Mechanical Constraints for Construction Equipment Prefabs

Description: some of the assets within the Construction Safety Simulator have mechanical aspects associated with their models. It was found that when trying to make animations it was difficult to do so as these mechanical systems did not have an intuitive way of interfacing with them. As a result, object constraints were applied to these systems in order to constrain their allowable behavior. This provided the more intuitive interface for the models.

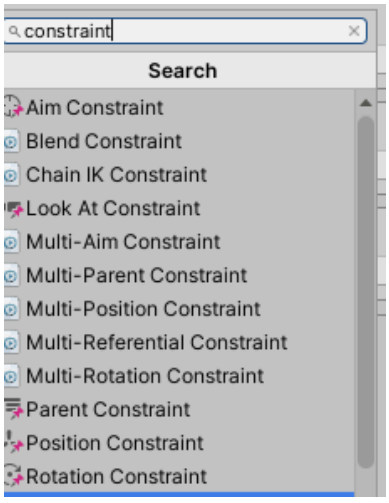
List of Affected Objects:

1. BackHoe
2. Dozer
3. DumpTruck
4. DumpTruck_Mini
5. FlatBed_A
6. FlatBed_B
7. Grader_B
8. PipeBender
9. PipeCrane_B
10. SemiTruck_A
11. Truck_A
12. Truck_B
13. Truck_C

Applying Constraints:

Constraints are simple to apply. Simply pick the prefab that you want to apply a constraint to. Open the prefab for editing. If this prefab is a direct import, I recommend adding it to a scene temporarily, unpacking the object completely, then making a new prefab from it. Rename the old prefab import afterward. This will enable you to add new elements to the newly created prefab.

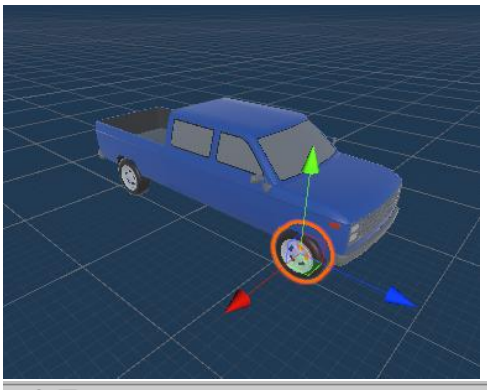
Once the prefab is open for editing, pick the desired mechanical element to constrain. Then on the inspector view of the element select 'Add Component' and type in 'Constraint'. This will generate a list of constraints that you can apply to a component:

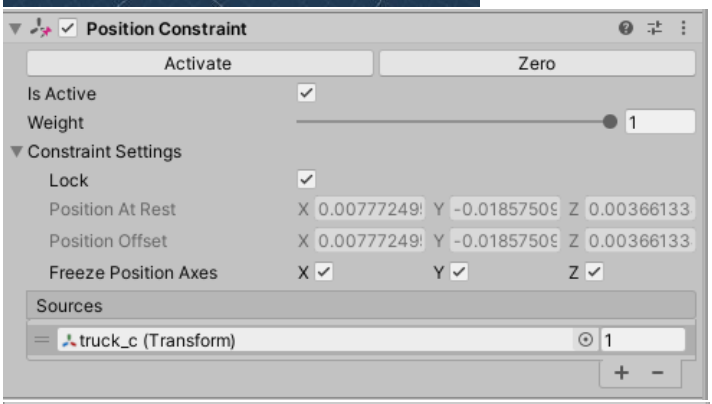


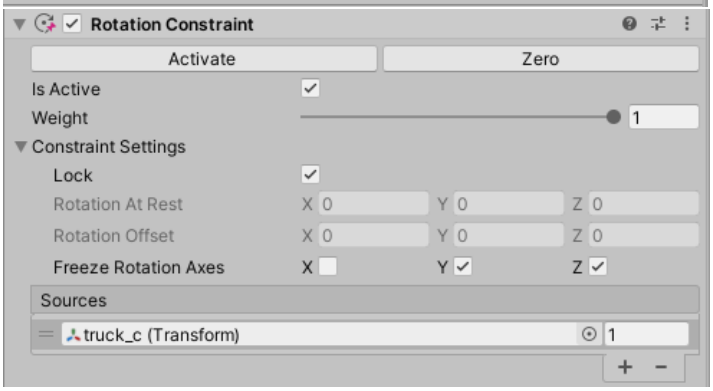
When the constraint has been added, apply the necessary data and save the object. I would recommend using prefabs for this. Applying the constraints to the prefabs means that the constraints will exist across all instances of the prefab in the simulation. Prefabs are like classes, and the objects in the scenes are instances of those prefabs.

