Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК_____

Наумик
Владислав, Игоревич

Модуль адаптивной генерации контента в компьютерных играх

АВТОРЕФЕРАТ
на соискание степени магистра технических наук
по специальности 1-40 80 02 «Системный анализ, управление и обработка информации»

_____

Научный руководитель
Сердюков Роман Евгеньевич
кандидат техн. наук, доцент

_____

Минск 2019

ВВЕДЕНИЕ

Современная игровая индустрия в большинстве своём перешла на «сервисную» бизнес-модель. В рамках этой модели пользователи при покупке видеоигры получают не законченный продукт, а некоторую его часть, к которой в дальнейшем в течение долгого времени будет поставляться дополнительный контент (*DLC*), что заставляет пользователя вновь возвращаться к ранее приобретённой игре и тратить на неё дополнительные время и деньги. Однако в этом случае разработчики «игр-сервисов» активно следят за развитием своего продукта и общаются со своей целевой аудиторией, что даём им возможность реализовывать в дополнительном контенте то, чего хотят игроки, тем самым решая вопрос персонализации.

В силу сложившихся обстоятельств вытекает необходимость создавать большое количество контента, что связано с дополнительными затратами времени и денег на его создание. Поэтому процедурная генерация контента становится всё более актуальным направлением исследований в сфере мультимедиа и, 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. Использование ПГК позволяет не только значительно понизить стоимость создания контента, но и решает проблему персонализации, приобретающую большую значимость в связи с увеличением количества потенциальных игроков.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель исследования

Целью настоящей работы является разработка метода адаптивной генерации контента в компьютерных играх на основе существующих методов процедурной генерации.

Задачи исследования

– разработать модель игрового контента, отражающую его структуру и содержание;

– разработать алгоритмы адаптивной генерации на основе разработанной модели контента;

– реализовать разработанные алгоритмов адаптивной генерации в какой-либо доступной среде.

Личный вклад соискателя

Соискателем выполнены все изложенные в работе разработки и исследования. Постановка задач и обсуждение результатов проводились совместно с научным руководителем и сотрудниками кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники. Обработка, интерпретация данных, а также выводы сделаны автором самостоятельно.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались на 54-ой научно-технической конференции магистрантов, аспирантов и студентов (Минск 2019);

Публикации результатов диссертации

Основные результаты диссертации опубликованы в двух статьях в сборниках материалов 54-й и 55-й научных конференций аспирантов, магистрантов и студентов

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Алгоритмы генерации игровых уровней

Генерация уровня состоит из следующих этапов:

– выделение области для генерации;

– разделение выделенной области на участки, которые будут использоваться для построения комнат;

– построение графа путей, узлами которого будут ранее подготовленные участки;

– расчёт оценок сложности на основе графа путей.

В начале генерации задаётся произвольный размер области в клеточном формате. Для разделения исходной области на участки был выбран алгоритм на основе *BSP*-деревьев, поскольку он позволяет получить структуру участков, которую можно использовать для генерации различных видов ландшафтов.

После окончания разделения исходной области готовые участки используются для построения комнат, которые в свою очередь будут использоваться для генерации графа путей.

Генерация графа путей состоит из двух этапов:

– 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.

После получения минимального остовного дерева, можно приступать к формированию оценок сложности для комнат. Алгоритм формирования оценок сложности следующий:

Шаг 1. Отобрать все вершины, которые имеют лишь одно связующее ребро.

Шаг 2. Выбрать случайным образом одну из отобранных вершин в качестве стартовой (вход) и присвоить ей оценку 0.

Шаг 3. Найти поочерёдно все возможные пути от стартовой вершины к остальным отобранным вершинам.

Шаг 4. Для вершин каждого возможного пути рассчитать число сложности по следующей формуле:

$$C_i = C_{i-1} + \frac{1}{d}, i = 1, .., N \quad (3.1)$$

где C_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): $i \bmod d = 0$, то $d = -d$

Шаг 5. Для каждой вершины найти среднее арифметическое её оценок сложности по всем возможным путям.

Шаг 6. Случайным образом выбрать вершину для выхода из подземелья и присвоить ей оценку 0.

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. В дальнейшем данные числа будут использоваться для генерации самой комнаты с противниками [1–А].

