Пояснительная записка

К проекту Академии Samsung

Для трека «Мобильная разработка»

«Combat Shooter»

Выполнил: Солопов Даниил Дмитриевич

Руководитель: Аршинский Вадим Леонидович

**Содержание**

[1. Идея приложения 2](#_Toc74607962)

[2. Функциональные требования 3](#_Toc74607963)

[3. Использованные технологии 4](#_Toc74607964)

[4. Проектирование пользовательского интерфейса 5](#_Toc74607965)

[5. Проектирование базы данных 6](#_Toc74607966)

[6. Архитектура приложения 7](#_Toc74607967)

[7. Результат реализации 8](#_Toc74607968)

[8. Дальнейшее развитие приложения 10](#_Toc74607969)

**1. Идея приложения**

**Идея проекта** заключается в реализации мобильного 2D платформера.

Игрок преодолевает препятствия перебегая с одной платформы на другую. Присутствует три вида платформ: лава, холод и земля. По лаве обычно перемещаются монстры – главные враги игрока. Монстры охраняют сокровища и в зависимости от размера сокровища и уровня, на котором находится главный герой, монстры становятся более выносливыми и способны привести игрока к проигрышу, так как монстры будут всячески противостоять сбору сокровища игроком. Игрок же в свою очередь имеет преимущество перед монстрами – оружие дальнего боя. Игрок может воспользоваться несколькими снарядами, которые он также может найти в сокровищах. Снаряды быстро заканчиваются, поэтому игроку стоит быть экономнее ведь если снаряды закончатся, то он будет в постоянной опасности от нанесения урона монстрами.

**Задачи:**

1. Ознакомление с графической частью мобильного приложения
2. Реализация алгоритма столкновений между объектами (коллизии)
3. Практическое применение объектно-ориентированного подхода при решении задач
4. Реализация игрового музыкального сопровождения
5. Использование сторонних решений, для увеличения скорости разработки

**Существующих аналогов** достаточно много, в особенности для мобильных приложений. Такие проекты как Mario, Broforce, Sonic Mania и Rayman Origins достаточно известны широкому кругу игроков.

**Целевой аудиторией** данного проекта являются игроки, имеющие симпатию к классическим 2D-платформерам. Размеры данной аудитории могут различаться, однако, достаточно большое число игроков являются поклонниками 2D-платформеров, зачастую из-за того, что большая часть мобильных игр являются 2D-платформерами.

**2. Функциональные требования**

1. Визуализация игровых объектов должна осуществляться с помощью текстур, которые расположены в открытом доступе в интернете
2. Определённые динамические элементы (снаряд, игрок, монстр) должны перемещаться по платформам, а игрок должен иметь возможность переходить с одной платформы на другую (прыжок).
3. Игрок должен иметь возможность атаковать монстров с помощью оружия дальнего боя (в данном случае – снаряд)
4. Для сбора сокровищ необходимо преодолевать препятствия (быстрое перемещение монстров, сложное расположение сокровищ и др.)
5. Должна быть возможность просмотра статистики игроков
6. Изменение уровней с повышением уровня сложности (например, с каждым уровнем скорость перемещения монстров увеличивается, а конечная точка может быть охраняема монстром).
7. Введения системы статистики пользователей, включая возможность авторизации или регистрации, а также просмотр рейтинга (10 лучших игроков).

**3. Использованные технологии**

Данный проект был реализован на базе фреймворка LibGDX, который позволяет разрабатывать высококачественные мобильные игры.

