

UPython

MIT License.

Unity 2019.x tested.

Author: [Haikun Huang](#)

My Lab: [DCXR](#)

Github: [Click Here](#)

This plugin is more like a bridge connecting Unity and Python. UPython allows you to write and run the python code in Unity C# directly. UPython also will stream back the python output to Unity C# and allows you to do further actions.

How does it work?

Essentially, UPython first will generate a .py file base on the python code you wrote in Unity C#. UPython then will call the python command at CMD window on Windows or Terminal on OS X.

Quick Start

UPython.cs class is the core class of UPython. It inherited MonoBehaviour which technically allows you to run multiple python threads at the same time.

Here is the quick start introduction:

- Make sure you have set up the python dev environment properly. To valid the setup, you can run the python command at CMD window on Windows or Terminal on OS X. If it works, which means your setup is good to go.

```
(base) ~ python
Python 3.7.1 (default, Dec 14 2018, 13:28:58)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

- Open the **UPython.cs** file, and change the variable **python_program**, make sure the **python_program** is pointing to your local python program, e.g.,

```
static public string python_program =
"/Users/haikunhuang/anaconda3/bin/python";
```

- You are ready to go.

APIs & Variables

- The folder used to store the python files UPython generated.

```
public string runningFolder = "../Python";
```

- The file used to store the python files UPython generated.

```
public string runningFile = "demo.py";
```

- If shows the debug info at console.

```
public bool showDebug = true;
```

- If UPython still running. (Reay Only)

```
public bool isRunning { get; private set; } = false;
```

- The python script buffer.

```
public List script = new List();
```

- The output buffer.

```
public List results = new List();

public List errs = new List();
```

- Store the variables with the '@' at the beginning of each line of the output stream.

e.g., @a=1

how to get the “a” variable?

e.g., string something = variables[“a”];

This dictionary helps you to retrieve the result more easily.

More detail please see the examples.

```
public Dictionary<string, string> variables = new Dictionary<string,
string>();
```

- Create a new Environment so that the previous result and script can be held by other references.

```
public void NewEnvironment();
```

- Build and run the python script.

If the callback function is not null, then the callback function will be invoked when the python script is done.

```
public void BuildAndRun(string args = “”, UnityAction callback = null);
```

- Run the python script from the given **runningFolder** & **runningFile** and without rebuild it.

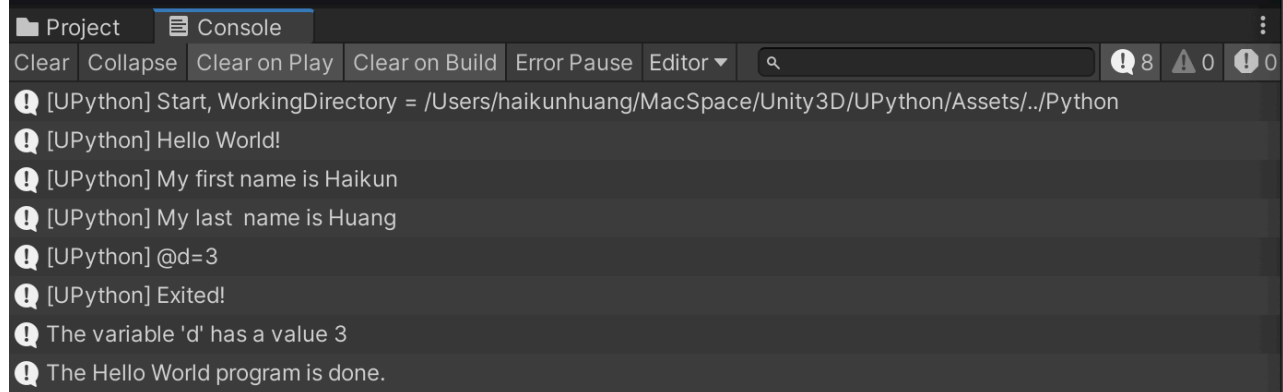
```
public void RunFromFile(string args = "", UnityAction callback = null)
```

Examples

1. HelloWorld

Must have a look, it shows the most basic functions of UPython.

Play and see the console



2. Parser

A helper tool helps you to convert the Python code into a UPython friendly format in Unity C#.

Copy your python code into the left box, and the converted code will be shown in the right box immediately.

