

Научная статья
УДК 004.94

ПРИМЕНЕНИЕ API PYTHON В 3D-МОДЕЛИРОВАНИИ BLENDER

Анастасия Юрьевна Чевардина¹, Владимир Викторович Побединский²
^{1,2} Уральский государственный лесотехнический университет,
Екатеринбург, Россия
¹ chevardinaayu@m.usfeu.ru
² pobed@e1.ru

Аннотация. В данной статье рассмотрена интеграция *API Python* в популярную систему для трехмерного моделирования *Blender*, достоинства и недостатки языка *Python*. Представлено средство интеграции *Python* в *Blender* – модуль *bpy*, в который входит набор классов, методов и функций, а также основные используемые подклассы. Приведены примеры применения кода на языке *Python* в *Blender*.

Ключевые слова: информационные технологии, 3D-моделирование, Python, Blender, скрипт

Original article

APPLICATION OF THE API PYTHON IN 3D MODELING BLENDER

Anastasia Yu. Chevardina¹, Vladimir V. Pobedinsky²
^{1,2} Ural State Forest Engineering University, Yekaterinburg, Russia
¹ chevardinaayu@m.usfeu.ru
² pobed@e1.ru

Abstract. This article discusses the integration of the Python API into the popular system for three-dimensional modeling Blender. Advantages and disadvantages of Python. A tool for integrating Python into Blender is presented - the *bpy* module, which includes a set of classes, methods and functions, as well as the main using subclasses. Examples of using Python code in Blender are given.

Keywords: information technology, 3D modeling, Python, Blender, script

Информационные технологии прочно укоренились в жизни и быту даже самых отстраненных от компьютерных технологий людей. В большинство электрических приборов сегодня встраивается программное обеспечение. Ежедневно выпускается огромное количество разнообразных приложений, а системы искусственного интеллекта совершенствуются на базе обработки

больших данных ежечасно. Средствами трехмерного моделирования, наряду с действительностью реального мира, создается виртуальная реальность, которая стремится к тому, чтобы ее сложно было отличить от действительности. Все перечисленные передовые технологии настоящего времени объединяют языки программирования, на которых они базируются. Одним из наиболее популярных и востребованных на сегодняшний день является *Python*.

Язык *Python* развивается уже более двух десятков лет. Однако наибольшую популярность он сыскал в последнее время. На данном языке программирования обрабатывают большие данные, разрабатывают приложения и веб-сайты, его используют для разработки искусственного интеллекта и тестирования других программ. Такая востребованность данного языка неслучайна. К достоинствам *Python* относят кроссплатформенность, минималистичный синтаксис, совместимость с другими языками программирования, например с *Java*, *C* и *C++*, а также обширная библиотека, предлагающая пользователям ряд готовых решений практических задач. Применению данных материалов на практике способствует поддержка модульной системы в *Python*. Таким образом, благодаря разработчикам языка программирования и другим пользователям в некоторых случаях можно подключить готовые модули и правильно их скомбинировать. *Python* относится к интерпретируемым языкам программирования, поэтому написанный на нем код исполняется с определенной задержкой и расходом дополнительных ресурсов компьютера, что относят к главному недостатку данного языка программирования [1].

Программный комплекс *Blender* является свободно распространяемым инструментом для трехмерного моделирования. В данную программную среду изначально входит большое количество режимов работы, инструментов, функций и расширений. Однако несмотря на этот факт в программный комплекс внедрен еще один очень полезный инструмент – интеграция с *API Python*, позволяющая производить многие преобразования в сцене программным путем. Для подобных работ заложен режим *Scripting*, представленный на рис. 1.

