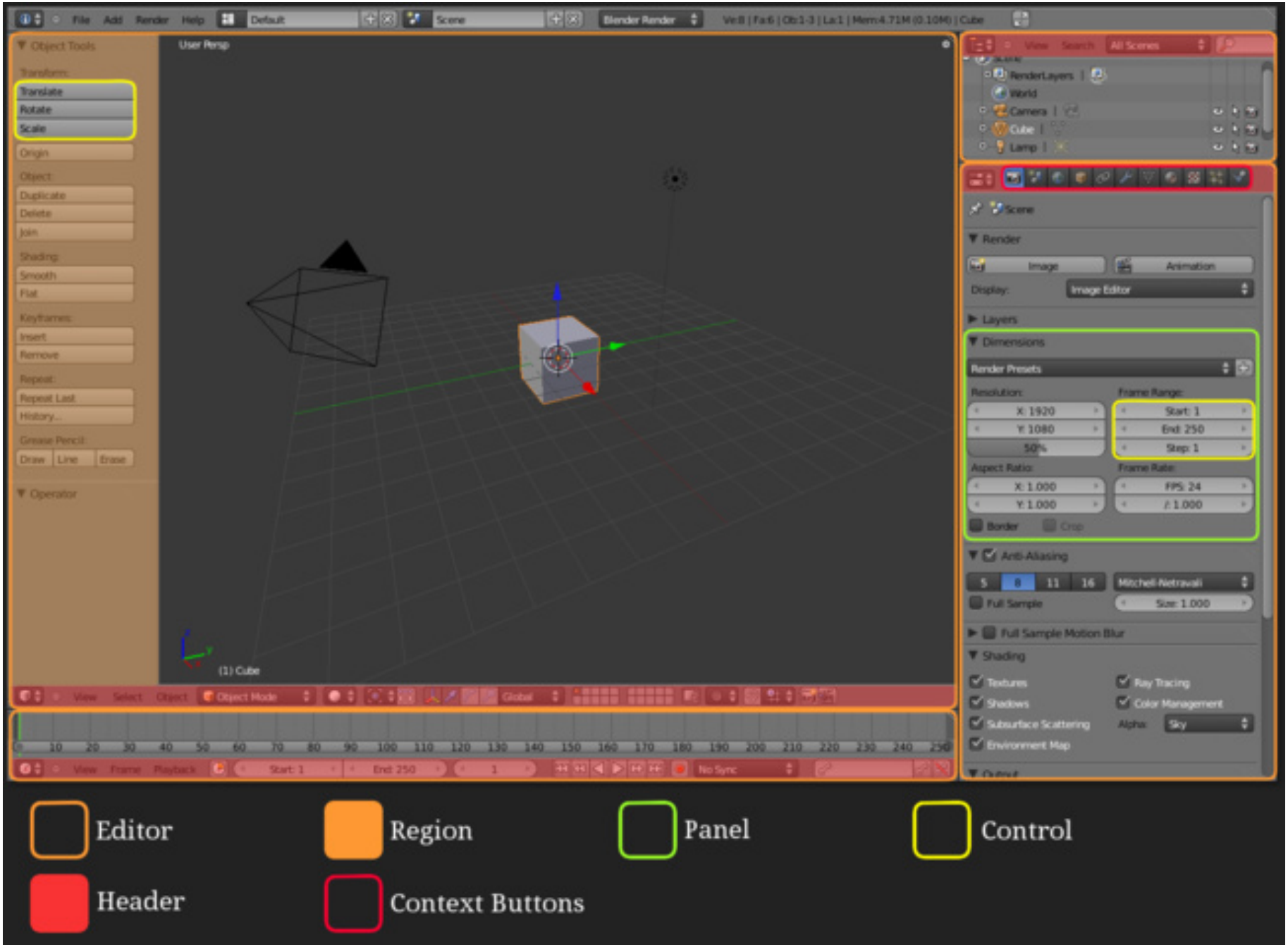
