**1 Вариант использования паттерна стратегия:**

Создание различных поведений для NPC. К примеру, в мирное время NPC может торговать, играть музыку или раздавать квесты. А при наступлении какого-нибудь ивента, поведение NPC меняется на враждебное, кто-то бьёт в рукопашную, кто-то используем магию или использует оружие (при условии, что уже было оружие у NPC). Или же NPC сначала получает поведение поиска оружия, а после нахождения, начинает искать цель и атаковать.

**2 Вариант использования паттерна стратегия:**

Создание различных видов алгоритмов для генерации ландшафта. Например:

1. Алгоритм с использованием процедурного шума
2. Алгоритм с использование заготовленных карт высот
3. Алгоритм на основе рандомных числе в качестве высот

К примеру, в игре есть 3 сцены. Первая генерируется на основе первого алгоритма для того, чтобы создавать открытый мир. Второй алгоритм используется для сюжетного прохождения как это задумал Level дизайнер. А третий алгоритм использовать для мест с повышенной ценностью предметов, так как из-за рандома местность может быть труднопроходимой.

**1 Вариант использования паттерна шаблонный метод:**

Создание базового класса Enemy с абстрактным методом OnAttack, а уже наследники сами реализовывают какие действия должны произойти. Например:

* Вампир медленно выкачивает жизни
* Оборотень бьёт слабо, но оставляет порезы, которые как дебафф висят несколько секунду и тратят жизни игрока.
* Маг просто выпускает магические снаряды. Также можно сделать базовый класс для магического снаряда и в наследниках по-разному реализовывать эффект магического снаряда.

**2 Вариант использования паттерна шаблонный метод:**

Различные виды реализации сохранений. Например, есть базовый класс SaveSystem. От него наследуются такие классы как:

* PositionSaveSystem – для сохранение позиции
* ImageSaveSystem – сохранение скриншотов или обложек точек сохранения как в
* InventorySaveSystem – сохранение инвентаря
* DialogueSaveSystem – сохранение состояний диалогов с NPC