

УДК 004.925

Лунтовская А.А.

Донецкий национальный технический университет
кафедра прикладной математики и информатики
E-mail:anastasia.luntovskaya@gmail.com

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ ОБЛАКОВ И РЕНДЕРИНГ НЕБА В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКОГО ПРОЦЕССОРА

Аннотация

Лунтовская А.А. Процедурная генерация облаков и рендеринг неба в реальном времени с использованием графического процессора. В статье рассмотрен процесс генерации и рендеринга облаков и реализация цикла смены дня и ночи для визуализации динамического 3D окружения в компьютерных играх с использованием графического процессора. Генерация текстуры облаков выполнена с помощью алгоритма шума Перлина. Алгоритм реализован в Unity3D на языке шейдеров Cg.

Постановка проблемы.

Генерация и представление облаков является одной из основных проблем рендеринга реалистичного неба. Для создания неба в играх традиционно используются два основных подхода. В статическом подходе одна или больше фотографий текстурируются на 3D модель неба. Например, текстура облаков может накладываться на голубую текстуру неба. Несмотря на то, что само небо в результате таких действий может быть представлено очень убедительно и реалистично, учитывая качество используемых фотографий, отсутствие динамики является существенным недостатком такого подхода. Второй подход предполагает процедурную генерацию облаков. Объемные методы пытаются моделировать 3D пространство, занимаемое облаком, плоскостные представляют облака с помощью 2D

текстуры, накладываемой на 3D модель неба. При условии, что зритель всегда остается на уровне земли, такой метод вполне может использоваться для создания реалистичного неба. В статье будет рассмотрен алгоритм генерации 2D текстуры облаков, которая впоследствии накладывается на купол неба.

На сегодняшний день алгоритм шума Перлина широко применяется для процедурной генерации туманов, облаков, огня, различных эффектов поверхности воды и земли. Существует множество реализаций шума Перлина на центральном процессоре, но с увеличением мощности графических процессоров становится актуальной реализация алгоритма на ГПУ, что значительно снижает временные затраты, поскольку алгоритм включает в себя достаточно большое количество вычислений, которые могут выполняться параллельно.

Анализ исследований и публикаций.

Кен Перлин [1] в 1983 году предложил алгоритм генерации процедурной текстуры псевдослучайным способом, который в компьютерной графике используется для увеличения реализма поверхности геометрических моделей.

Функция шума имеет несколько важных характеристик:

- для каждой входной позиции вычисляется повторяющееся псевдослучайное значение;
- диапазон генерируемых значений известен и чаще всего принимается за $[-1,1]$;
- очевидные повторяющиеся узоры исключаются;
- пространственная частота инвариантна смещению [2].

Реализация шума Перлина в пиксельном шейдере имеет ряд преимуществ:

- требуется меньше памяти для текстур;
- текстура имеет высокую периодичность, шаблон узора не повторяется часто;
- значительно превосходит результаты ЦПУ по времени;
- более высокое качество интерполяции, чем то, что доступно при аппаратной фильтрации текстур, таким образом, результат

получается более сглаженным;

- появляется возможным создание четырехмерного шума, который является незаменимым для анимационных аффектов.

Паллистер [3] выдвинул идею объединения четырех текстур, созданных с помощью алгоритма шума Перлина, для создания облаков. Первая текстура называется первой октавой. Она используется для задания общей формы облаков. Вторая октава генерируется в меньшем масштабе, чем первая, смешивается с первой для придания большей детальности. Аналогично третья и четвертая октавы генерируются в еще меньшем масштабе, чтобы сделать результирующую текстуру еще более детальной. Для анимации текстуры Паллистер предложил непрерывно модифицировать текстуру, интерполируя ее со вторым набором текстур динамически генерируемого шума. Дубе [4] использовал аналогичный подход с восьмью октавами и добавил алгоритм трассировки лучей в шейдере, чтобы вычислить реалистичное освещение в каждом пикселе. Проблема здесь заключается в том, что такие шейдеры требуют слишком много вычислительных ресурсов. Такое представление облаков, таким образом, делает их недоступными для экранов с высоким разрешением.

Цели и задачи статьи.

