Техническая документация

Roguelike

Лямин Владимир, Ершов Владислав, Фомина Виктория

Декабрь 2021

Содержание

1 Общее об архитектуре Roguelike

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

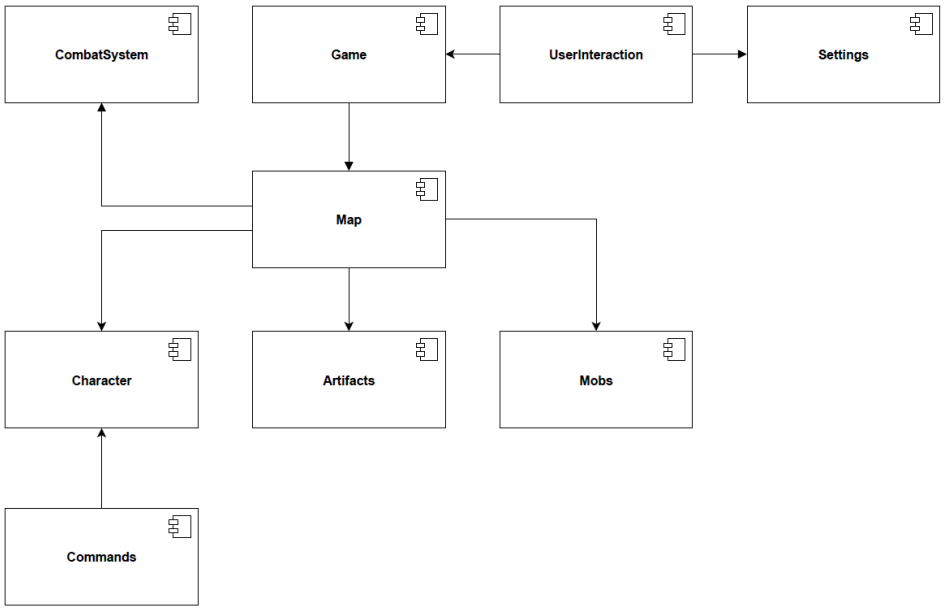

Рис. 1: Диаграмма компонентов

[1] https://github.com/victoriafomina/Software-Design/tree/main/Roguelike

2 Компонент Map

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

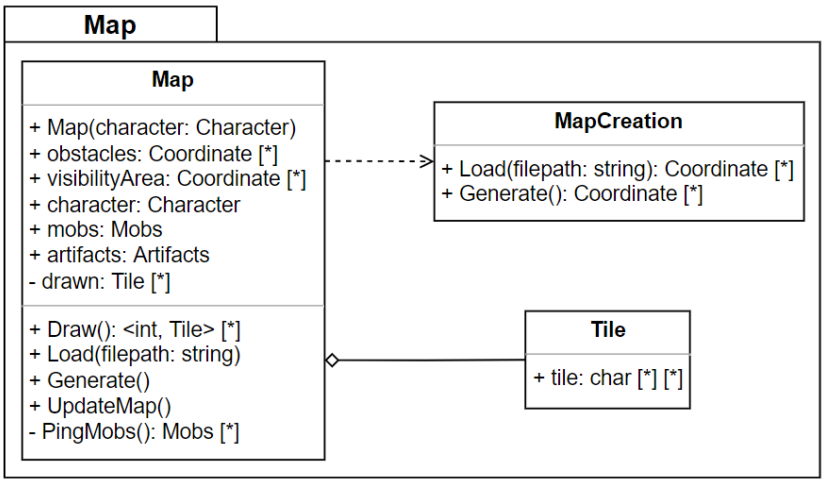

Рис. 2: Диаграмма классов компонента Map

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

При наступлении события levelCompleted (карта подписывается на него при создании) генерируется карта для нового уровня, персонаж переходит на новый уровень.

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

Метод Draw возвращает коллекцию пар, состоящую из номеров изменившихся тайлов и самих тайлов. Для отрисовки мобов метод Draw вызывает метод PingMobs, который возвращает список мобов, находящихся в области видимости персонажа.

