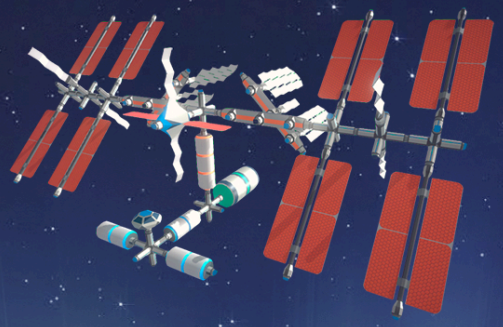
Space Station DESIGNER



Modding Guide

Table of Contents

Version Number	Revision Date	Description
0.5.15	January 16th, 2024	Published a new set of modding tools that is compatible with Space Station Designer v. 0.5.15 and updated the affected samples files in mod.io to version 0.5.15.
0.5.0	August 5th, 2023	Minor amendments to the document. Published a new set of modding tools that is compatible with Space Station Designer v. 0.5.0 and updated the affected samples files in mod.io to version 0.5.0.
0.4.0	October 10th, 2021	Initial revision released with Space Station Designer v. 0.4.0.

Introduction

Welcome to the official modding guide to Space Station Designer!

Space Station Designer is a construction and management game that puts you in charge of creating and expanding a network of orbital outposts in Earth and Lunar orbits. It is the second large game from Polar Motion, the creators of Buzz Aldrin's Space Program Manager. The game is under heavy development, but it's already available for players who are interested in trying it out and give us feedback.

If you'd like to give Space Station Designer a spin, you can either:

1. Register for the Playtest version in the Steam page. Approval of Playtest requests normally take less than 24 hours.
2. Adding it to your itch.io library.

The versions from both online stores are in sync and offer the same functionality, including a 'Campaign' and a 'Sandbox' mode.

The 'Campaign' mode allows you to perform a wide range of activities such as recruiting and managing astronauts, building and installing modules, providing commercial services, manufacturing commercial goods, acquiring and commercializing industrial R&D technologies, leasing modules, and a lot more!

In contrast, the ‘Sandbox’ mode offers you a more chilled experience by providing a vast array of modules and resupply spacecraft so that you can come up with your own creative designs without worrying about any constraints.

Elements that can be modded

At the time of this writing, Space Station Designer allows you to mod the following elements:

1. **Astronauts:** You can create your own astronauts for the ‘Campaign’ mode, including their own custom names and portraits. Modded astronauts are only relevant in the ‘Campaign’ mode.
2. **Resupply spacecraft:** You can create your own 3D models featuring resupply spacecraft that can be attached to the space station. Please note that modding of resupply spacecraft is only available in the ‘Sandbox’ mode.
3. **Space station modules:** You can create your own 3D models featuring existing or fictional space station modules, and define their various attributes such as electricity, energy consumption or the number of docking ports for resupply spacecraft that they provide. Modded space station modules are available in both the ‘Campaign’ and ‘Sandbox’ modes.

The Modding Tools

Pre-requisites and installation

Modding in Space Station Designer is conducted by using the Unity Editor, which can be downloaded for free. In order to be able to create your own mods, please download Unity 2021.3.26f1 (we recommend you do this via the Unity Hub, which can be found here) and open the official ‘SSX_ModdingTools’ project that we provide, which can be found as a zip file here. The current version of the modding tools is 0.5.15.

Structure of the SSX_ModdingTools Unity project

As seen in figure 1, the SSX_ModdingTools Unity project features a very simple folder structure:

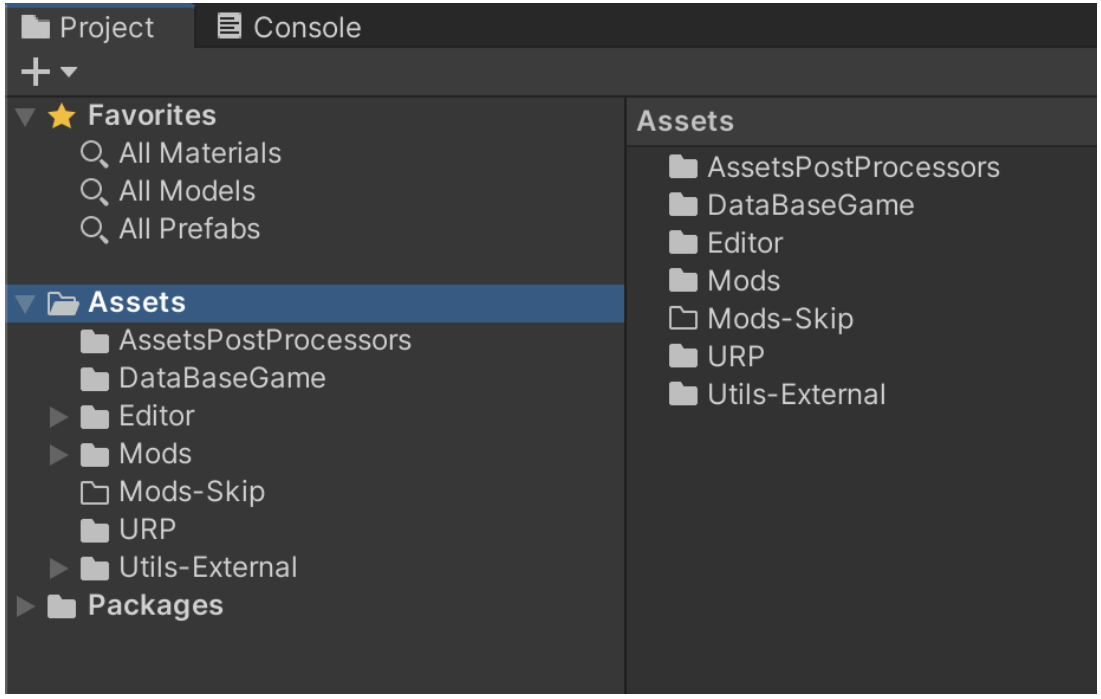


Figure 1: The SSX_ModdingTools Unity project folder structure.

Under the mods folder, there's a series of seven sub-folders (figure 2) that feature various examples that are already available in [Space Station Designer's mod.io page]:

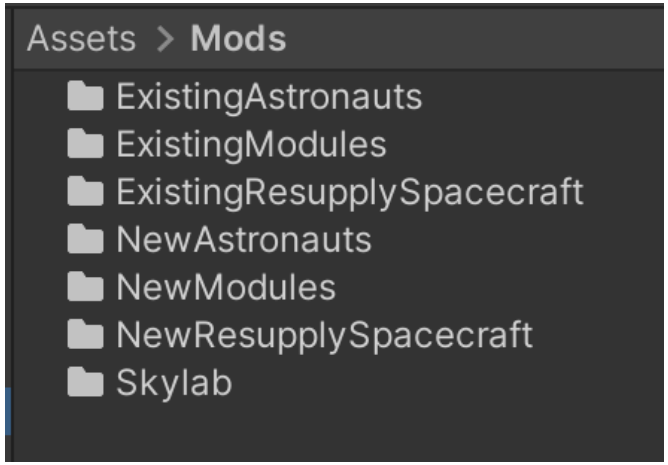


Figure 2: The various sample mods provided under the SSX_ModdingTools project.

These mods are:

- **Existing Astronauts:** This mod changes the profile pictures and names of two astronauts that ship with the base game. You can find out more about this sample in this subsection.
- **Existing Modules:** This mod replaces two existing modules (‘Cassegrain Concentrator' and 'Propulsion Module (Medium)') with a new pair featuring their own textures and materials. You can find out more about this sample in this subsection.
- **Existing Resupply Spacecraft:** This mod replaces the ‘Space Shuttle’ resupply spacecraft with a more realistic version, featuring its own set of textures and materials. You can find out more about this sample in this subsection.
- **New Astronauts:** This mod add two new astronauts from the Apollo era to your roster: Pete Conrad and Alan Bean. This includes their portraits and names and, due to the way astronauts work in Space Station Designer, they are only available in the Campaign mode. You can find out more about this sample in this subsection.
- **New Modules:** This mod adds two new space station modules: ‘Ceramics Production Module Pink’ and ‘Chemicals Production Module Red’. They aren’t part of the base game, and they come with their own stats. You can find out more about this sample in this subsection.
- **New Resupply Spacecraft:** This mod adds the Apollo resupply spacecraft to the list of options in both the Campaign and the Sandbox modes. The model comes with its own set of materials and textures, which give it a realistic style. You can find out more about this sample in this subsection.

