

# АДАПТИВНАЯ ГЕНЕРАЦИЯ ПРОТИВНИКОВ В ИГРАХ ЖАНРА ACTION RPG

Процедурная генерация контента (ПГК) уже давно нашла своё место во многих игровых жанрах, однако по-настоящему раскрыть себя она смогла именно в Action RPG. В данном жанре ПГК используется как для построения ландшафтов игровых уровней, так и для генерации противников, о которой в дальнейшем и пойдёт речь.

## ВВЕДЕНИЕ

Генерация противников подразумевает собой подбор групп противников из имеющегося множества заготовленных шаблонов и впоследствии расположения их на разных участках игрового уровня. В зависимости от ландшафта, группы могут располагаться как близко друг к другу, так и находиться на значительном расстоянии. Это также может зависеть от уровня сложности, который нужно обеспечить игроку. В данной статье не будет делаться упор на размещении противников на игровой карте, а будет рассмотрен способ подбора противников для какой-то ключевой точки на игровом уровне с определённым уровнем сложности.

## I. ПОСТАНОВКА ЗАДАЧИ

Итак, предположим, что в некоторой игре рассматриваемого жанра сложность определяется числом от 0 до 1. Также мы имеем определённую точку на игровом уровне с некоторым числом сложности, для которого нужно подобрать противников. Обозначим данное число  $D$ . Данную задачу можно свести к задаче линейного целочисленного программирования:

$$E = \sum_{i=1}^n d_i x_i \rightarrow D, \sum_{i=1}^n x_i \in [b_{min}, b_{max}] \quad (1)$$

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.

## II. НАХОЖДЕНИЕ ЕДИНИЧНОЙ СЛОЖНОСТИ

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 ( $t_p$ ), с временем, необходимым противнику, чтобы уничтожить игрока ( $t_e$ ):

$$d = \frac{t_e}{t_p} \quad (2)$$

Расчёты выполняются по следующим формулам:

$$t_p = \frac{Health_e}{V_p * Damage_p} \quad (3)$$

$$t_e = \frac{Health_p + \frac{R_{hit} * Health_e}{Damage_p} + R_{passive} * t_p}{V_e * Damage_e} \quad (4)$$

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

## III. ВЫВОДЫ

Таким образом, разработанный метод позволяет генерировать противников, подстраивая сложность игрового процесса под пользователя. Система (1) решается уже известными методами решения задач целочисленного линейного программирования.

## Список литературы

1. Procedural Dungeon Generation Algorithm [Электронный ресурс]. – Режим доступа: <https://www.gamasutra.com>.
2. Системный анализ и исследование операций: учебно-метод. пособие / Б. В. Никульшин, Т. В. Тиханович – Минск: БГУИР, 2003. – 138 с.

*Наумик Владислав Игоревич*, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, [vladislav.naumik@mail.ru](mailto:vladislav.naumik@mail.ru).

*Научный руководитель: Сердюков Роман Евгеньевич*, доцент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, [serdyukov@gmail.com](mailto:serdyukov@gmail.com).