Метод UpdateMap вызывается при обновлении уровня и при перемещении персонажа (в этом случае необходимо у класса Mobs вызвать метод OnCharacterMove) (на эти события класса Character карта подписывается при создании).

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.

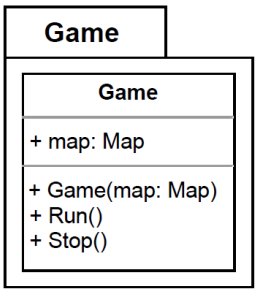

Рис. 3: Диаграмма классов компонента Game

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

Класс Game при создании подписывается 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. Mob наследуются от интерфейса IMob – классы AgressiveMob, CowardlyMob и PassiveMob реализуют его, определяя мобов с различным поведением. Каждый моб имеет свой набор свойств (Properties). Также каждый моб обладает состоянием IState – его реализуют классы NormalState и AlarmState. Состояние моба может меняться в зависимости от изменения его здоровья.

Состояние моба, его текущее положение, а также текущее положение персонажа задают вектор движения моба.

IMobsTypeFactory – абстрактная фабрика по созданию мобов различных типов. AliensTypeFactory и DemonsTypeFactory – конкретные фабрики, позволяющие создавать тематические наборы мобов.

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).

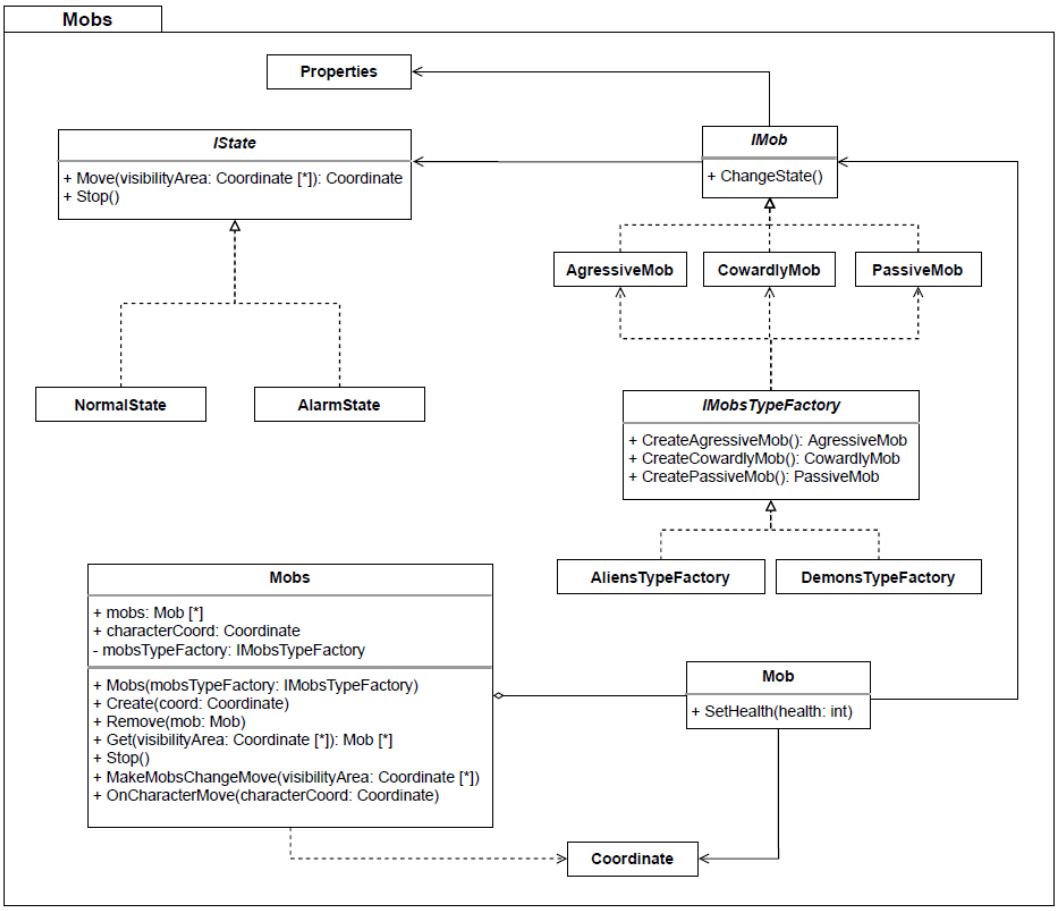

Рис. 4: Диаграмма классов компонента Mobs

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 лежит ли он на карте или взят персонажем. Артефакт имеет свойства, которые показывают как его надевание влияет на изменение свойств персонажа (например, +2 к силе).