- **Skylab:** This mod brings together elements from the previous examples and shows how to put them together as part of a single package. More specifically, it includes the 'Skylab' space station module, the 'Apollo' resupply spacecraft, and a list of all nine astronauts that were involved in the Skylab missions during the 1970s. You can find out more about this sample in this subsection.

Generating mods

All new mods need to be placed in a new subfolder under *Assets/Mods*. If you open the SSX_ModdingTools Unity project for the first time, you'll find that there's a folder for each one of the seven samples that we provide (see figure 2). These sample mods have already been uploaded to mod.io but, if they were your creations, you'd have to build them in order to upload them and share them with the community. This can be done by selecting *Space Station Designer => Mod Support => Build All Mods* from the tools menu (figure 3).

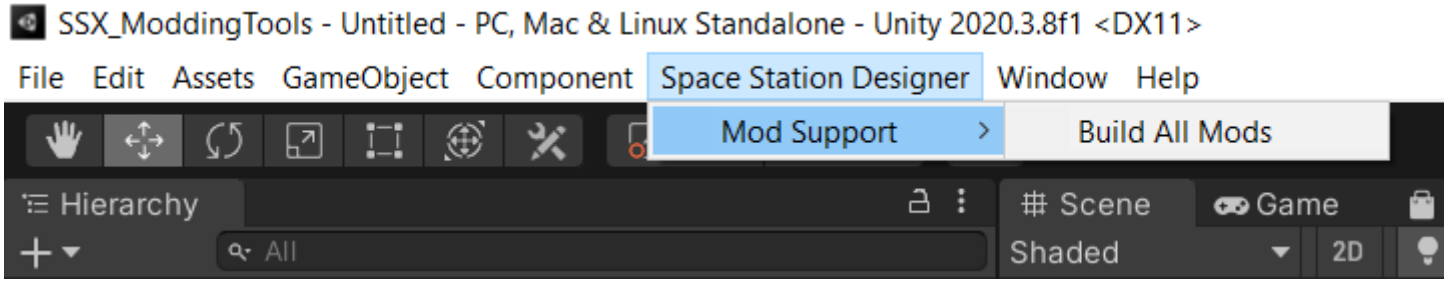


Figure 3: Tools menu for generating all mods residing under the Assets/Mods subfolder..

The build time will depend on the number of mods and the amount of data contained in each one of them. As a test, go ahead and select *Build All Mods*. A progress bar similar to the one shown in figure 4 will show up in the center of the screen.

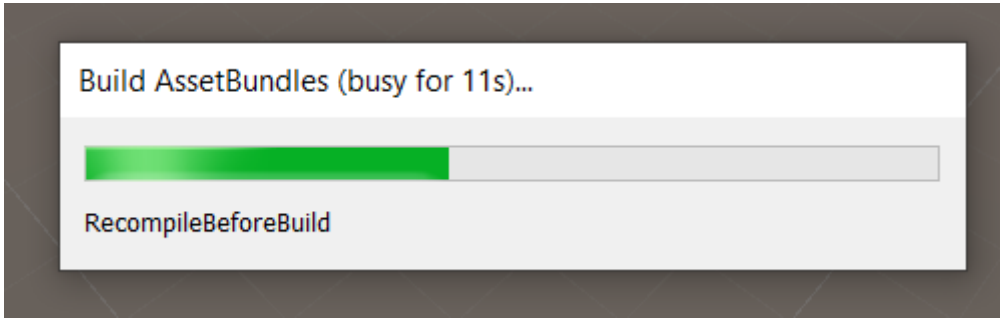


Figure 4: Progress bar shown during the mods generation process.

Once the process is over, you should be able to see a *Mods_Output* folder at the root level (figure 5), which contains one folder per mod (figure 6).

Name	Date modified	Type	Size
Assets	01/10/2021 18:49	File folder	
Library	03/10/2021 20:54	File folder	
Logs	01/10/2021 18:53	File folder	
Mods_Output	03/10/2021 16:31	File folder	
Packages	29/09/2021 22:44	File folder	
ProjectSettings	29/09/2021 22:44	File folder	
Temp	03/10/2021 20:54	File folder	
UserSettings	29/09/2021 22:44	File folder	
.vsconfig	29/09/2021 22:58	VSCONFIG File	1 KB
Assembly-CSharp.csproj	01/10/2021 18:51	C# Project file	53 KB
Assembly-CSharp-Editor.csproj	03/10/2021 16:21	C# Project file	53 KB
SSX_ModdingTools.CSharp.csproj	11/12/2018 19:02	C# Project file	8 KB
SSX_ModdingTools.CSharp.Editor.csproj	11/12/2018 19:02	C# Project file	10 KB
SSX_ModdingTools.csproj	22/01/2019 21:25	C# Project file	20 KB
SSX_ModdingTools.Editor.csproj	22/01/2019 21:25	C# Project file	24 KB
SSX_ModdingTools.sln	01/10/2021 18:51	Microsoft Visual St...	2 KB

Figure 5: Location of the Mods_Output folder.

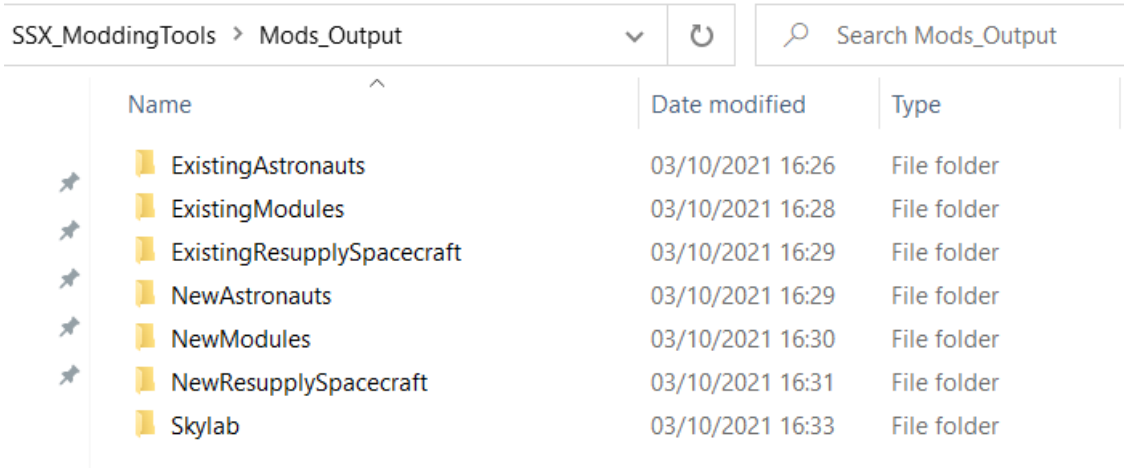


Figure 6: Contents of the Mods_Output folder after building all the provided sample mods.

Since building the sample mods may take a non-trivial amount of time, you might want to move all the sample mods folders under the *Mods_Skip* subfolder and work on a single mod at the time (see figure 7).