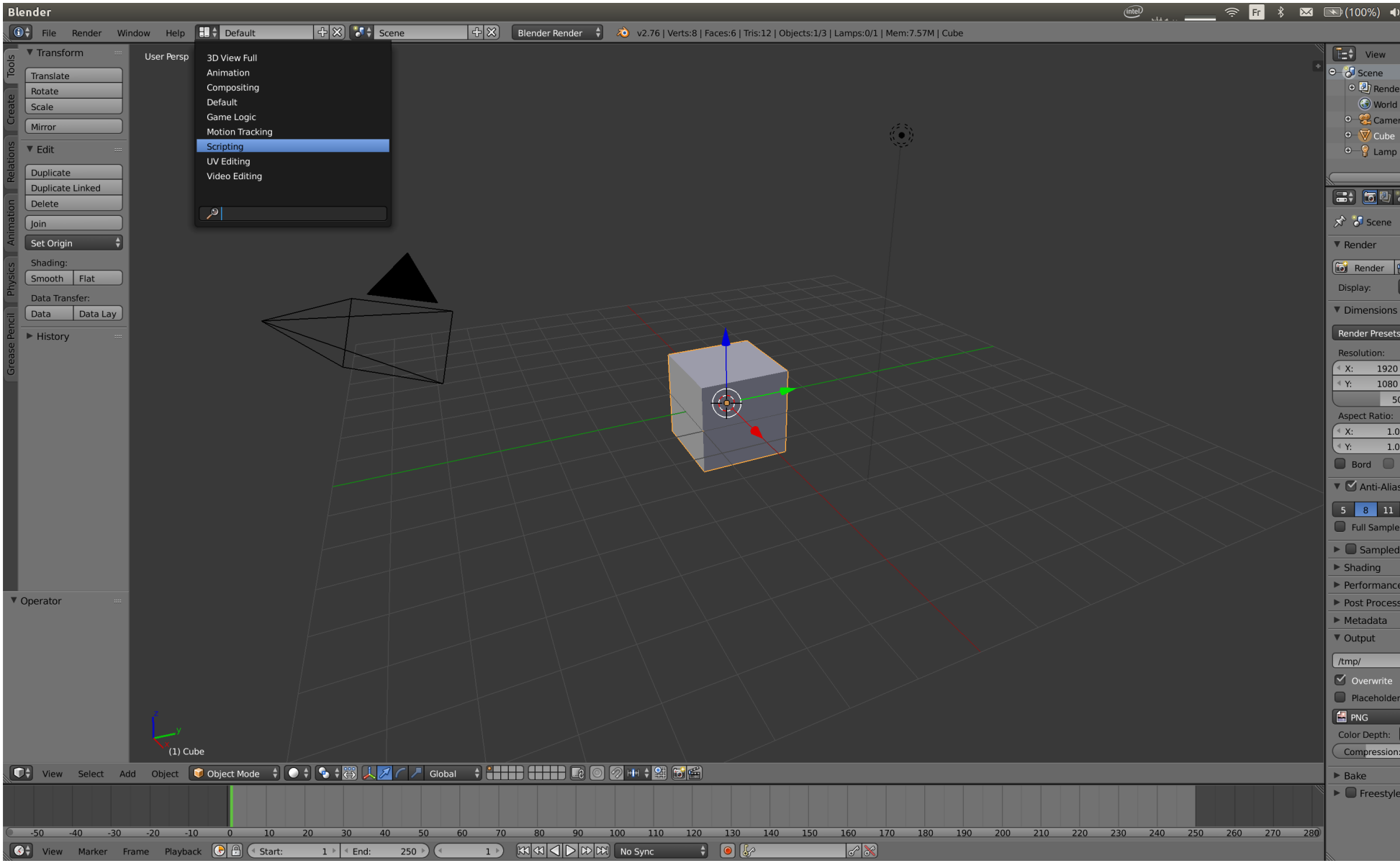
Présentation de Blender/Python