Необходимо осуществить процесс создания реалистичных изображений облаков и неба, выполнить генерацию текстуры облаков с помощью алгоритма шума Перлина, реализовать цикл смены дня и ночи для использования в компьютерных играх. Должно быть обеспечено быстроедействие, достаточное для представления таких эффектов окружения в реальном времени.

Реализация алгоритма.

Для реализации анимированных облаков нами были сгенерированы 4 октавы шума Перлина на разных частотах и амплитудах. В общем случае количество октав влияет на реалистичность конечного изображения, но для создания облаков

четырёх октав вполне достаточно.

Каждая октава рассчитывается отдельно, а затем, сложив все эти функции, мы получим результирующую – в достаточной степени гладкую и хаотичную. Создавая текстуры шума несколько раз и объединяя затем в одну, мы добиваемся более реалистичного представления облаков. Однако после такой процедуры сглаживание шума все же требуется. Функция сглаживания возвращает цвет пикселя в зависимости от исходного цвета и цвета соседних пикселей.

Сгенерированную таким образом текстуру облаков необходимо поместить на сферу, иначе облака будут казаться бесконечно уходящими вдаль.

Как было сказано выше, всего генерируется 4 октавы шума. Анимация облаков, таким образом, осуществляется простой интерполяцией от одного шума к другому. Чаще всего функция интерполяции принимает 3 параметра. Первые два, a и b , – параметры, между которыми производится интерполяция, последний – некоторое число в диапазоне от 0 до 1, от которого будет зависеть результат интерполяции: в случае, если параметр равен 0, то возвращается a , если равен 1, то возвращается b , иначе возвращается некоторое значение между a и b . Нужно также учесть, что октавы шума Перлина должны быть анимированы с разной скоростью: октава с большей амплитудой шума должна анимироваться медленнее октавы с меньшей амплитудой шума.

Для представления неба используются 3D модель купола неба, которая размещается над виртуальным миром и всегда центрирована на пользователе. Рендеринг купола неба осуществляется с помощью трех текстур, смешиваемых между собой в зависимости от времени суток. Текстура с голубым градиентом используется для дневного неба, а текстура, представляющая звезды, и текстура с темно-синим градиентом – для ночного.

Сгенерированная текстура облаков накладывается на купол неба. В зависимости от плотности облаков, текстура будет сделана частично прозрачной, позволяя куполу неба быть видимым. Каждая из этих моделей рендерится с помощью

пиксельных шейдеров на языке Cg. Шейдер для облаков вызывается один раз за кадр для каждого пикселя.

Предусмотрены два контролируемых параметра, влияющих на выполнение шейдера: количество облаков и время суток. Время суток меняется перемещением соответствующего ползунка, который позволяет пройти полному суточному циклу. Для задания количества облаков используются соответствующие кнопки, всего предусмотрено 4 степени концентрации облаков.

Результаты работы представлены на рисунке 1.



Рисунок 1 – Результаты генерации облаков

Выводы и перспективы.

Реализация рассмотренного алгоритма процедурной генерации облаков позволяет создавать реалистичное изображение. Поскольку алгоритм реализован с использованием графического процессора, следует ожидать, что в результате достигнуто быстрое действие, достаточное для генерации

изображения облаков в реальном времени. Полученные результаты будут использованы для создания динамического игрового 3D окружения.

Список литературы

6. Perlin, K. Implementing Improved Perlin Noise. – GPU Gems: programming techniques for high-performance graphics and general-purpose computation (Matt Pharr ed), 2005 [электронный ресурс]

http://http.developer.nvidia.com/GPUGems/gpugems_ch05.html

7. Green, S. Implementing improved Perlin noise. – GPU Gems 2: programming techniques for high-performance graphics and general-purpose computation (Matt Pharr ed), 2005 [электронный ресурс]

http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter26.html

8. Pallister, K. Generating Procedural Clouds Using 3D Hardware. – Game Programming Gems 2 (Mark Deloura, ed.), Charles River Media, 2001, pp. 463-473.

9. Dube, J. “Realistic Cloud Rendering on Modern GPUs” in Game Programming Gems 5 (Kim Pallister, ed.), Charles River Media, 2005, pp. 499-505.