УДК 004.925 **Технические науки**

Качурин А.В., *магистрант кафедры «Тепловые двигатели и гидромашины»*
Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия
Ильичев В.Ю., *к.т.н., доцент кафедры «Тепловые двигатели и гидромашины»*
Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия

ОСНОВЫ ФОРМИРОВАНИЯ ВИЗУАЛЬНОЙ КАРТИНЫ ШУМА ПЕРЛИНА ПРИ ПРОГРАММИРОВАНИИ НА ЯЗЫКЕ PYTHON

Аннотация: Статья посвящена описанию принципов формирования визуального эффекта, называемого шумом Перлина. Создана программа на языке Python, позволяющая с использованием пользовательских функций и модуля Perline-Noise создавать трёхмерную картину шума Перлина, вид которой зависит от начальных условий и прочих описанных в работе аргументов функций. Данная программа позволяет не только продемонстрировать формирование шума Перлина, так и самостоятельно произвести разнообразные численные эксперименты с использованием описанной технологии. Осуществлённая работа также позволяет исследовать некоторые часто применяемые на практике визуальные и вычислительные возможности языка Python и поэтому может быть полезной при его изучении.

Ключевые слова: визуальный эффект, шум Перлина, программирование, библиотека Perline-Noise, язык Python.

Annotation: The article is devoted to describing the principles of forming a visual effect called Perlin noise. A Python program has been created, which allows

using user-defined functions and the Perline-Noise module to create a three-dimensional picture of Perlin noise, the appearance of which depends on the initial conditions and other arguments of the functions described in the work. This program allows not only to demonstrate the formation of Perlin noise, but also to independently perform various numerical experiments using the described technology. The work carried out also allows you to investigate some of the often used in practice visual and computational capabilities of the Python language and therefore can be useful in studying it.

Keywords: visual effect, Perlin noise, programming, Perline-Noise library, Python language.

Введение

В последнее время разрабатывается всё большее количество математических методов создания компьютерной графики [1], широко используемых для многих целей:

1. повышения реализма структуры поверхности виртуальных объектов;
2. генерации новых визуальных эффектов;
3. имитации физических процессов и явлений.

Так называемый шум Перлина [2] представляет собой визуальную структуру, обладающую как случайными свойствами, так и некоторой упорядоченностью, степень которой можно изменять с помощью применения специальных программных средств. Такие искусственно созданные структуры нашли наибольшее применение в компьютерной графике для имитации реалистичных природных текстур, например образов облаков, огня, горных и лесных массивов. Также возможна генерация турбулентных потоков и сильно неупорядоченных структур, например броуновского движения частиц. Можно сказать, что шум Перлина используется повсеместно в визуальных сценах, сгенерированных компьютером (CGI) [3].

Данный тип текстуры генерируется по псевдослучайному алгоритму путём применения специального вида интерполяции. Одним из наиболее

распространённых алгоритмов при создании визуальных эффектов является так называемый классический шум Перлина, созданный ещё на заре мультимедиа в 1983 г. [4]. С тех пор алгоритм получил реализацию на разных языках программирования и развивался с целью создания всё более сложных и максимально реалистичных объектов. Были придуманы алгоритмы создания функции шума любой необходимой размерности.

Создание классического шума Перлина в 1-D включает в себя 3 этапа:

- создание прямой, на которой через определённый шаг строятся векторы, имеющие случайное направление;
- вычисление по алгоритму Перлина координат точек на плоскости или пространстве между созданными векторами;
- интерполяция координат точек.

Наиболее современной библиотекой для вычисления координаты шума Перлина в 1-D на языке программирования Python является модуль perline-noise, позволяющей вычислять изменение значения функции шума в зависимости от координаты.

Целью данной работы является исследование влияния аргументов функции шума Перлина на вид получаемого при этом изображения, а также комбинирования нескольких функций шума с разными параметрами для создания разных типов визуальных структур.

Материал и методы исследования

Как было сказано выше, для получения шума Перлина в 1-D создаётся прямая, разделённая псевдослучайно направленными векторами (рис. 1). Направление векторов зависит от заданных начальных условий, в качестве которого используется целое число [5].

