

Архитектура и шаблоны проектирования

**Меня хорошо видно
&& слышно?**



Защита проекта

Тема: Реализация игрового сервера для игры “Космический бой”



Разгоняев Андрей

Вещудий разработчик
ООО «МТС Диджитал»



План защиты



Цели проекта

Что планировалось

Используемые
технологии

Что получилось

Схемы/архитектура

Выводы

Цели проекта

- | | |
|----|------------------------------------------------------------------------------------------|
| 1. | Закрепить знания в части шаблонов проектирования. |
| 2. | Спроектировать и реализовать приложение, отвечающее требованиям промышленной разработки. |
| 3. | Освоить новые технологии и инструменты. |

Что планировалось

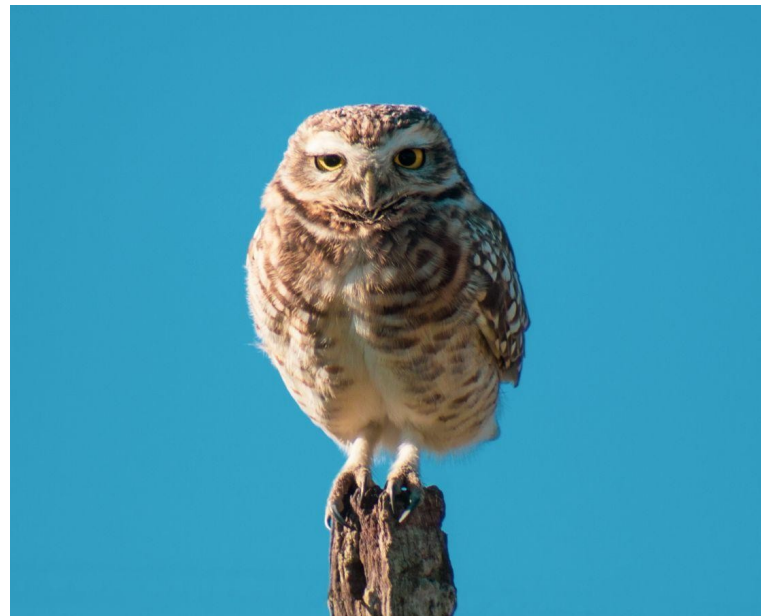
1. На практике познакомиться с миром game-dev, по возможности используя предыдущий опыт backend разработки.
2. Попытаться реализовать систему управления игровыми объектами на основе микросервисной архитектуры.
3. Особое внимание уделить использованию шаблонов проектирования, создать расширяемый и поддерживаемый продукт.

Используемые технологии

- | | |
|----|----------------------------------------------------------------------------|
| 1. | JSON Web Token как инструмент аутентификации и авторизации. |
| 2. | Swagger как средство документирования API. |
| 3. | REST API как удобный способ взаимодействия web приложений. |
| 4. | WebSocket как быстрый способ доставки клиенту часто изменяемой информации. |

Что получилось

1. Сервис аутентификации и авторизации.
2. Сервис управления игровыми объектами.
3. Простое клиентское приложение для отображения игрового поля.



Сервис аутентификации и авторизации

1. Принимает запросы на создание игры.
2. Принимает запросы на предоставление токена участника игры для аутентификации и авторизации в других сервисах системы.
3. Хранит информацию об учетных данных пользователей, выполняет аутентификацию пользователей на основе этих данных.
4. Хранит информацию о ролях пользователей и привилегиях, включенных в каждую роль, выполняет авторизацию пользователей, добавляет данные об авторизации в возвращаемый токен.

1.0

OAS3

http://localhost:4001/swagger/v1/swagger.json

Game

POST /create-game

Parameters

No parameters

Request body

application/json

{ "user", "admin" }

Execute

Clear

Token

POST /generate-game-token/{gameId}

Parameters

Name	Description
gameId • required	
string (path)	Game1

Request body

application/json

{ "username": "admin", "password": "adminpassword" }

Execute

Clear



Сервис управления игровыми объектами

1. Принимает команды для управления игровыми объектами. Клиент должен передать токен, в котором хранятся идентификаторы игры и пользователя, список разрешенных команд.