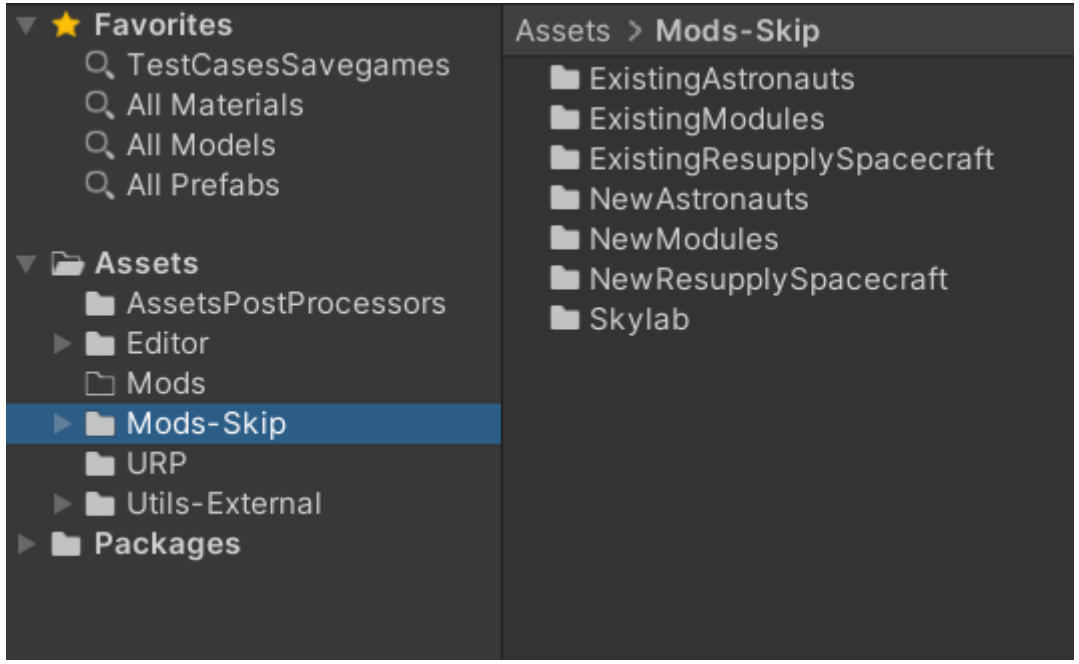


Figure 7: Location of the Mods_Skip folder.

For example, if you want to create a realistic version of [the Mir space station] (which would go really well with the modded version of the Space Shuttle, by the way!), you'd create a sub-folder named *Mir* under the *Mods* folder (figure 8):

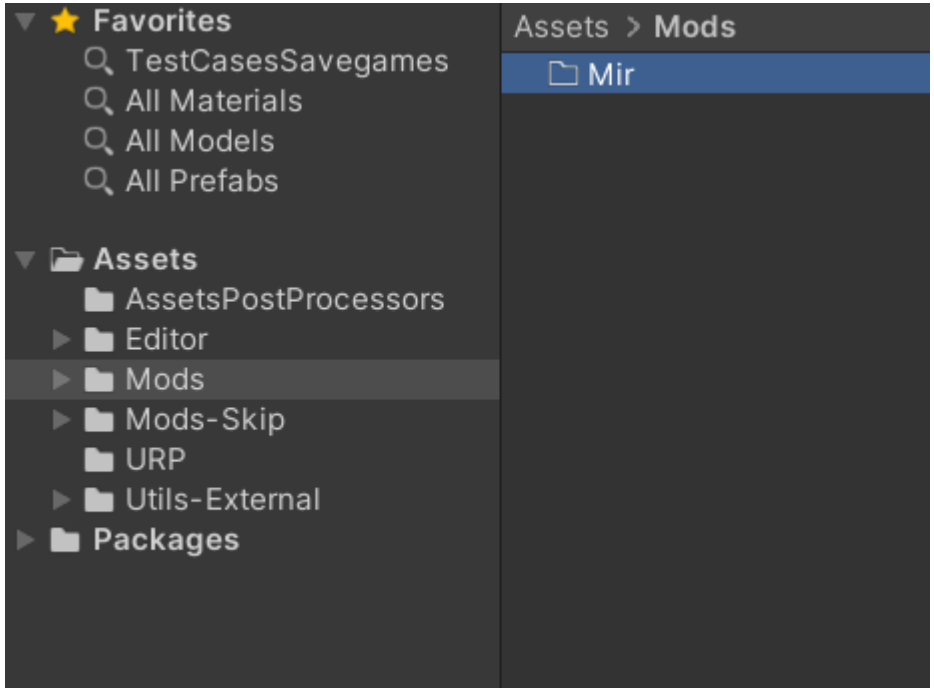


Figure 8: Adding a folder for a new mod named 'Mir'.

In the next subsections, we'll cover the details behind all the official samples, so that you can use them as a starting point for your own mods.

Modding example #1: Replacing existing astronauts

Note: Modded astronauts can only be used in the Campaign mode.

The folder for this sample is under *Assets/Mods/ExistingAstronauts* (figure 9).

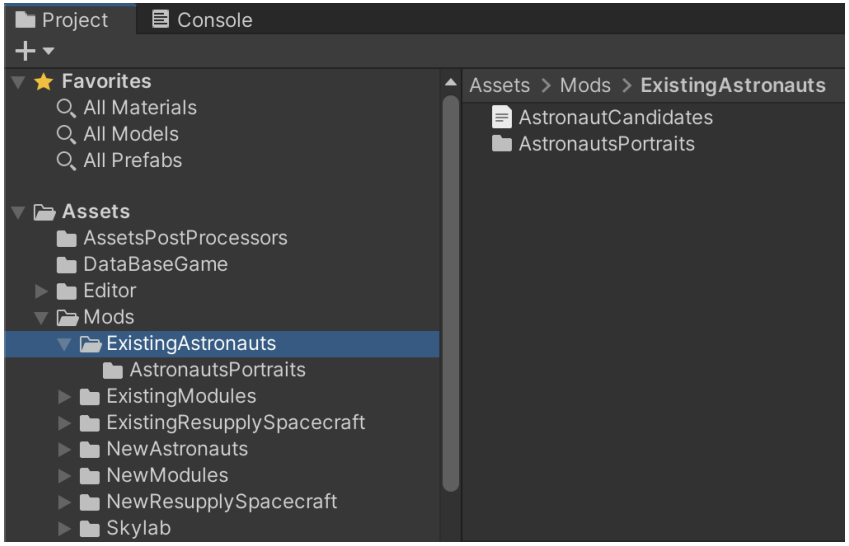


Figure 9: Contents of the *ExistingAstronauts* sample mod folder.

Replacing existing astronauts requires:

1. A subfolder named *AstronautsPortraits* featuring the portraits of all the astronauts that you'd like to replace.
    a. The portraits must adhere to the following conventions:
        i. The name should be AstronautPortrait-XYZ, where XYZ is the astronaut's ID.
        ii. The portraits must be image files (we recommend using the PNG file format) of size 165x190 px.
2. A file named AstronautCandidates.xml, featuring a unique ID for the astronauts, a name, and a gender.

If you're replacing existing astronauts, make sure you select an ID that is already in use by the game. The existing list of astronauts candidates can be found under *Assets/DataBaseGame/AstronautCandidates.xml*.

Modding example #2: Adding new astronauts

Note: Modded astronauts can only be used in the Campaign mode.

The folder for this sample is under *Assets/Mods/NewAstronauts* (figure 10).

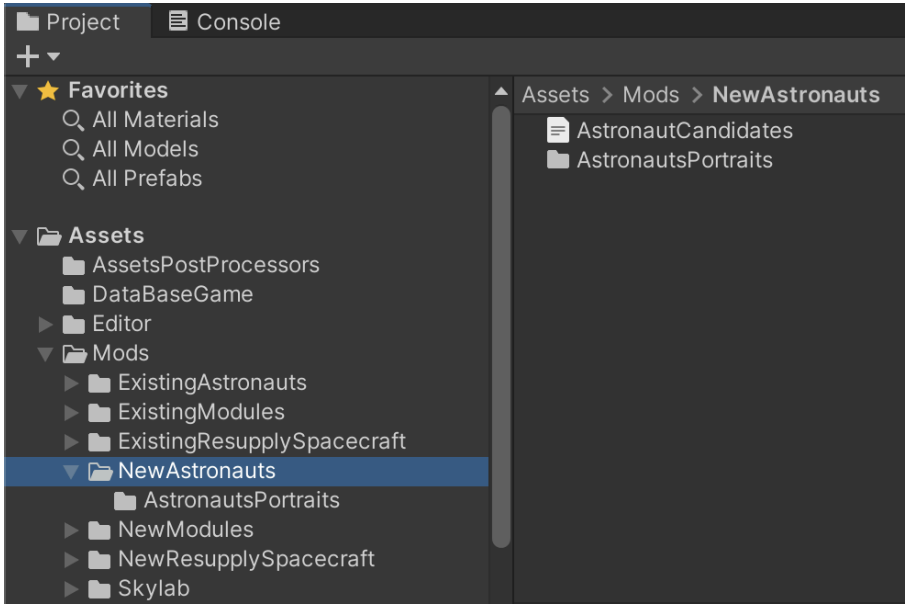


Figure 10: Contents of the *NewAstronauts* sample mod folder.

