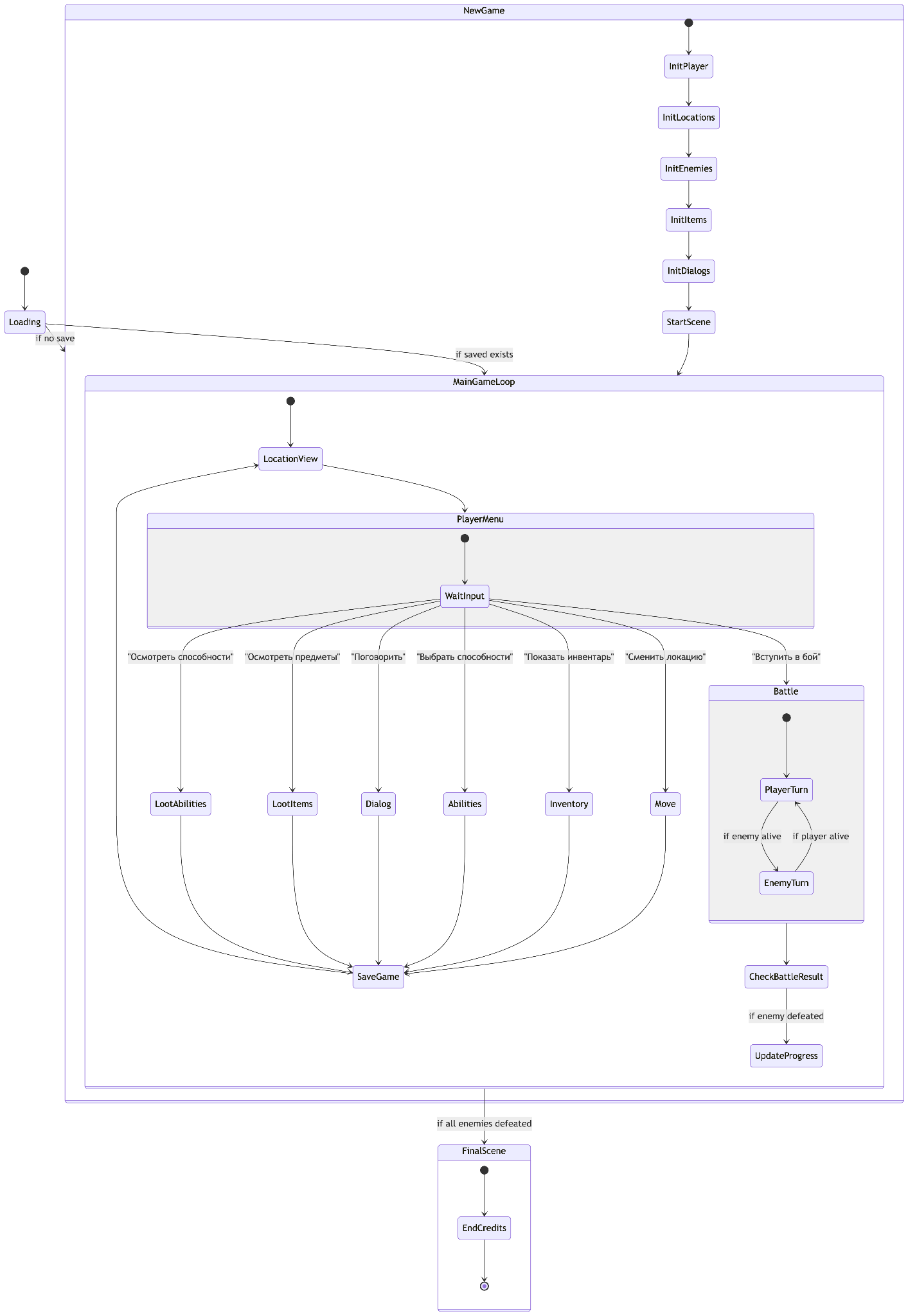
**Архитектура**

1 Игровые механики

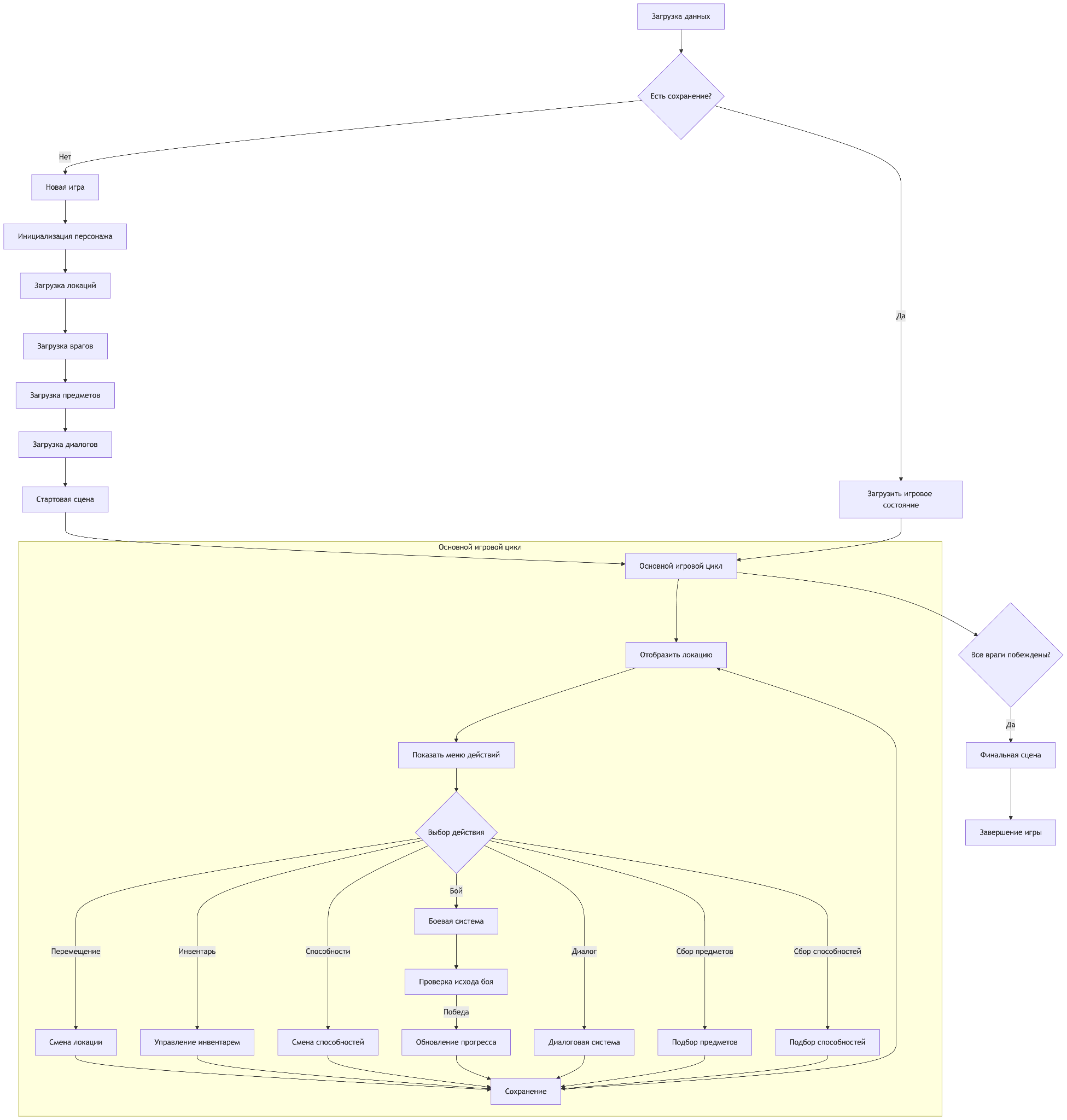
1.1 Описание игровых механик

В данной игре реализованы несколько ключевых игровых механик, которые обеспечивают взаимодействие игрока с окружающим миром, а также с NPC и противниками. Основные механики включают:

