Maxscript Workshop 1

Full video tutorial support for Maxscript is available on BlackBoard (B8: Lectures).

The “Listener” window is very helpful as it shows what Maxscript commands are currently being executed when you use the GUI. This is a good way to help figure out how to automate a process that you are doing by hand; although there are exceptions where it might not show what you need. Fortunately there is copious documentation in the “Maxscript Help” resource. As an additional benefit of looking at this, there are some capabilities you might discover whilst perusing that are only available through Maxscript and that are not accessible via the GUI.

Task 1 - Teapot Spiral (Warm-up)

1a - Try evaluating the following script. You can either type it into a “New Script” and run it with “Tools->Evaluate All” (Ctrl+E); or, you can enter and evaluate it line-by-line by using the white text area in the “Listener” window. If you choose this latter approach (which is useful for debugging) then you can run multiple lines by selecting them and tapping the “Enter” key on the **number pad**.

```
resetMaxFile()
for k in 0.0 to 360.0 by 10.0 do (
   teapot position:[k,0,0] scale:[k/100,k/100,k/100] \
rotation:(angleAxis k [0,0,1])
)
```

1b - Adapt this script to construct and place 10 cubes in a circle.

Task 2 - Planetary System

Create three spheres (Sun, Earth and Moon) then Ctrl+click each each so all are selected. The order you click them in will be the order they feature in the currently selected ObjectSet when it is converted to an array $[1..n]. Note, if only one items is selected then $ just represents that item (not an ObjectSet containing only a single item as one might expect).

The currently selected object, or selection of objects, is given by $
e.g.

```
if (classof $ == ObjectSet) then (
   messagebox ("There are "+($.count as string)+" objects selected")
) else (
   messagebox ("Just one object selected:\n"+($ as string))
)
```

Use the following script to animate your planetary system so that the Moon orbits the Earth, and the Earth orbits the Sun.

```
if (classof $ != ObjectSet) then (
   messagebox("Need at least two selected objects.")
) else (
   for obj_id in 2 to $.count do (
      $[obj_id].parent = $[obj_id-1]
   )
   animate on (
      for obj in $ do (
         at time animationRange.start (
            rotate obj (angleAxis 0 [0,0,1])
         )
         at time animationRange.end (
            rotate obj (angleAxis 360 [0,0,1])
         )
      )
   )
)
```

Be sure you are clear about how this script is working before moving on.

Note, if your animation does not loop smoothly (i.e. the planets accelerate at the beginning and slow down to a stop at the end of the loop), check the rotations in the "Graph Editors" displays (particularly the "Track View - Curve Editor…"). Consider, what would make your animation loop without pausing?

Adjust the "Default In/Out Tangents for New Keys" setting (just to the right of the "Set Key" button) as appropriate. Note, before generate new keys it's usually best to remove the exiting ones. You can either select them in the time line and press delete, or to delete them all:

```
slidertime=0; deleteKeys objects #allKeys
```

Task 3 - Autoworm

In the lecture I showed you how to skin and rig a worm in 3DSMax (like you did last week in Blender). Start by understanding the workflow (i.e. quickly do it yourself):

```
1 - Make a Geometry->Cylinder
2 - Make some appropriate Systems->Bones
3 - Apply the Modifier->Skin to the cylinder
4 - Add the bones to the modifier
5 - Create an animation by keyframing the bones
```

Your task this week is to entirely automate the above process in a single Maxscript.

The first line of the script should be:

```
resetMaxFile(#noPrompt)
```

This will clear 3DSMax of any existing content. From here your script should complete the above steps to create and rig a cylinder and make an appropriate animation. To give you some clues as to how to achieve this, here are some tips:

```
B1 = BoneSys.createBone [0,0,0] [10,0,0] [0,0,1]
```

Whilst you can create bones manually i.e. bone() these are really just geometric object placeholders. So that they have the properties we want (i.e. a fixed length and known start and end points, the `BoneSys` helper functions are very convenient. Note, you will still need to setup the hierarchy (e.g. `B2.parent = B1`).

```
mySkin = skin()
addmodifier myCylinder mySkin
```

Modifiers can be created just like any other object. The above commands create a new skin modifier object (mySkin) and attach it to a geometry object (myCylinder). However, the options within a modifier are sometimes only available when the modifier's associated window is active. To make this happen you need to:

```
max modify mode
modPanel.setCurrentObject mySkin
```

Then using the helper function in skinOps you should then be able to do things like add a bone to the skin. e.g.

```
skinOps.addBone mySkin B1 1
```

Finally, to make keyframes automatically, you can use "`animate on`" (as in the planetary system example). Also, not that your script should **rotate** bones rather than trying to **move** them (when you manually drag a bone you are actually rotating it about its joint or "**pivot point**").

```
at time 10 rotate B2 (angleaxis -45 [0,1,0])
```