Adding new astronauts is very similar to replacing existing ones, which was described in the previous section. The only restriction is that you need to use an ID that is not already in use by the game. Please note that if the player selects multiple mods and they happen to be using the same IDs for their astronauts, the behaviour of the game will be undefined.

Modding example #3: Replacing existing space station modules

The folder for this sample is under *Assets/Mods/ExistingModules* (figure 11).

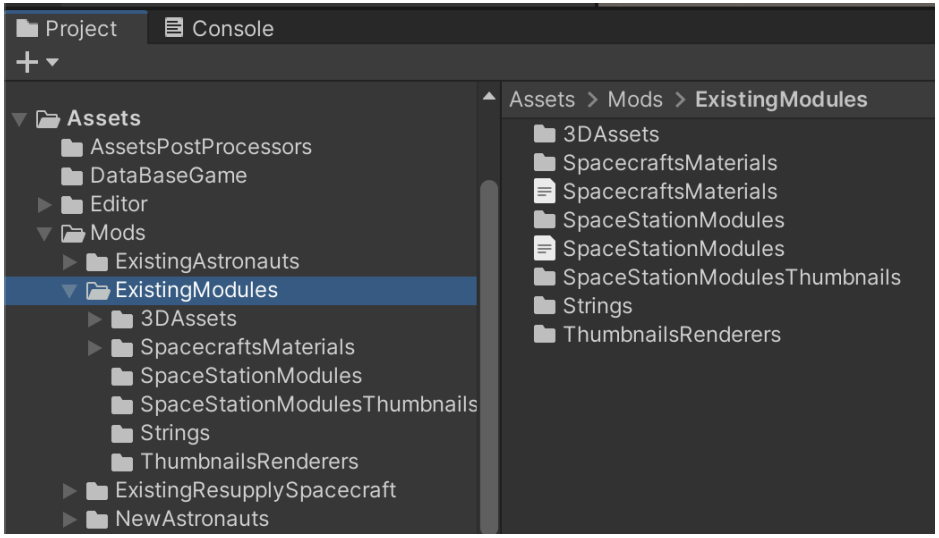


Figure 11: Contents of the *ExistingModules* sample mod folder.

The mod provides replacements for the 'Cassegrain Concentrator' and 'PropulsionModule (Medium)' modules featuring more realistic materials and is comprised by the following folders:

- **3DAssets:** Contains the FBX files for both modules. The contents of this folder are ignored when generating the output for this mod.
- **SpacecraftsMaterials:** Contains all the materials used by this mod. The details behind materials will be discussed in a future subsection.
- **SpaceStationModules:** Contains all the Unity Prefabs for the modules included in this mod. The details behind how to assemble these Prefabs will be discussed in a future subsection.
- **SpaceStationModulesThumbnails:** Contains thumbnails images for all the modules included in this mod. The details behind how to automatically generate these images will be discussed in a future subsection.
- **Strings:** Contains the various strings files required by this mod.
- **ThumbnailsRenderers:** Contains a Unity scene and its associated script file. This Unity scene is used in order to generate the thumbnails inside the SpaceStationModulesThumbnails subfolder. The details behind how to configure this scene will be discussed in a future subsection.

You can also find the following data files at the root level:

- **SpacecraftsMaterials.xml:** Defines which materials need to be used for each modded spacecraft depending on its current state. The details behind this file will be discussed in a future subsection.
- **SpaceStationModules.xml:** Defines the attributes for the modded space station modules. The details behind this file will be discussed in a future subsection.

Anatomy of a space station module and creation of Prefabs

The FBX files for both modules provided by this mod can be found under *Assets/Mods/ExistingModules/3DAssets/SpaceStationModules* (see figure 12)

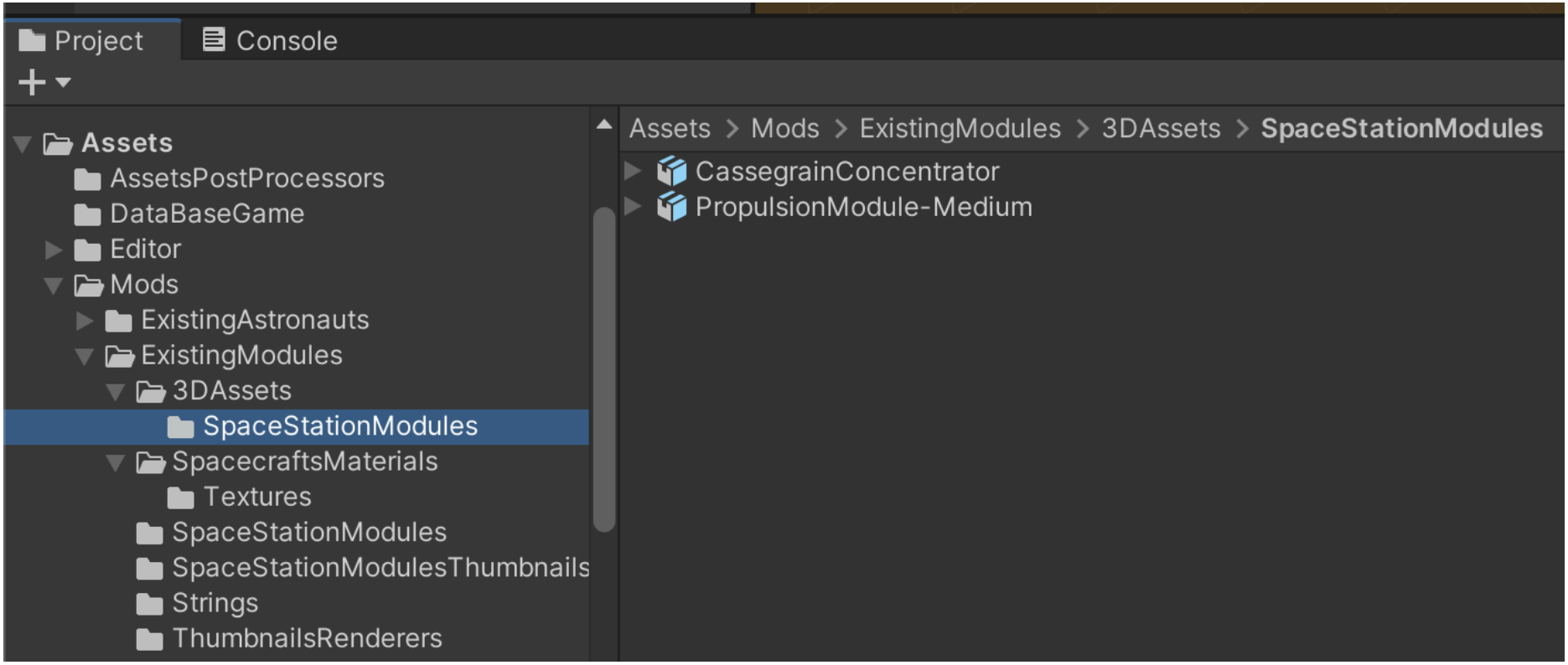


Figure 12: The FBX files under the *Assets/Mods/ExistingModules/3DAssets/SpaceStationModules* folder.

