Идея и актуальность:

Идея создать приложение “SketchIt” возникла, когда мне захотелось поиграть в игру “Gartic Show”, суть которой основывается на том, что стример рисует, а зрители угадывают нарисованное. Меня расстроило, что роль рисующего была доступна только для стримеров, и мне оставалось только угадывать. Появилась идея создать свое приложение, в котором каждый сможет и нарисовать, и угадать, что нарисовано.

Актуальность приложения в том, что оно помогает развивать творческие и логические способности, а также улучшать коммуникативные навыки. Игра позволяет проявить фантазию и креативность, а также улучшить навыки визуального восприятия и ассоциативного мышления. Кроме того, игра может быть интересной формой досуга и общения с друзьями или коллегами в онлайн-формате.

Цель и задачи:

Цель нашего проекта - создать MVP приятной и увлекательной многопользовательской игры, которая позволяет игрокам проявлять свои творческие способности и навыки угадывания.

Для достижения цели, нужно выполнить ряд задач:

* Создать гибкую архитектуру приложения с межсетевым взаимодействием;
* Создать удобный UI интерфейс;
* Проработать UX;
* Создать устойчивую систему клиент-серверного взаимодействия.

Функционал:

Список основных функций приложения:

* Вход / Регистрация пользователь;
* Система сессий;
* Поиск комнаты (группы) для игры;
* Трансляция изображения;
* Возможность просмотра вариантов игроков;
* Возможность отправки вариантов;
* Система рисующего игрока.

Работа приложения:

1. Процесс регистрации;
2. Процесс авторизации;
3. Процесс подбора игроков (для игры требуется 5 человек);
4. Игра.

Структура:

* GameContoller – отвечает за игровую логику;
* GameLayoutBridge – посредник между GameCotroller и UI (паттерн Mediator);
* ConsoleLogger – один из возможных способов логирования системы;
* Ilogger – абстракция для систем логирования;
* Server – класс ответственный за сетевое взаимодействие;
* Application – переопределение стандартного класса Application, entry point;
* ChatLayoutAdapter – RecyclerView Adapter для чата;
* ChatRecycleView – переопределение RecyclerView, для возможности рисования через него;
* PaintView – View, на котором происходит рисование;
* GameMenuActivity – класс activity игрового меню;
* LoginActivity - класс активности аутентификации;
* MainMenu – класс игровой активности;
* RegistrationMenu – класс активности регистрации.

Планы на будущее:

* Добавление аналитики;
* Изменение API (с поддержкой sockets);
* Добавить различные кисти и эффекты;
* Улучшить UI/UX дизайн;
* Обновить список слов;
* Поддержка разного количества игроков.