

Техническая документация

Roguelike

Лямин Владимир, Ершов Владислав, Фомина Виктория

Декабрь 2021

Содержание

1	Общее об архитектуре Roguelike	2
2	Компонент Map	3
3	Компонент Game	4
4	Компонент Mobs	4
5	Компонент Artifacts	5
6	Компонент Commands	6
7	Компонент Character	7
8	Компонент CombatSystem	8
9	Компонент UserInteraction	8
10	Компонент Settings	8

1 Общее об архитектуре Roguelike

Диаграммы компонентов и классов¹ описывают архитектуру игры Roguelike. При проектировании выбор был сделан в пользу компонентного архитектурного стиля. Получилось девять компонентов (рис. 1): Map, Game, Mobs, Artifacts, Commands, Character, CombatSystem, UserInteraction и Settings.

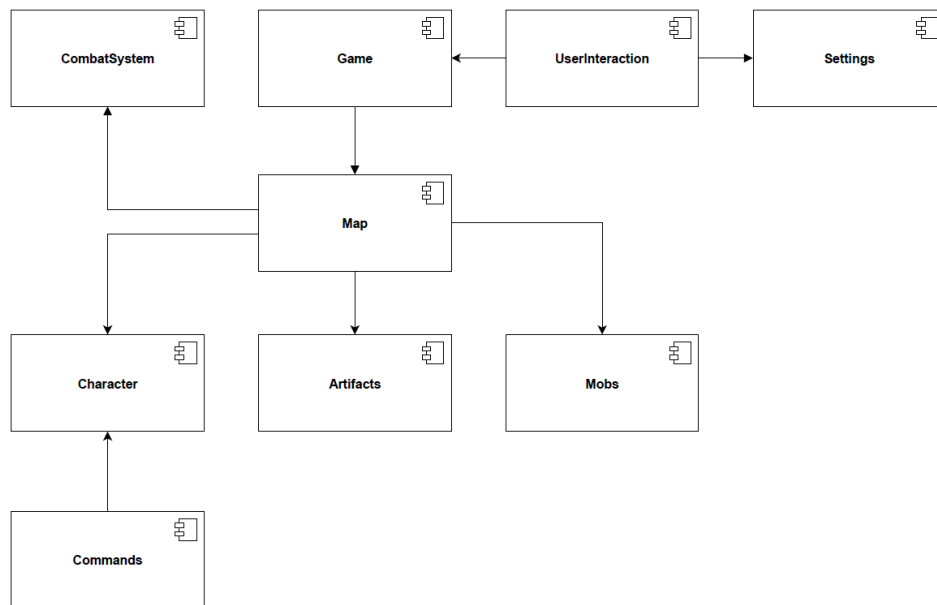


Рис. 1: Диаграмма компонентов

¹<https://github.com/victoriafomina/Software-Design/tree/main/Roguelike>

2 Компонент Map

Задача компонента Map (рис. 2) заключается в генерации карты (пещер, озер, стен и т.д. – препятствий) из файла и случайным образом, а также создании мобов, отслеживании их местоположения, создании артефактов.

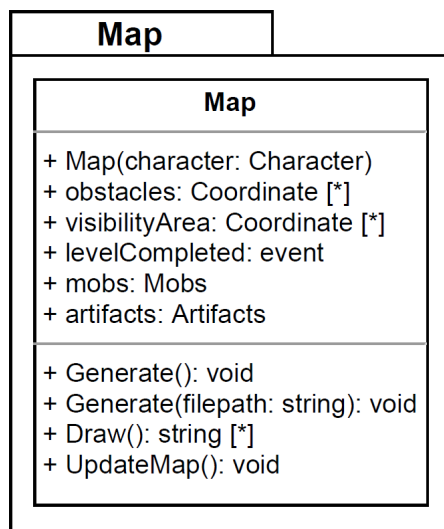


Рис. 2: Диаграмма классов компонента Map

Карта подписана на событие moveDone персонажа (Character), обновляется при его наступлении.

При наступлении события levelCompleted генерируется карта для нового уровня, персонаж переходит на новый уровень.

