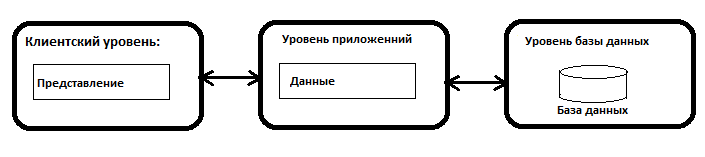
Проект Bystander

* 1. Определить тип приложения:

Bystender – насыщенное клиентское приложение(игра от первого лица) для выполнения преимущественно на ПК.

1.2.Выбрать стратегию развёртывания:

В качестве стратегии развёртывания наиболее подходящей является n-уровневое развёртывание.



1.3.Выбор технологий:

Так как проект представляет собой игру-головоломку от первого лица, то возникает необходимость в реализации большого игрового мира с разнообразными моделями. Для достижения этой цели наиболее подходящим и удобным является Unuty 3D. Для хранения сохранённых игр будут использоваться базы данных SQL, которые будут размещаться на ПК пользователя.

1.4.Атрибуты качества:

1) Игра должны потреблять небольшое количество оперативной памяти(до 1Гб), для того, чтобы сделать её доступной для пользователей с различным оборудованием.

2)Игра должна настраивать разрешение под различные мониторы без потери чёткости и пропорциональности.

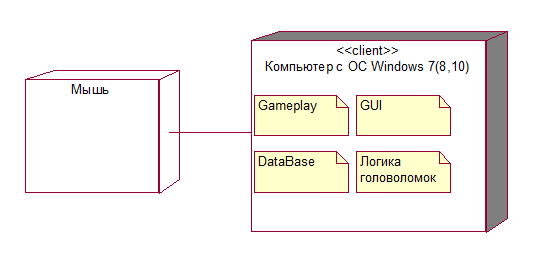
3) В игре должен присутствовать удобный графический интерфейс, для упрощения игрового процесса, чтобы любой человек смог в ней разобраться.

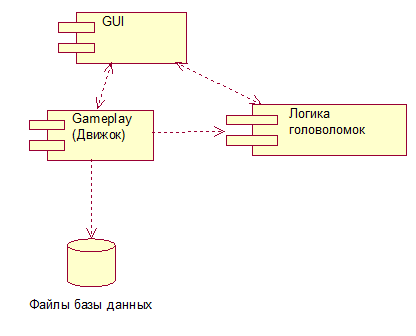
4) Надёжность: сохранённая игра должна воспроизводиться без потерь, с идеальной прогрузкой моделей.

1.5. Сквозная функциональность:

В проекте будет присутствовать механизм управления конфигурациями, для его реализации будет разработан отдельный UI доступный из меню «Настройки», при помощи которого можно будет изменять параметры конфигурации.

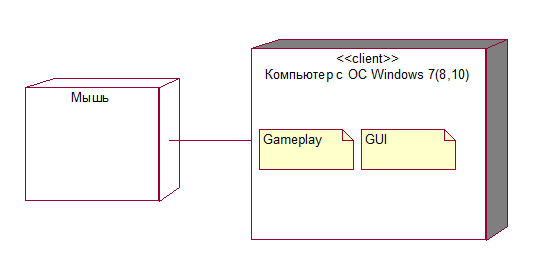
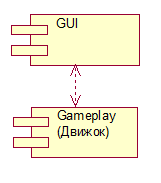
1.6.Архитектура To Be

(Диаграмма развёртывания)

 (Диаграмма компонентов)

ЧАСТЬ 2

2.2 Диаграммы развёртывания, компонентов и классов после первого спринта:



ЧАСТЬ 3(ВЫВОД):

3.2. Отличия и их причины:

Как видно невооружённым взглядом диаграмма «As is» является своего рода “поддиаграммой” «To be». Их отличия главным образом обусловлены тем, что за время первого спринта была реализована лишь часть функционала, который необходим для полноценного игрового процесса. В частности ещё не были реализованы головоломки и соответствующая им логика.

3.3. Для улучшения архитектуры необходимо провести ещё один спринт, за который будет добавлен функционал. Данный функционал сделает игру полноценной, но ещё не обеспечит полноту сюжета и достаточного разнообразия. Также, для улучшения читабельности кода, необходимо провести рефакторинг. Ах ну да, для экономии времени при написании кода, можно использовать паттерны проектирования.