

## РАЗРАБОТКА ТЕХНОЛОГИИ ДЛЯ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ ВИРТУАЛЬНЫХ ПРОСТРАНСТВ НА UNREAL ENGINE 4

*Ч.Т. Куулар, Т.В. Монгуш, В.А. Коровкин*  
*Томский политехнический университет*  
E-mail: ctk1@tpu.ru

### **Введение**

В настоящее время представляет большой интерес изучение различных методов для генерации виртуальных пространств. Круг применения этих методов представляется большим, начиная от генерации пространств для сферы мультипликации и анимации, заканчивая построением целых массивов помещений для нужд видеоигровой индустрии [1].

Процедурная генерация – это автоматическое создание какого-либо медиа контента с помощью различных вычислительных алгоритмов. Другими словами, процедурная генерация представляет собой программное обеспечение, которое может создавать контент самостоятельно или согласно установленными разработчиками правилами [1]. Под контентом в данном разрезе понимается создание уровней, ландшафтов, схем помещений, текстур, предметов, и так далее.

На данный момент существует большое количество различных алгоритмов для создания процедурных пространств. В основном большая часть этих алгоритмов посвящена либо созданию классических лабиринтов, либо генерацию рельефа [1], результаты которых очень впечатляют. Ярким примером является алгоритм, способный создавать деревья в зависимости от окружающих условий, будь то бассейн озера или реки, или возвышенности гор [4]. Часть алгоритмов, предназначенных для построения лабиринтов, опирается на построение деревьев [2], часть на случайной расстановке объектов генерации, какие-то алгоритмы могут использовать желаемые параметры и вводить так скажем соревнования между объектами генерации, как например указано в источнике №3 [3]. Все эти алгоритмы конечно не дают полностью готовое пространство, однако значительно упрощают жизнь разработчику, когда основная рутинная часть работы выполняется за пару минут [1]. В данной работе будет представлена реализация программы, генерирующая классический лабиринт, при помощи алгоритмов построения остоного дерева.

Остовное дерево – это дерево, подграф графа, с тем же числом вершин что и у исходного графа. Говоря неформально, оставное дерево получается из исходного графа удалением максимального числа рёбер, входящих в циклы, но без нарушения связности графа [7]. Целью данной работы является создание программы, которая бы генерировала лабиринт, вообще без участия разработчика.

### **Разработка программы**

Целью разработки является создание программы для генерации лабиринта из комнат и коридоров. Задача может быть декомпозирована на подзадачи, а именно, необходимо решить:

1. каким образом расставлять комнаты,
2. как связать комнаты,
3. необходимо физически связать коридорами.

Для решения поставленных задач были применены различные графовые алгоритмы. Для начала необходимо расставить комнаты, для этого был использован алгоритм Киркпатрика, для построения – триангуляции Делоне. Вкратце обозначим что, случайная расстановка комнат даёт хаотичные результаты, порой невозможно в принципе построить связные маршруты между комнатами, расставленными подобным образом, триангуляция решает данную проблему, внося структуру в расстановку комнат.

Триангуляция Делоне – это картинка, множества точек  $S$ , где для любого треугольника любые точки, не являющиеся вершинами из  $S$ , лежат вне окружности, описанной вокруг треугольника [6]. Пример триангуляции представлен на рисунке 1.



Рис. 1. Пример триангуляции Делоне

Для построения триангуляции используется алгоритм Киркпатрика. Алгоритм работает следующим образом:

1. разбиение исходного множества точек на подмножества;
2. объединение оптимальных триангуляций:
  - a. нахождение пары точек, отрезки которых в совокупности с построенными триангуляциями образуют выпуклую фигуру;
  - b. соединение точек в отрезки, он выбирается в качестве начало обхода;
  - c. обход:
    - i. построение описанной вокруг отрезка окружности до какой-либо точки;
    - ii. соединение с точкой той части отрезка, которая не была соединена с ней;
    - iii. отрезок проверяется на пересечение с другими отрезками триангуляции;
    - iv. найденный отрезок становится новым началом обхода;
    - v. повторение обхода.

