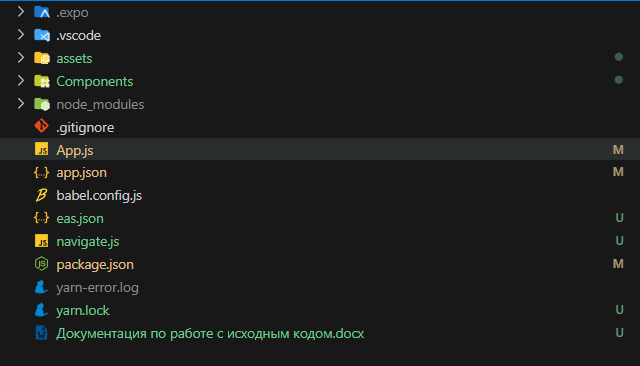
**Документация по работе с исходным кодом**

**Файлы проекта**

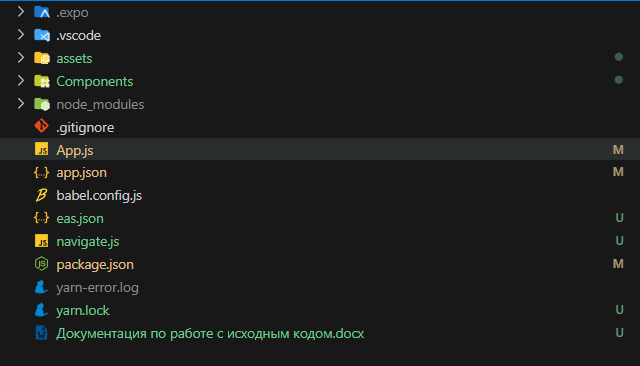
При разработке использовалось IDE [Visual Studio Code](https://code.visualstudio.com/download)

Так выглядит корневая папка проекта:



Для работы с проектом понадобится:

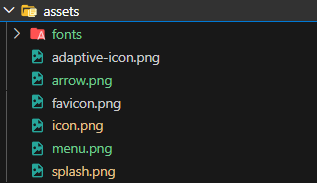
App.json



chat.js и menu.js, находящиеся в папке Components



А также папка assets



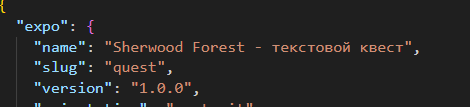
**Смена иконки приложения**

1. Вам необходимо зайти в папку assets
2. Далее вам необходимо изменить icon.png на новою иконку

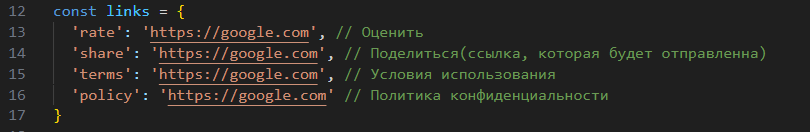
**Картинка должна быть в формате 1:1 и называться icon.png**

**Смена названия приложения**

1. Вам необходимо зайти в файл app.json
2. Далее вам необходимо изменить параметр name на новое название

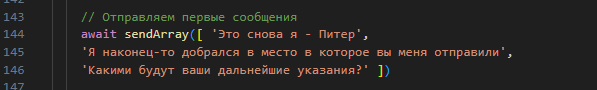
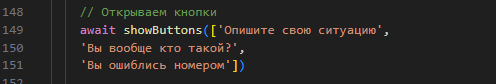


**Смена ссылок**

1. Вам необходимо зайти в файл menu.js, который находится в папке Components (./Components/menu.js)
2. Там в начале кода вы сможете увидеть такие строки: 

Ссылки на этих строках(“https://google.com”) вы можете редактировать. Каждая ссылка подписана справа (ее функция).

**Смена сюжета**

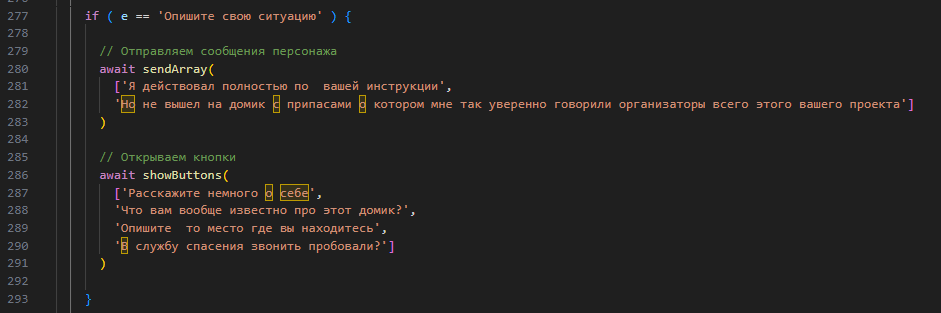
1. Вам необходимо зайти в файл chat.js, который находится в папке Components (./Components/chat.js)
2. Для начала вам нужно добавить первые сообщения(закадровый голос). Для этого посмотрите на строку 12, там вы увидите список фраз, которые вы можете поменять на свои.
3. Далее нужно настроить то, с чего начнется сюжет, для этого перейдите на строку 179, вы увидите еще один список, это список самых первых сообщений, его вы также можете поменять .
4. Чуть ниже на строке 184 есть еще один список, это список кнопок, которые откроются после отправки первых сообщений, их вы также можете менять. 
5. Перейдем к основному сюжету, откройте строку 314, после нее идет множество условий, отвечающих за сюжет.
6. Для создания собственного сюжета вам необходимо удалить все со строки 316 по строку 486
7. Далее вы можете создавать сюжетные модули(см. ниже)

