Архитектура

Общие сведения о системе

Система описывает архитектуру игры Roguelike, в которой игрок может управлять персонажем, перемещаясь по миру где персонаж игрока может находить различные предметы или мобов - обитателей игрового мира, с которыми можно сражаться.

Представлена возможность влиять на характеристики персонажа посредством экипировки. Характеристики влияют на различные аспекты игры.

Главное меню предоставляет возможность начать новую игру или продолжить старую.

Композиция

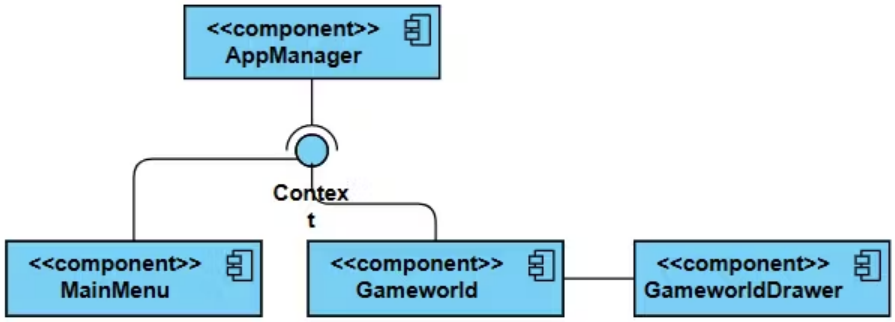


Диаграмма компонентов и её текстовое описание.

Основные компоненты:

- Менеджер
- Главное меню
- Игровой мир

Компоненты игровой мир можно разбить на модель и художника её рисующую, просто из-за большого количества задач которые они выполняют, но они очень тесно связаны.

Менеджер

Центральная компонента, управляет остальными, состоит из одного класса - manager. Транслирует в текущий контекст (главное меню или игровой мир) действия игрока и меняет состояния в зависимости от результата.

Главное меню

Главное меню предоставляет возможность начать новую игру или продолжить старую.

Простой контекст, потому рисует себя сам.

Игровой мир

Отражает весь геймплей:

- Взаимодействие с инвентарём
- Перемещение игрока, его взаимодействие с объектами карты
- Действия мобов

И изменяет это все, исходя из действий пользователя.

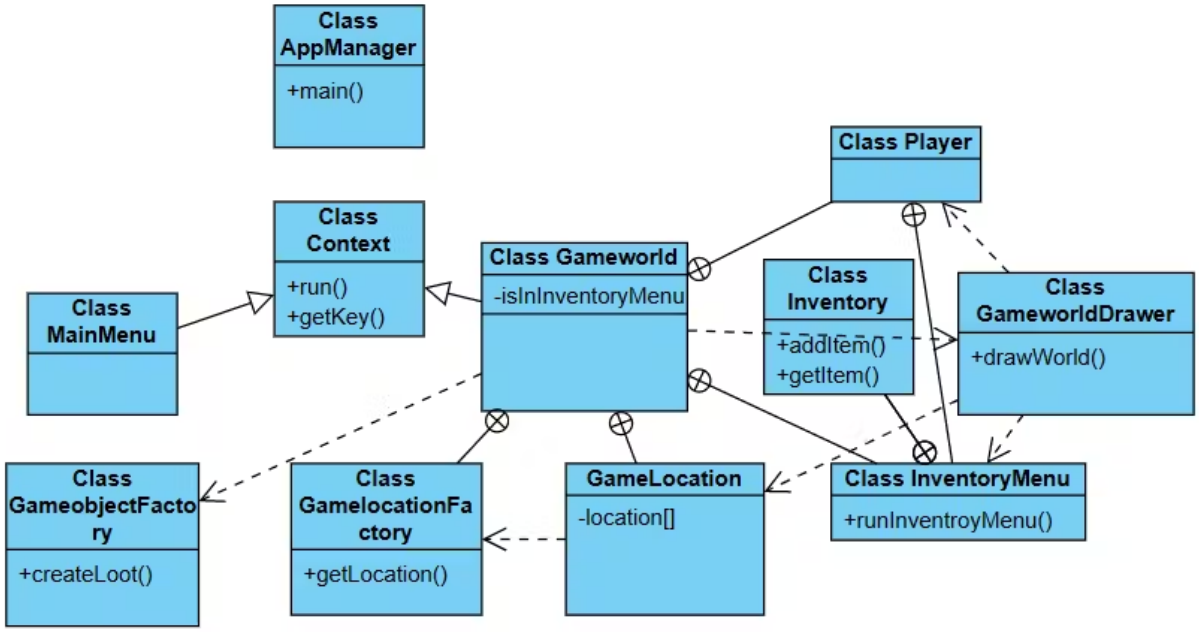
Диктует художнику когда нужно что-то отрисовать.