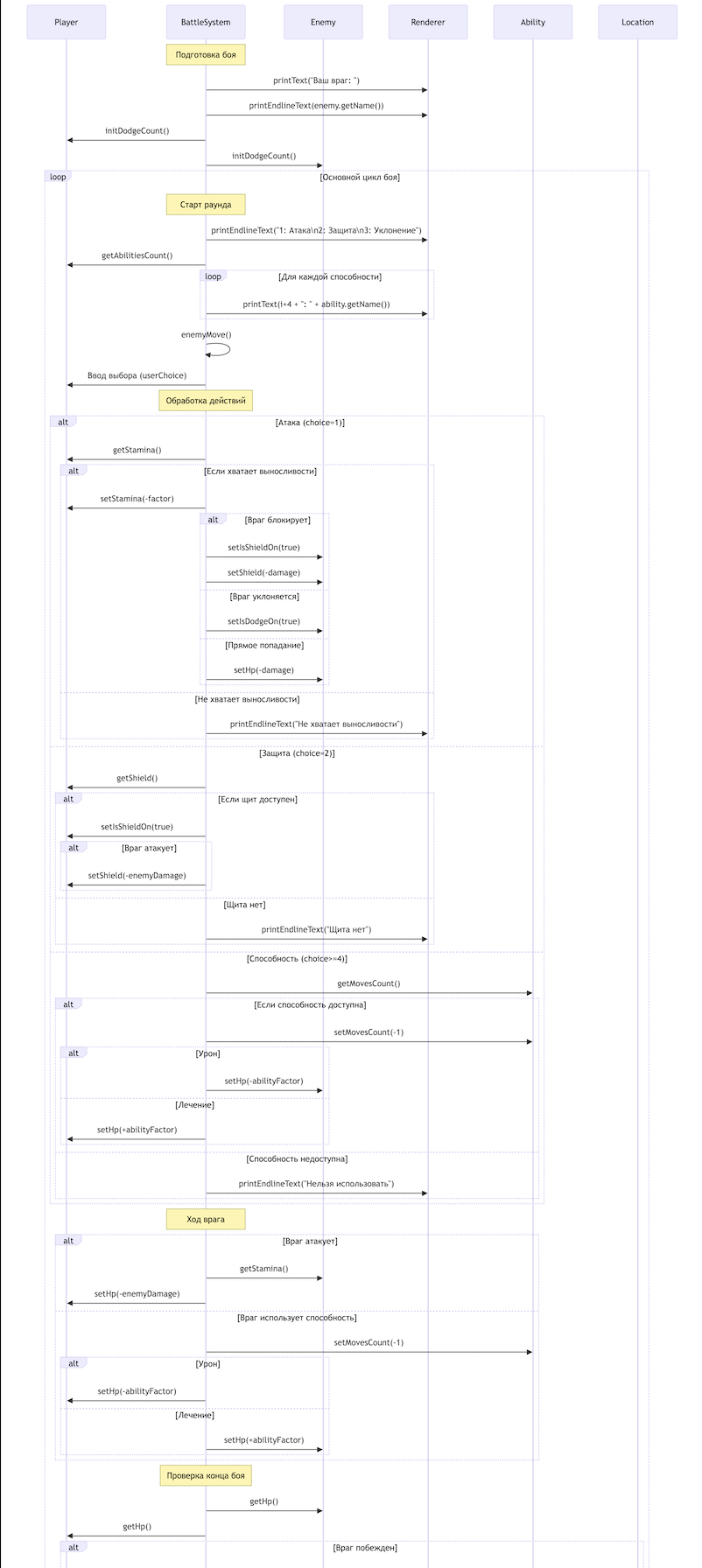
1. Боевая система: Игроки и враги могут использовать способности и предметы для нанесения урона друг другу. Боевая система учитывает параметры персонажей, такие как здоровье (HP), урон (damage), выносливость (stamina), щит (shield) и количество уклонений (dodgeCount). На параметры персонажей также влияют выбранные способности и оружие, дающие постоянные или временные эффекты.
2. Инвентарь: Игрок может собирать предметы, которые могут быть использованы в бою или для восстановления здоровья. Инвентарь позволяет добавлять, удалять и использовать предметы.
3. Способности: Персонажи могут иметь различные способности, которые могут быть использованы в бою. Способности могут оказывать различные эффекты на противников или на самого персонажа.
4. Диалоги: Игрок может взаимодействовать с NPC через диалоги, выбирая различные варианты ответов.
5. Исследование мира: Игрок может перемещаться между различными локациями, каждая из которых имеет свои описания и выборы для дальнейшего перемещения.