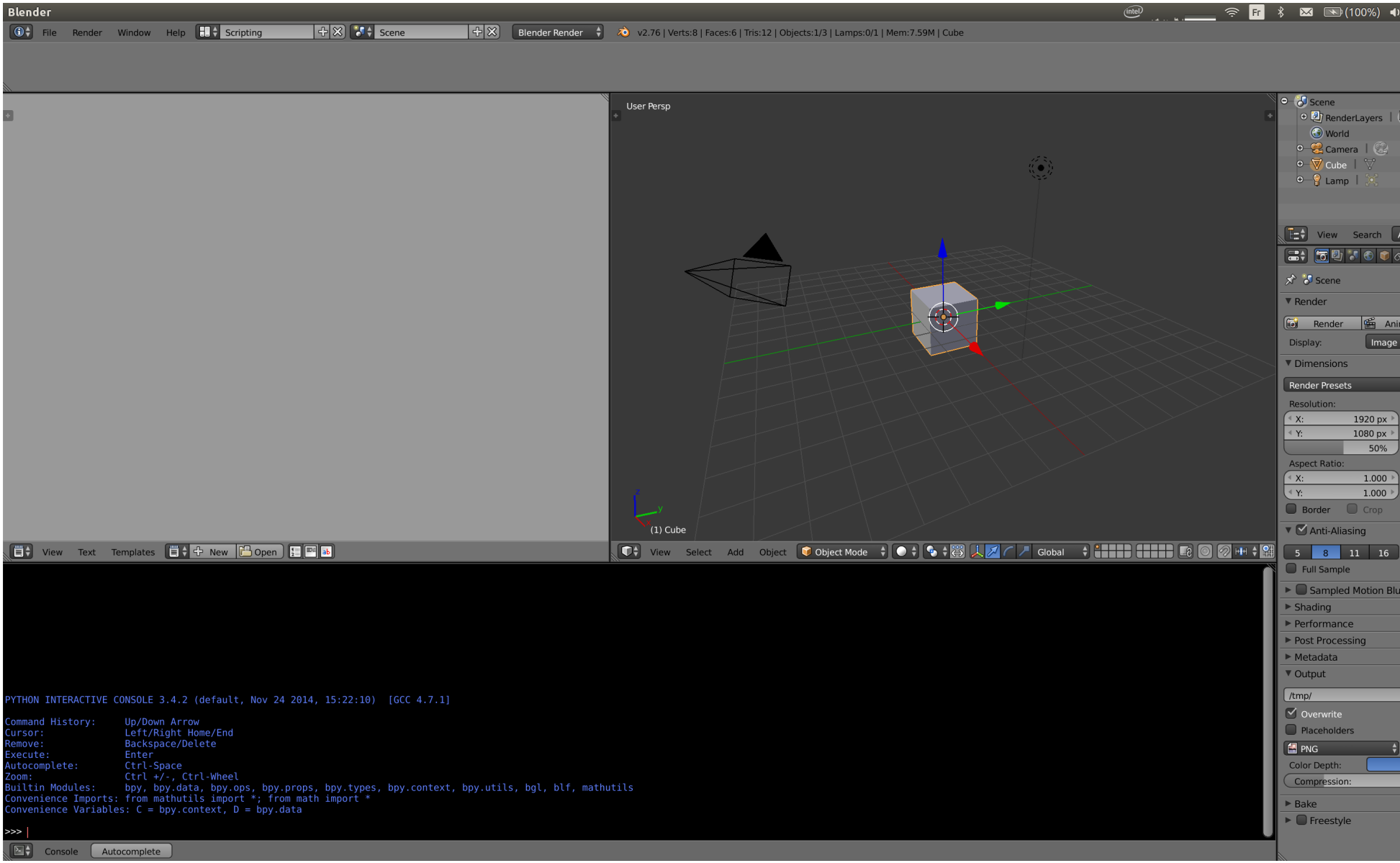
Etienne, Pijus, Raphaël

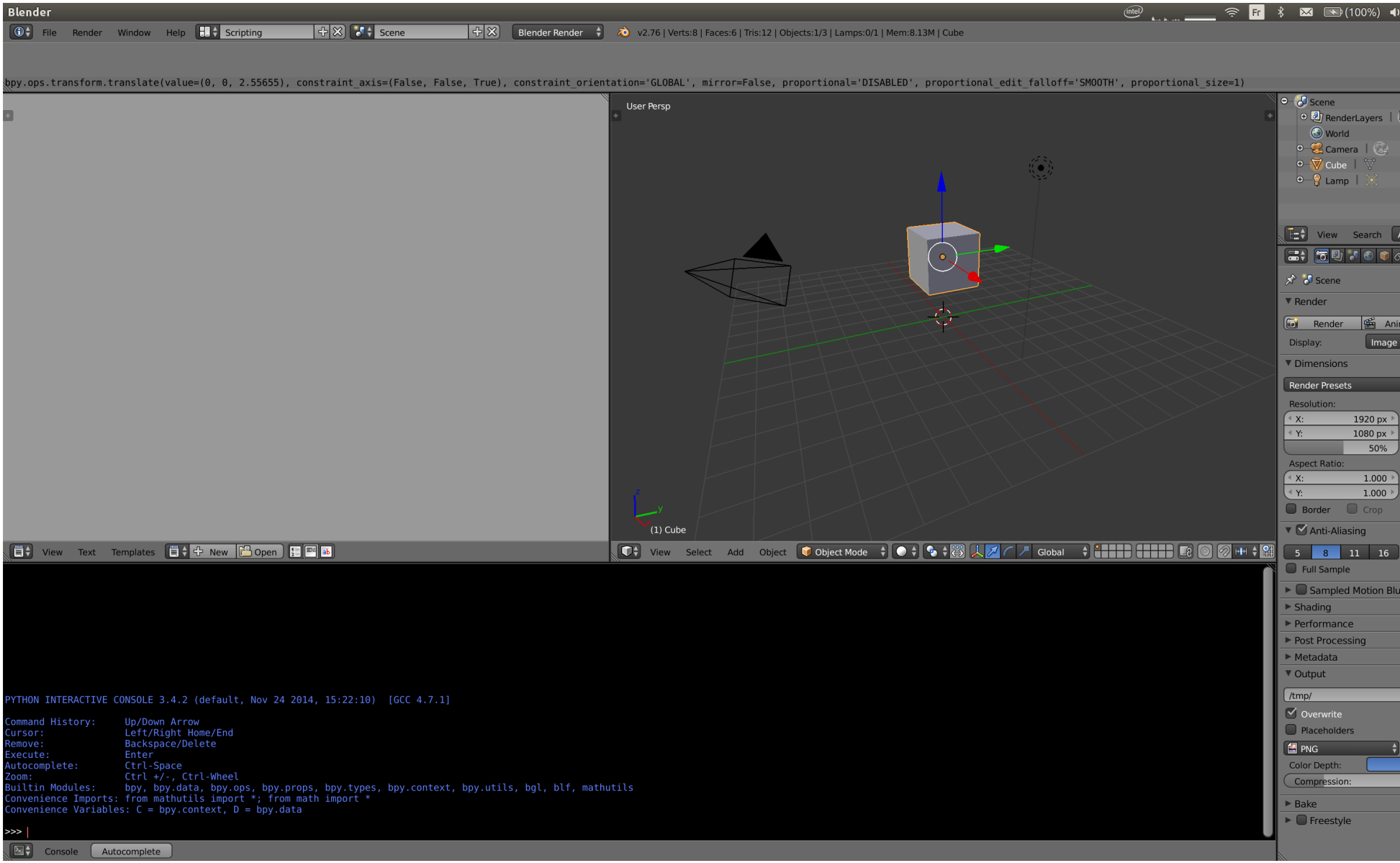
12 novembre 2015


Editor
Region
Panel
Control
Header
Context Buttons

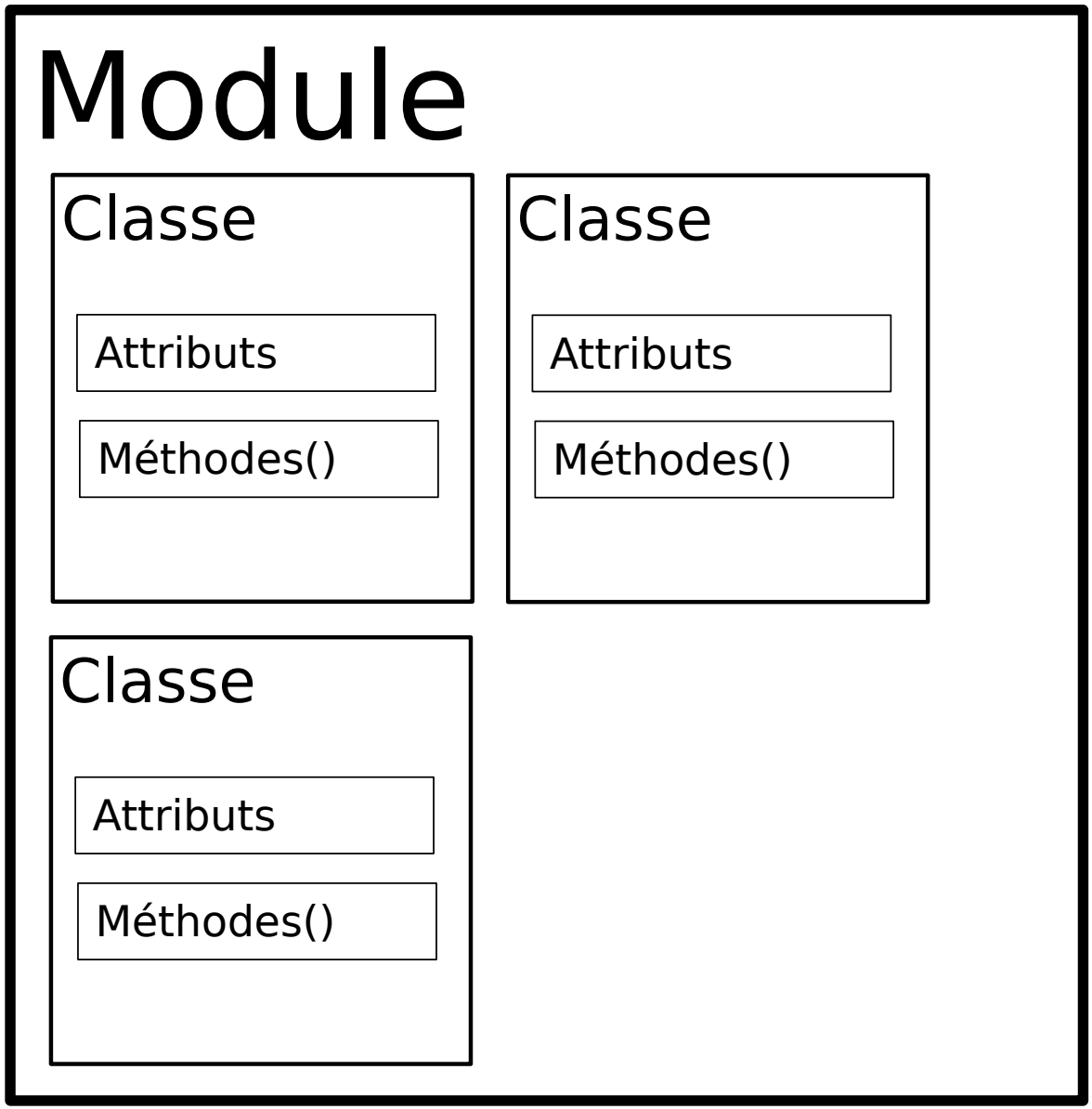
- Usage de la souris et du clavier
- Console et erreurs


Blender
File
Render
Window
Help
Default
Scene
Blender Render
v2.76 | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Lamps:0/1 | Mem:7.57M | Cube
3D View Full
Animation
Compositing
Default
Game Logic
Motion Tracking
Scripting
UV Editing
Video Editing
User Persp
Transform
Translate
