

e-Yantra Robotics Competition (eYRC-2016) Task 1 – Model a Terrain

Introduction to Blender Python API:

Blender is a powerful 3D application offering a broad range of essential tools, including Modeling, Rendering, Animation, Texturing, many types of Simulations and Game Creation.

Through scripting, Blender can be extended in real-time via Python, which is an impressive high level and open source language.

Learning Python:

To learn Python, you can refer to the link: <http://www.learnpython.org/> and you can download the software required for learning Python from <https://www.python.org/downloads/>

Blender Python:

You have been working on Blender interface, where you did editing and modification of objects by navigating in the 3D space with the use of mouse movement and keyboard shortcuts. The screen in which you have worked till now is shown in Figure 1:

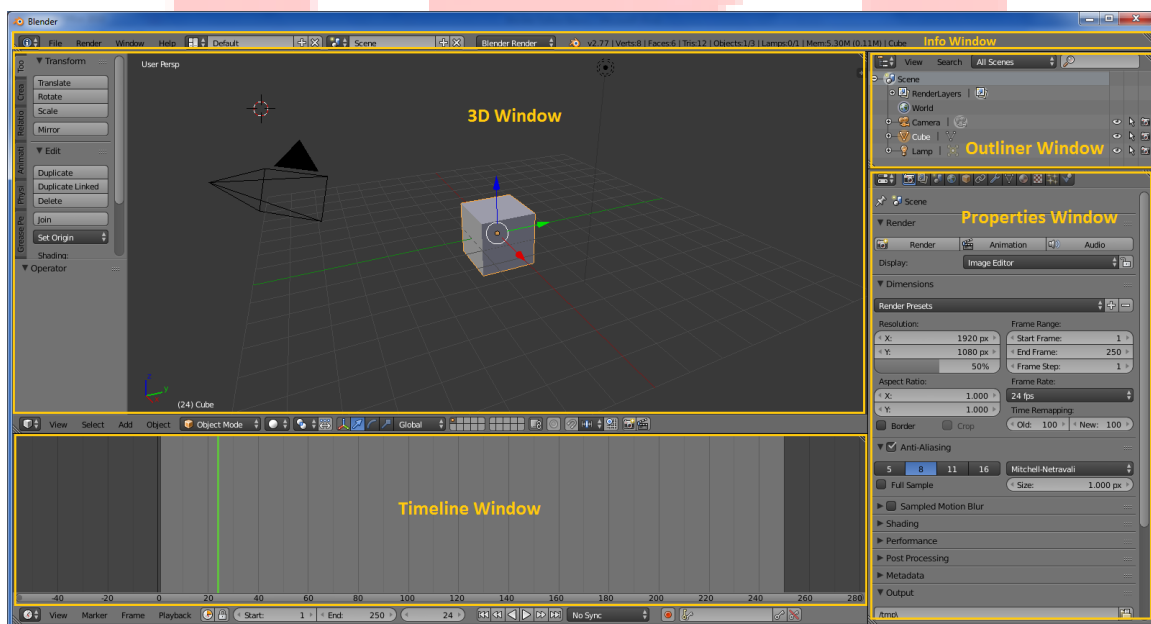


Figure 1

Python accesses Blender's data in the same way as the animation system and Blender interface; this implies that any setting that can be changed via a button can also be changed from Python. This enables you to create objects and other things using a scripting interface:

- Blender uses user-friendly Python scripting language.
- It provides extensive programming interface to match almost all the functionality of the application (hence called Blender Python API or BPY API).

The two most common ways to execute Python scripts in Blender are by using the

- Built-in Text Editor or

- Blender Python Console

A quick list of helpful things to know before starting:

- Blender uses Python 3.x; some 3rd party extensions are not available yet.
- The Python console is great for testing one-liners. It also has auto-completion so you can inspect the API quickly.
- Button tool tips show Python attributes and operator names. This will help you to write the Python script only you should know logic which you want to apply.
- Right clicking on buttons and menu items directly links to API documentation.
- For more examples, the text menu has a templates section where some example operators can be found.

Accessing Blender Python Console:

1. Using Mouse:

Change any editor type (3D, Properties, Outliner, Timeline or Info window) to Python Console by clicking on the editor type selector that is, highlighted icon shown in Figure 2 and selecting Python console.