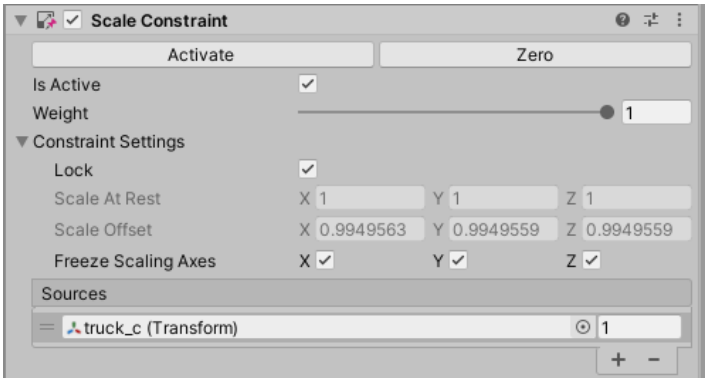
Examples:

1. **Wheels:** no translation or scaling, limit rotation to single or double axis. Use parent as reference transform. In this example this is the main truck assembly. Here rotation is allowed along the x-axis.

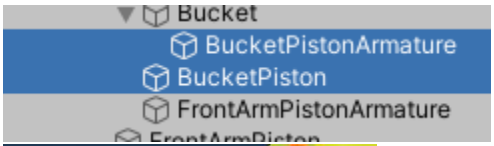


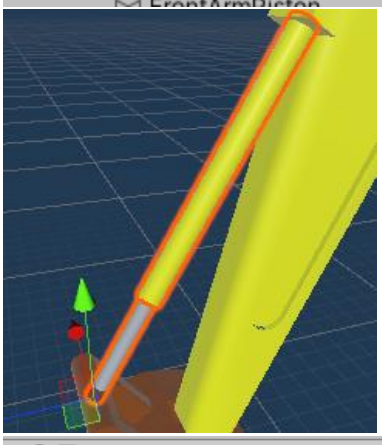


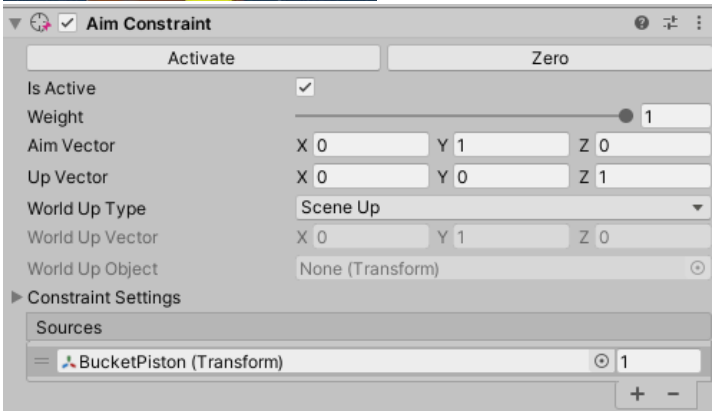


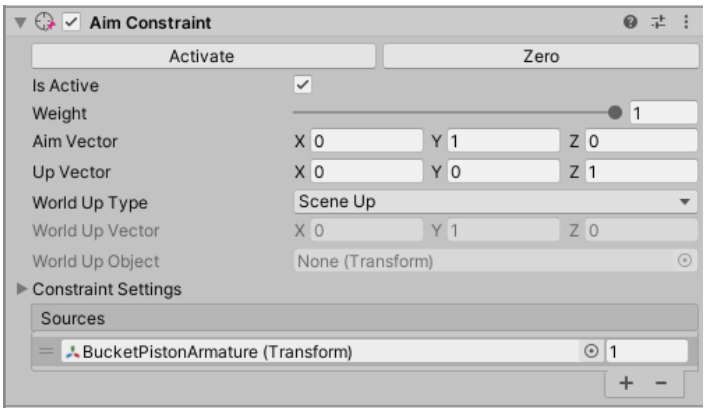


2. **Pistons:** Pistons have two elements: the inner shaft and the outer housing. They must continuously point to one another while the machine moves. This is an example of a piston setup. Each of the elements has an 'Aim Constraint' that forces it to point at the other element continuously. Note that the Aim and Up vectors need to be set properly for the constraints to work.








Aim Constraint
Activate
Zero
Is Active
Weight
1
Aim Vector
X 0
Y 1
Z 0
Up Vector
X 0
Y 0
Z 1
World Up Type
Scene Up
World Up Vector
X 0
Y 1
Z 0
World Up Object
None (Transform)
Constraint Settings
Sources
BucketPistonArmature (Transform)
1