***Описание документа.***

***Настоящий документ является продолжением документа plan I DD-MM-YYYY.docx.***

Мы будем выделять отдельные части проекта(игры), т.к. нельзя просто взять и сделать игру. А, нужно «кристаллизовать» процесс, т.е. разбить его на части, задачи и этапы.

Весь проект(игра) состоит из отдельных частей:

- **сервер**. Это нужно что бы игроки могли обмениваться друг с другом данными, так уж устроен интернет. Для хранения профилей игроков (очки, сыгранные игры и т.д.);

- **клиент.** Это та - самая игра которая открывается в браузере, когда мы заходим, например, в игру во вКонтакте.

- **графика**. Тут понятно, мы должны понимать, что графика рисуется отдельно от программирования, тут всё просто вроде.

- **бизнес-логика**. Т.е. её механизмы всякие, идеи и прочие, например, «Игрок может отправить приглашение другому пользователю» - это логика, сюда не входит графика, и даже программный код, а только именно процесс, логика, описанная на бумаге или в электронном документе. «Научно» это называется **бизнес-логика**, от английского business-logic (дело и логика).

Далее будет описан **План,** в конце плана будет готовый продукт, которым можно пользоваться и радоваться.

План мы разделим на этапы. **Этап** – это будет набор того, что нужно сделать, что бы можно было выполнить какую-то отдельную, целостную часть проекта(игры). Например, создание чата в игре, мы можем описать несколькими задачами, но все их мы включим в ОДИН этап, так, что если этап будет выполнен, то мы получим готовый, рабочий чат, но не больше того. Т.е. мы не будем включать в этап, ничего кроме чата.

Надо сказать, что в процессе разработки, каждый новый этап будет сталкиваться с тем, что сделанное ранее не работает, и надо будет чинить, это нормальная практика.

Каждый этап будет описан в следующем виде.

Этап №: **Название Этапа**

**Результат:**

Конечный результат этапа. Что мы сможем увидеть, потрогать, потыкать…

*Задачи:*

№ - Называние задачи – описывает, что делаем. (*Код – кодовое название)*

Тут описание - зачем делается эта задача, к чему мы придем и что получим, что можно будет увидеть, потыкать.

Этап 1. **Автоматические авто-тесты.**

**Результат:** Будет создан код, эмитирующий действия конечного пользователя, он будет создавать игры, писать в чат и т.д. и проверять всё это сам, в случае, когда, что-то не работает, он нам сообщит об этом. Зачем это надо? Ну, во-первых, интернет меняется, и в любой момент старый код может перестать работать, хотя такое редко бывает, а во-вторых: после каждых изменений в коде, мы будем запускать эти тесты и они проверят, что ничего не сломалось, очень нужная вещь. Описаний задач не будет, по их названию будет понятно, какую область мы тестируем.

*Задачи:*

1 – Функционал для создания авто-тестов. (*ATF*)

Базовый код, с помощью которого мы и будем создавать авто-тесты.

2 – Тесты чата. (*ChatTest*)

3 – Тестирование игры по приглашению. (*InvitationsTest*)

4 – Тестирование приглашений. (*InvitationsTest*)

5 – Тестирование случайной игры. (*RandomGameTest*)

6 – Тестирование Игры с роботом. (*VsRobotGameTest*)

7 – Тестирование основной страницы, на наличие элементов.

8 – Тестирование страницы игры, на наличие элементов.

9 – Тестирование страницы рейтинга, на наличие элементов.

Этап 2. **Нагрузочное тестирование: Робот Krispi.**

**Результат:** результатом должно быть выявление узких мест сервера, что бы в будущем мы выдержали высокую пиковую нагрузку. Дело в том, что изначально, без нагрузки, невозможно сказать всё ли работает достаточно устойчиво к нагрузке, поэтому мы будем эмулировать вызовы к серверу на авторизацию, создание игры и т.д. и смотреть сколько, какое действие занимает времени. А Криспи, это я просто так называл робота, который нам поможет в этой задаче. ☺

1 – Создать базовый функционал. (*RobotKrispi*)  
Этот функционал ничего не будет делать, но даст возможность за 5-10 минут авторизоваться псевдо-игроком, создать игру, запросить кол-во онайл пользователей и т.д. Мы сделаем робота Криспи, которого можно будет попросить что-то сделать, и он это сделает. Например, авторизуется на сервере.

2 – Чат (Chat).

3 – Случайная игра

4 – Приглашения

5 – Рейтинг

6 – Игра с роботом

7 – Авторизация и информация.

8 – Состояние игрока

Этап 3. **Нагрузочное тестирование: robot Monkey.**

**Результат:…**

1 – Базовый функционал. (*RobotMonkey*)

Этап 4. **Выкладка во Вконтакте**

**1 - Оформление приложения;**

**2 – Оформление группы.**

Этап 5. **Доработки**

1 – Кнопка еще

2 – Фото оппонента

3 – Подсказки

4 – Приглашения друзей, кнопка плюс.

Этап 6. **Ручное тестирование**

Этап 7. **Хостирование проекта.**

Этап N. **Статистика**

Этап 8. **Презентационные данные.**

1 – AppData.

Этап 9. **Анализ бэклог задач и возможно создание из него задач.**

**Этап 10. Отказоустойчивость сервера.**1 – Автоперезапуск и оповещние;

2 – Пингер и оповещение;