

## Домашнее задание. Неделя 01. Руководство

**Задача:** С помощью maya.cmds и Python напишите скрипт, который создаст полигональную сферу, переместит ее в позицию [X: -10, Y: 0, Z: 0] и создаст анимацию перемещения сферы в позицию [X: 10, Y: 0, Z: 0]. Анимация должна начаться в 1-м кадре и закончиться в последнем, который вы видите на вашем timeline.

Обратите внимание на следующие команды, именно они помогут вам с выполнением данной задачи: **polySphere**, **move** / **setAttr** / **xform**, **setKeyframe**, **playbackOptions**.

Давайте разберем эту часть вместе, решение второй части будет предоставлено уже вам самим. Но если она будет вызывать у вас трудности, то паниковать не стоит. В конце следующей недели я вышлю видео с решением ДЗ и подробным объяснением.

Попробуйте выполнить эту часть задания вручную, прямо через меню Maya создайте полигональную сферу, переместите в позицию [X: -10, Y: 0, Z: 0], поставьте ключи на каналах translate. Передвиньте TimeSlider в последний кадр (по умолчанию 120), и поставьте ключи на новые значения [X: 10, Y: 0, Z: 0]. Обратите внимание на то, какие команды у вас появились в Script Editor History. У меня это выглядит следующим образом.

```
polySphere -r 2 -sx 20 -sy 20 -ax 0 1 0 -cuv 2 -ch 1;
// Result: pSphere1 polySphere1 //
setAttr "pSphere1.translateX" -10;
if( `getAttr -k "pSphere1.tx" ` || `getAttr -channelBox "pSphere1.tx" ` )setKeyframe "pSphere1.tx";
// Result: 1 //
currentTime 120 ;
setAttr "pSphere1.translateX" 10;
if( `getAttr -k "pSphere1.tx" ` || `getAttr -channelBox "pSphere1.tx" ` )setKeyframe "pSphere1.tx";
// Result: 1 //
```

Если убрать все лишнее, то получится:

```
polySphere -r 1 -sx 20 -sy 20 -ax 0 1 0 -cuv 2 -ch 1;
setAttr "pSphere1.translateX" -10;
setKeyframe "pSphere1.tx";
currentTime 120 ;
setAttr "pSphere1.translateX" 10;
setKeyframe "pSphere1.tx";
```

Обратите внимание, что для установки значения **translateX** использовалась команда **setAttr**. Это потому что я вбил значение вручную в **ChannelBox**. Если бы я передвинул объект с помощью манипулятора в сцене - выдалась бы команда **move**. Вы можете использовать любую из них, но самой продвинутой командой, связанной с изменением позиции объекта в пространстве - является **xform**. Рассмотрите примеры на Python как эта команда используется. Она вам очень пригодится в будущем.

Ну а пока мы воспользуемся тем, что нам выдал ScriptEditor, и перепишем весь код на Python с небольшими правками (команды, которые выдает ScriptEditor - не всегда полностью актуальны, иногда нам нужно поменять параметры либо просто удалить).

```
import maya.cmds as cmds

# Чтобы узнать первый кадр и последний на шкале TimeLine, мы
# можем воспользоваться командой playbackOptions
minTime = cmds.playbackOptions(query=1, minTime=True)
maxTime = cmds.playbackOptions(query=1, maxTime=True)

cmds.polySphere(r=1, name="MySphere")

# используем переменные minTime и maxTime чтобы передвинуть
# ползунок в нужный кадр для установки ключа
cmds.currentTime(minTime)
cmds.setAttr("MySphere.translateX", -10)
cmds.setKeyframe("MySphere.translateX")

cmds.currentTime(maxTime)
cmds.setAttr("MySphere.translateX", 10)
cmds.setKeyframe("MySphere.translateX")
```

Давайте попробуем улучшить наш код. Если мы обратим внимание на документацию по команде **setKeyframe**, мы можем увидеть что у команды есть параметр **time**. Т.е. мы можем не передвигать в реальности ползунок чтобы установить ключ на новом кадре а лишь указать время в команде **setKeyframe** (на самом деле передвижение ползунка может очень сильно затормозить код, т.к. сцены могут быть тяжелые и скрипту придется ждать пока сцена обновится, и лишь затем продолжить). Также у этой команды есть параметр **value**, т.е. мы можем сразу указать и значение для канала **translateX**.

```
cmds.setKeyframe("MySphere.translateX", time=minTime, value=-10)
cmds.setKeyframe("MySphere.translateX", time=maxTime, value=10)
```

Чтобы сделать нашу программу еще более продвинутой, мы сохраним имя нашей сферы в переменную, чтобы нам не пришлось каждый раз прописывать это имя в последующих командах. Для начала нам нужно провести небольшой тест. Попробуйте выполнить следующую команду:

```
import maya.cmds as cmds
sph = cmds.polySphere(r=1, name="MySphere")
print sph
```

Эта команда, помимо создания самой сферы с именем **MySphere** и радиусом 1, выведет в ScriptEditor такой результат: **[u'MySphere', u'polySphere1']**. Это говорит о том, что команда вернула список с именами нод. Нам же нужно только первое имя - **"MySphere"**, поэтому в конце команды **polySphere** мы должны поставить **[0]** - это значит что когда мы присвоим переменной **sph** эту команду, в этой переменной будет храниться имя **MySphere**. Мы используем ноль потому что индексация списков начинается с нуля.

Чтобы использовать эту переменную в других командах, мы можем использовать знак **+**, чтобы совмещать строки. Т.е. прежняя строка **"MySphere.translateX"** может быть изменена на **sph + ".translateX"**. Это очень удобно т.к. тогда при написании кода нам не нужно будет запоминать реальные имена нод, а лишь имена переменных. Вот что должно получиться в итоге:

```
import maya.cmds as cmds

sph = cmds.polySphere(r=1, name="MySphere")[0] # а н а л о г и ч н о [u'MySphere',
u'polySphere1'][0] - т.е. "MySphere"

minTime = cmds.playbackOptions(query=1, minTime=True)
maxTime = cmds.playbackOptions(query=1, maxTime=True)

cmds.setKeyframe(sph + ".translateX", time=minTime, value=-10)
cmds.setKeyframe(sph + ".translateX", time=maxTime, value=10)
```