We can analyze the hierarchical structure of the CassegrainConcentrator module by dropping it into an empty scene (figure 13):

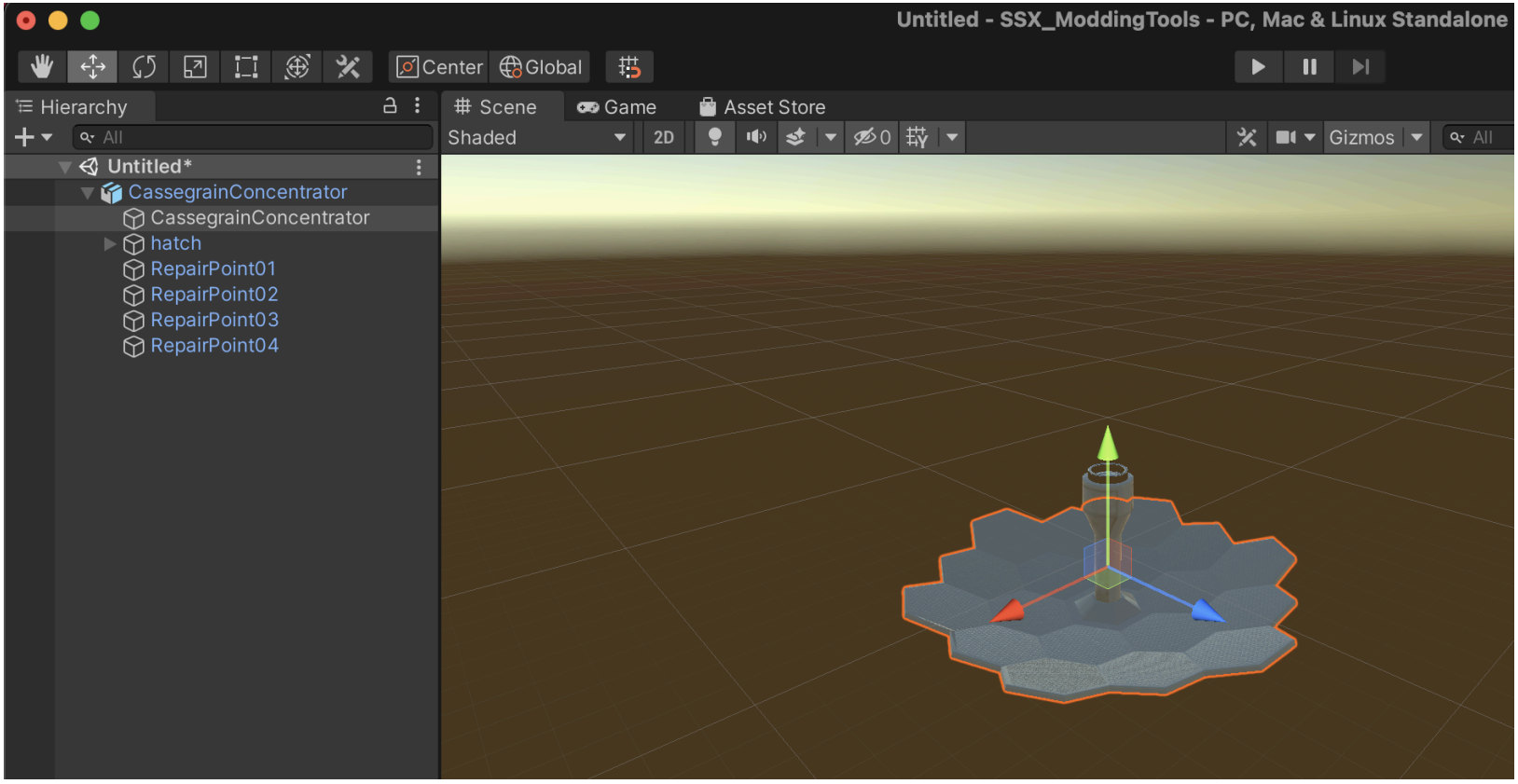


Figure 13: Hierarchy of the 'CassegrainConcentrator' module.

Modules typically adhere to the following structure:

- One or more sub-nodes featuring a mesh. In this case, these are *CassegrainConcentrator* and *hatch*.
- Four sub-nodes named *RepairPoint01*, *RepairPoint02*, *RepairPoint03*, and *RepairPoint04*. These nodes are optional, and are used in order to determine the positions of the animated astronauts when showing the module repair animated sequence in the Campaign mode.

In order to be able to be used by the game, they need to be converted to Unity Prefabs first. This can be accomplished by:

- Adding as many collider components to the main mesh node as needed:
    - These colliders are used in order to determine if the module is colliding with other modules and/or resupply spacecraft.
    - In the case of the CassegrainConcentrator, two box colliders are enough in order to create a bounding volume around its geometry (see figure 14).
- Adding a RigidBody component to the main mesh node (see figure 15).
- Adding and position a DockingPort node. They Y-axis (i.e., the green one) should be pointing outwards (see figure 16).
- (Optional) If the source FBX file doesn't have them, you can manually add RepairPoint nodes, which will be used when showing the module repair animated sequence in the Campaign mode.

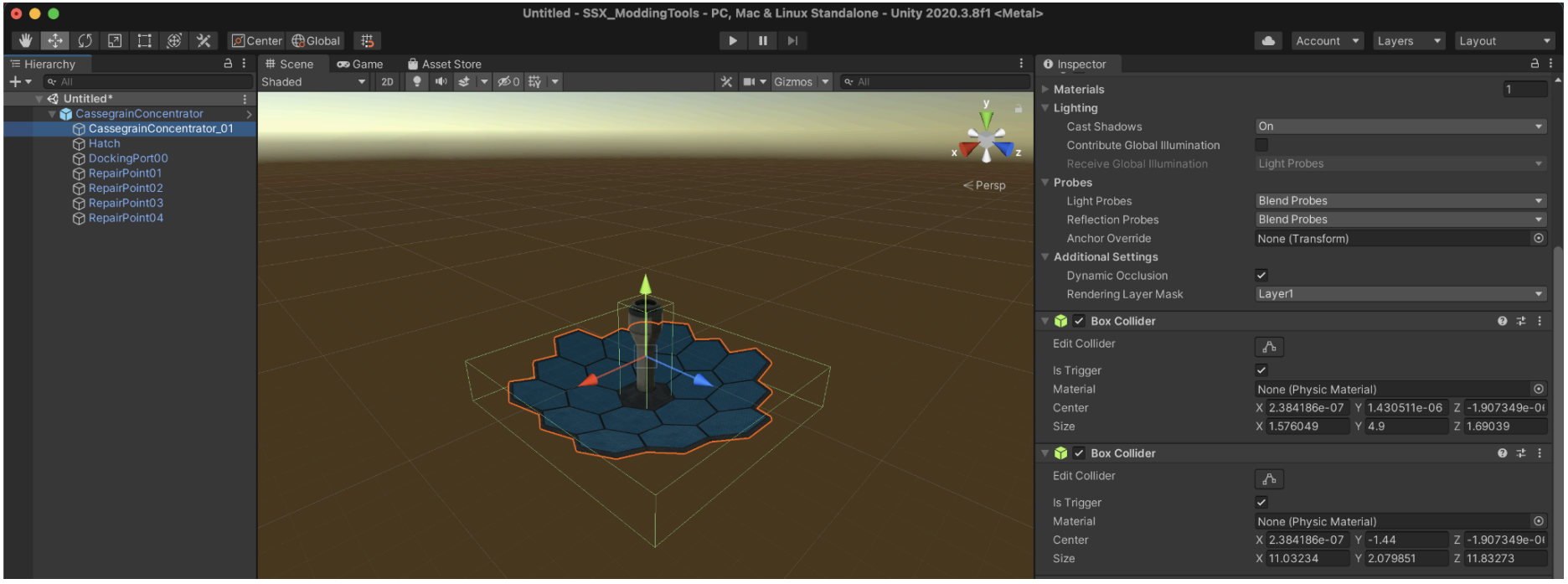


Figure 14: Box colliders attached to the 'CassegrainConcentrator' module.

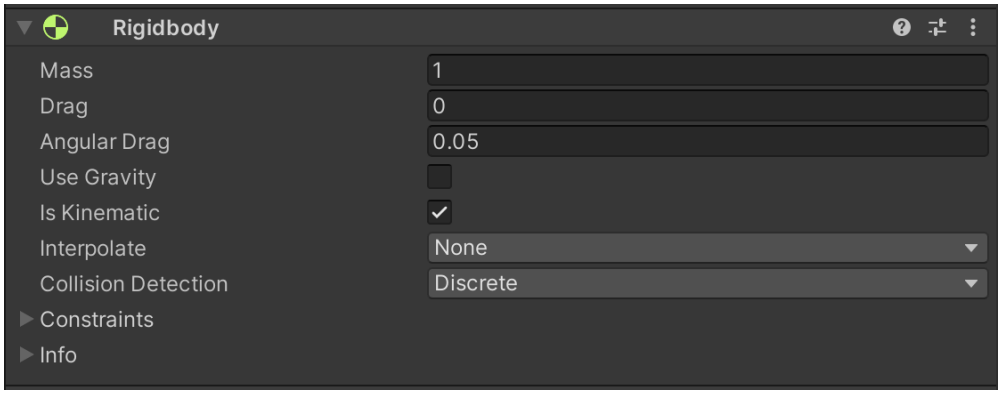


Figure 15: The RigidBody component attached to the main mesh node.