Художник игрового мира

Компонента отвечающая за отрисовку интерфейсов других компонент и игрового мира, исходя из состояния игры рисует меню инвентаря, карту и объекты на ней.

Логическая структура

Диаграмма классов и её текстовое описание, разбито по компонентам.



Менеджер

AppManager

- Начинает работу приложения (в нем находится метод main)

- Инициализирует контексты
- Управляет контекстами, переключает их и запускает через метод run()

Главное меню

MainMenu

Один из двух возможных контекстов (второй - Игровой Мир), принимает выбор игрока, является контекстом, то есть реализует его основной метод run.

Игровой мир

GameWorld

Хранит состояние текущей игры:

- Персонажа
- Локацию и объекты на ней
- Статус меню инвентаря

В методе run() реализует события игрового мира, считывает ввод игрока и реализует геймплей(описан в компоненте)

GameLocationFactory

Создаёт карты, добавляет на них объекты.

GameLocation

Хранит карту и объекты на ней.

GameObjectFactory

Создаёт игровые объекты (GameObject), в том числе выдаёт им уникальные идентификаторы, чтобы проще этими объектами было управлять.

InventoryMenu

Меню инвентаря, отвечает за взаимодействие игрока с инвентарём, переключается когда игрок нажимает i. В нём игрок может экипировать выбранный по номеру предмет.

Игровой мир запускает метод runInventoryMenu() когда нажата соответствующая клавиша.

GameObject

Игровые объекты, которые могут быть расположены на карте, этот класс реализуют

- Игрок (PlayerCharacter)
- Добыча (Loot)
- Моб (монстр)

PlayerCharacter

Хранит координаты где находится игрок, его характеристики и инвентарь. Взаимодействует с предметами (item).

Inventory

Хранит предметы (item) полученные игроком.

Loot

Хранит координаты где находится добыча, игрок может взаимодействовать с ней кнопкой e, забирая её в инвентарь.

Mob

Хранит координаты где находится монстр, его характеристики.

Item

Предмет, который лежит в инвентаре может быть экипирован в ‘слот’, например, righthand или lefthand.

Художник игрового мира

GameworldDrawer

Рисует игровой мир методом drawWorld(), вызывается после действия игрока.

Рисует меню инвентаря если он был вызван.

Берет информацию о карте и рисует её.

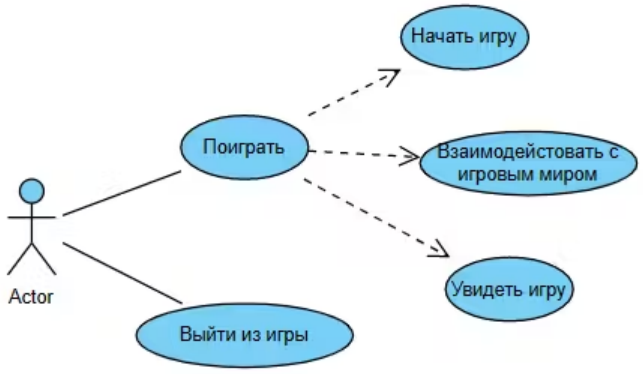
Architectural drivers

- Технические ограничения
  - Консольная графика. В качестве графических элементов доступны только ASCII символы, которые могут иметь разные цвета и принадлежать какому-то одному шрифту;
  - Управление с клавиатуры. Все взаимодействие пользователя с программой будет происходить посредством клавиатурных сочетаний, использовать мышь или другие устройства ввода нельзя.
- Качественные характеристики
  - Игра должна работать под Windows и Linux, желательно минимизировать усилия, необходимые для поддержки каждой ОС;
  - Набор функциональных требований будет пополняться, поэтому игра должна быть легко расширяемой.
- Ключевые функциональные требования
  - Игровая карта
    - Генерация случайной игровой карты
  - Игровой персонаж
    - Характеристики: здоровье, сила атаки
    - Инвентарь, содержащий слоты для экипировки или свободное место для других предметов;
    - Игровые предметы, которые можно подобрать на карте и носить в инвентаре;

- Экипировка, которую можно надеть на персонажа, что повлияет на его характеристики.