Для эффективного управления ресурсами используется visibilityArea – mobs и артефакты генерируются по границе области видимости персонажа.

Метод Draw возвращает текущее представление карты в виде коллекции строк.

Метод UpdateMap вызывается при обновлении уровня и при перемещении персонажа (в этом случае необходимо у класса Mobs вызвать метод OnCharacterMove).

3 Компонент Game

Задача компонента Game (рис. 3) заключается в управлении игровым процессом. Вызов метода Run запускает процесс игры. Этот метод постоянно вызывает метод Draw класса Map, что позволяет в режиме реального времени отрисовывать карту и объекты на ней.

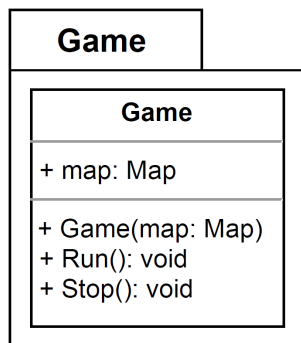


Рис. 3: Диаграмма классов компонента Game

Метод Stop ставит игру на паузу, останавливая всех мобов (метод Stop класса Mobs).

Класс Game подписан на событие персонажа died. В случае наступления этого события вызывается метод Run – игра начинается заново.

4 Компонент Mobs

Задача компонента Mobs (рис. 4) заключается в создании и управлении мобами.

Одним из основных классов компонента является Mob – класс моба. Моб обязан быть какого-то типа (IMobType), иметь состояние (IState), а также хранить текущие координаты.

Состояние моба может меняться в зависимости от изменения его здоровья. Состояние моба, его текущее положение, а также текущее положение персонажа задают вектор движения моба.

Класс MobsTypeFactory занимается созданием различных типов мобов. Так как типов мобов ограниченное количество, то в случае попытки создать уже существующий тип моба возвращается ссылка на ранее созданный объект класса соответствующего типа.

Класс Mobs занимается управлением движения мобов (например, мобы, находящиеся вне области видимости персонажа, засыпают). Также класс Mobs позволяет усыпить всех мобов (метод Stop) и разбудить мобов, находящихся в области видимости (метод Run).