Rotate
Scale
Mirror
Edit
Duplicate
Duplicate Linked
Delete
Join
Set Origin
Shading:
Smooth
Flat
Data Transfer:
History
Operator
(1) Cube
View
Select
Add
Object
Object Mode
Global
Render
Dimensions
Resolution:
Aspect Ratio:
Anti-Alias
Sampled
Shading
Post Process
Metadata
Output
/tmp/
Overwrite
PNG
Color Depth:
Bake
Freestyle
View
Marker
Frame
Playback
Start:
End:
No Sync


Blender
File Render Window Help Scripting Scene Blender Render
v2.76 | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Lamps:0/1 | Mem:7.59M | Cube
User Persp
(1) Cube
View Text Templates New Open
View Select Add Object Object Mode Global
PYTHON INTERACTIVE CONSOLE 3.4.2 (default, Nov 24 2014, 15:22:10) [GCC 4.7.1]
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
>>>
Console Autocomplete
Scene
RenderLayers
World
Camera
Cube
Lamp
View Search
Render
Display: Image
Dimensions
Render Presets
Resolution:
X: 1920 px
Y: 1080 px
50%
Aspect Ratio:
X: 1.000
Y: 1.000
Border Crop
Anti-Aliasing
5 8 11 16
Full Sample
Shading
Performance
Post Processing
Metadata
Output
/tmp/
Overwrite
Placeholders
PNG
Color Depth:
Compression:
Bake
Freestyle