1.2. UML state

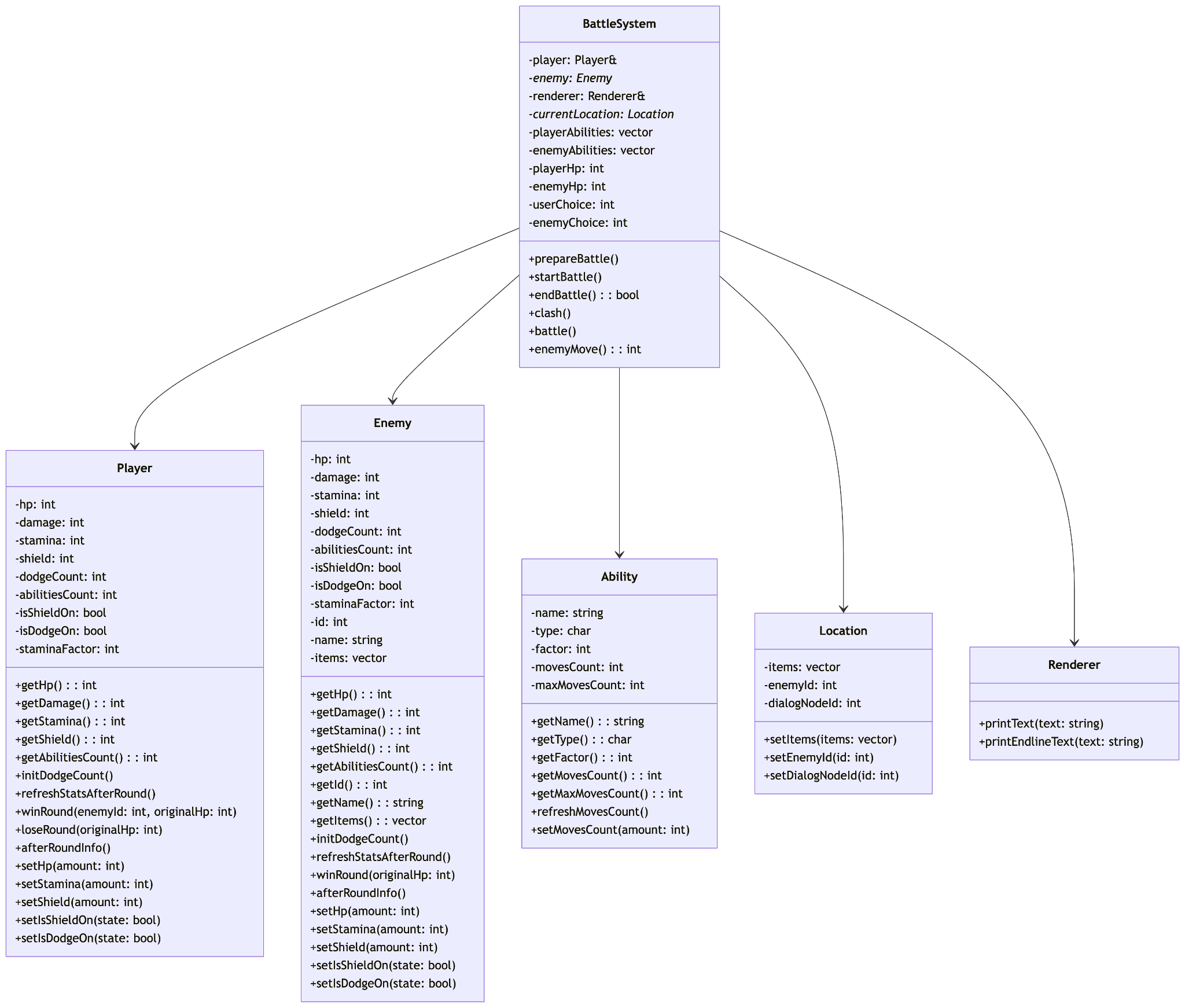
1.3. UML Activity



1.4. UML Sequence(боевка)



1.5. UML class(боевка)



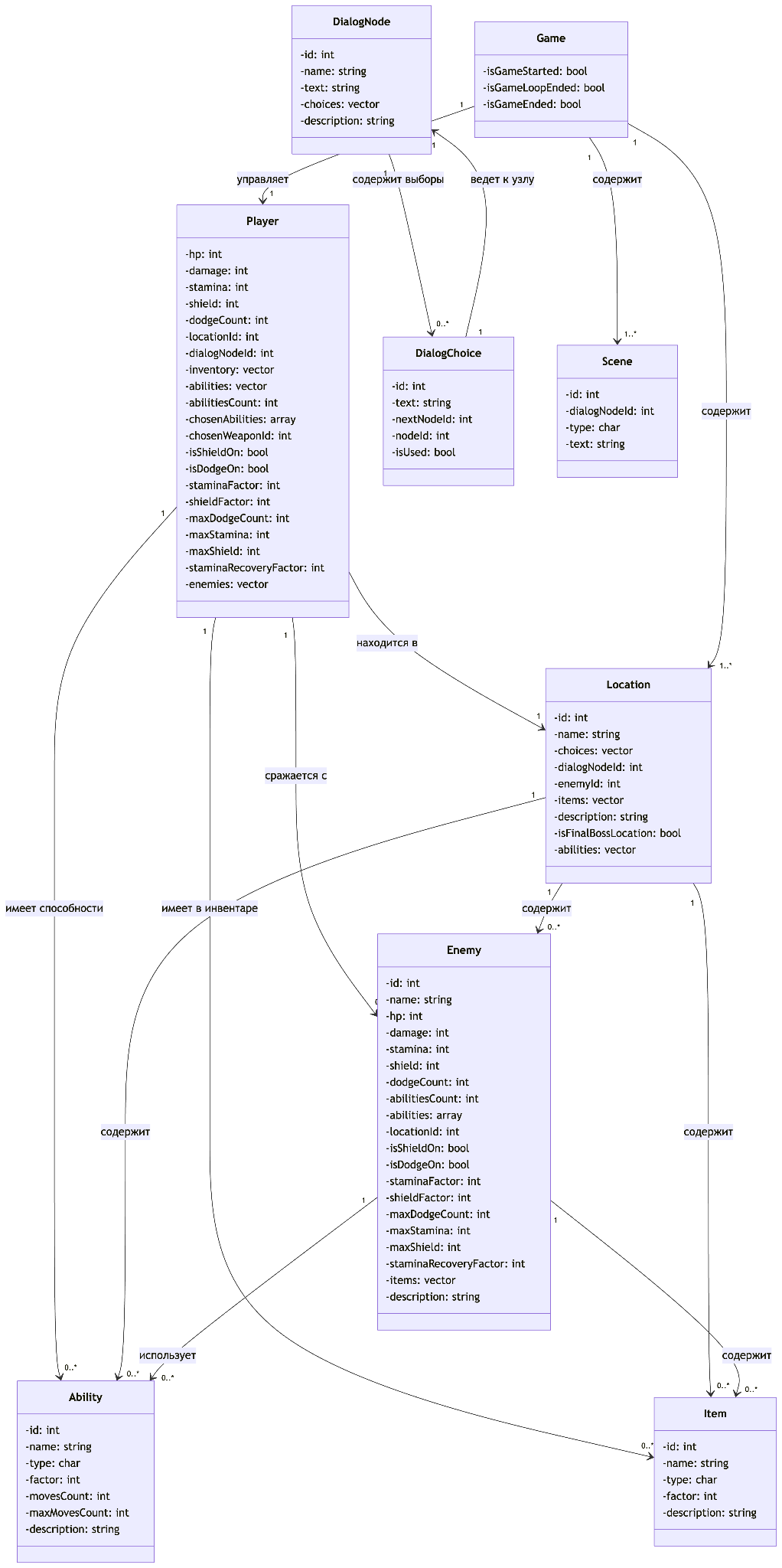
2 Общая архитектура игры

2.1. Подсистемы

В данной архитектуре игры можно выделить несколько ключевых подсистем, каждая из которых отвечает за определенные аспекты игрового процесса. Основные подсистемы включают:

1. **Подсистема рендеринга (Renderer)**:
   1. Отвечает за вывод информации на экран, включая текст, графику и интерфейс.
   2. Взаимодействует с другими подсистемами для отображения состояния игры (например, текущее местоположение игрока, диалоги и т.д.).
2. **Подсистема управления игрой (Game)**:
   1. Объединяет все другие подсистемы и управляет игровым процессом.
   2. Следит за выполнением логики игры и сохранением состояния.
   3. Отвечает за запуск игры и её завершение.
3. **Подсистема боёвки(BattleSystem)**:
   1. Управляет состоянием боя.
   2. Определяет элементы боя.
4. **Игрок (Player)**:
   1. Представляет собой персонажа игрока, который может взаимодействовать с окружающим миром.