Далее необходимо построить минимальное остовное дерево. Для его построения используется алгоритм Каракалла. Алгоритм работает следующим образом: упорядочивает рёбра графа по возрастанию метрики евклидова расстояния между вершинами, и поочерёдно оставляет те рёбра, которые не образуют цикл в графе [7].

В завершении, необходимо соединить сгенерированные комнаты между собой по оставшимся путям при помощи алгоритма A\*, чтобы в итоге помещение выглядело целым на плане-схеме. Алгоритм делает следующее:

- в начале смотрит соседние точки;
- проверяет были ли эти точки посещены ранее алгоритмом;
- находит не посещенные точки;
- посещает их, занося их в список посещённых;
- в течение прохода алгоритм просчитывает евклидово расстояние до целевой точки, а также уже пройденный путь, чтобы выбрать оптимальный маршрут [8].



Рис. 11. Пример работы

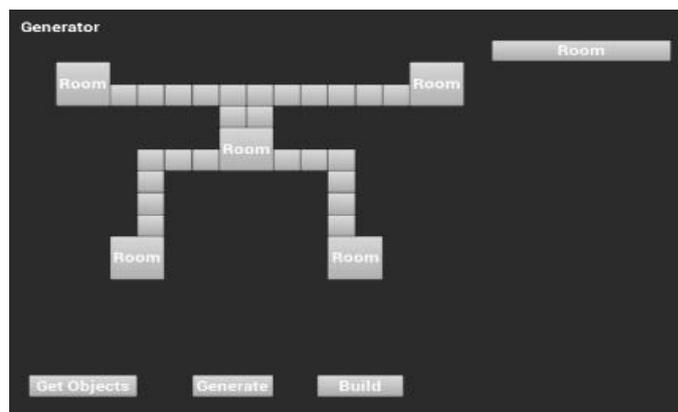


Рис. 3. План схема

### Заключение

На основе графовых алгоритмов была создана программа, которая генерирует точки расположения комнат в пространстве. Был написан модуль построения коридоров между комнатами. А также модуль визуализации построенного виртуального пространства. Разработка была проведена на языке C++. Пример на рисунке 2.

### Список использованных источников

1. Процедурная генерация уровней [Электронный ресурс]: Режим доступа: URL: <https://habr.com/ru/post/418685/> (дата обращения 11.03.2021)
2. Использование BSP-деревьев для создания игровых карт [Электронный ресурс]: Режим доступа: URL: <https://habr.com/ru/post/332832/> (дата обращения 11.03.2021)
3. Процедурная генерация планов помещений [Электронный ресурс]: Режим доступа: URL: <https://habr.com/ru/post/184818/> (дата обращения 11.03.2021)
4. Как процедурная генерация помогает создавать открытые миры [Электронный ресурс] Режим доступа: URL: <https://dtf.ru/gamedev/169117-kak-procedurnaya-generaciya-pomogaet-sozdavat-otkrytye-miry> (дата обращения 11.03.2021)
5. Скворцов А.В. Триангуляция Делоне и ее применение/ А.В. Скворцов. – Томск: Издательство томского университета, 2002, 128с.
6. Скворцов А.В. Алгоритмы построения и анализа триангуляции/ А.В. Скворцов, Н.С. Мирза. – Томск: Издательство томского университета, 2006, 168с.
7. Минимальное остовное дерево. Алгоритм Краскала [Электронный ресурс]: Режим доступа: URL: [http://e-maxx.ru/algo/mst\\_kruskal](http://e-maxx.ru/algo/mst_kruskal) (дата обращения: 19.10.2020)
8. Алгоритм A\* для новичков. [Электронный ресурс]: Режим доступа: URL: <http://masters.donntu.org/2013/fknt/buga/library/AStar.htm> (дата обращения: 19.10.2020)