В. И. Наумик

ГЕНЕРАЦИЯ УРОВНЕЙ

Процедурная генерация контента является одним из наиболее актуальных и активно развивающихся направлений исследований в сфере мультимедиа, в частности в индустрии видеоигр. Под процедурной генерацией контента (ПГК) понимают автоматическое и полуавтоматическое создание и динамическое изменение различных составляющих частей игр, в том числе игровых объектов и уровней, двумерной и трехмерной графики, эффектов, звуков, музыки, персонажей, сюжетов и др.

Введение

Несмотря на высокую реиграбельность уровней, созданных с помощью ПГК, зачастую они становятся скучными со временем, т.к. контент генерируется с помощью случайных величин. Поэтому в своей магистерской работе я рассматриваю генерацию контента, которая будет учитывать действия игрока. Игрок должен постоянно испытывать чередующиеся простоту и сложность во время игры, иначе он либо будет скучать, либо слишком уставать.

Сам модуль будет состоять из двух частей: генератор уровней и генератор NPC (противников). Генератор уровней будет выстраивать подземелье с комнатами и коридорами, а генератор NPC в последствии будет располагать противников по созданным комнатам так, чтобы обеспечить игроку возможность как отдыхать, так и испытывать сложности при прохождении.

I. Основной алгоритм

Поскольку мы имеем дело с абстрактным понятием «сложность», то его необходимо каким-то образом преобразовать в число. Наилучший вариант – число от 0 до 1. Происходить это будет на этапе создания подземелья. Когда будет построен граф, узлами которого будут главные комнаты, каждому узлу будет поставлено в соответствие число сложности, на основе которых на месте узлов будут сгенерированы комнаты с противниками, а затем по графу будут построены коридоры.

Основной алгоритм построения графа следующий:

- случайным образом разместить необходимое число главных комнат;
- использовать сгенерированный набор комнат для создания связного графа с помощью триангуляции Делоне;
- полученный граф использовать для создания минимального остовного дерева. В качестве весов использовать расстояния между узлами [1].

Теперь, когда у нас есть граф для будущего подземелья, можно присвоить вершинам оценку сложности:

- отобрать все вершины, которые имеют лишь одно связующее ребро;
- выбрать случайным образом одну из отобранных вершин в качестве стартовой (вход) и присвоить ей оценку 0;
- найти поочерёдно все возможные пути от стартовой вершины к остальным отобранным вершинам;
- для вершин каждого возможного пути рассчитать число сложности по следующей формуле:

$$C_i = C_{i-1} + \frac{1}{d}, i = 1, .., N \qquad (1)$$

  где i – оценка сложности i-ой вершины в пути, N – число вершин, d – константа, означающая число вершин, через которое оценка пойдёт обратно на убыль (задаётся вручную): если $i \ mod \ d = 0$, то $d = -d$.
- для каждой вершины найти среднее арифметическое её оценок сложности по всем возможным путям;
- случайным образом выбрать вершину для выхода из подземелья и присвоить ей оценку 0.

II. Выводы

Таким образом значения сложности не просто случайно назначаются каждой комнате, а высчитываются на основании вероятности зайти в ту или иную комнату, что делает эти значения достаточно объективными. Данную схему можно реализовать на различных игровых движках и языках программирования.

Список литературы

1. Procedural Dungeon Generation Algorithm [Электронный ресурс]. – Режим доступа: https://www.gamasutra.com.

Наумик Владислав Игоревич, магистрант кафедры информационных технологий автоматизированнных систем Белорусского государственного университета информатики и радиоэлектроники, vladislav.naumik@mail.ru.

Научный руководитель: Сердюков Роман Евгениевич, доцент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, serdyukov@gmail.com.