Introduction to Python in 3DS Max 2015

Initial Implementation:

In this tutorial I will show you how to get started with implementing python in 3ds Max. I will be using the PyCharm IDE for this tutorial, but the essential steps should be applicable to any IDE.

Checklist of required software:

1.) PyCharm - http://www.jetbrains.com/pycharm/download/index.html (or any IDE of your choice)

2.) 3DS Max 2015

3.) Max python Plugin https://github.com/arturleao/YCDIVFX_MaxPlus/archive/master.zip

Step1 : Install the plugin and save it to a directory of ur choosing (eg:C:\YCDIVFX_MaxPlus-master)

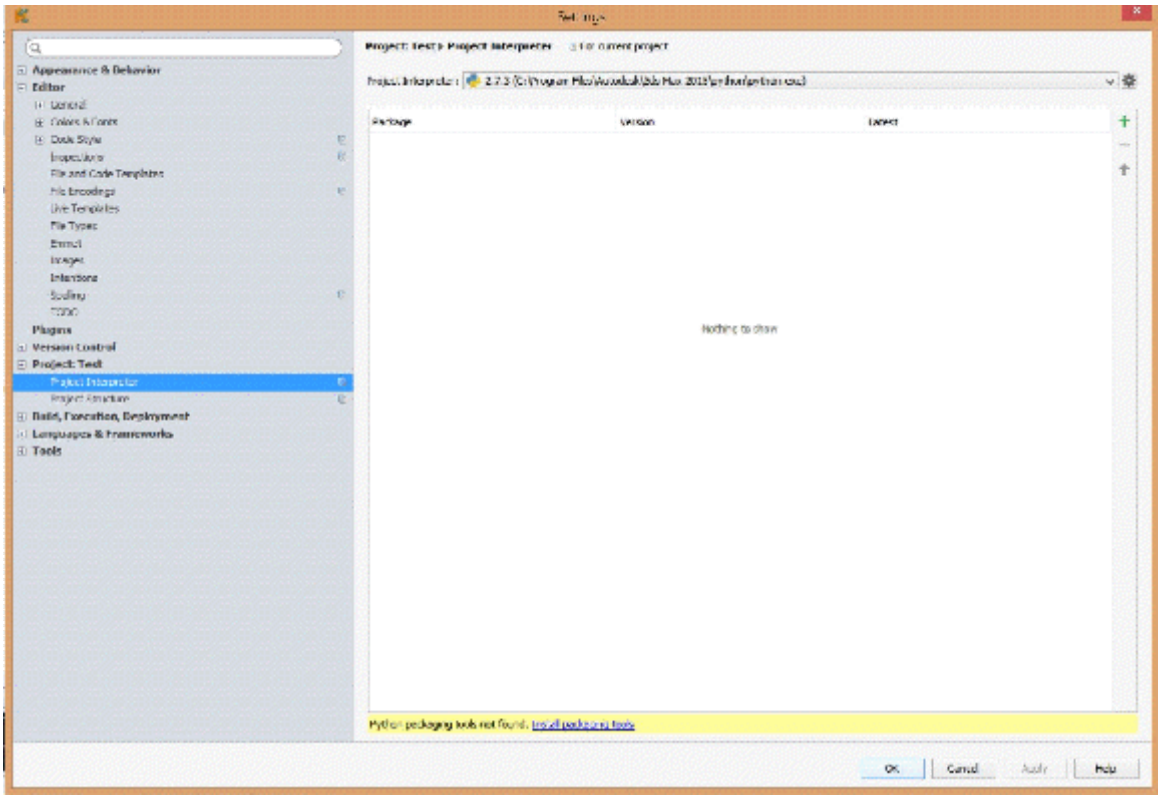
step 2: open up PyCharm and create a new project, create a new python file(eg. sierpinksy.py)

step 3: Set up the python interpreter,

a.) go to file->settings->Project->Project interpreter

b.) click the project interpreter drop down button and navigate to the 3ds max root folder, and select the python.exe(this is the interpreter) (eg:C:\Program Files \Autodesk\3ds Max 2015 \python\python.exe)

