

# The creation of simple plants by the help of Python script in Blender

Pavel Pokorný

Faculty of Technology in Zlín  
Thomas Bata University  
Mostni 5139, 760 01 Zlín,  
Czech Republic  
pokorny@ft.utb.cz

## ABSTRACT

Blender is the open source software for 3D modeling, animation, rendering, post-production, interactive creation and playback. This software is fully free (with source code under GNU GPL) to use and distribute within any educational, professional or commercial environment. We can use some modeling methods implemented in Blender or Python scripts for the creation of our artwork. Blender has implemented an embedded Python interpreter, which allows it to run scripts written in that language. These scripts can use the Blender Python API to access the program's internals and greatly expand Blender's functions. In this paper, a simple Python script for the plants generation has been described and parameters and a random factor have been set.

## Keywords

Programming, Python script, Blender

## 1. INTRODUCTION

Blender is an integrated suite of tools enabling the creation of and replay of linear and real-time, interactive 3D content. It offers full functionality for modeling, rendering, animation, post-production, game creation and playback with the singular benefits of crossplatform operability and a download file size of less than 4.8 MB. [Ble05a] [Ble05b]

Python is an interpreted, interactive, object-oriented programming language. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level of dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems (X11, Motif, Tk, Mac, MFC). New built-in modules are easily written in C or C++. Python is also usable as an extension language for applications that need a programmable interface.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Rqwgtu'r'tqeggf lpi u'KDP": 2/: 8; 65/26/8*  
*WSCG'2006, January 30-February 3, 2006*  
*Plzen, Czech Republic.*  
Copyright UNION Agency – Science Press

## 2. ABOUT BLENDER

The user communicates with Blender via the keyboard and the mouse, the program gives feedback via the screen and its windowing system (Figure 1). Blender's interface allows using three mouse buttons and a wide range of hotkeys.

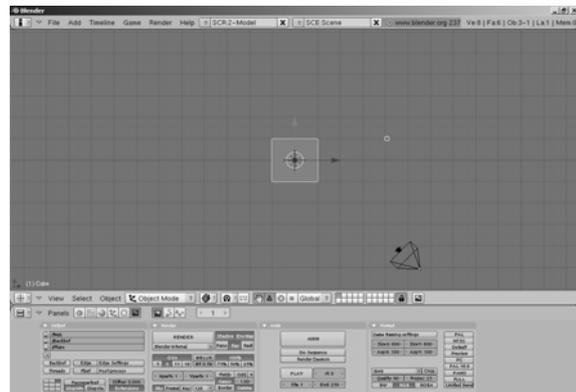


Figure 1. The Blender scene.

The 3D viewport window shows the environment of the current scene in the default viewing direction, but we can change it (Blender offers us to work in three-dimensional or two-dimensional space).

When we create a scene in Blender, we start with a few basic elements that include simple cube object, a camera and a light. We can add more cameras and more lights, but this scene is ready for rendering too.

### 3. BLENDER SETTING FOR THE PYTHON

Blender supports more types of windows. Each window in Blender has a specific appearance and function. We can switch the type of window and with this we can set the basis for our work. We use the window type “Text editor” to write any texts of comments to our projects or program code in Python language.

Blender has implemented Python compiler with its commands and syntax. [Ble05a] Therefore in many scripts we do not need any Python compiler. We compile and run our Python script by the command “Run Python Script” in menu File of its window.

### 4. THE CONCEPTION OF PYTHON SCRIPTS IN BLENDER

Each project in Blender is composed of blocks. These blocks are interconnected. Examples of these blocks are Scene (in hierarchy it is at the outside and it contains others objects), World (information about the environment – colors, background, mist, stars, etc.), Object (information about the position, scale and rotation; it contains most of the elements in the scene – meshes, curves, lamps, cameras, etc.), Material (information about the surface of objects; it can contain the texture).

This paper’s name is “The creation of simple plants”. Each plant is composed of mesh objects. The substance of this script is a mesh creation. The creation of a simple mesh object is described in following steps: creation of a new mesh, setting number of vertices and add them to this mesh, generation a face for these vertices and append to the mesh and the last step is to add the mesh into the Blender scene.

### 5. THE PYTHON SCRIPT FOR THE CREATION OF SIMPLE PLANTS

Above I have described a simple Python script. Now we will generate a simple plant with some leaves. Each leaf of this plant can be a separated rectangular mesh object.

The leaves does not look realistic (they are rectangular), but it is sufficient for us. This plant is universal – the real faces of leaves give us a material setting. We add material with a texture to all of leaves. This texture will have two layers – the bitmap picture with any leaf and its alpha map. In material menu we set the alpha channel. After this, the final picture with materials of background and world is shown in the Figure 5. Here I use a scanned leaf of a buckhorn.

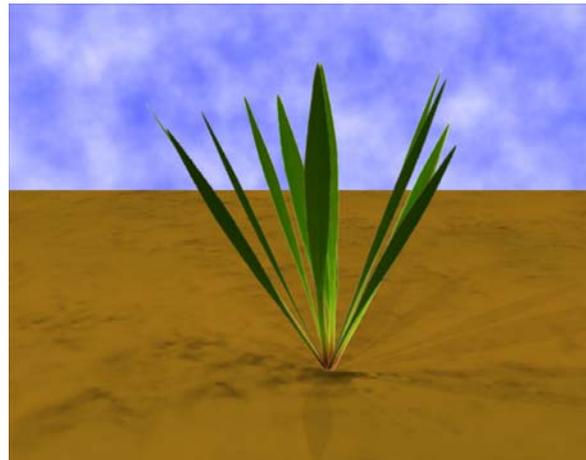


Figure 5. The bitmap – a final render of my scene with a simple plant.

Not all the plants are the same - we want to generate different plants. In our example we can change some parameters – the number and the size of leaves and angles to  $x$ ,  $y$  and  $z$  axis. To this we can add a random factor. The randomize part gives us a different face for the same plants.

I have designed a simple window with the use of the buttons. These buttons set parameters for each leaf. The each of leaves is generated in the same place and after this is transformed (scale  $x$  and  $y$ ,  $x$  and  $z$  rotation, random factors) into the final position. We can set these transformations with corresponding buttons. With this we can generate varied plants.

### 6. CONCLUSIONS

In this paper there, the usage of the Python scripts in Blender has been presented. These subjects are taught at our university within the Computer Graphic.

I have illustrated a programmable simplicity of Python scripts. This example (random plants generator) shows the use of mesh commands in Python language.

### 7. REFERENCES

- [Ble05a] Blender development page at <http://www.blender.org>
- [Cer02a] Černohous, P. Problems with the Python scripts in Blender at <http://www.grafika.cz>
- [Lut03a] Lutz, M., and Ascher, D. Learn Python. Prague, Grada press, 2003.
- Ostrava, pp. 267-270, 2004, <http://www.ineer.org>
- [Pok05a] Pokorný, P., and Sysel, M. Using Python script in Blender. In: DAAAM, Opatia, pp. 305-306, 2005, <http://www.daaam.com>
- [Pyt05a] Python official page at <http://www.python.org>