2. Управляет игровыми объектами на основе очереди команд, по таймеру пересчитывает положение объектов на основе их положений и скорости.

3. Контролирует границы игрового поля и столкновения объектов на основе двух смещенных систем окрестностей.

4. Уведомляет клиента о перемещении объектов.

Otus-project-server1.0OAS3

http://localhost:4000/swagger/v1/swagger.json

Authorize

Message

POST /message/{objectId}/{operationId}

Parameters

Cancel

Reset

Name	Description
objectId * required string (path)	<input type="text" value="Object1"/>
operationId * required string (path)	<input type="text" value="StartMove"/>

Request body

application/json

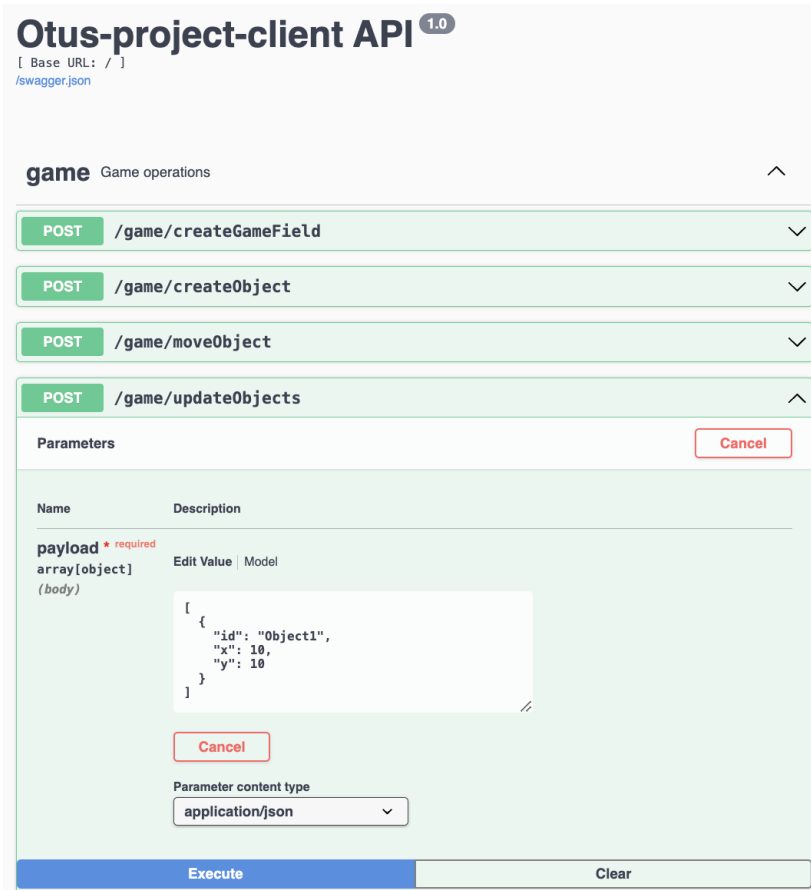
```
"{"InitVelocity": "1,1"}"
```

Execute

Clear

Клинтское приложение для отображения игрового поля

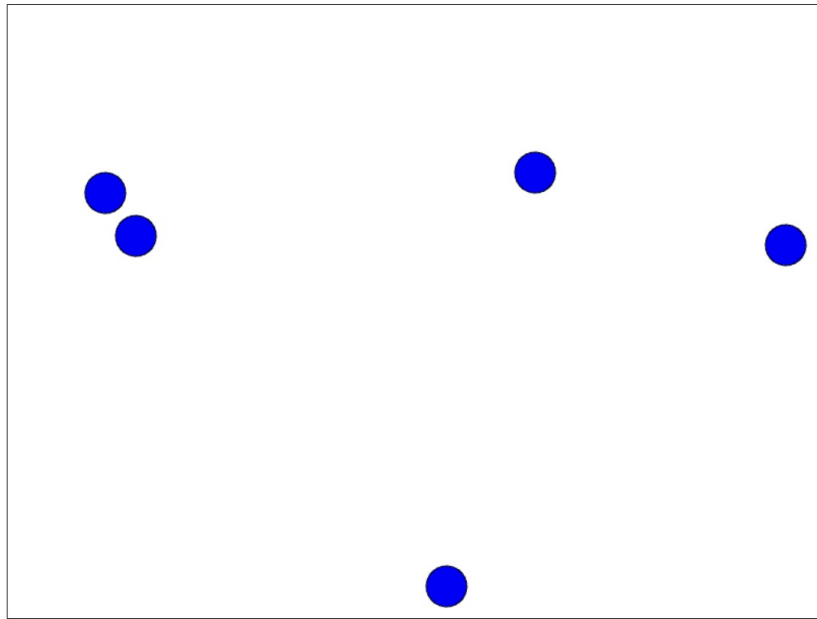