Settings Window: Project Interpreter Settings



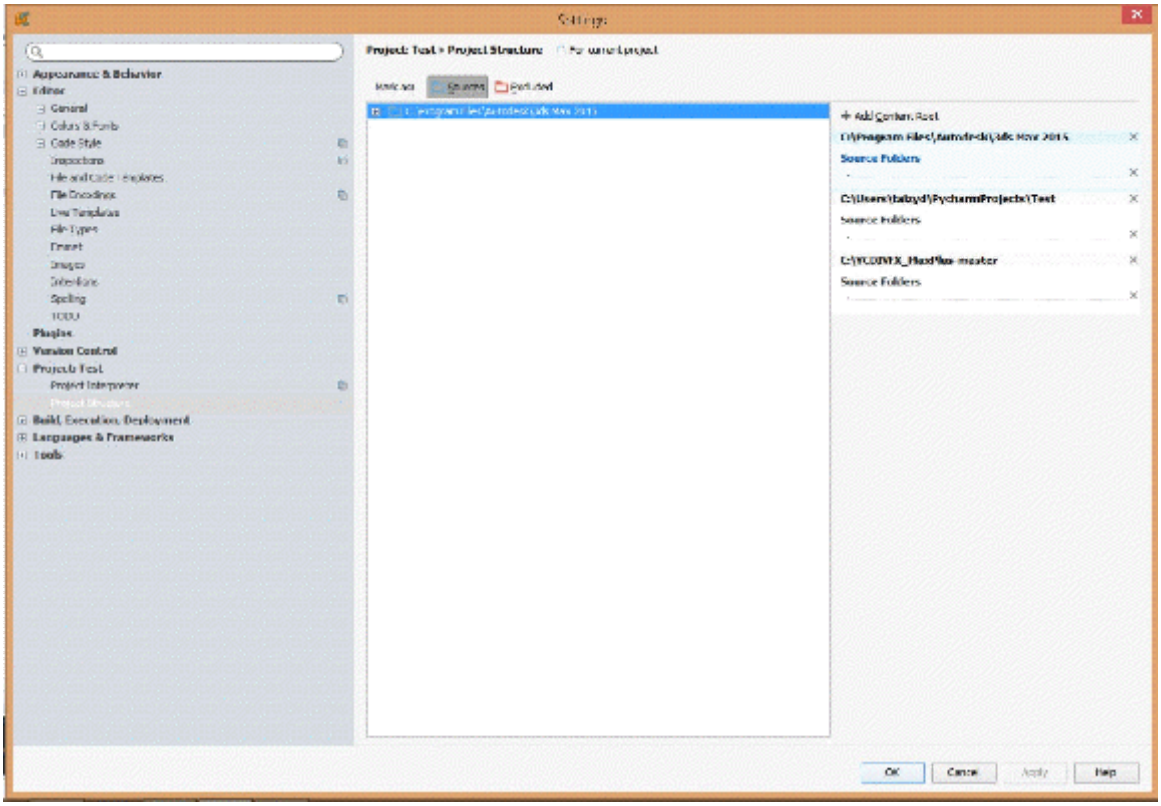
step 3: Set up the project directories:

a.) Click on the Project Structure, click the "'+" icon next to (add content root) and navigate to your project folder. eg(C:\Users\taizyd\PycharmProjects\Test), Now, click on "source" to load the directory.

b.) Click on "+" again and this time navigate to your 3dsMax root directory(eg:C: \Program Files\Autodesk\3ds Max 2015). Click on "source" to load it.

c.)Click on "+" one more time and navigate to the directory where you saved the Plug-in(eg:C:\YCDIVFX_MaxPlus-master).Click on source to load it.

Settings Window:Project Structure Setting



Testing Communication:

Now go back to your project window and create a new python file, we will use this as our "Hello World" program to test if our code is communicating with 3ds Max correctly.