```
from sklearn import datasets
digits=datasets.load_digits()
print(digits.data.shape)
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[0])
plt.show()
```

```
List<string> pyText = new List<string>();
pyText.Add("from sklearn import datasets");
pyText.Add("digits=datasets.load_digits()");
pyText.Add("print(digits.data.shape)");
pyText.Add("import matplotlib.pyplot as plt");
pyText.Add("plt.gray()");
pyText.Add("plt.matshow(digits.images[0])");
pyText.Add("plt.show()");
```

3. Run In UI Text

An example of running the Python code in real-time.

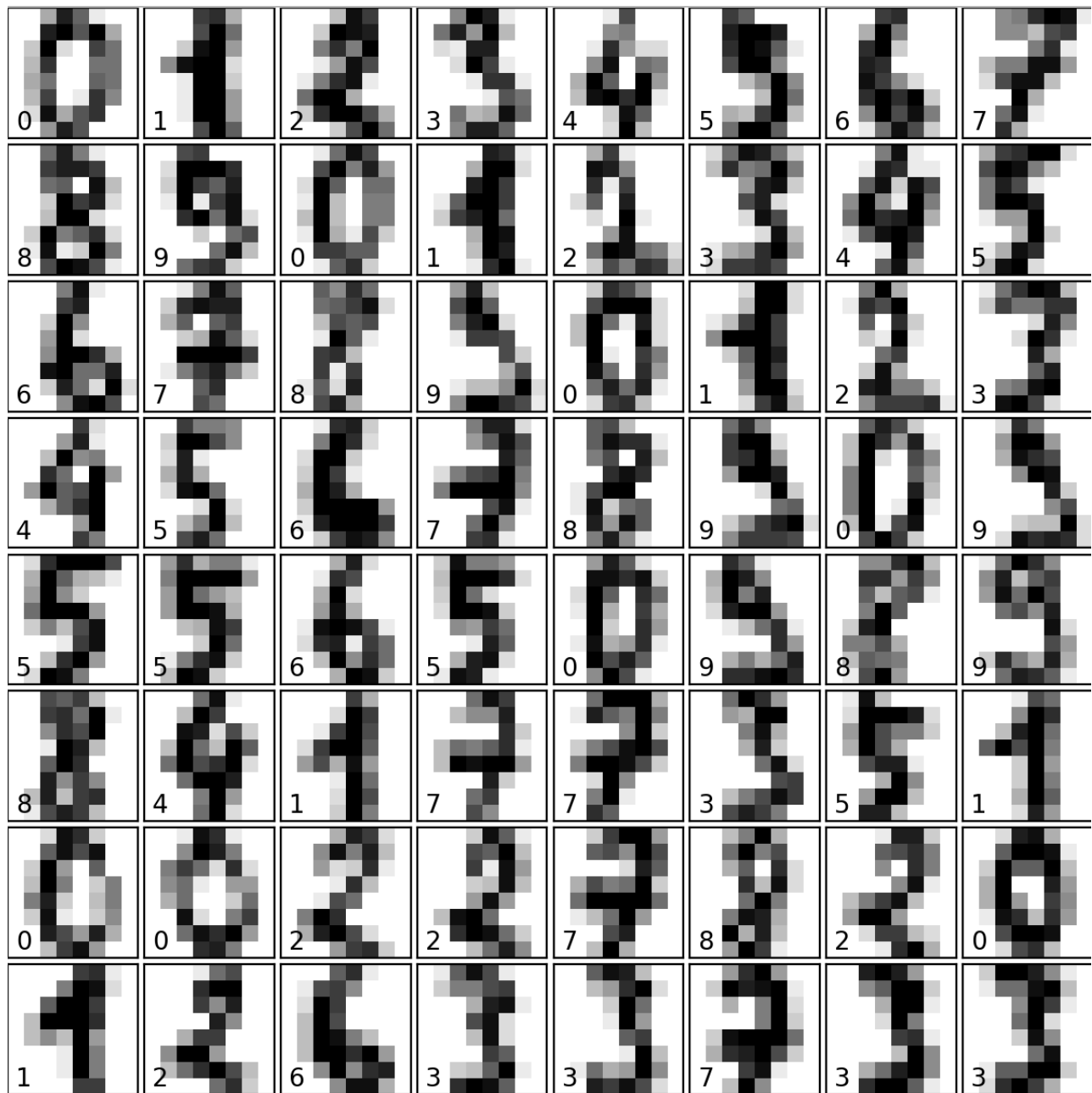
Input your python code in the input box and click Run. The result will show in the console. Also, the figure will pop out if there have any.

```
from sklearn import datasets
digits=datasets.load_digits()
print(digits.data.shape)
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[0])
plt.show()
```

Run

4.Use Case

A use case example, just run it and see what happen.



5. Embedding Example

An example of embedding the generated data into the Python code by UPython.
Useful for generating the cost curve after each simulation.

UPlot helper function has been provided. Please take a look at the new Embedding Example Demo.

```
public void SimplePlot(List xdata, List ydata, string title="", string xLabel="",  
string yLabel = "")
```

Advantages

- Easy to integrate your work between Unity and Python.
- UPython Embeds all the variables into the Python code. So you don't need to care how to pass the variables to the external Python code.
- Easy to retrieve the results after running UPython by looking up the "variables" dictionary.
- You should be able to write and run any CMD or Terminal runnable scripts. (Not only Python.)

Limitations

- Low performance.
- Looks stupid.