1. Принимает команды от игрового сервера по созданию игрового поля, созданию объектов на поле, перемещению конкретных объектов или обновлению состояния всех объектов.
2. Отрисовывает в браузере игровое поле и движение объектов в реальном времени с помощью WebSocket



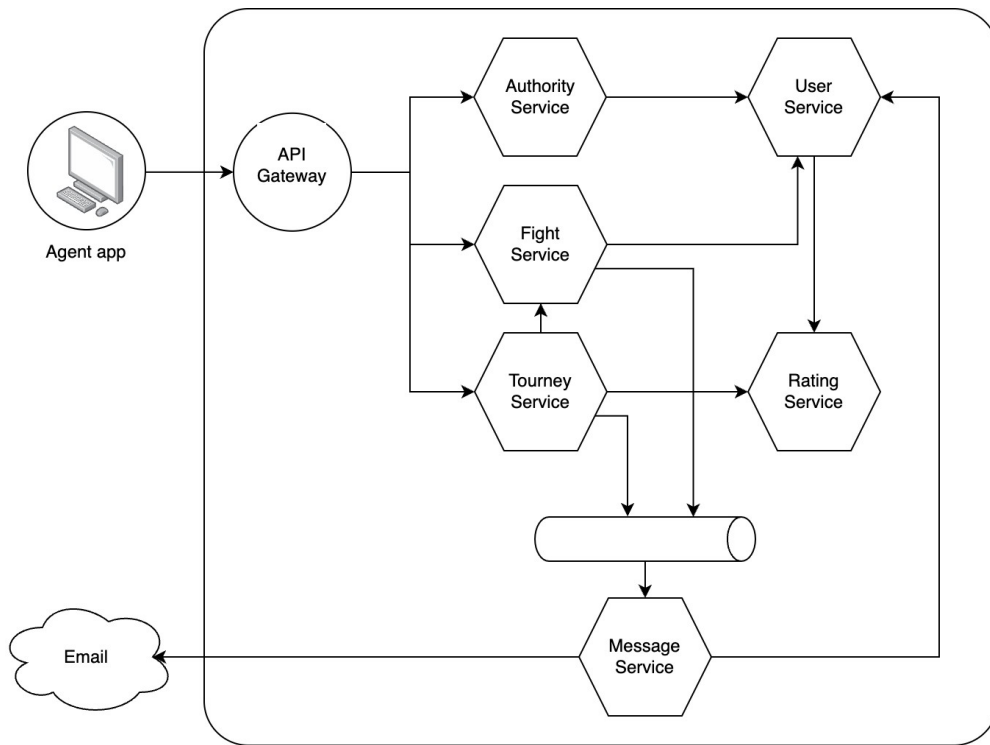
Клинтское приложение для отображения игрового поля

1. Принимает команды от игрового сервера по созданию игрового поля, созданию объектов на поле, перемещению конкретных объектов или обновлению состояния всех объектов.
2. Отрисовывает в браузере игровое поле и движение объектов в реальном времени с помощью WebSocket

Game Client



Возможная архитектура решения





Выводы

1. Удалось закрепить полученные знания, используя в проектной работе большое количество паттернов проектирования.
2. Удалось создать расширяемую систему управления игровыми объектами.
3. Удалось применить новые технологии и инструменты.

**Спасибо за
внимание!**