Code Snippet:

For now, jsut copy the code as it is, in the next example I will give an introductory explanation of the 3ds Max subclasses that you will need to know for some basic operations

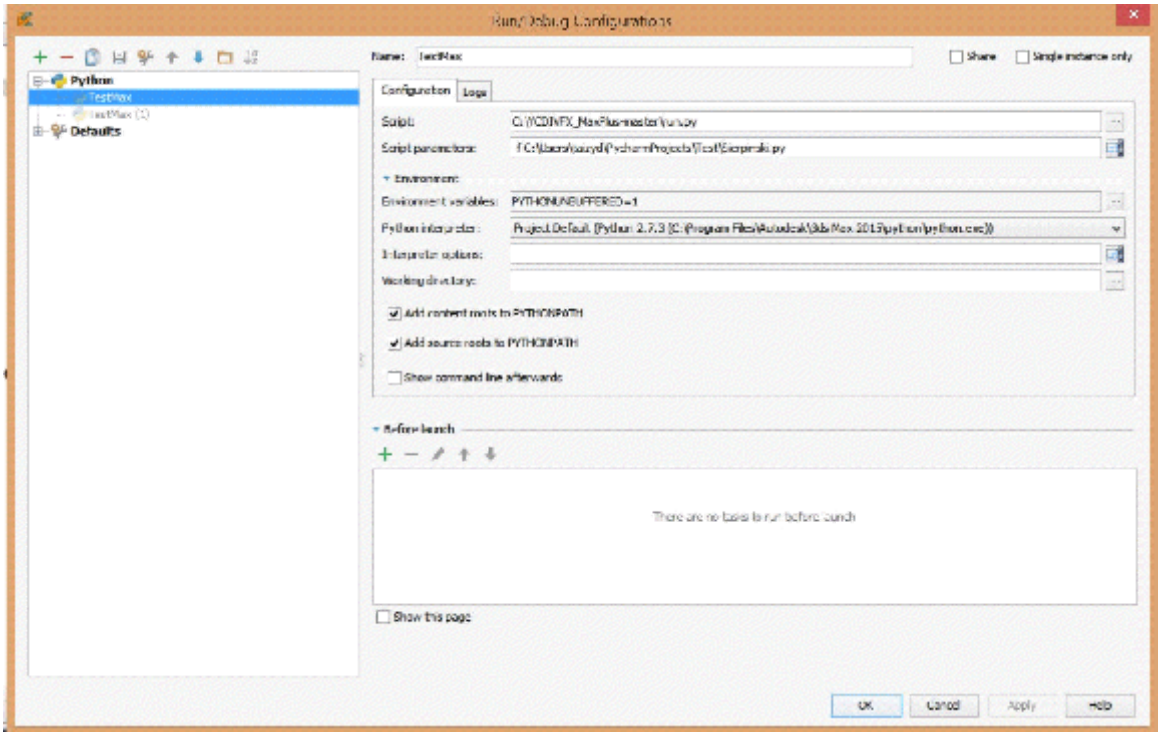
```
__author__ = 'taizyd'
from random import randint
import MaxPlus

MaxPlus.Core.WriteLine("Hello World")
```