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.

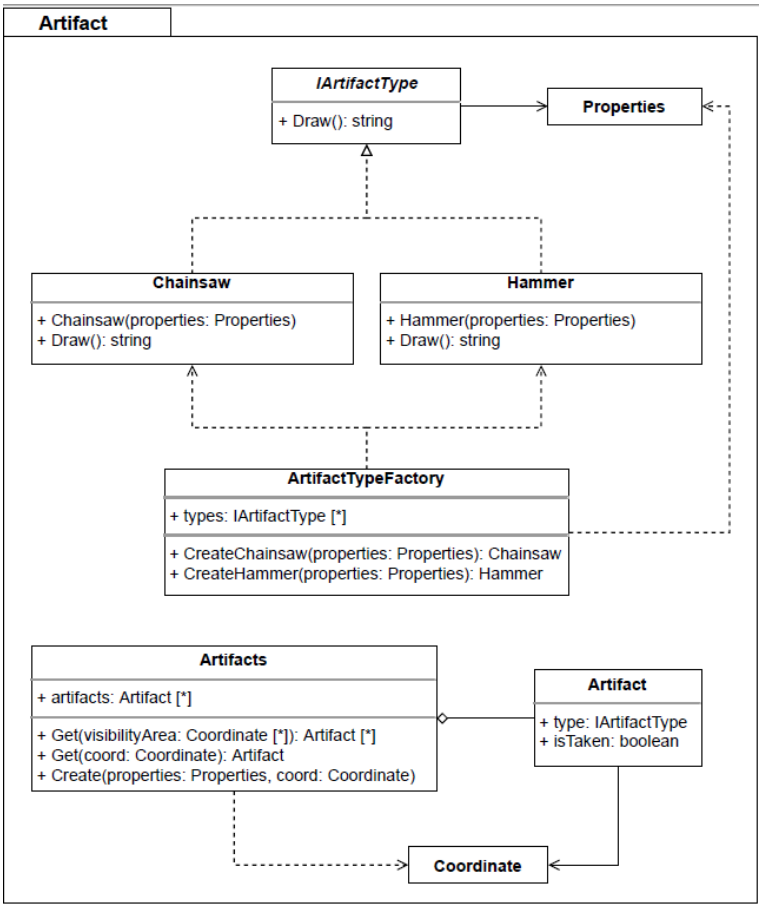

Рис. 5: Диаграмма классов компонента Artifacts

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

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), PutInInventoryArtifactCommand (команда, позволяющая положить артефакт в инвентарь) и RemoveFromInventoryArtifactCommand (команда, позволяющая удалить артефакт из инвентаря).