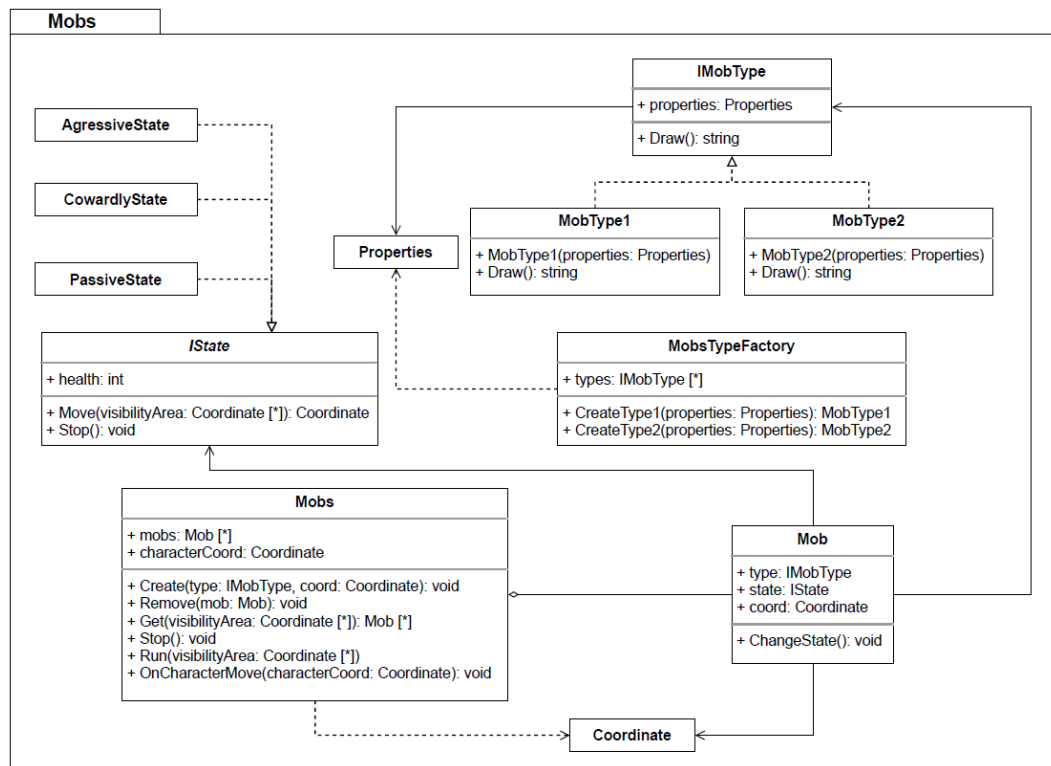


Рис. 4: Диаграмма классов компонента Mobs

5 Компонент Artifacts

Задача компонента Artifacts (рис. 5) заключается в создании артефактов.

Одним из основных классов компонента Artifacts является Artifact – класс артефакта. Артефакт обязан быть какого-то типа (IArtifactType), хранить свои координаты и показывать взят ли он персонажем или нет.

Класс ArtifactsTypeFactory занимается созданием различных типов артефактов. Так как типов артефактов ограниченное количество, то в случае попытки создать уже существующий тип артефакта возвращается ссылка на ранее созданный объект класса соответствующего типа.

Класс Artifacts занимается созданием артефактов, позволяет получать коллекцию артефактов, находящихся в области видимости персонажа, а также получать артефакт по его координате.

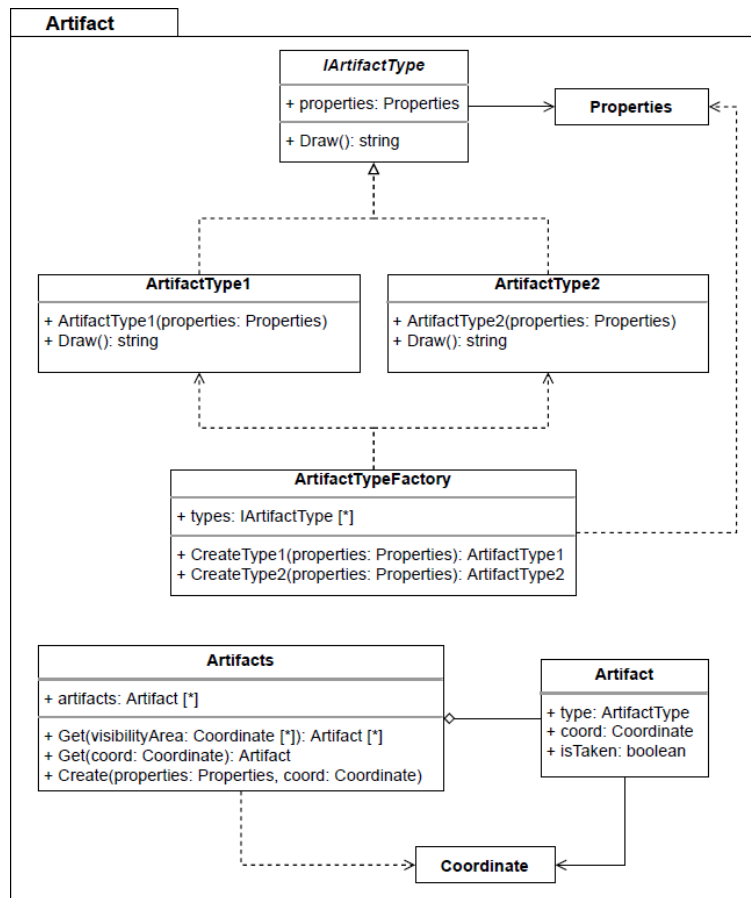


Рис. 5: Диаграмма классов компонента Artifacts

6 Компонент Commands

Задача компонента Commands (рис. 6) заключается в выполнении команд игрока таких как UpdateCoordinateCommand (команда перемещения персонажа), PutOnArtifactCommand (команда взятия артефакта), PutOffArtifactCommand (команда снятия артефакта).

