

Задание 2: проектирование Roguelike

Жилкин Феодор

Смирнов Александр

26 января 2021 г.

Схема работы

Начало игры

Игра начинается с генерации игрового мира (т.е. карты). Пользователь может выбрать загрузку карты из файла или же рандомную генерацию карты.

Персонажи

Мы играем за Героя (который наследуется от родительского класса Персонаж). Также в игре представлены разного рода mobs, которые также являются персонажами. Всех персонажей объединяют следующие **особенности** (поля, свойства):

- **HP** – уровень здоровья;
- **Damage** – количество наносимого урона;
- **Resist** – количество защиты;
- **Equipment** – экипировка (список из предметов, которые находятся на персонаже);
- **Level** – текущий уровень персонажа.

и следующие **действия** (поведение):

- **Attack** – возможность атаковать другого персонажа;
- **Move** – возможность перемещаться по карте.

Свойства

Свойство **Damage** персонажа рассчитывается по 3-м параметрам:

- количество очков атаки оружия;
- количество очков атаки боеприпасов (для оружия дальнего боя);
- количество очков атаки самого персонажа.

Свойство **Resist** персонажа рассчитывается по 2-м параметрам:

- количество очков защиты всей брони на персонаже (Armor);
- количество очков защиты самого персонажа.

Герой

Герой (Hero), за которого мы играем также имеет **инвентарь** (Inventory) – список вещей, которые герой носит вместе с собой.

Мобы

Мобы (Mob) являются теми же персонажами, что и наш, герой, но они выступают в качестве главных антагонистов в игре (именно против них мы и будем отважно сражаться в темных залах подземелья, не боясь потерять свою честь или даже жизнь...).

Помимо этого у мобов нет инвентаря. У мобов есть 3 состояния и 3 поведения описывающие эти состояния:

- **Агрессивный режим** (Rage Mode) и агрессивная стратегия поведения;
- **Пугливый режим** (Coward Mode) и пугливая (отступающая) стратегия поведения;
- **Пассивный режим** (Passive Mode) и пассивная стратегия поведения.

В зависимости от количества здоровья мобы могут переходить в разные режимы и преследовать соответствующую модель поведения (стратегию). Например, если у изначально агрессивно настроенного моба малое количество здоровья, то он переходит в трусливый режим и бросается в бегство. Снова в нападение жалкий трус пойдет только тогда, когда уровень его здоровья повысится.

За такую смену состояния (режима) и поведения (стратегии) отвечают паттерны «Состояние» и «Стратегия» соответственно.

Мобы могут быть двух типов:

- мобы ближнего боя с оружием ближнего боя;
- мобы дальнего боя с оружием дальнего боя.

За создание конкретного моба отвечает паттерн «Абстрактная фабрика». Абстрактная фабрика MobAbstractFactory является родителем двух дочерних фабрик:

- фабрика создания моба ближнего боя;
- фабрика создания моба дальнего боя.

Например: игра приняла решение создать 2 моба ближнего боя и 3 моба дальнего боя, тогда фабрика мобов дальнего боя исполнит метод CreateMob() 3 раза (CreateMob() случайно исполняет один из методов CreateBowMob(), CreateCrossbowMob() и т.д., каждый из этих методов создает экземпляр класса Mob и “дает ему в руки” соответствующее оружие). Аналогично работает фабрика мобов ближнего боя.

Предметы

За создание предметов на карте (которые можно в дальнейшем будет подобрать нашему Герою) отвечает паттерн «Абстрактная фабрика». ItemAbstractFactory является родителем двух фабрик по созданию предметов:

- ArmorFactory

- WeaponFactory

Каждая из этих фабрик создает случайное оружие или случайную броню, которые наш Герой может найти в сундуках, на полу в подземельях или поднять с побежденного им моба.

Броня имеет тип один из перечисления `ArmorType`, у каждого типа брони разный уровень защиты, что влияет на конечный уровень защиты персонажа.

Оружие может быть дальноточным и ближним, каждый из подвидов оружия имеет тип один из перечисления `RangedWeaponType` и `MelleWeaponType` соответственно. Также для оружия дальнего боя существует определенные боеприпасы, которые подходят только к соответствующему оружию. Боеприпасы (из перечисления `AmmoType`) также имеют определенное количество очков атаки, которое потом повлияет на конечный наносимый урон.

Оружие и броня могут быть выполнены из определенного материала (из перечисления `MaterialType`). Материал также влияет на уровень атаки или защиты конкретного предмета. Игрок управляет игрой с помощью клавиатуры, перед началом игры пользователь может задать клавишам команды. Взаимодействие с пользователем реализовано с помощью паттерна «Команда». После нажатия определенной клавиши (к которой была привязана команда), проверяется возможность исполнения этой команды (`CanExecute()`) и команда исполняется (`Execute()`).

Приключения нашего героя заканчиваются, когда у Героя кончается запас здоровья (Герой умирает – `Die()`), в противном случае игра будет длиться бесконечно, генерируя все новые миры.