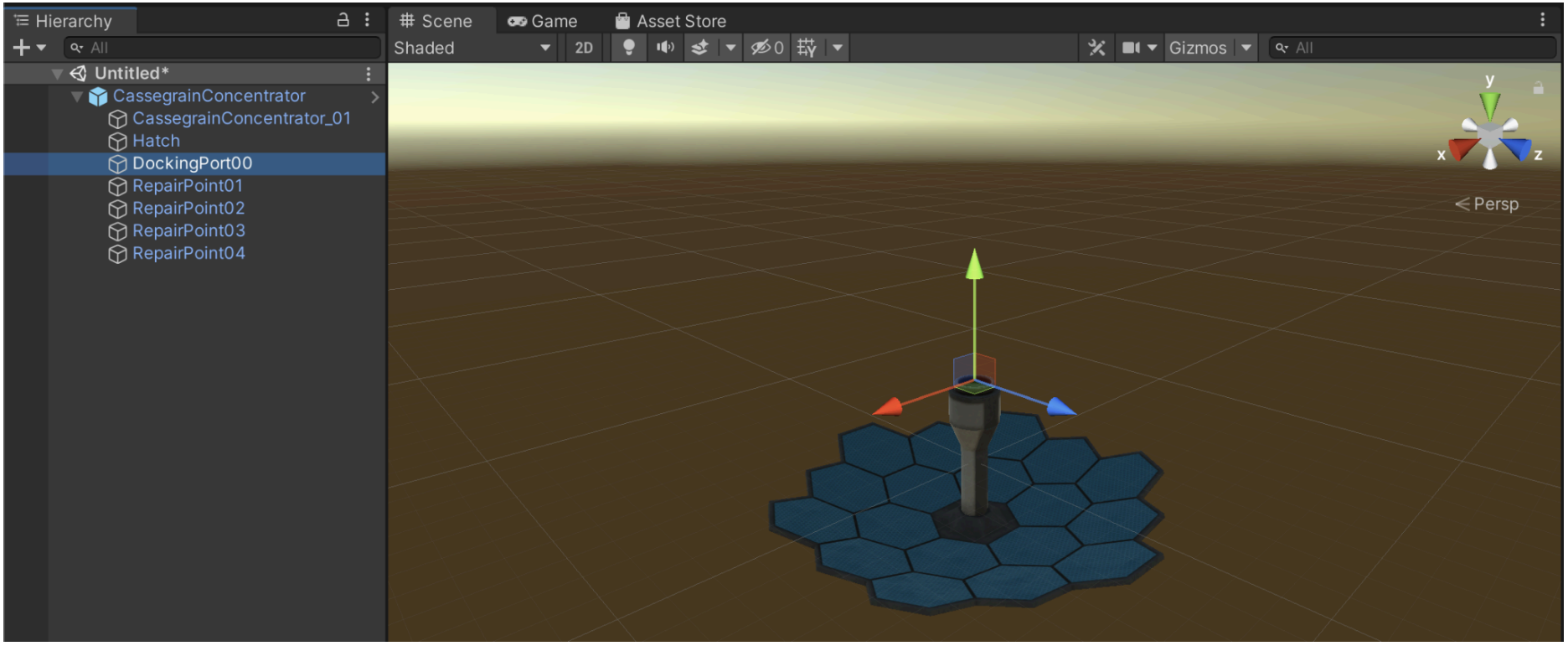


Figure 16: The only docking port featured by the 'CassegrainConcentrator' module.

Once you've finished adding all these various properties to the module, drag the GameObject into the *SpaceStationModules* subfolder, which will create a Unity Prefab (see figure 17) that can be processed by the game.

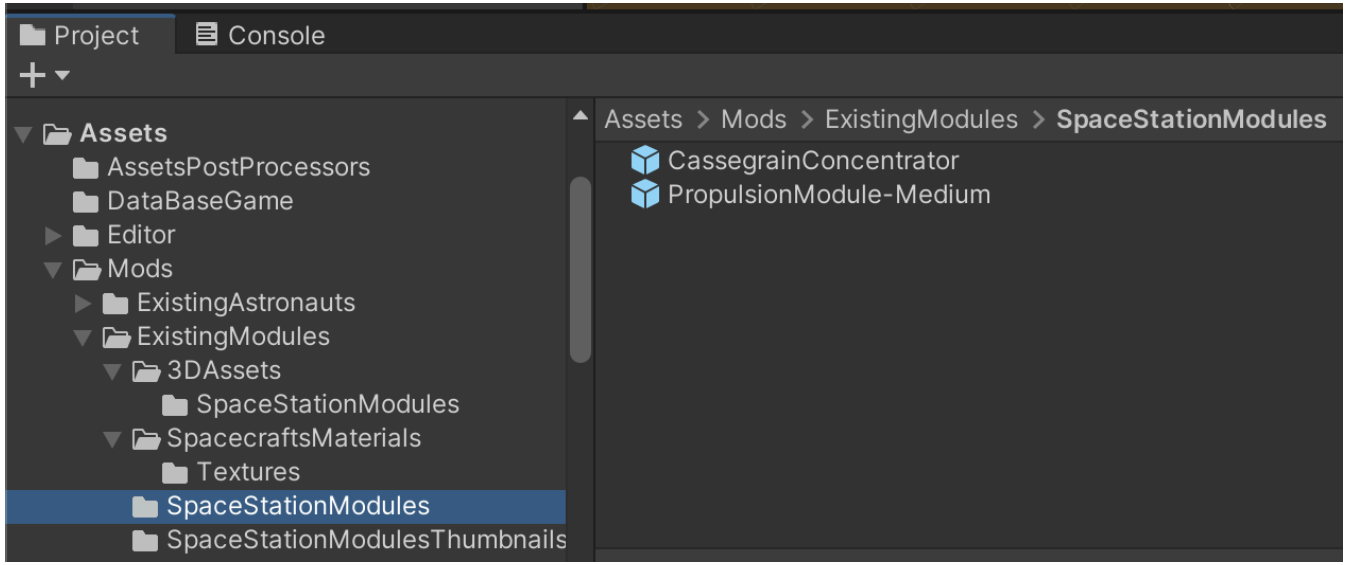


Figure 17: Space station modules prefabs under the 'ExistingModules' sample mod.

Creating and configuring spacecrafts materials

Modules can have one of the following different states depending on the context:

- **Active:** This state is used when a module is active and operational.
- **Inactive:** This state is used when a module is manually deactivated by the player.
- **InactiveDueToMissingConnectionWithCoreModule:** This state is used when the connection route between a module and the station's Core Module has been interrupted due to a malfunctioning module in the path.
- **InactiveAndAssignedToProductionLine:** This state is used when a manufacturing module belongs to a production line and has been manually deactivated by the player.
- **ExperiencingFailure:** This state is used when the module has experienced a failure and needs to be repaired.
- **Inoperative:** This state is used when a module that has experienced a failure couldn't be repaired.
- **PartiallyInoperative:** This state is used when a module that has experienced a failure has been partially repaired.

- **BeingReplaced:** This state is used when a module that either has failed and couldn't be repaired or that has extended past its service life is being replaced.
- **ReachedEndOfLifetime:** This state is used when a module has extended past its service life.
- **SelectedAsPartOfAProductionLineOrLease:** This state is used when selecting modules that are going to be included in a lease contract or a production line.
- **BeingPlacedAndInstalled:** This state is used when placing a new module in the station.
- **Colliding:** This state is used when the module that is being installed is colliding with either another module or a docked resupply spacecraft.
- **Hover:** This state is used when the mouse cursor is hovering over a module.
- **NotSuitableForProductionLineLeaseOrSpaceTouristsGroup:** This state is used when selecting modules that are going to be included in a lease contract or a production line in order to convey the fact that they are not suitable for selection.
- **PartOfExistingProductionLineOrLease:** This state is used when selecting modules that are going to be included in a lease contract or a production line in order to convey the fact that they are already part of an existing production line or a leasing contract.

