Техническое задание

# Введение:

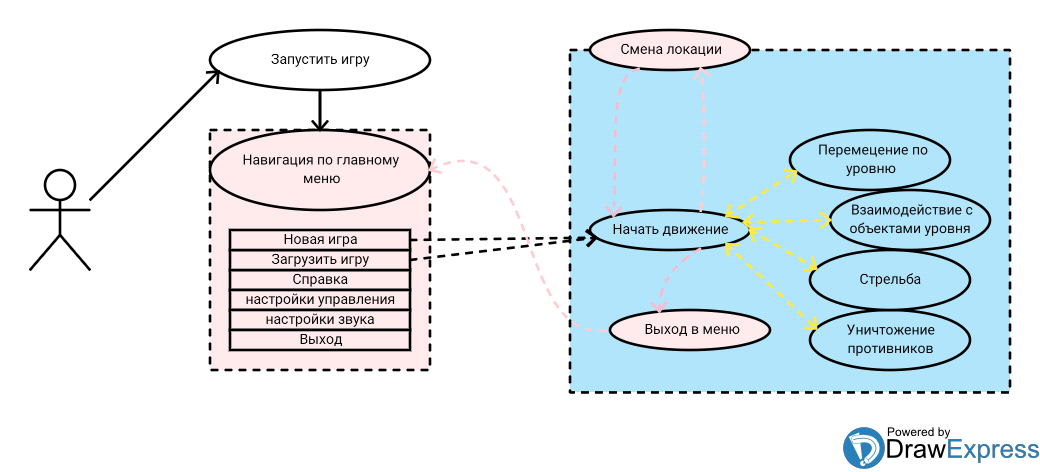
1. Рабочее название – «YetAnotherProject».
2. Жанр – Side scroller
3. Платформа - (Windows / Linux)???
4. Сроки выполнения:
   1. 2 модуль – Программа минимум - // число
   2. 3 модуль – Программа оптимум - // число

# Задачи:

1. Минимум
   1. Статичный уровень
   2. Возможность перемещаться
   3. Возможность взаимодействия с миром
   4. Возможность убийства противников
2. Оптимум
   1. Подбор оружия
   2. инвентарь
   3. AI
   4. Динамическая сцена
   5. Звук
3. Максимум
   1. Сохранения
   2. Онлайн
   3. Мобилки

# Функциональные требования:

На данной UML-диаграмме представлены основные возможности игрока:

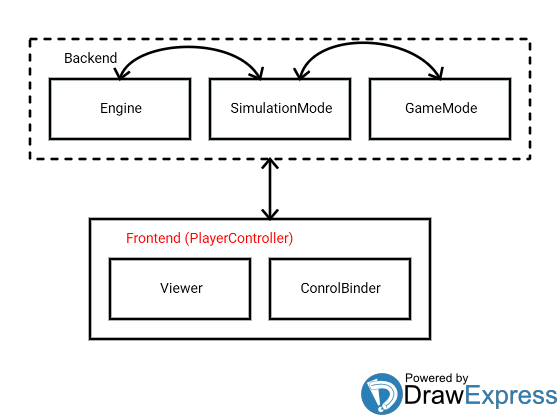


Для обеспечения которых проект должен:

1. Обеспечивать возможность запуска:
   1. Главного меню
   2. Игровой сессии
2. Обеспечивать следующие функции:
   1. Общие:
      1. Обрабатывать приходящие от пользователя команды
      2. Обеспечивать возможность смены локации (меню – уровень,   
         уровень – уровень)
      3. Обеспечить возможность кастомизации управления / звука
      4. Обеспечивать возможность выхода
      5. Сохранять внутреннее состояние при выходе
   2. Игровые:
      1. Обновлять состояние игрового поля
      2. Отображать состояние игрового поля
      3. Отображать состояние подконтрольного персонажа
      4. Отображать состояние экипировки персонажа
      5. Отображать состояние активных заданий

# Основные компоненты

На данной диаграмме представлены основные компоненты системы.



Где:

1. Engine – сущность, отвечающая за загрузку необходимых для игровой сессии ресурсов и обеспечивающая игровой мир тиками.
2. SimulationMode – Базовая точка входа, в игру, в которой описывается создание загрузочного экрана и главного меню. Отвечает за снятие текущего уровня с обработки и загрузку нового, является точкой, к которой привязывается Viewer и ControlBinder.
3. GameMode – он же уровень, сущность, описывающая базовую инициализацию игрового мира и создание глобальных виджетов. Занимается контролем игровой ситуации.
4. PlayerController – Сущность, занимающаяся взаимодействием с пользователем и привязанная к подконтрольной внутриигровой сущности Avatar.  
   1. Viewer – Сущность, привязанная к создаваемому приложением окну и занимающаяся обновлением экрана по команде из SymulationMode.
   2. ControlBinder – занимается регистрацией эвентов клавиатуры.

ЗАМЕЧАНИЕ:

Фронтенд (помимо PlayerController’а) включает в себя еще и поле типа Facade, лежащее внутри каждого инстанса ActorComponent.

# Диаграмма основных классов и компонент

