Construction Safety Assets

Preface:

This is not an all-encompassing guide for performing 3D modelling. It merely depicts design choices, reasoning for those choices, and provides small basic ideas to encourage good development of models.

Blender:

Blender 2.8+ was chosen as the primary platform for extracting, editing, and importing models into the Unity project as it is entirely free to use. Further, the tools internal to Blender are professional level, and many independent films have been produced with the software that rival the visual fidelity produced by such prominent film companies as DreamWorks and Pixar. Blender has excellent tools for working with meshes, developing materials for the meshes, performing scene construction, generating animations, post-processing effects, and creating the final renditions be they either photos or movies. We use a small subset of these functions.

Creating Individual Assets:

Blender provides tools for importing various types of models into its internal representation. This was one of the primary reasons for its selection. The project had a mock-up of a construction scene built within SketchUp, and we wanted to use those same assets within the simulation environment. However, SketchUp models cannot be directly imported into Unity, and the mock-up environment was created in such a way that the individual assets could not be split into separate components, as SketchUp would only export a singular mesh model. We needed a was to split this mesh into separate components for import into the simulation in Unity. So, the entire mock-up scene was imported into Blender using the '.fbx' format.

The mesh editing tools of Blender were used to take the SketchUp model and break it down into individualized models that we could use for importing into Unity. To perform this, we entered 'Edit' mode in Blender for the very large object, selected those components from smaller objects (like vehicles, construction equipment, etc.), and then split those geometries away from the larger one using the 'Split' functionality provided by Blender in 'Edit' mode. These split off geometries were then renamed and exported to additional Blender files that only contained the individualized construction equipment components we needed for importing into Blender. This same process can be carried out in the future with any other large 3D modelled scenes, provided those scenes can be imported into Blender in some fashion.

Sub-Dividing Individual Models:

For models that have mechanical elements such as wheels, pistons, or any other system of components that move relative to each other, these models need to be split into sub-components. For example, on a civilian vehicle it may be expected that the wheels rotate about their axis when the

vehicle is in motion. If the meshes were left as a single entity for the model this would be next to impossible to perform. In these instances, the models were further broken down into sub-components that represented the mechanical systems and their relations to each other.

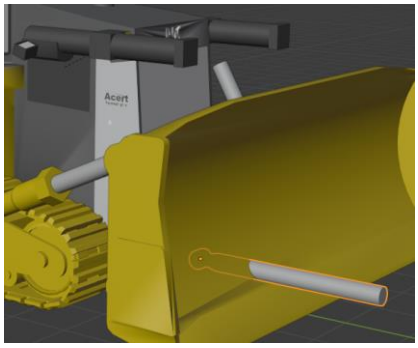
Note that this implies that there may be a hierarchical structure for the reference frames of each of these components. This is synonymous with skeletal structures in organic models. As an example, a finger is connected to a hand, which is connected to a wrist, which is connected to a lower arm, and so on. In these cases, the models are split into the separate armature components, and then are placed in this hierarchical tree type structure by parenting their reference frames to the reference frame that they are attached to.

Doing this stipulates that certain aspects of the models need to have specific attention paid to them. Since the sub-models are in a tree structure, we need to pay particular attention to the rotation and scale of the reference frame. So, if you are making alterations to the sub-model through its frame, ensure that you apply the modification to the mesh when complete. This resets the model to a 'unity' reference frame, which is key in having models that will respond correctly in Unity. This can be done in Blender when in 'Object' mode by selecting the sub-model and looking at the Rotation and Scale parameters for it. The rotation should be zero, and the scale should be one. You can apply the rotations and scales by selecting the object and pressing 'ctrl+a'. This will bring up a popup window that will allow you to do so.

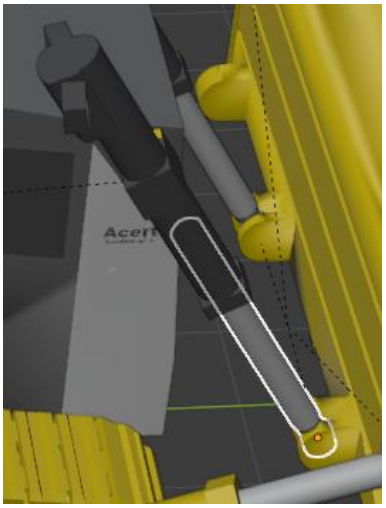
Piston Sub-Models:

For the piston models, we can apply constraints to both the outer housing and the inner shaft such that each component will always remain axially aligned to the other when in movement. However, to get this to work, **you must ensure that the sub-model is axially aligned to its reference frame**. This is critical in ensuring that this constraint will work in both Unity and Blender. This means that, when the constraints are not applied, the sub-models for the components of the systems will be axially aligned with the World reference frame, like the whole of the model. This will look very strange as the piston will not be aligned properly. But we are concerned with how the model will align when the constraints are applied, and for this to work properly they must be axially aligned to their respective reference frames.

If you have questions about this, open the Dozer model in Blender. You will see that the pistons have constraints for their parts. You can turn the constraints off and see what the axial alignment looks like. Here is an example of a piston with the constraints turned off:



and the constraints turned on:



Importing to Unity:

It is recommended to export the Blender models to a '.fbx' format file, and then import this into Unity. This is not a requirement. Unity has the capability of importing Blender files into its framework by default. However, internally Unity actually converts the Blender file to '.fbx' format in the process. The .fbx format is a smaller file size than the .blend format Blender uses, and so it is recommended to use the .fbx file format to limit the size of the ConstructionSafetySimulator Git project. There is no benefit to having the Blender files within this project, as we have the separate repository ConstructionSafetyAssets for tracking the externally developed assets.

Additionally, certain aspects of the model developed within Blender will not translate to Unity. The 3D meshes, reference frames, bone structures, and animations will transfer into Unity. Both Unity and Blender have a graphical input system designed to make materials in the context of a system of mathematical components that represents a graph. These are not compatible with each other. This means if you develop one of these in Blender, it will not transfer to Unity. In general, complicated materials from Blender will not transfer to Unity on a one-to-one functionality. They may need some touch up in Unity, or the development of an entirely new material. Constraints applied to the models in Blender will also not transfer to the models in Unity. This means you will need to reapply them using Unity's system of constraints. This does not mean you should not bother applying constraints in Blender. On the contrary, you should absolutely do this so you can test the application of a constraint on aspects of the model and ensure they work properly before moving them into Unity. This is a time saver.