**Сюжетный модуль**

Сюжетным модулем я называю условие, которое исполняется после ответа игрока

**Сюжетные модули обязательно должны находиться внутри функции plot(там, где они находятся в исходном коде бета-приложения)**

Вот пример сюжетного модуля:



Принцип работы:

* Игрок как-то отвечает
* Этот ответ проходит через условие и если ответ игрока совпадает с тем, который указан в условии, то персонаж как-либо отвечает, а затем у пользователя открываются новые кнопки с вариантами ответов

Чтобы создать свой модуль вам нужно воспользоваться структурой:

if ( e == тут ожидаемый ответ ) {

await sendArray(

[тут вы через запятую перечисляете все сообщения от персонажа]

)

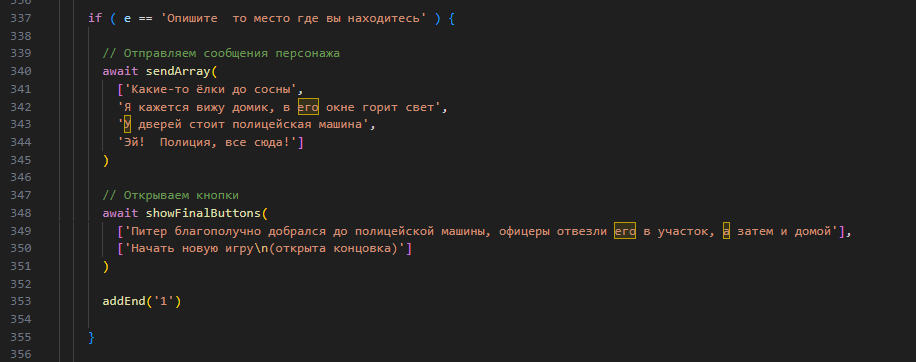
await showButtons (

[тут вы через запятую перечисляете все кнопки, которые появятся после сообщений персонажа]

)

}

Если служебный модуль должен будет открыть концовку или выдать поражение, то структура будет немного отличаться:



if ( e == тут ожидаемый ответ ) {

await sendArray(

[тут вы через запятую перечисляете все сообщения от персонажа]

)

showFinalButtons (

[тут вы указываете финальный закадровый голос],

['Начать новую игру\n(открыта концовка)' или

'Начать новую игру\n(поражение)']

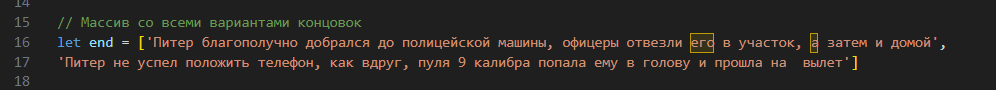
)

Тут если была открыта концовка вам необходимо добавить такую строку: addEnd(тут нужно указать уникальный индекс концовки **в кавычках**)

}

**!!!Важно**

**Все концовки должны быть добавлены в список на строке 16**

****

**Индексы всех концовок должны быть добавлены в файл menu.js в список на строке 11(все элементы должны идти через запятую). На фото пример с одной концовкой под индексом 1.**

****

**На этом фото пример с тремя концовками под индексами 3, 7 и 11**

****

**Если при завершении игры(поражение/новая концовка) возникают проблемы с размещением сообщений(сообщения поменялись местами в чате и т.п.) измените эту часть кода:**

showFinalButtons (

[тут вы указываете финальный закадровый голос],

['Начать новую игру\n(открыта концовка)' или

'Начать новую игру\n(поражение)']

)

**На эту:**

showFinalButtons (

[тут вы указываете финальный закадровый голос],

['Начать новую игру\n(открыта концовка)' или

'Начать новую игру\n(поражение)'],

true

)

**Закадровый голос**

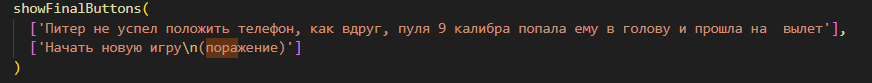
Закадровый голос отличается от остальных сообщений только цветом. Чтобы сделать какое-либо сообщение закадровым голосом нужно открыть строку 13 в файле chat.js и вписать в список текст сообщения которое вы хотите сделать закадровым голосом. На фото пример, где сообщение “Это снова я - Питер” будет являться закадровым голосом.