Роли и случаи использования;

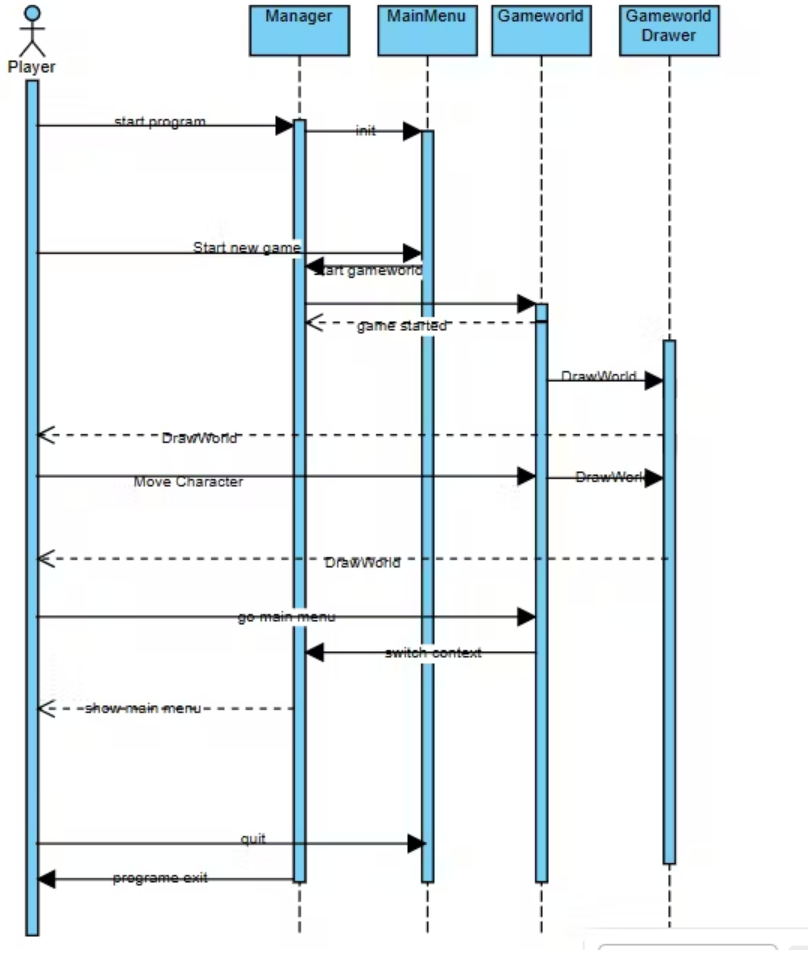
Пользователь может играть единственную роль - роль игрока.



Случаи использования:

Взаимодействия и состояния

Диаграммы последовательностей и конечных автоматов и их текстовое описание



Пользователь хочет поиграть в игру, он запускает программу, управляющий класс инициализирует основные классы.

Пользователь начинает новую игру загружается игровой мир, компонента художник рисует его.

Пользователь поиграл, например просто сделал шаг, игровой мир на это отреагировал и вызвал художника отрисовать актуальную картину.

Пользователь вышел в главное меню, менеджер поменял контекст и вышел из игры.

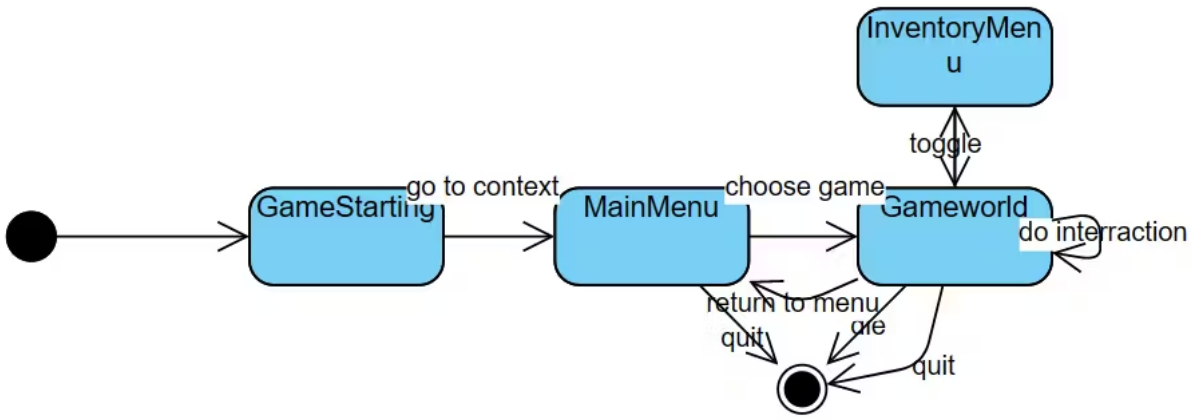


Диаграмма конечных автоматов отражает связь между ключевыми состояниями.

Запсукается игра, подставляется первый контекст - главное меню, из которого можно перейти в игровой мир, в котором можно взаимодействовать с объектами на локации или перейти в меню инвентаря. Из меню инвентаря можно вернуться к геймплею, в котором можно погибнуть, вернувшись в главное меню. Из приложения можно выйти как и из главного меню так и с игрового