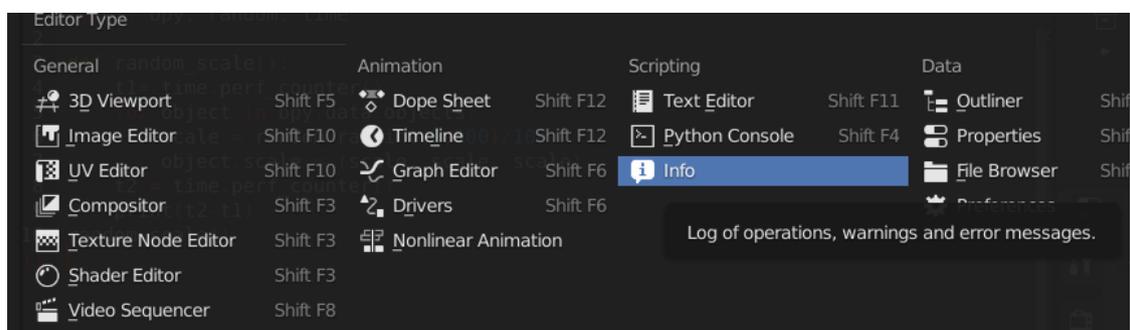


Рис. 1. Режим работы *Scripting* в меню выбора редакторов

В режим *Scripting* входят:

- текстовый редактор;
- консоль Python;
- справка.

Модуль, в который входит набор классов, методов и функций, относящихся к *Blender* в *Python*, называется *bpy*.

Можно выделить основные подклассы модуля *bpy*:

– *bpy.context* – определение вызываемой части пользовательского интерфейса (сцена/объект/коллекция);

– *bpy.data* – доступ к внутренним данным *Blender*;

– *bpy.path* – путь к данным *Blender*;

– *bpy.types* – все типы, работающие в *Blender*. Наиболее часто используется для регистрации операторов и элементов пользовательского интерфейса;

– *bpy.ops* – доступ *Python* к вызывающим операторам, включая операторы, написанные на *C*, *Python* или макросы. Операторы имеют ряд своих классов, таких как *texture*, *scene*, *sculpt*, *uv*, *particle*, *mesh*, *object* и др. Также можно создавать собственные подклассы, но важно помнить, что в *Blender* операторы контекстно зависимы, то есть зависят от того места, откуда происходит вызов данного оператора;

– *bpy.props* – свойства и параметры объектов в *Blender*;

– *bpy.utils* – содержит служебные функции, специфичные для *Blender*, но не связанные с внутренними данными блендера [2].

Модуль *bpy* устроен таким образом, чтобы программно можно было обратиться к любому объекту, его свойству или параметру и задать ему необходимые настройки.

Программный комплекс *Blender* изначально позволяет создавать и анимировать модели различной сложности и гибко настраивать пользовательский интерфейс. Однако работа с трехмерной графикой достаточно трудоемкая и занимает большое количество времени. Задачу автоматизации в *Blender* позволяет решить *Python*. С помощью программного кода можно обратиться к любой точке/кривой/мешу/объекту/сцене или коллекции объектов и задать размеры, локализацию или вращение, установить связи и генерацию новых элементов в рабочей области. Данные возможности значительно облегчают и ускоряют работу пользователя в тех случаях, если однородных объектов в сцене большое количество, либо для процесса моделирования требуется периодическая генерация элементов с заданными параметрами. Также для удобства пользователя можно создать собственную панель с необходимыми полями, параметрами и кнопками. В принципе своем предустановленные и свободно распространяемые расширения в *Blender* являются теми же скриптами, написанными на языке *Python*. Разница между *add-on* (расширением блендер) и скриптом состоит в том,

что *add-on* имеет специальное описание и структуру, где указан автор, применение и прочие параметры.

Примеры применения скрипта на языке *Python* в программном комплексе *Blender* представлены на рис. 2–4 [2].