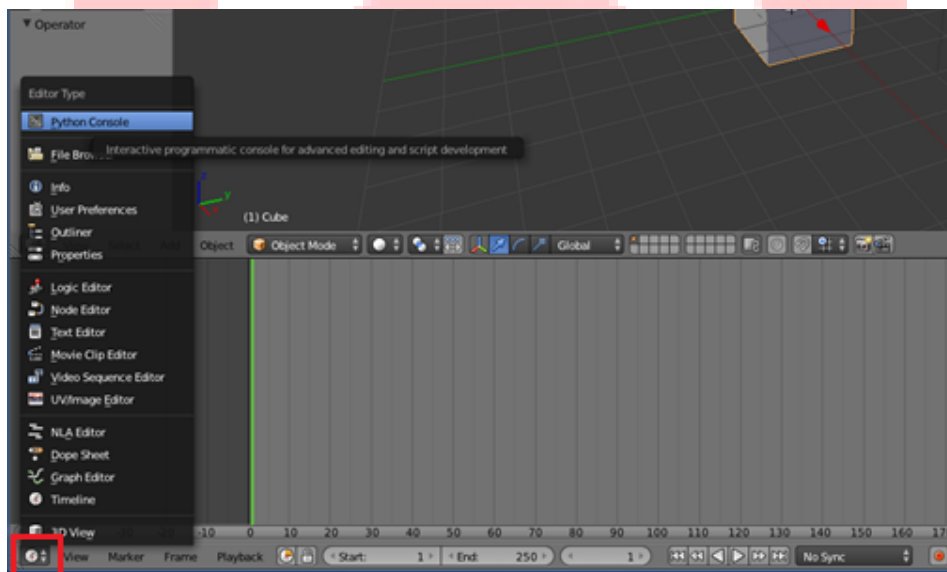


Figure 2

2. Shortcut:

Press Shift-F4 in the editor to change it to Python Console.

After this you will see the Python Console interface as shown in Figure 3:

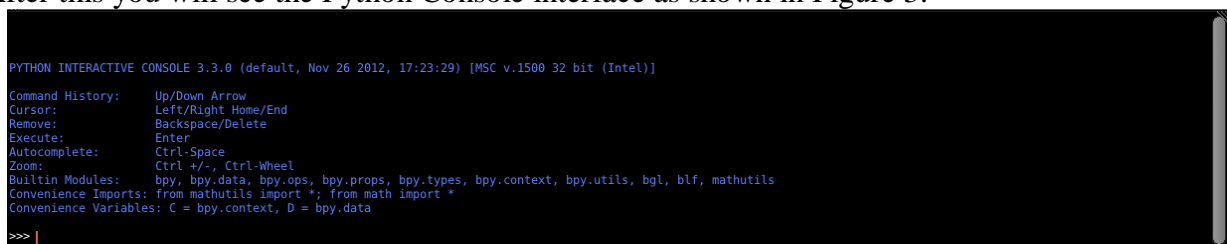


Figure 3

The command prompt is typical for Python 3.x, the interpreter is loaded and is ready to accept commands at the prompt >>>

The Python Console is used for typing in snippets and for testing the commands to get immediate feedback.

```
Command History: Up/Down Arrow
Cursor: Left/Right Home/End
Remove: Backspace/Delete
Execute: Enter
Autocomplete: Ctrl-Space
Zoom: Ctrl +/-, Ctrl-Wheel
Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> bpy.ops.mesh.primitive_cube_add()
{'FINISHED'}

>>> |
```

Figure 4

Notice the blue output `{'FINISHED'}` in Figure 4 what you see is the result of the command.

The Python console is great for testing one-liners. It also has auto-completion so you can inspect the API quickly.

Auto-completion:

Type `bpy.` and then press Ctrl-Spacebar and you will see the Console auto-complete feature in action shown in Figure 5.

```
Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> bpy.
    app
    context
    data
    ops
    path
    props
    types
    utils

>>> bpy.|
```

Figure 5

You will notice that a list of sub-modules inside of `bpy` appears. These modules encapsulate all that we can do with Blender Python API and are very powerful tools.

Accessing Python Text Editor:

1. Using Mouse:

Change any editor type (3D, Properties, Outliner, Timeline window etc.) to Text Editor by clicking on the highlighted icon shown in Figure 2 and selecting Text Editor.

2. Shortcut:

Press Shift-F11 in the editor to change it to Text Editor.