**LibGDX** - фреймворк для создания игр и приложений, написанный на Java с использованием C и C++ и позволяющий писать кроссплатформенные игры и приложения, используя один код (<https://ru.wikipedia.org/wiki/LibGDX> )

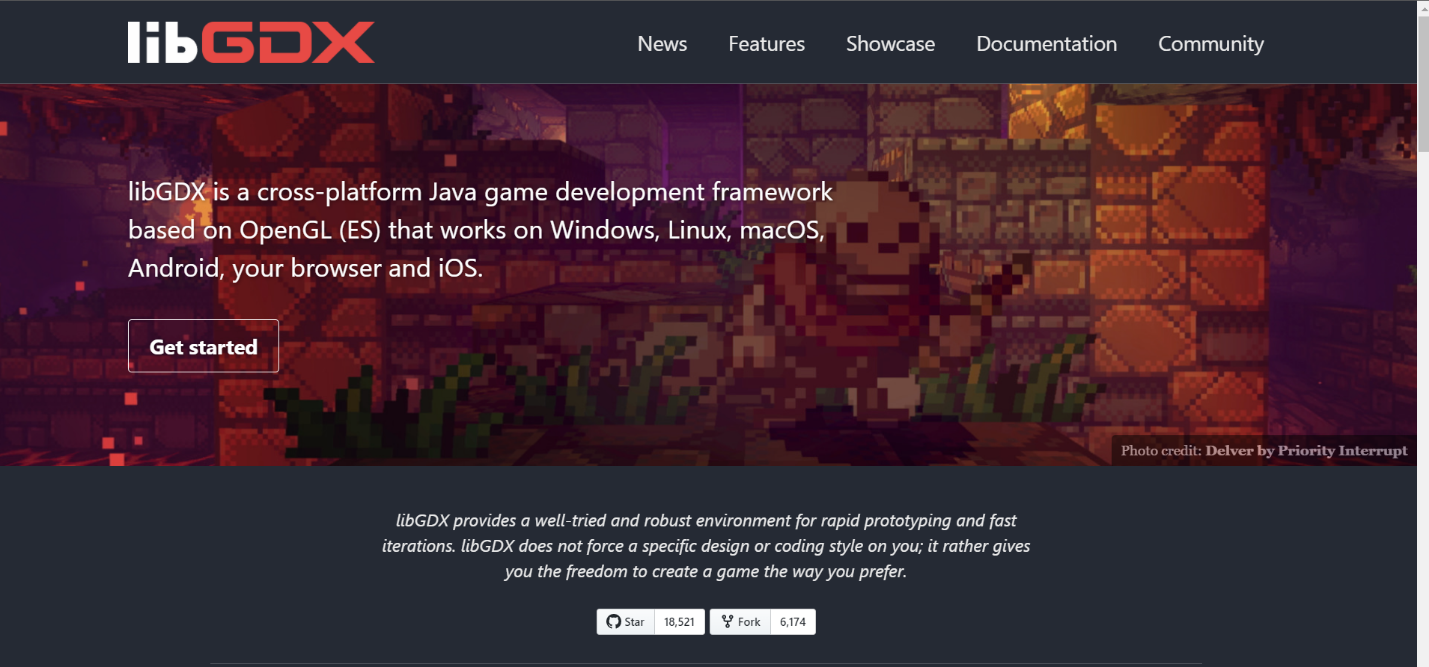


Рисунок 1 – Главная страница официального сайта libGDX

Официальный сайт libGDX расположен по адресу <https://libgdx.com/>

Данный фреймворк широко используется для создания 2D-платформеров, однако может быть использован и для 3D игр.

С помощью данного фреймворка можно с лёгкостью взаимодействовать с графическими средствами Android, например загружать текстуры и выводить их на экран, с помощью рендеринга по определённым правилам (динамические передвижения, обработки коллизий и др.). Также, данный фреймворк позволяет с легкостью обращаться к звуковым ресурсам, что способствует быстрой реализации звукового окружения.

Пиксельные изображения и музыкальные треки, используемые в игре, были скачаны из Интернета. Данные ресурсы распространяются в свободном доступе.

**4. Проектирование пользовательского интерфейса**

Макеты шаблонов пользовательского интерфейса для регистрации/авторизации и игрового меню, представлены на рисунках 2-3.