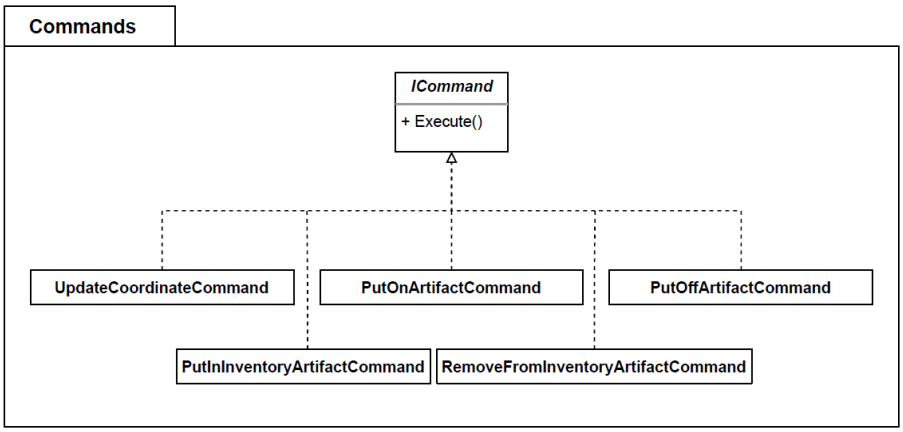

Рис. 6: Диаграмма классов компонента Commands

7 Компонент Character

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

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

Компонент Game при создании подписывает персонажа на событие карты levelCompleted – это позволяет обновлять уровень персонажу.

Персонаж знает какие артефакты надеты, а какие находятся в инвентаре.

При увеличении опыта персонажа наступает событие experienceIncreased. На это событие при создании персонажа подписывается метод LevelUp(). В случае если персонаж набрал достаточно опыта для перехода на следующий уровень, то этот переход осуществляется.

8 Компонент CombatSystem

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

Здоровье мобов и персонажа уменьшается в процессе боя. Изменение здоровья зависит от того насколько “прокачены” у соперника(-ов) такие характеристики как сила, ловкость и т.д.

За победу над мобом опыт персонажа увеличивается. То насколько он увеличится зависит от такого насколько сильным был противник(-и) (т.е. зависит от силы, ловкости моба(-ов)-соперника(-ов) и т.д.).

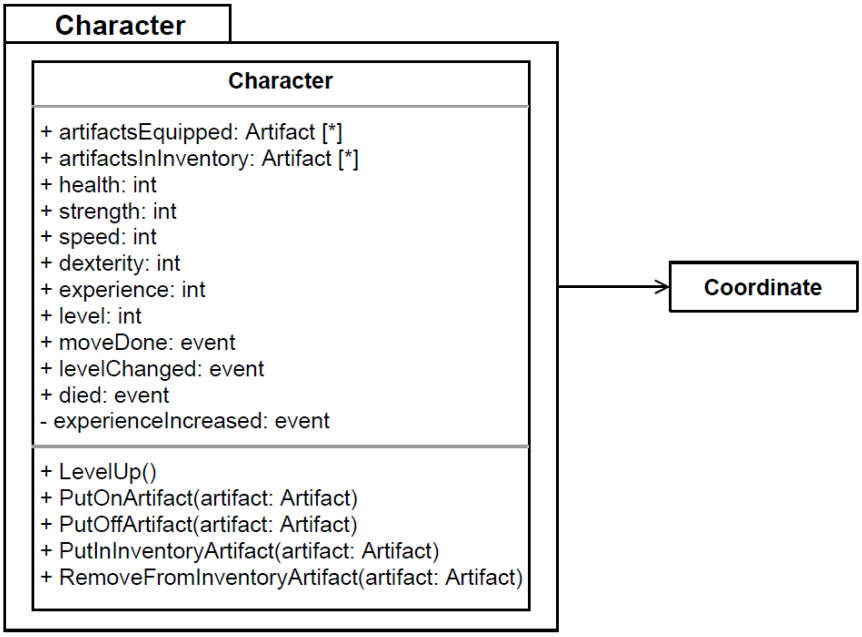

Рис. 7: Диаграмма классов компонента Character

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.

11 Компонент Utils

В этом компоненте объявлен вспомогательный класс Coordinate, который используется в нескольких компонентах.

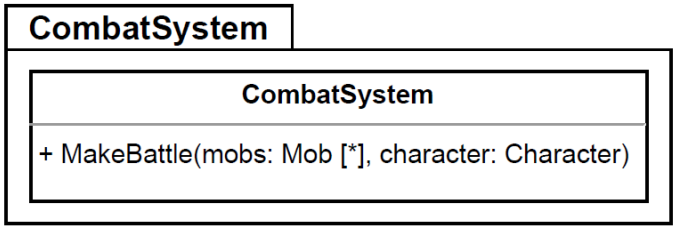

Рис. 8: Диаграмма классов компонента CombatSystem