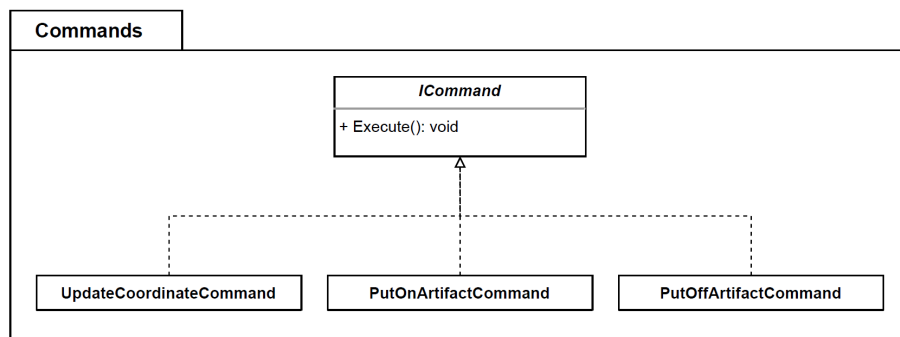


Рис. 6: Диаграмма классов компонента Commands

7 Компонент Character

Задача компонента Character (рис. 7) заключается в управлении персонажем.

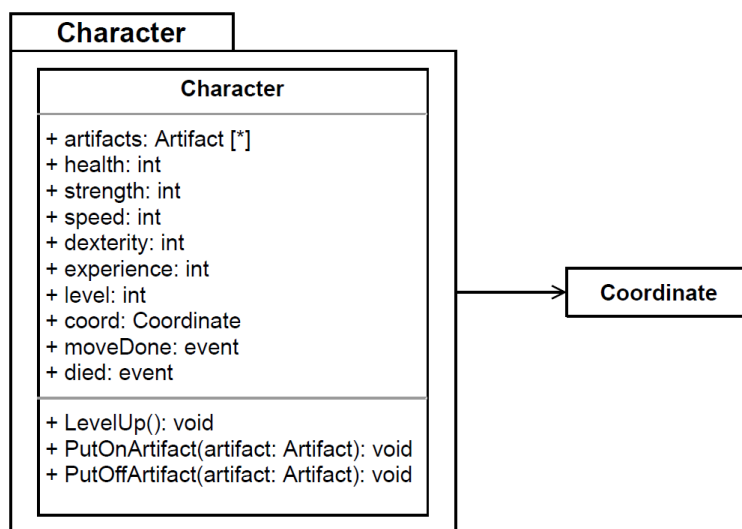


Рис. 7: Диаграмма классов компонента Character

Событие moveDone позволяет оповестить подписчиков о том, что персонаж переместился (карта перерисует игровое поле).

Персонаж подписан на событие карты levelCompleted – это позволяет ему обновлять уровень.

8 Компонент CombatSystem

Задача компонента CombatSystem (рис. 7) заключается в управлении процессом боя между персонажем и мобами, находящимися на одной игровой клетке с персонажем.

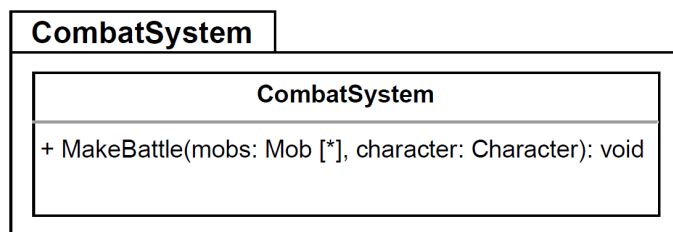


Рис. 8: Диаграмма классов компонента CombatSystem

9 Компонент UserInteraction

Задача компонента UserInteraction заключается в управлении игрой посредством использования компонента Game, а также в управлении настройками (компонент Settings).

10 Компонент Settings

Задача компонента Settings заключается в управлении настройками игры – поставить на паузу, убавить звук фоновой музыки и т.д.