* to -5.
   c. Change the position X value of *gearShell_Left* to -10.
   d. Change the position X value of *gen* to -7.
   e. Change the position Z value of *rod* to 3,6.
   f. Change the position Z value of *rodHolder* to 2,5.
   g. Change the position Z value of *rodTip* to 9,2.
9. Select ALL children under *turbine_targets* and uncheck the Mesh Renderer component from Inspector panel.
10. If the tags are not created yet, add 6 new tags namely parts1, parts2, parts3, parts4, parts5, parts6.
    NOTE: These tags have to start from 1 and have to be in order. See the note on step 15 regarding the prefix.
11. From the Inspector panel, change the tags of the following 7 children under *turbine*:
    a. Change *blades* tag into parts2.
    b. Change *gear1* and *gen* tag into parts6.
    c. Change *gearShell_Left* tag into parts3.

   d. Change *rod* tag into parts4.
   e. Change *rodHolder* tag into parts5.
   f. Change *rodTip* tag into parts1.
12. Attach *receiver.js* of the same 7 objects mentioned in step 11. You can quickly do this by selecting those 7 objects and drag the script to the Inspector panel.
13. Attach *mainController.cs* to *turbineMainController* via drag and drop from Project panel to Hierarchy panel.
14. Select *turbineMainController* and from the Inspector panel, change the Parts Amount value to 7.
    NOTE: In this scene, we only have 6 different parts (tags). However, due to some reasons and bugs, this value has to be 1 value more. So if your scene has 16 parts, this variable has to be set to 17.
15. Still on the Inspector panel, type in *parts* in the Tag Prefix.
    NOTE: This variable should be the same as the prefix of your tags. Since this scene is tagging with parts1, parts2, etc., this variable should be *parts*. For example, if you are tagging your objects with Group_1, Group_2, etc., then this variable should be G*roup_*.
16. Run it and start testing it by clicking the 'Detach 1 Seq.' button from the Game tab.

5. Troubleshooting

You may have done all the 16 steps mentioned in chapter 4 and yet the set up does not run as it should have been. The very first thing that you want to do is to review the notes that we have supplied under every step. Notes on step 4 and 7, for example, are often taken for granted yet could be the cause for failure. To make things easier, we will list down those notes in this section to act as a checklist for your troubleshooting:

- Make sure that no model is a parent or child when you exported them.
- Before you duplicate your objects in step 6, make sure that they are named uniquely. No two objects may share one identical name when you exported them from the modelling software.
- After you duplicate your object, make sure that you do NOT change any name except adding *_targets* to the parent of the duplicated object.
- Add precisely *_targets* to the clone. *_Targets* or *_target* will not work. If you are a programmer, you are free to access *receiver.js* and create custom suffix this by modifying line 32.
- Make sure that the tags you created starts with suffix number 1. It will not work if you number the tag as 01, 001 or the likes. It has to start with 1.
- Still related to the previous point, make sure that you skip no number.
- In relation to step 14, make sure to type in 1 value higher than the parts you would like to have, in the Parts Amount variable. In *SampleScene.unity* this value has to be 7 because the scene has 6 parts. So if you have 10 parts, you have to type in 11 to this variable.
- Make sure that in step 15, you have typed in identical name as your tag's prefix. This variable is case sensitive and so is Unity tagging system.

We hope this checklist could help you solve your problems. Do contact us for feedbacks and more troubleshooting issues you need to tackle. Our contact is listed on Chapter 8 of this documentation.

6. Taking It Further

The best way to enhance your visualization is to create appealing GUI. This tool comes with a paper-themed GUI, which is included to give you a better idea how you can push the graphics element. You can explore this by opening the *SampleScene_Further.unity* located inside the *TakingItFurther* folder. We have to remind you once again, understanding this scene will not be easy if you are totally unfamiliar with Unity GUI system.

For the interaction, you can also combine this tool with our other product, the Camera Orbit Tool. This will enable you to rotate the object, as well as zoom it in and out.

Lastly, if you are a programmer, you can change the type of input to dis/assemble your objects. For example, you can use this tool on a tablet and code it so that when users spread fingers, the objects will be disassembled.

7. Future and Previous Updates

By purchasing this tool, you are entitled for free future updates. The current version is 1.1 and it has the following features:

- The scripts are able to record rotation, enabling rotating movement during detaching / attaching parts.
- Minor bug fixes and cleaning up.

8. Support

We are really thankful for your support by purchasing this tool, please continue supporting us by sending us feedbacks, complaints (or even curses!) to support@oluss.com and we will get back to you in 48 working hours. We do not have phone support right now.