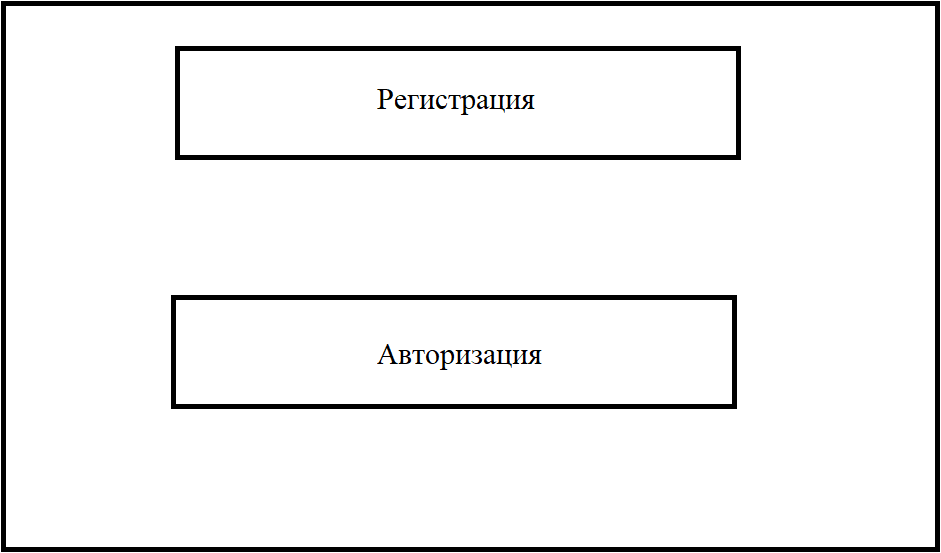


Рисунок 2 – Макет шаблона пользовательского интерфейса при входе в мобильное приложение

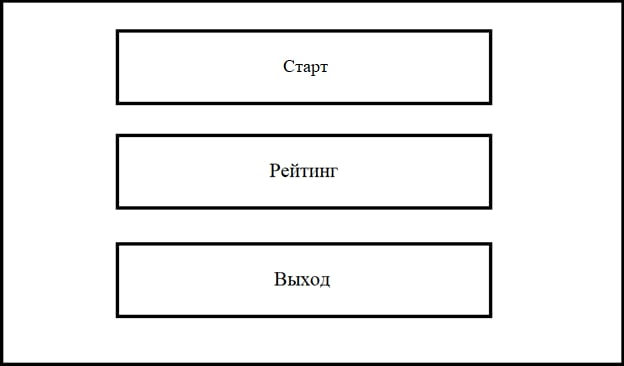


Рисунок 3 – Макет шаблона пользовательского интерфейса при авторизации пользователя

**5. Проектирование базы данных**

При входе в приложение пользователю необходимо ввести свои регистрационные данные (логин и пароль) или зарегистрироваться, тем самым дополнив базу данных пользователей.

Работа с базой данных осуществлена с помощью SQLite, поскольку данный инструмент позволяет значительно ускорить процесс развёртывания базы данных как локально, так и с возможностью развернуть её удалённо. На рисунке 4 представлена модель базы данных, которая используется в мобильном приложении.

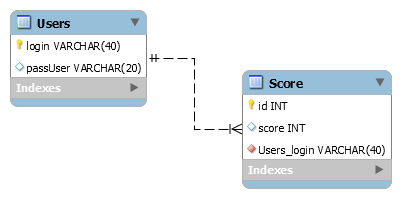


Рисунок 4 – Модель данных

**6. Архитектура приложения**

Классы в приложении представляют собой отдельные сущности, каждый из которых реализует свой определённый функционал и предоставляет другим возможность его использовать. Диаграмма классов представлена на рисунке 5.