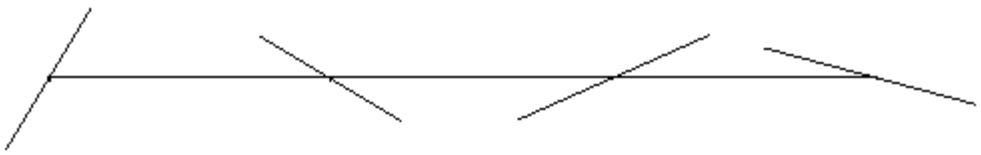

Рис. 1. Векторы псевдослучайного направления

На следующем этапе создаётся (подобно алгоритму кривых Безье) волнистая линия, касающаяся последовательно всех векторов. Однако, данная волнистая линия является слишком плавной, чтобы напоминать структуру природных объектов. Далее необходимо проделать следующую операцию: увеличивается разрешение исходной картины (данное разрешение называют октавой – если шаг между псевдослучайными векторами уменьшается в 2 раза, то это вторая октава). Если на рис. 1 присутствуют 3 отрезка, то на второй октаве число отрезков увеличивается до 6. Затем для всех точек картины повышенного разрешения повторяется операция построения векторов и касательной к ним кривой, и данная кривая складывается с первоначальной. Если действовать по приведённому алгоритму и далее, то на третьей октаве складываются уже 3 кривых.

При сложении 4 и более октавных кривых результирующая картина получается уже несколько «ступенчатой» и напоминает присущее природным объектам распределение параметров.

Описанный алгоритм генерирования координат 1-D шума Перлина реализован в библиотеке perline-noise для языка Python. С помощью применения разного количества октав и начальных условий для каждой оси координат, можно создать также 2-D или 3-D картину шума Перлина, а комбинированием вклада нескольких октав в одну координату получают сложные картины шума. Для решения этой задачи была составлена программа, выполняющая следующую последовательность действий (на примере генерации 3-D изображения):

1. импорт функции PerlinNoise из библиотеки perlin_noise;

2. импорт функции Axes3D [6; 7] из библиотеки mpl_toolkits.mplot3d [8] и функции pyplot из библиотеки matplotlib для создания трёхмерного графика;

3. создание функций генерирования координат точек шума Перлина для каждой оси координат, каждая из которых позволяет выбрать количество используемых октав и начальные условия для генерации;

4. создание переменной, означающей количество точек, на которое разбивается отрезок по каждой оси координат;

5. создание трёх пустых массивов, в которые будут заноситься сгенерированные координаты шума Перлина по каждой из осей x, y, z;

6. перебор в цикле последовательно каждой точки осей координат, генерирование координат точек шума Перлина с помощью функций п. 3 и занесение каждой координаты в соответствующий массив п. 5;

7. создание фигуры и трёхмерной области графика;

8. рисование точечного графика по массивам координат с возможностью выбора размера и цвета точек, типа желаемого маркера.

Примеры расчётов

Приведём результаты, получающиеся при расчёте примеров с разным набором начальных условий для вычисления координат шума Перлина и числом используемых октав. Вначале используем одинаковое количество октав (например, 10) и начальные условия (задаётся целым числом, например, 100) для всех осей координат. 3-D график, выводимый при этом программой, представлен на рис. 2. Условный размер точек принят равным 0.01, количество точек 1000.

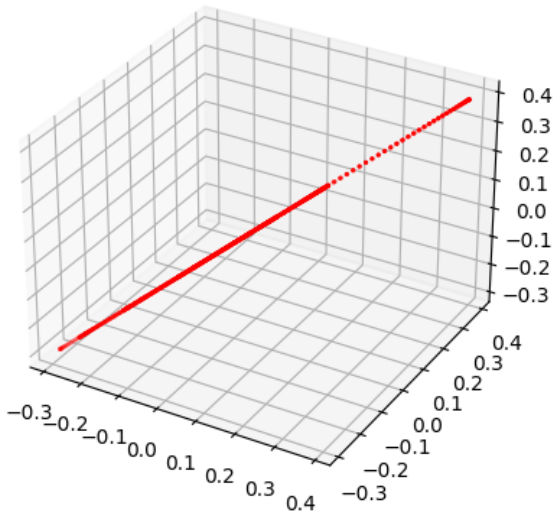

Рис. 2. Шум Перлина, формируемый при одинаковых начальных условиях для осей координат

Наглядно видно, что график является прямым, то есть все координаты изменяются по линейному закону.