   2. Хранит информацию о состоянии здоровья, уроне, инвентаре и способностях.
5. **Враг (Enemy)**:
   1. Представляет собой противника в игре, который может атаковать игрока.
   2. Хранит информацию о состоянии здоровья, уроне и способностях.
6. **Способности (Ability) и Предметы(Item)**:
   1. Определяют действия, которые могут выполнять игроки и враги, а также предметы, которые могут быть использованы в игре.
7. **Эффекты(Effect)**:
   1. Определяют эффекты оружия или способности.
8. **Сцены(Scene)**:
   1. Определяют заскриптованные диалоги.

2.2. UML class всей архитектуры



3 Физическая организация

[Кривда](https://docs.google.com/spreadsheets/d/1IrsFpxw5w274Vn2qOfbhGmMHgEhDYJtQBimCX7FS84w/edit?gid=0#gid=0)

/GameProjectRoot

├── main.cpp

├── /core

│ ├── Game.cpp

│

├── /player

│ └── Player.cpp

│

├── /enemy

│ └── Enemy.cpp

│

├── /combat

│ ├── Ability.cpp

│ └── Effect.cpp

│

├── /items

│ └── Item.cpp

│

├── /dialog

│ ├── DialogNode.cpp

│ └── DialogChoice.cpp

│

├── /location

│ ├── Location.cpp

│ ├── LocationChoice.cpp

│

├── /ui

│ └── Renderer.cpp