

# ПРОГРАММИРОВАНИЕ ПРОЦЕДУРНОЙ ГРАФИКИ С ПОМОЩЬЮ ПРОГРАММНОГО ПАКЕТА HOUDINI

*В.А.Коровкин, ассистент ОИТ ИШИТР,  
Д.И. Мингалеев, студент гр. 8ИМ01,  
Томский политехнический университет  
E-mail: [dim19@tpu.ru](mailto:dim19@tpu.ru)*

## Введение

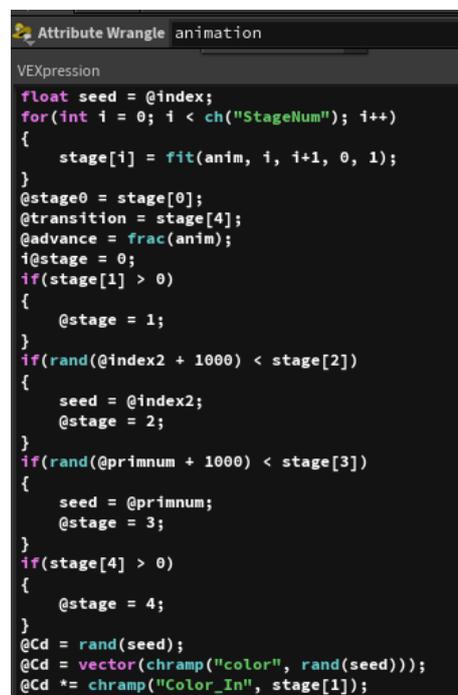
Процедурная генерация контента (ПГК) — это группа особых методов и вычислительных алгоритмов. С помощью которых можно создать различные трехмерные объекты, изменяя различные их параметры. Данными объектами могут являться – дома, растения, ландшафт, мебель и т.д.

Построение моделей различных объектов методом ПГК — актуальная на сегодня задача, которая находит применение во многих сферах, связанных с компьютерной графикой, например, в создании компьютерных игр, спецэффектов, кино, мультфильмов. Этот метод используется тогда, когда нужно получить большое множество моделей с уникальными характеристиками, которое нельзя получить в разумные сроки и при разумных трудозатратах обычным способом построения вручную. Другими словами, активное использование алгоритмов и методов ПГК позволяют сократить ресурсы разработки и оптимизировать труд 3D-художника [1].

## Программное обеспечение Houdini

Houdini – представляет собой профессиональный программный пакет для работы с трёхмерной графикой [2].

Главным отличием данного ПО от других 3D редакторов, таких как: Blender, 3DS Max, 3D Coat, Maya, Cinema4D и остальных подобных ПО заключается в том, что он является средой визуального программирования. Основной минимальной структурой визуального программирования являются ноды (Nodes). Они способны обрабатывать определенные данные соответствующие типу выбранной ноды и возвращать результат в соответствии с данным типом. Также возможно использовать ноду, в которой можно написать часть кода, которая, например, будет отвечать за переход между стадиями анимации, как показано на рисунке 1.



```
Attribute Wrangle animation
VEXpression
float seed = @index;
for(int i = 0; i < ch("StageNum"); i++)
{
    stage[i] = fit(anim, i, i+1, 0, 1);
}
@stage0 = stage[0];
@transition = stage[4];
@advance = frac(anim);
i@stage = 0;
if(stage[1] > 0)
{
    @stage = 1;
}
if(rand(@index2 + 1000) < stage[2])
{
    seed = @index2;
    @stage = 2;
}
if(rand(@primnum + 1000) < stage[3])
{
    seed = @primnum;
    @stage = 3;
}
if(stage[4] > 0)
{
    @stage = 4;
}
@Cd = rand(seed);
@Cd = vector(chramp("color", rand(seed)));
@Cd *= chramp("Color_In", stage[1]);
```

Рис. 9. Код для перехода между стадиями анимации.

Возможности программного обеспечения Houdini, как говорилось ранее, достаточно обширны. В работе создавались процедурно генерируемые планеты, замки, деревья, заполнение стен и крыши дома

кирпичами и черепицей, и тому подобное. Достаточно загрузить базовую форму, создать логику из нод и вынести нужные параметры для настройки процедурно генерируемого объекта в отдельную специальную ноду. Она позволяет не искать нужный параметр по всему дереву логики. При традиционном подходе в графическом редакторе дизайнер вручную расставляет объекты на сцене. На ручное заполнение пространств больших объемов может уходить не только несколько дней, но и несколько месяцев монотонной неэффективной работы [3, 4]. На рисунке 2 приведен пример заполнения стен и крыши дома кирпичами и черепицей.



Рис. 2. Заполнения стен и крыши дома кирпичами и черепицей.

Также процедурная генерация обладает еще одним полезным свойством – экономия ресурсов компьютера. Размещение объектов вручную в пространстве занимает много ресурсов компьютера, так как каждый объект получает свою координату в пространстве. Информация о сгенерированной растительности или о кирпичах на стене здания сохраняется в отдельную карту и занимает меньше ресурсов. Настроив параметры генерации один раз можно легко создавать различные сочетания растительности, различных объектов, которые разбросаны по карте и в дальнейшем передать результат художнику для того что бы генерация выглядела более естественно [5].

### Преимущества ПО Houdini

Данное ПО довольно грубое по отношению к задаче моделирования одного объекта (создание высоко полигональной модели). Для данной задачи подойдет любой другой редактор трёхмерной графики, например, Blender, 3D Max и т.д. Основное направление Houdini – генерация различных эффектов, симуляция поведения различных материалов, так как не требует установки сторонних плагинов.

### Заключение

Планируется дальнейшее изучение ПО Houdini и создание логики процедурно-генерируемых ассетов (моделей) и их последующего экспорта в игровой движок Unreal Engine 4.

### Список использованных источников

1. Процедурная генерация контента. – Текст: электронный // Прикладная Математика и Информатика: [сайт]. – URL: <https://nauchkor.ru/uploads/documents/587d362e5f1be77c40d588b6.pdf> (дата обращения: 05.03.2021).
2. Houdini (графическая программа) – Текст: электронный: [сайт]. – URL: [https://ru.wikipedia.org/wiki/Houdini\\_\(графическая\\_программа\)](https://ru.wikipedia.org/wiki/Houdini_(графическая_программа)) (дата обращения: 05.03.2021).
3. Процедурное создание зданий. – Текст: электронный: [сайт]. – URL: <https://habr.com/ru/post/424203/> (дата обращения: 06.03.2021).
4. Процедурная генерация случайных замков в Houdini. – Текст: электронный: [сайт]. – URL: [https://pikabu.ru/story/protsedurnaya\\_generatsiya\\_sluchaynykh\\_zamkov\\_v\\_houdini\\_7237859](https://pikabu.ru/story/protsedurnaya_generatsiya_sluchaynykh_zamkov_v_houdini_7237859) (дата обращения: 05.03.2021).
5. Как процедурная генерация помогает создавать открытые миры. – Текст: электронный: [сайт]. – URL: <https://ddf.ru/gamedev/169117-kak-procedurnaya-generaciya-pomogaet-sozdavat-otkrytye-miry> (дата обращения: 07.03.2021).