Даже при незначительном изменении начальных условий (при значениях 100, 101, 102) график существенно изменяется (рис. 3).

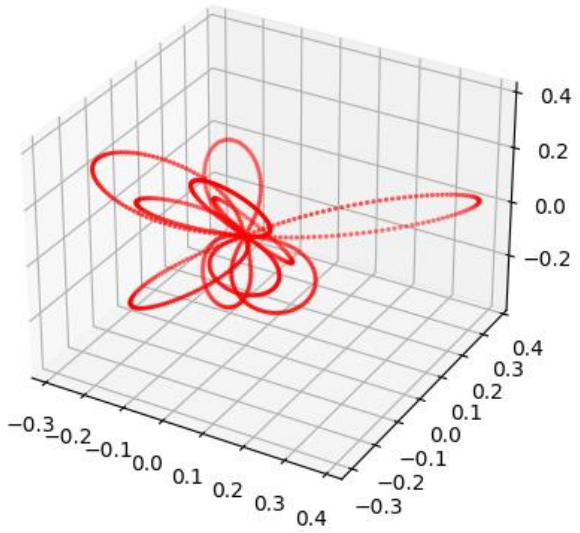

Рис. 3. Шум Перлина, формируемый при отличии начальных условиий для осей координат

Получаем некую сложную траекторию, изменяющуюся в тех же пределах, что и рис. 2. Попробуем теперь больший размер генерируемых точек - 1 (для увеличения наглядности выводимой картины) и одновременно задать небольшое, но разное количество октав для осей координат (1, 2, 3) и разные начальные условия (100, 200, 300). При этом программой выводится картина, также сильно отличающаяся от предыдущих (рис. 4).

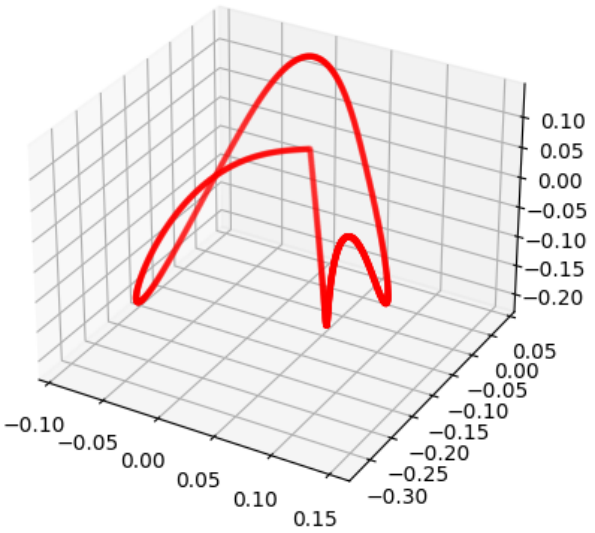

Рис. 4. Шум Перлина при различии количества октав по осям координат

Возможны и другие варианты искусственно сгенерированных с помощью функции Шум Перлина PerlinNoise картин. Таким образом, варьируя начальные условия и количество октав, можно получить очень интересные для разных отраслей науки и техники массивы данных (например, имитирующие разные виды течений – ламинарных, турбулентных) [9].

Например, с использованием той же графической библиотеки Matplotlib и четырёхкратного применения функции PerlinNoise с разным количеством октав, можно получить картину в 2-D, напоминающую облака (рис. 5). Используя библиотеки вывода графики, отличные от Matplotlib, можно получить и другие разные интересные визуальные эффекты [10].