After this you will see the Text Editor as shown in Figure 4:

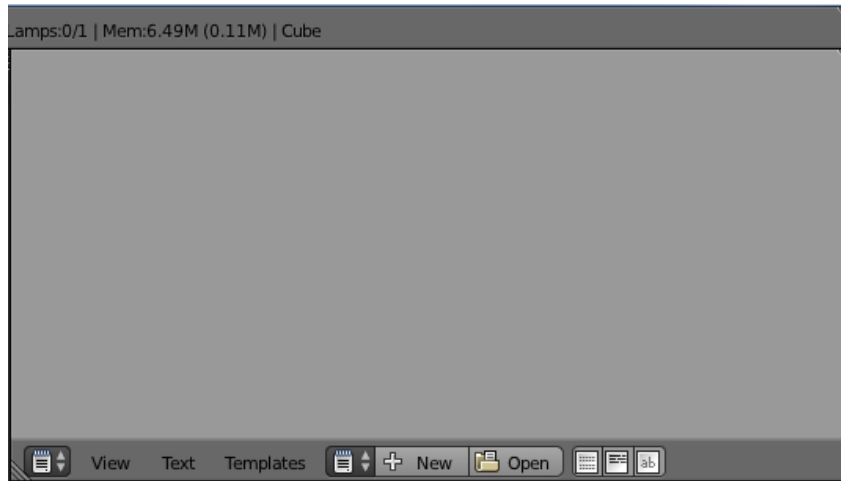


Figure 4

Click on New button to start scripting. After clicking on New Button you can see the interface as shown in Figure 5.

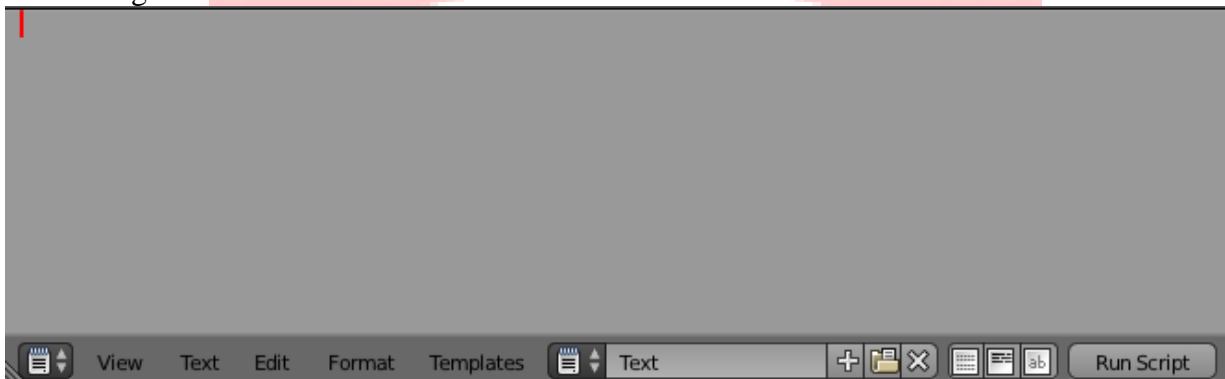


Figure 5

To run the script, click on Run Script button or press ALT + P. By default the script written in Text editor is saved as Text.py if you save the blend file. To rename the file, click on text area and rename the file.

To save the Python file separately, click on view menu and select Save As shown in Figure 6.

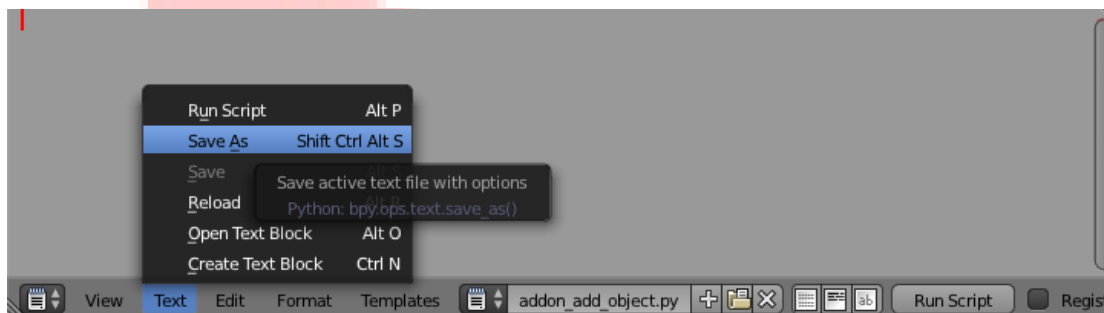


Figure 6

To write and debug the script in Text Editor, follow the instructions given below:

In the Text editor, type `print("Hello")` and then click on "Run Script" button to run the code or press ALT+P.

