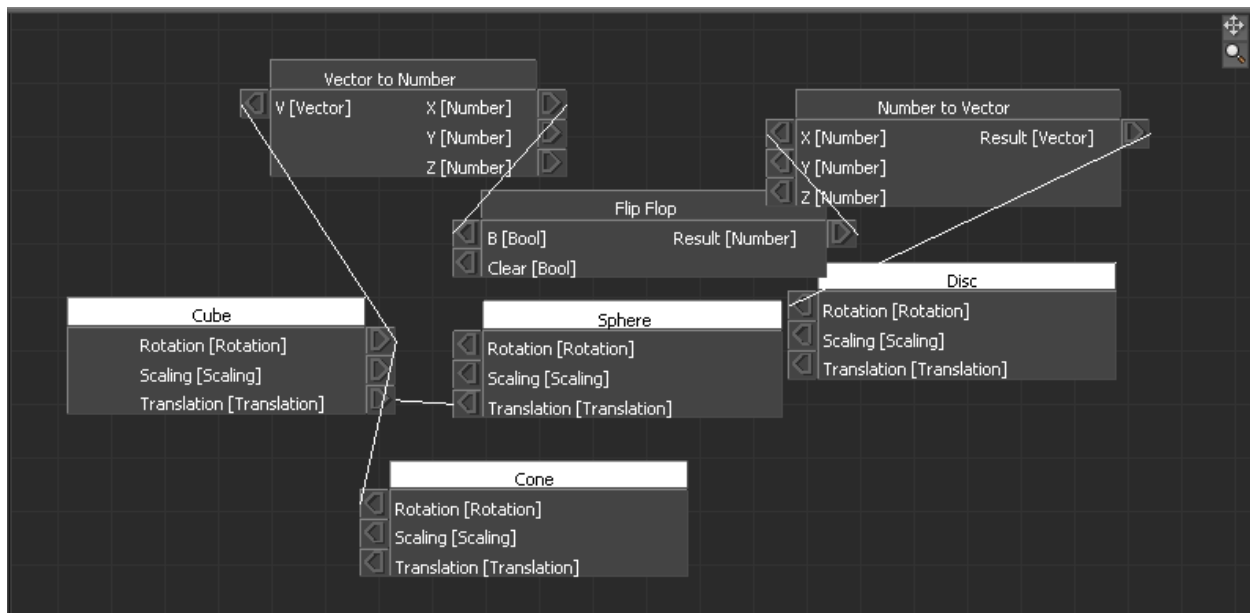


## Module 8: Constraints Assignment



### Traversing the relation constraint connections

Working with the scene file RelationConstraintScene.fbx, we are going to create a script that traverses all the boxes in the relation constraint and follow the connections, so that we know how to access this type of information.



### A. Accessing the Relation Constraint

Find all the constraints in the scene, print out something like this:

```
*****
***These are the constraints in the scene***
*****
Constraint Name -> Relation
```

### B. Determining the types of objects

Finding the objects in the Relation Constraint, and determining if they are a sender, operator, or a receiver:

```
*****
***These are the 7 boxes in the Relation Constraint***
*****

Cube is a Sender Box
Disc is a Receiver Box
Sphere is a Receiver Box
Cone is a Receiver Box
Flip Flop is a Function (operator) Box
Vector to Number is a Function (operator) Box
Number to Vector is a Function (operator) Box
```

## C. Determining the connections

Finding out what is Animation Node is connected to what Animation Node between the objects and operators:

```
*****
***These are the connections between the boxes in the Relation Constraint***
*****

-----Cube Box connected Animation Nodes-----
OUT: Cube (Translation) > Sphere (Translation)
OUT: Cube (Rotation) > Cone (Rotation)
OUT: Cube (Rotation) > Vector to Number (V)
-----Disc Box connected Animation Nodes-----
IN: Disc (Rotation) > Number to Vector (Result)
-----Sphere Box connected Animation Nodes-----
IN: Sphere (Translation) > Cube (Translation)
-----Cone Box connected Animation Nodes-----
IN: Cone (Rotation) > Cube (Rotation)
-----Flip Flop Box connected Animation Nodes-----
OUT: Flip Flop (Result) > Number to Vector (X)
IN: Flip Flop (B) > Vector to Number (X)
-----Vector to Number Box connected Animation Nodes-----
OUT: Vector to Number (X) > Flip Flop (B)
IN: Vector to Number (V) > Cube (Rotation)
-----Number to Vector Box connected Animation Nodes-----
OUT: Number to Vector (Result) > Disc (Rotation)
IN: Number to Vector (X) > Flip Flop (Result)
```