The game provides enough flexibility to define unique materials for every modded spacecraft (i.e., space station module or resupply spacecraft) for every state. This is very rarely needed, though, but the option is there in case you want to be very granular when it comes to specifying materials.

Note that the spacecraft that come with the game make use of a single material, but the modding system provides enough flexibility in order to allow the use of multiple materials. Please refer to the materials file under *Assets/Mods/ExistingModules/SpacecraftsMaterials.xml* and use it as a starting point for your own mods.

Generating the spacecrafts thumbnails

The thumbnail images for all the various spacecrafts (i.e., space station modules and resupply spacecraft) are stored under the *Assets/Mods/ExistingModules/SpaceStationModulesThumbnails* folder, which contains a set of PNG files of 327x172 px.

The thumbnails are generated automatically by a custom tool that is under the *Assets/Mods/ExistingModules/ThumbnailsRenderers* folder. In this folder, there's a Unity scene named *ThumbnailsRendererSpaceStationModules_ExistingModules.unity* which makes use of a script file named *ThumbnailsRendererSpaceStationModules_ExistingModules.cs* (see figure 18).

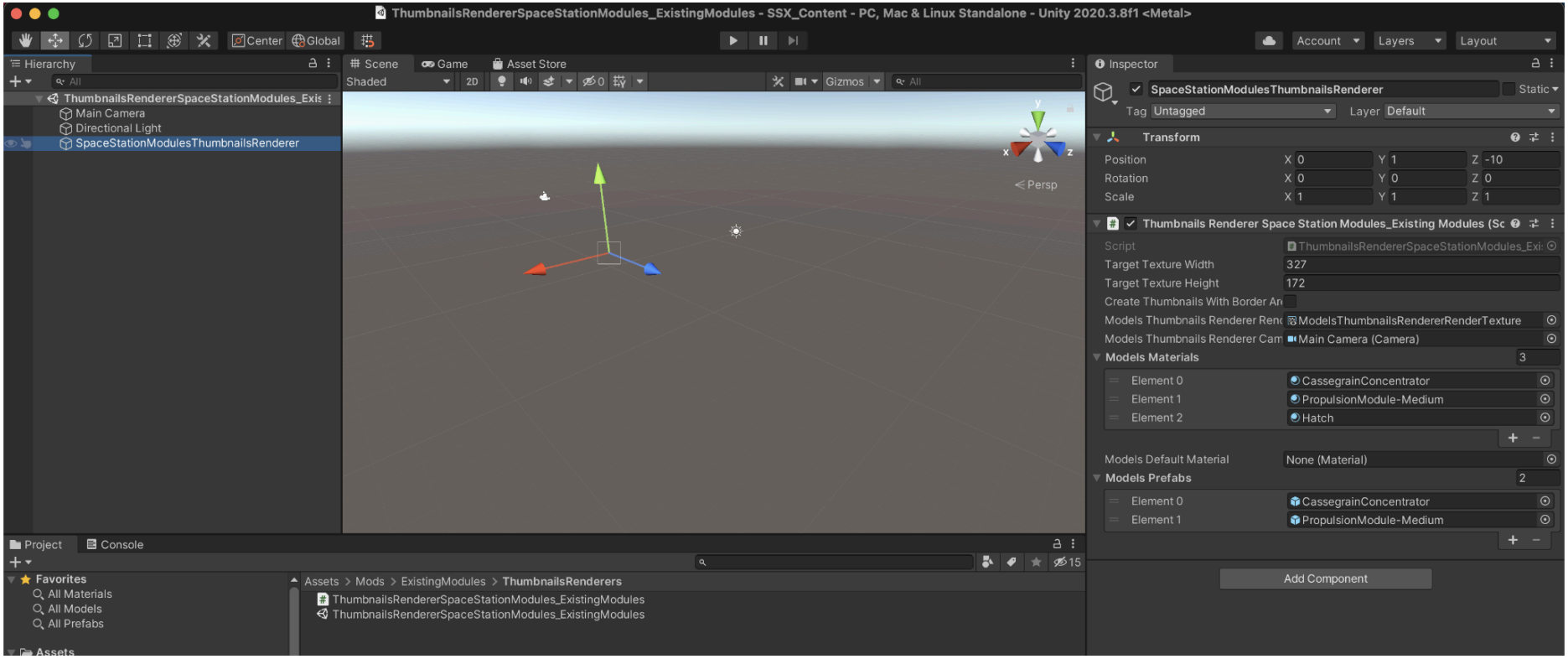


Figure 18: The *ThumbnailsRendererSpaceStationModules_ExistingModules.unity* sample scene.

By entering play mode, the tool will go through the list of 'Models Prefabs' in the SpaceStationModulesThumbnails and create a thumbnail for each model. The output folder for the PNG files, along with position and rotation of each individual spacecraft, is defined in the *ThumbnailsRendererSpaceStationModules_ExistingModules.cs* file, which you can use as a starting point for your mods.

(Optional) Editing the modules' stats and their display names

By default, modded spacecraft (i.e., space station modules and resupply spacecraft) will use the stats and display names defined by the base game. However, if you want to, you can change either of them when creating mods. More specifically:

- The stats for space station modules are under *SpaceStationModules.xml* file. For reference, the stats of all the modules that come with the base game can be found under *Assets/DataBaseGame/SpaceStationModules.xml*. In this sample mod, we've modified the cost of the 'Cassegrain Concentrator' and made it cheaper than in the base game.
- The display names for space station modules are under *Strings/Strings-EN-ExistingModules*. There's no need to specify strings when modifying existing modules but, as an example, we've changed the display name of the 'Cassegrain Concentrator'.

Modding example #4: Adding new space station modules

The folder for this sample is under *Assets/Mods/NewModules* (figure 19).

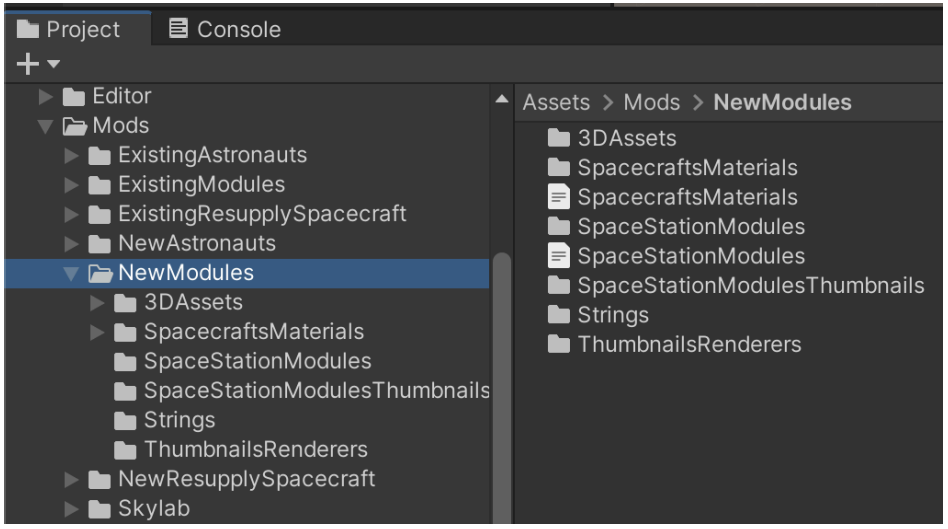


Figure 19: Contents of the *NewModules* sample mod folder.

The process for creating new modules is very similar than the one used when replacing existing ones, which was covered in a previous subsection. You'll also need to:

- Add the various stats associated to each module under the *SpaceStationModules.xml* file. For reference, the stats of all the modules that come with the base game can be found under *Assets/DataBaseGame/SpaceStationModules.xml*.
- Add the display names of all the modules under a strings file. This file should be located under *Assets/Mods/MyModFolder/Strings*. For the moment, Space Station Designer only supports the English language. Make sure you name your file as *Strings-EN-XYZ.xml*, with XYZ being the name of the mod (in this example, the file is named *Strings-EN-NewModules.xml*).