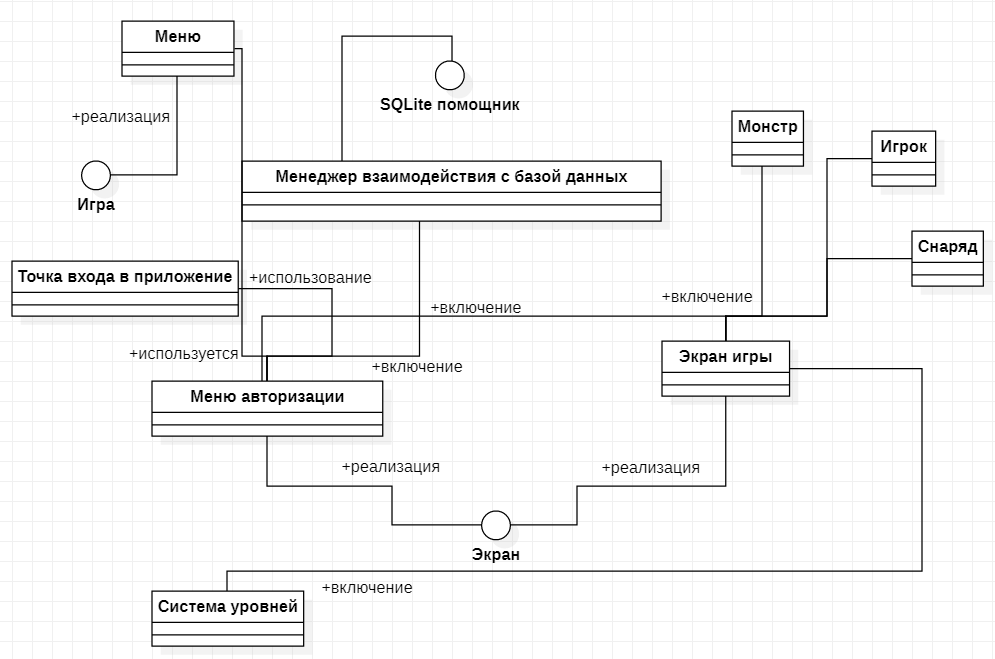


Рисунок 5 – UML-диаграмма классов

**7. Результат реализации**

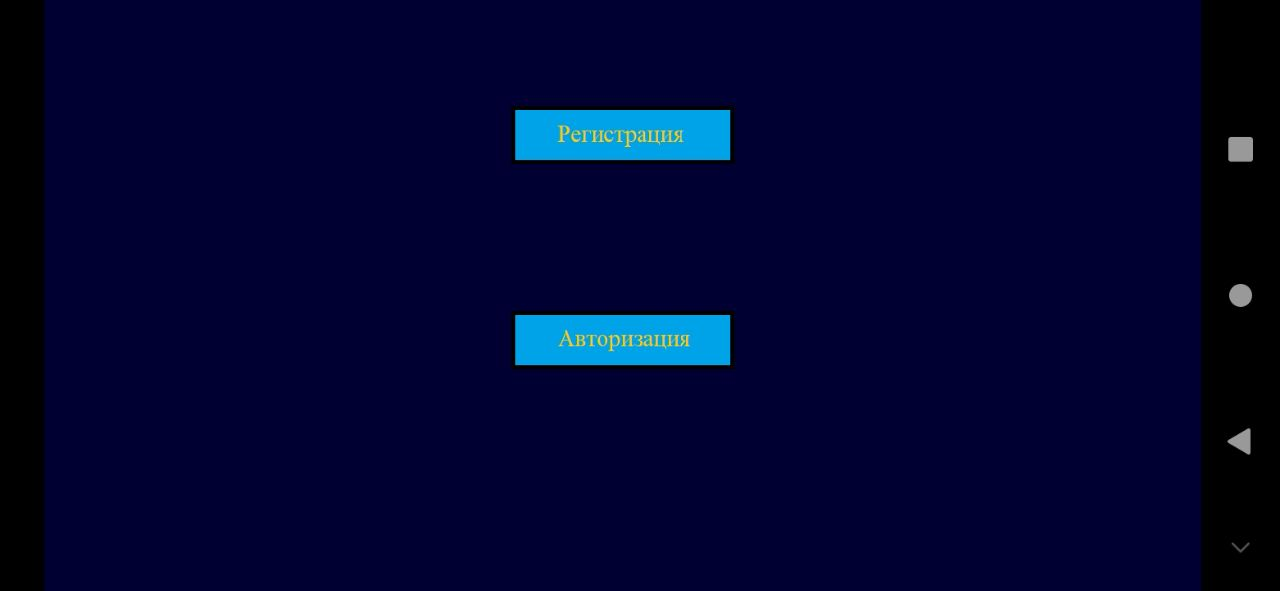


Рисунок 6 – Меню регистрации и авторизации

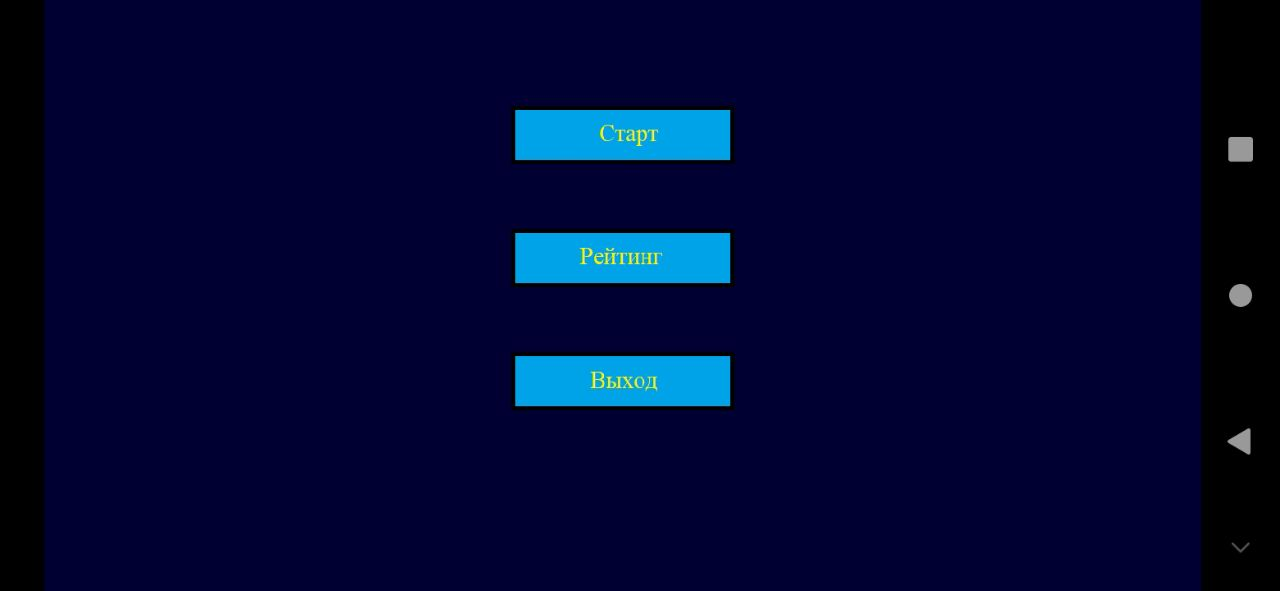


Рисунок 7 – Главное игровое меню



Рисунок 8 – Игра



Рисунок 9 – Статистика пользователей

**8. Дальнейшее развитие приложения**

В дальнейшем развитие данного приложения планируется реализовать более гибкую систему уровней. Вполне возможно, что для осуществления данной цели логику изменения уровней будет решено перенести в отдельный модуль, который будет предоставлять возможность создавать свои “карты”, которые могут быть представлены машинной и передаваться по каналам связи (загрузка карт) для последующего использования другими игроками (например, после обновления игры можно будет добавить новые карты). Также, можно реализовать центральный сервер, который будет предоставлять клиентской части приложения доступ к базе данных, на которой будут хранится регистрационные данные каждого пользователя. То есть – централизовать хранение данных и обеспечение асинхронной их обработки с учётом возможных ошибок и проблем (например, два пользователя с одним и тем же логином). Физика перемещений может быть улучшена применяя более глубокую математическую составляющую, а коллизии могут быть более гладкими, за счёт реализации анимаций столкновений, прыжков, перемещения и т.п.

Ссылка на исходный код приложения: <https://github.com/DanSoW/Combat-Shooter-GameAndroid>