Алгоритмы генерации неигровых персонажей

На предыдущем этапе генерации уровня были получены оценки сложности, которые будут использоваться в качестве входного параметра для алгоритма подбора противников.

Итак, основные вопросы, которые необходимо решить алгоритму подбора противников, это какие виды и сколько противников разместить в той или иной комнате. Ключевыми параметрами в этом случае являются оценка сложности и минимальное и максимальное количество противников в группе. Данную задачу можно свести к задаче линейного целочисленного программирования:

$$\begin{gathered} E = \sum_{i=1}^{n} d_i x_i \to D, \\ \sum_{i=1}^{m} x_i \le b_{max}, \\ \sum_{i=1}^{m} x_i \ge b_{min}, \end{gathered} \tag{3.2}$$

$$x_i - \text{целое}, \qquad i = 1, .., n$$

где x_i – количество противников i-го вида;

d_i – единичная сложность противников i-го вида;

b_{min} и b_{max} – минимальное и максимальное число противников на данном уровне, задаваемые в качестве параметров генерации. Далее рассмотрим нахождение единичной сложности видов противников.

Единичная сложность вида противника отображает сложность одной единицы данного вида противника по отношению к игроку. Данную величину, основанную на сравнении характеристик вида противника и персонажа игрока, необходимо рассчитывать в ключевые моменты игры, такие как начало нового уровня или получение новой экипировки. Расчёт данного показателя основан на сравнении времени, необходимого игроку, чтобы уничтожить противника (tp), с временем, необходимым противнику, чтобы уничтожить игрока (te):

$$d = \frac{t_e}{t_p} \tag{3.3}$$

Разумеется, такой подход актуален только для тех игр, в которых присутствуют шкалы здоровья игрока и противников (может быть неявно), но у большинства игр так или иначе они есть. Дальнейшие расчёты выполняются по следующим формулам:

$$t_p = \frac{Health_e}{V_p * Damage_p} \tag{3.4}$$

$$t_e = \frac{Health_p + \frac{R_{hit} * Health_e}{Damage_p} + R_{passive} * t_p}{V_e * Damage_e} \tag{3.5}$$

где $Health_e$ ($Health_p$), $Damage_e$ ($Damage_p$), V_e (V_p) – величина здоровья, величина чистого урона и скорость атаки противника (игрока) соответственно, а $R_{passive}$ и R_{hit} – пассивная регенерация (количество восполняемого здоровья) и регенерация от атак здоровья игрока.

На основании выведенных формул можно для каждого вида доступных противников в ключевые моменты рассчитывать их единичные сложности. Система (1) решается уже известными методами решения задач целочисленного линейного программирования, к примеру, методом ветвей и границ [2–А].

ЗАКЛЮЧЕНИЕ

В ходе работы над магистерской диссертацией были изучены методы генерации игрового контента и на их основе был разработан модуль адаптивной генерации игрового контента. Программный модуль представляет собой реализацию методов адаптивной генерации локаций и противников в виде компонентов игрового движка *Unity 2019*. В рамках данного модуля были выполнены все требования, выдвигаемые методам генерации контента, а именно:

– вариативность, обеспечивающая генерацию уровней различного типа;

– гибкость, предоставляющая широкий спектр настроек алгоритмов;

– отслеживание ошибок генерации (тупиков, возможности выйти за пределы уровня и др.);

– баланс между ресурсоёмкостью и реалистичностью;

– возможность интеграции контента, созданного вручную;

– высокая степень реиграбельности.

На основании полученных результатов была проверена работоспособность разработанных методов и модуля в целом.

Все задачи, поставленные в рамках исследования, были выполнены.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1–А. Наумик, В. И. Генерация уровней / В. И. Наумик // Информационные технологии и управление: материалы 54-й научной конференции аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. – Минск: БГУИР, 2018. – С. 46.

2–А. Наумик, В. И. Адаптивная генерация противников в играх жанра Action RPG / В. И. Наумик // Информационные технологии и управление: материалы 55-й научной конференции аспирантов, магистрантов и студентов, Минск, 22 – 26 апреля 2019 г. – Минск: БГУИР, 2019. – С. 61.