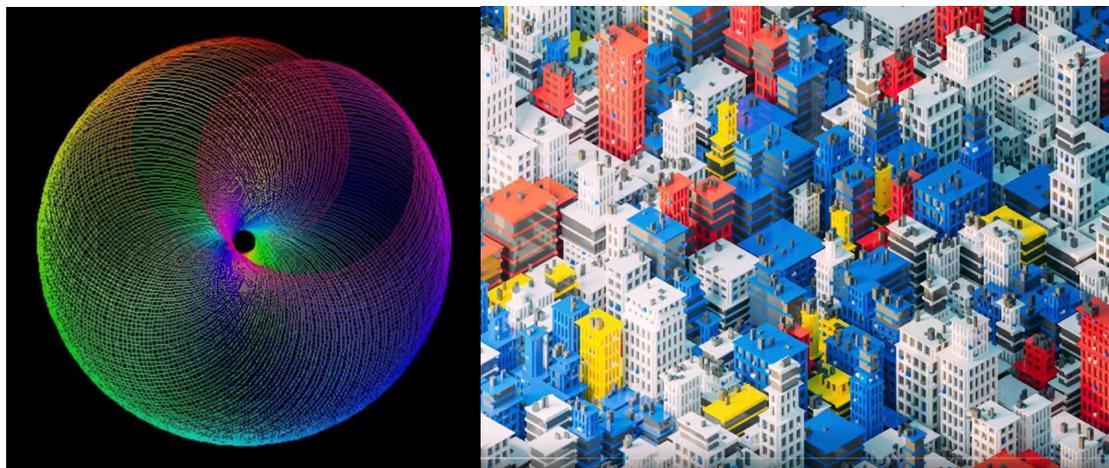


Рис. 2. Примеры применения кода на языке *Python* в *Blender*

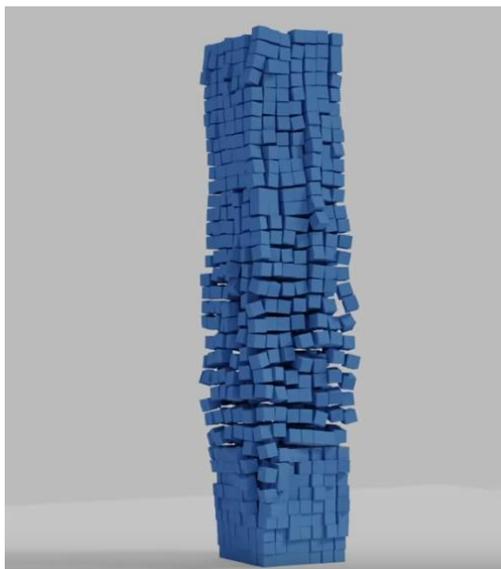


Рис. 3. Пример применения кода на языке *Python* в *Blender*

Можно сделать вывод, что режим *Scripting* в *Blender* позволяет выполнять программным путем базовые действия, связанные с элементами классического моделирования, анимацией сцены, настройкой имитации света и рендера, накладыванием текстуры на объект и другие основные инструменты работы *Blender*, при этом позволяя автоматизировать трудоемкий процесс построения 3D-модели средствами *API Python*. *Blender* дает возможность обратиться к любому объекту в рабочей области, будь то отдельная точка, грань, полигон или группа объектов, чтобы произвести над ними

определенные действия. Помимо этого можно генерировать не только графическую модель как таковую и связанные с ней инструменты такого типа, как свет или рендер, но и сами элементы интерфейса *Blender*, например панели или вкладки с интересующими пользователя параметрами. В совокупности слияние этих двух инструментов дают практически безграничные возможности разработчику и местами облегчают ему труд по настройке и созданию большого количества объектов в сцене как при помощи кода, так и при помощи уже заложенных инструментов и готовых модулей.

Список источников

1. Язык программирования Python для начинающих [Электронный ресурс]. URL: <https://pythonworld.ru/> (дата обращения: 14.09.2023).
2. Blender 3.6 Python API Documentation // Blender [Электронный ресурс]. URL: <https://docs.blender.org/api> (дата обращения: 20.09.2023).
3. How to create graphics using Python // youtube.com [Электронный ресурс]. URL: <https://www.youtube.com/shorts/eaMDDseIL7E> (дата обращения: 05.10.2023).
4. Artistic Coding in Blender by David Mignot // youtube.com : [сайт]. URL: <https://clck.ru/37bDEP> (дата обращения: 05.10.2023).
5. 3D Programming with Python and Blender for Physics Simulations // youtube.com [сайт]. URL: <https://clck.ru/37bDLH> (дата обращения: 05.10.2023).