Modding example #5: Replacing existing resupply spacecraft

Note: Modded resupply spacecraft are available in both the Sandbox and Campaign modes. However, the number of crew seats cannot be changed. Doing so will result in undefined behaviour.

The folder for this sample is under *Assets/Mods/ExistingResupplySpacecraft* (figure 20).

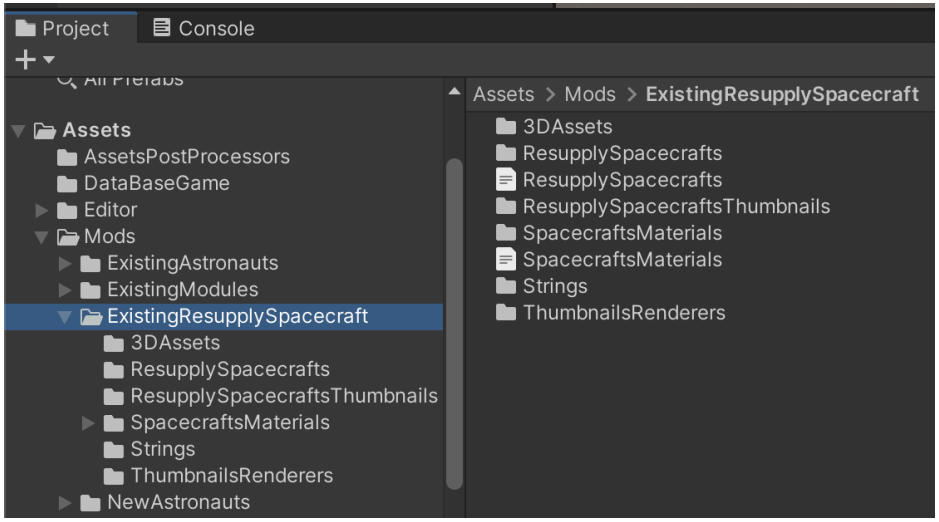


Figure 20: Contents of the *ExistingResupplySpacecraft* sample mod folder.

The process for modding existing resupply spacecraft is very similar to the one that needs to be followed for space station modules, which was covered in this section. More specifically:

- Prefabs of the resupply spacecraft need to be added under the *ResupplySpacecrafts* folder.
- Thumbnails of the resupply spacecraft need to be added under the *ResupplySpacecraftsThumbnails* folder. The process for creating these thumbnails is the same than the one used for space station modules. Please refer to the *ThumbnailsRenderers* folder in order to understand how this was achieved for this sample mod.
- An XML file named *ResupplySpacecrafts.xml* **must** to be provided. This file contains the various stats of the modded resupply spacecraft. Please note that it's perfectly possible to omit the stats of any modded resupply spacecraft. However, the *ResupplySpacecrafts.xml* file is not optional and must be included, even if it doesn't feature any stats. For reference, the stats of all the resupply spacecraft that come with the base game can be found under *Assets/DataBaseGame/ResupplySpacecrafts.xml*.

Modding example #6: Adding new resupply spacecraft

Note: New modded resupply spacecraft (i.e., resupply spacecraft that doesn't replace one that comes with the base game) only show up in the Sandbox mode. They will be ignored in the Campaign mode.

The folder for this sample is under *Assets/Mods/NewResupplySpacecraft* (figure 21).

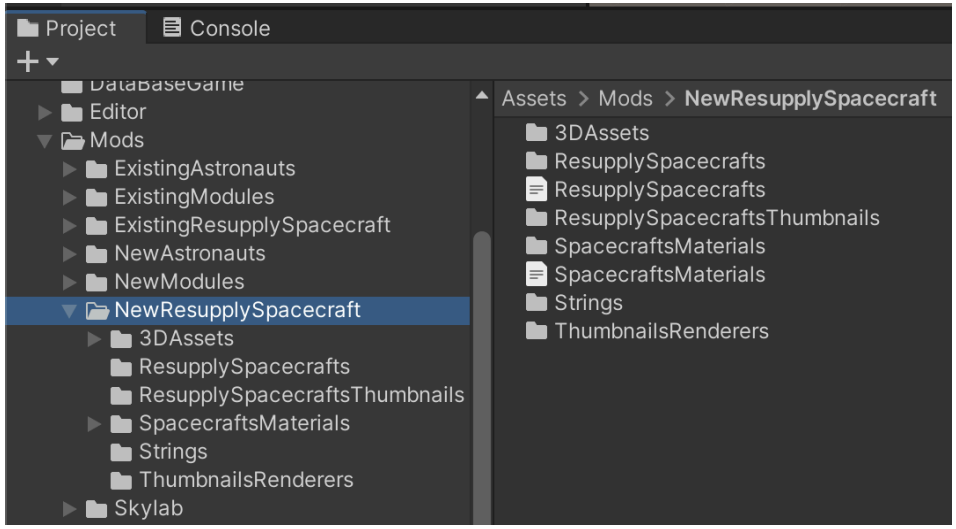


Figure 21: Contents of the *NewResupplySpacecraft* sample mod folder.

The process for adding new resupply spacecraft is similar to the one used for modifying existing ones, which was covered in a previous section. This sample mod adds the 'Apollo' resupply spacecraft, which features a high-poly model with a realistic material. The spacecraft has four submeshes that require different materials. Please refer to the *Assets/Mods/NewResupplySpacecraft/SpacecraftsMaterials.xml file* in order to understand how to setup a resupply spacecraft featuring multiple materials.

Modding example #7: Tying it up all together in Skylab

The folder for this sample is under *Assets/Mods/Skylab* (figure 22).

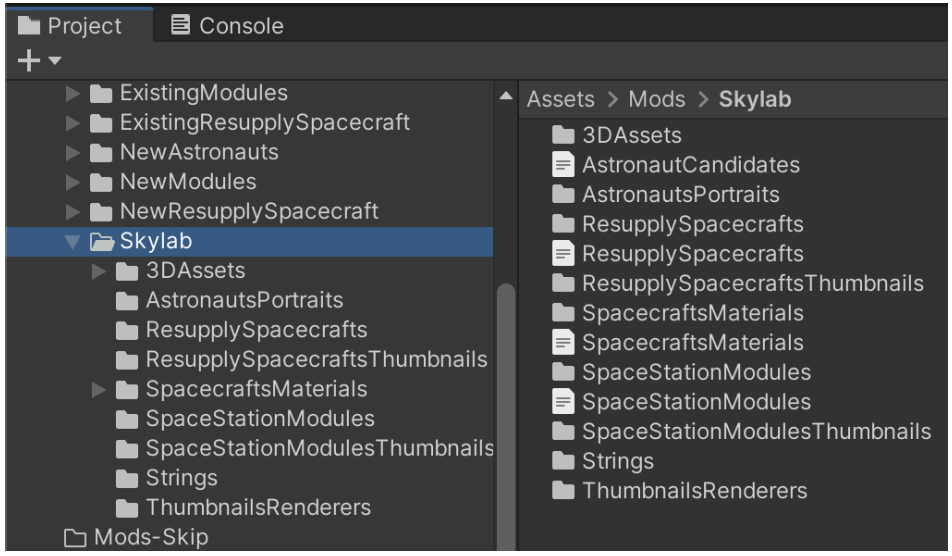


Figure 22: Contents of the *Skylab* sample mod folder.

This sample mod shows how to add a new resupply spacecraft (Apollo), a new space station module (Skylab), and various astronauts as part of the same mod package. The Skylab mod is just a compilation of existing mods and doesn't add anything new, but it's worth including it as a reference so that you can use it as a starting point for your own ambitious creations.

Conclusion

This modding guide has provided all the information required in order to tweak existing elements (astronauts, space station modules and resupply spacecraft) of Space Station Designer and add new ones. If you find any mistakes in this guide and/or have any feedback, please let us know by using any of the following channels:

- The ‘Space Station Designer’ forums on Steam.
- The Space Station Designer forums on itch.io.
- Twitter (@polarmotion).