Above statement should print Hello but where you can find the output?

It is the System Console which is available in the menu options of Info window of Blender where you can catch the startup message, error messages and debugging print statements from your Python scripts.

You can toggle the display of the System Console from the Blender interface, as illustrated in Figure 7:

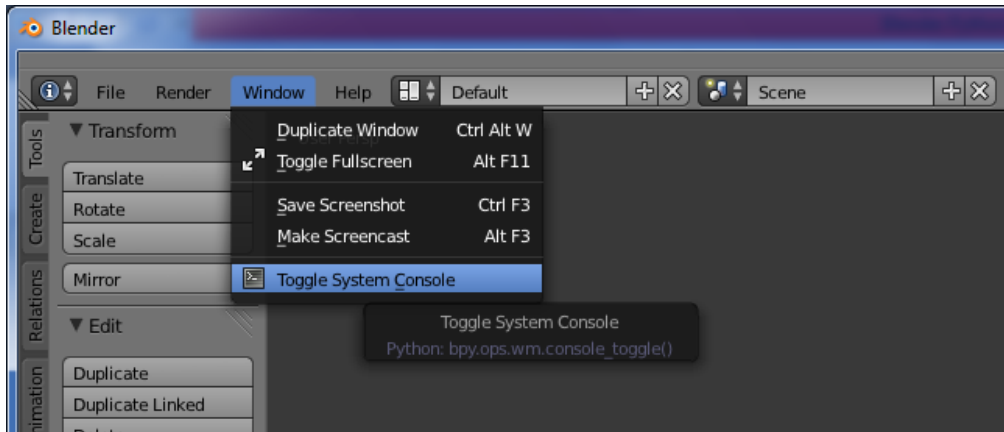


Figure 7

You can now see the output to your code as shown in Figure 8:

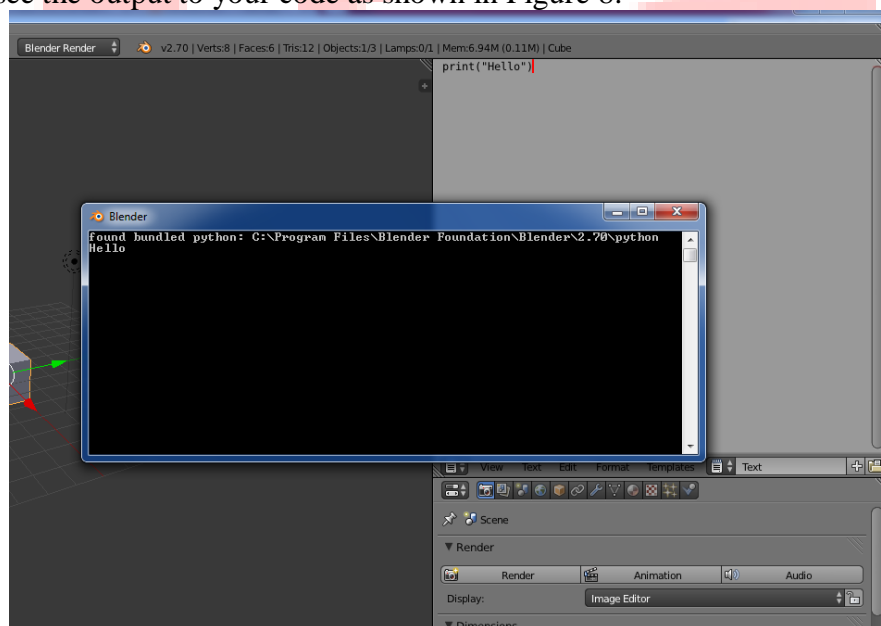


Figure 8

References:

Refer to the given link to know more about Blender:

Blender Python API Documentation:

http://www.blender.org/api/blender_python_api_current/

https://www.blender.org/api/blender_python_api_2_65_5/info_quickstart.html

https://www.blender.org/api/blender_python_api_2_63_release/info_tips_and_tricks.html

Links to know about other modules:

http://www.blender.org/api/blender_python_api_current/search.html?q=bmesh

http://www.blender.org/api/blender_python_api_current/mathutils.geometry.html