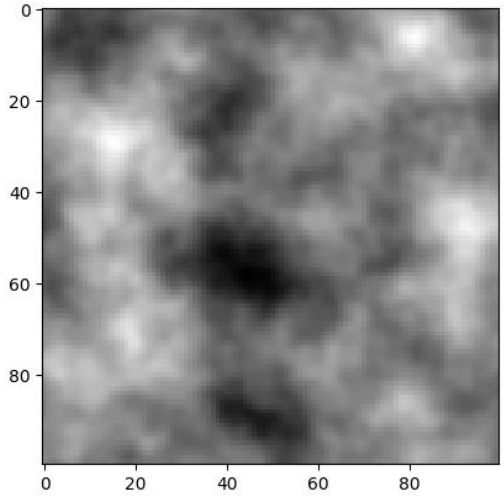

Рис. 5. Шум Перлина, имитирующий вид облаков в 2-D.

Заключение

Таким образом, в работе кратко описана методика генерации шума Перлина и сферы его практического применения.

Цель работы была полностью достигнута: разработана программа на языке Python, позволяющая выводить визуальную трёхмерную картину формируемого шума Перлина при различном сочетании аргументов функции из библиотеки perlin_noise для каждой из осей координат. Также приведена полная последовательность действий, выполняемых программой, описаны используемые в ней команды и подключаемые библиотеки функций.

В ходе дальнейших исследований предполагается применение библиотеки perlin_noise для создания более сложных графических образов. При этом помимо рассмотренных в статье, необходимо применение дополнительных графических средств, например, использование фреймворков [11].

В данной статье рассмотрены только некоммерческие, свободно распространяемые программные средстве, что является важным по причине того, что они доступны всем заинтересованным исследователям, и их освоение не требует какой-либо сложной подготовки. Как правило, средства языка Python являются интуитивно понятными. Демонстрация разработанной программы (и ряда других, разработанных авторами, например, [12]) студентам машиностроительного вуза, а также получение разных картин шума Перлина при вариации начальных условий и аргументов функций программы, показала высокую заинтересованность обучающихся в освоении современных информационных технологий, способствует развитию творческого мышления и повышает интерес к самостоятельным исследованиям.

Библиографический список:

1. Ильичев В.Ю. Разработка программы для исследования аттрактора Лоренца и ее использование. // Сложные системы. 2021. № 1 (38). С. 56-63.

2. Слеповичев С.О. Шум Перлина как способ получения компьютерных спецэффектов природных явлений. // Инновационное развитие. 2017. № 2 (7). С. 32-33.

3. Гребенкин Д.А. Аналитический обзор применения CGI графики в сфере киноиндустрии. // Моя профессиональная карьера. 2020. Т. 2. № 11. С. 237-247.

4. Антипов В.А., Коковкина В.А. Формирование базы синтезированных изображений с использованием модели шумовых воздействий DSPA. // Вопросы применения цифровой обработки сигналов. 2016. Т. 6. № 4. С. 866-869.

5. Perlin-noise 1.7 [Электронный ресурс]. URL: https://pypi.org/project/perlin-noise/ (Дата обращения 24.08.2021).

6. Ильичев В.Ю., Качурин А.В. Создание программ на языке Python для исследования множества Мандельброта. // E-Scio. 2021. № 5 (56). С. 362-371.

7. Ганков М.С., Ильичев В.Ю. Разработка программ на языке Python для графической интерпретации точечных отображений. // Научное обозрение. Технические науки. 2021. № 3. С. 15-20.

8. Ильичев В.Ю., Гридчин Н.В. Визуализация масштабируемых 3d-моделей с помощью модуля Matplotlib для Python. // Системный администратор. 2020. № 12 (217). С. 86-89.

9. Никитин Н.В., Пиманов В.О. Суперкомпьютерное моделирование - путь к пониманию турбулентности. // В сборнике: Суперкомпьютерные технологии в науке, образовании и промышленности. Москва, 2017. С. 163-170.

10. Коротков Н.И., Григорьева И.И. Реализация индивидуального проекта школьника с использованием графических библиотек Python. // Вестник ТОГИРРО. 2018. № 2 (40). С. 93.

11. Козич П.А., Кизянов А.О., Глаголев В.А. Реализация Web приложения с помощью CherryPy на языке программирования Python. // Постулат. 2019. № 1-1 (39). С. 66.

12. Ильичев В.Ю. Использование рекурсивных функций для создания фрактальной графики средствами языка Python. // Системный администратор. 2021. № 3 (220). С. 92-95.