Now, Click on Run-> Edit Configuration

a.) Under Script: set the directory of the run.py file in your plugin directory

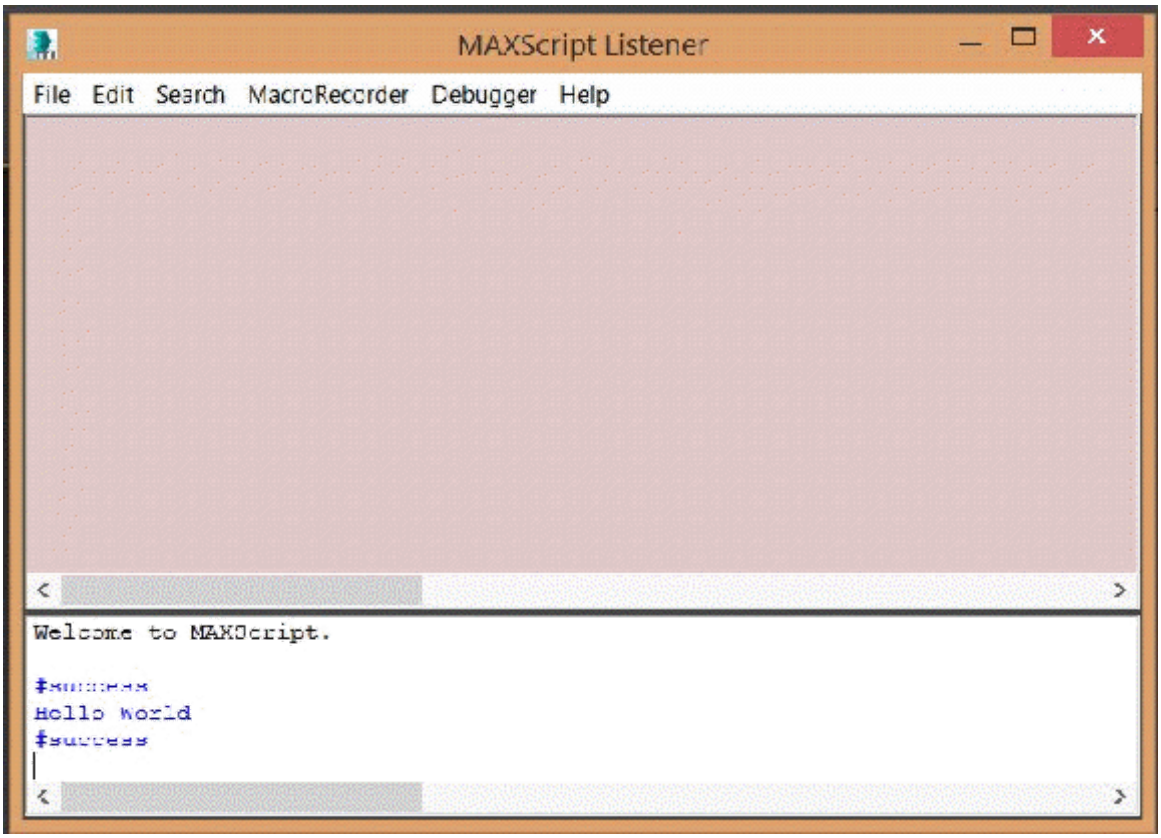
b.) Under Script parameter: type -f and then the path of YOUR script. (This will need to be set for every script that you create)



Now open up 3DS Max 2015

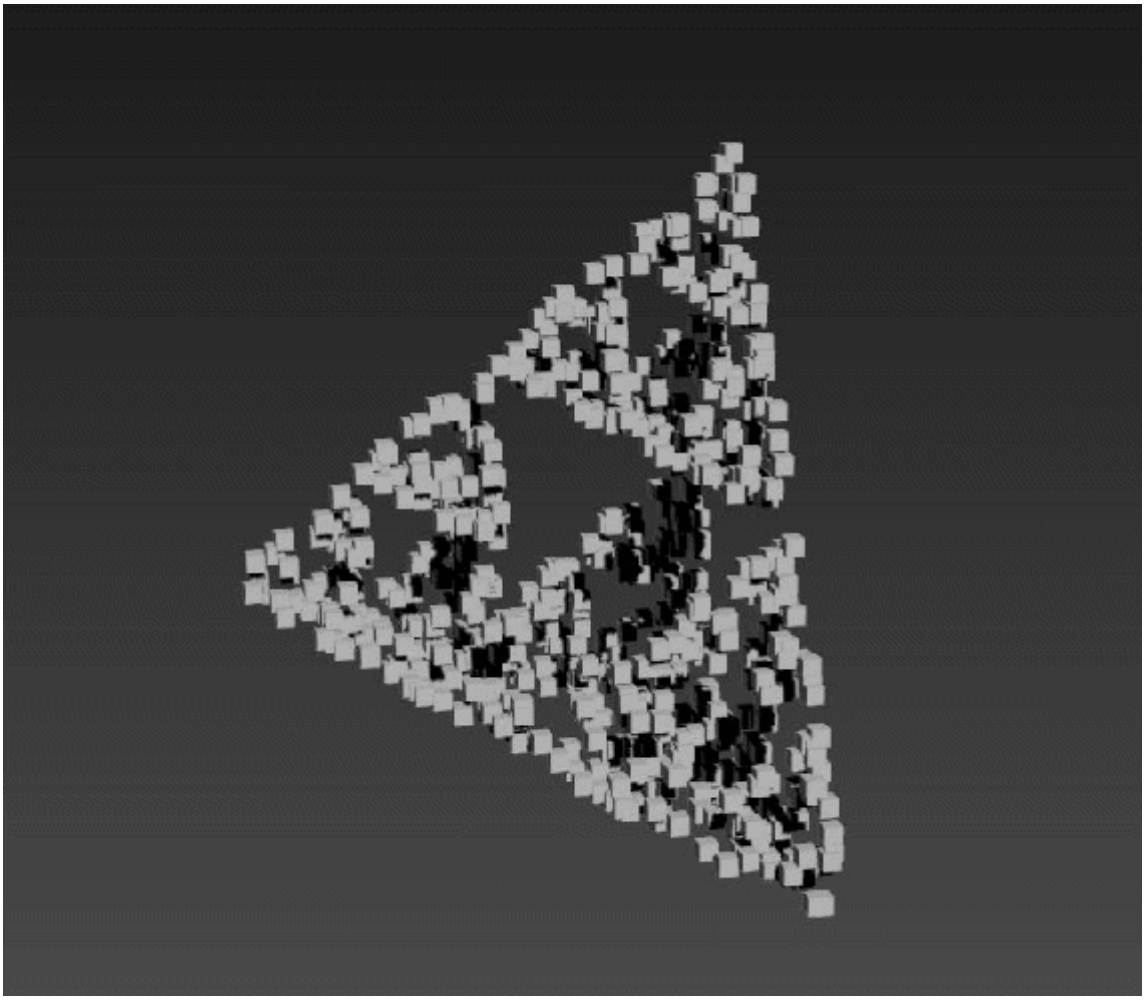
Go back to PyCharm.Now press "Run" or the little green arrow in the top right corner of the project window.