Blender
File Render Window Help Scripting Scene Blender Render v2.76 | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Lamps:0/1 | Mem:8.13M | Cube
bpy.ops.transform.translate(value=(0, 0, 2.55655), constraint_axis=(False, False, True), constraint_orientation='GLOBAL', mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1)
User Persp
(1) Cube
Scene
RenderLayers
World
Camera
Cube
Lamp
View Search
Render
Display: Image
Dimensions
Render Presets
Resolution:
X: 1920 px
Y: 1080 px
50%
Aspect Ratio:
X: 1.000
Y: 1.000
Border Crop
Anti-Aliasing
5 8 11 16
Full Sample
Sampled Motion Blu
Shading
Performance
Post Processing
Metadata
Output
/tmp/
Overwrite
Placeholders
PNG
Color Depth:
Compression:
Bake
Freestyle
View Text Templates New Open
View Select Add Object Object Mode Global
PYTHON INTERACTIVE CONSOLE 3.4.2 (default, Nov 24 2014, 15:22:10) [GCC 4.7.1]
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
>>>
Console Autocomplete

```
1 bpy.ops.transform.translate(value=(0, 0, 2.55655),
constraint_axis=(False, False, True),
constraint_orientation='GLOBAL', mirror=False,
proportional='DISABLED', proportional_edit_falloff='SMOOTH
', proportional_size=1)
```


Module
Classe
Attributs
Méthodes()
Classe
Attributs
Méthodes()
Classe
Attributs
Méthodes()

Un fichier **monmodule.py** avec une classe **MaClasse**.
Dans un autre fichier :

```
1  import monmodule

3  objet = monmodule.MaClasse()
```

Un fichier **monmodule.py** avec une classe **MaClasse**.
Dans un autre fichier :

```
1  from monmodule import MaClasse

3  objet = MaClasse()
```

Un fichier **monmodule.py** avec une classe **MaClasse**.
Dans un autre fichier :

```
1  from monmodule import MaClasse, MaClasse2

3  objet = MaClasse()
```

Un fichier **monmodule.py** avec une classe **MaClasse**.
Dans un autre fichier :

```
1  from monmodule import *

3  objet = MaClasse()
```

- Scripts : développement rapide
- Addons : ce qu'il faudra avoir à la fin
  - Il faut ajouter des infos telles que nom, version, ...

Addon = Module

```
1 bl_info = {"name": "My Test Addon", "category": "Object"}
  def register():
3     print("Hello World")
  def unregister():
5     print("Goodbye World")
```

Du code simple...

```
1 import bpy

3 scene = bpy.context.scene
  for obj in scene.objects:
5     obj.location.x += 1.0
```

...À l'addon

```
bl_info = {
    "name": "Move X Axis",
    "category": "Object",
}

import bpy


class ObjectMoveX(bpy.types.Operator):
```

```
class ObjectMoveX(bpy.types.Operator):
    """My Object Moving Script"""      # blender will use this as a tooltip for menu items and buttons.
    bl_idname = "object.move_x"        # unique identifier for buttons and menu items to reference.
    bl_label = "Move X by One"         # display name in the interface.
    bl_options = {'REGISTER', 'UNDO'}  # enable undo for the operator.

    def execute(self, context):        # execute() is called by blender when running the operator.

        # The original script
        scene = context.scene
        for obj in scene.objects:
            obj.location.x += 1.0

        return {'FINISHED'}            # this lets blender know the operator finished successfully.

```

```
2 def register():
      bpy.utils.register_class(ObjectMoveX)
4

6 def unregister():
      bpy.utils.unregister_class(ObjectMoveX)
8

10 # This allows you to run the script directly from blenders
       text editor
   # to test the addon without having to install it.
12 if __name__ == "__main__":
      register()
14
```

Tabulations

4 espaces

Espaces

Opérateurs entourés d'espaces :

```
variable = 'valeur'
ceci == cela
1 + 2


```

Non :

```
variable='valeur'
ceci==cela
1+2
```

Espaces

Sauf...

- Groupes d'expressions

```
a = x * 2 - 1
b = x * x + y * y
c = ( a+b ) * ( a-b )
```

- "=" dans les arguments d'une fonction

```
def fonction(arg='valeur'):
resultat = fonction(arg='valeur')
```

- parenthèses

```
2 * (3 + 4)
```

Noms de variables

Lettres uniquement, en minuscule :

```
for x in range(10):
    print(x)

i = get_index() + 12
print(ma_liste[i])
```

Underscores autorisés pour : modules, variables, fonctions et méthodes.

Pour les noms de classes : lettres uniquement, une majuscule par mot :

```
class CeciEstUneClasse:
    def methodiquement(self):
        pass
```

- RTM
  `http://www.blender.org/manual/`
- PEP-8
  http ://sametmax.com/le-pep8-en-resume/