Go to 3ds Max, click on MAXScript-> Maxscript listener Window. and you should see.



Now, that we have PyCharm communicating with Max, lets try and create something a little more visual.

Creating the Sierpinski Fractal



Brief:

This tutorial assumes that you are comfortable with python, but new to the 3ds Max subclasses used in python. The first thing to understand is that python does not work directly with 3ds Max, it uses a plugin created by autodesk, to convert python to maxscript and vice versa. In this tutorial I will give a basic explanation of how to create and object tweak its parameters. In the end we will use code to create what is known as a sierpinski fractal (image shown above).

Code:

```
__author__ = 'taizyd'
from random import choice
from random import randint
import MaxPlus

pnts = [ [0,0,1], [1,0,-1], [-1,0,-1], [0,1.5,-0.2] ]
seed = [0,0,0]
randchoice = [2.0, 2.5, 3.0]
data = []

randnum = choice(randchoice)


def midpoint(p1,p2):
    tp1 = (p2[0] + p1[0]) / randnum
    tp2 = (p2[1] + p1[1]) / randnum
    tp3 = (p2[2] + p1[2]) / randnum
    temp = [tp1, tp2, tp3]
    return temp

def pickpnt(pnts):
    return choice(pnts)


for n in range(1000):

    pickpnt(pnts)
    seed = midpoint(pickpnt(pnts), seed)
    data.append(seed)
    obj = MaxPlus.Factory.CreateGeomObject(MaxPlus.ClassIds.Box)
    obj.ParameterBlock.Length.Value = 0.05
    obj.ParameterBlock.Width.Value = 0.05
    obj.ParameterBlock.Height.Value = 0.05
    node = MaxPlus.Factory.CreateNode(obj)
    node.Move(MaxPlus.Point3(seed[0], seed[1], seed[2]))
```

Maxplus:

```
import MaxPlus
```

Maxplus is what gives us access to the 3ds Max libraries in python. Simply put you can't do anything without this file, so be sure to import it in all your python files that are being used with max.

3ds Max Subclass

Creating an object is a simple process, the following lines of code creates a box and positions it at the coordinate specified by the seed.

```
obj = MaxPlus.Factory.CreateGeomObject(MaxPlus.ClassIds.Box)
obj.ParameterBlock.Length.Value = 0.05
obj.ParameterBlock.Width.Value = 0.05
obj.ParameterBlock.Height.Value = 0.05
node = MaxPlus.Factory.CreateNode(obj)
node.Move(MaxPlus.Point3(seed[0], seed[1], seed[2]))
```

The 2nd, 3rd and 4th lines are used to specify the parameters of the object created, these parameters will vary depending on the type of object created.

The last two lines of code, is what actually creates the object and positions it in 3D space.

These subclasses can only be used if the "MaxPlus" plugin is imported.

This has been a brief introduction into getting started with using python in 3DS MAX, I hope you find this helpful and happy coding!