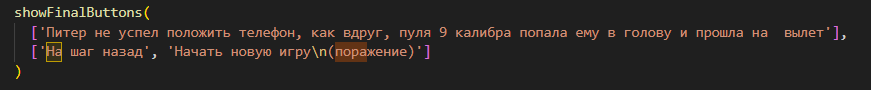
**“На шаг назад”**

Если вы хотите, чтобы после завершения игры была возможность вернуться на шаг назад вам нужно изменить часть кода(см. пример)

Без возможности вернуться на шаг назад:



С возможностью вернуться:



То есть нужно изменить второй параметр из

**['Начать новую игру\n(поражение)']**

в

**['На шаг назад', 'Начать новую игру\n(поражение)']**

Больше ничего изменять не нужно

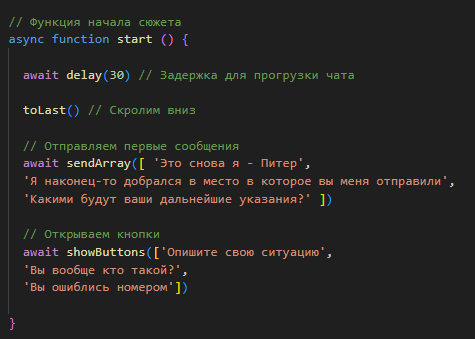
**Создание apk файла**

1. Для начала установите [node.js](https://nodejs.org/en/download)
2. После установки вам нужно через терминал открыть корневую папку проекта
3. После вам необходимо зарегистрироваться на сайте [expo](https://expo.dev/)
4. Когда вы зарегистрируетесь запустите в терминале команду **npx expo logout**, а затем **npx expo login** и введите необходимые данные
5. Далее запустите команду **eas build -p android --profile preview**
6. Далее вам будет нужно следовать инструкциям в терминале

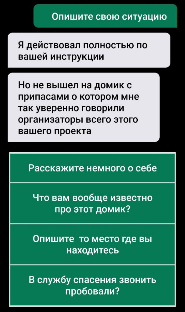
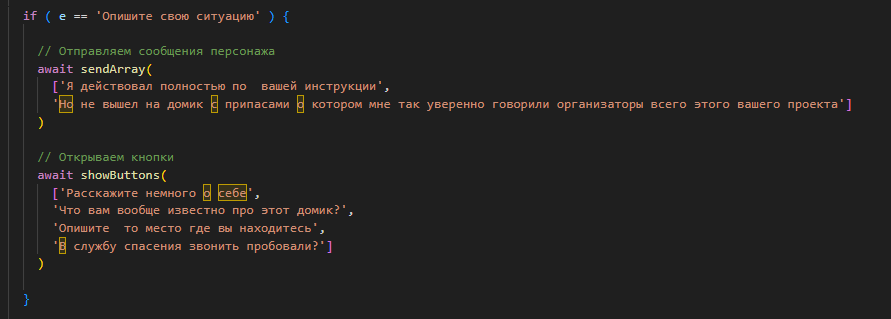
**Примеры**

Рядом с отрывком кода указан скрин, который показывает, что данный код делает

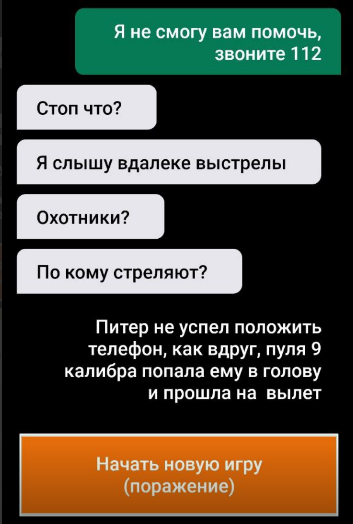
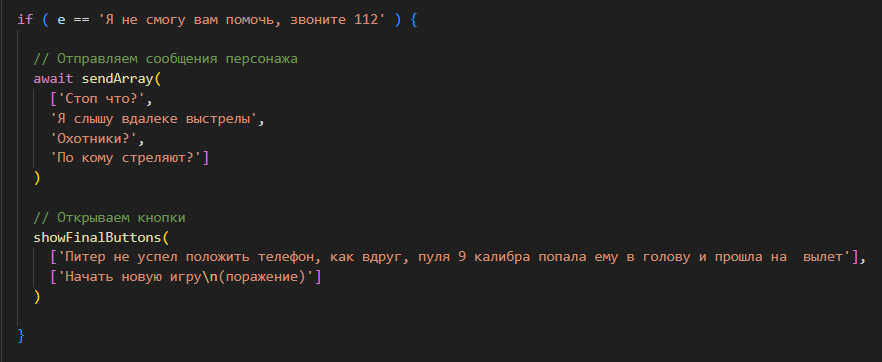
1. **Первые сообщения**

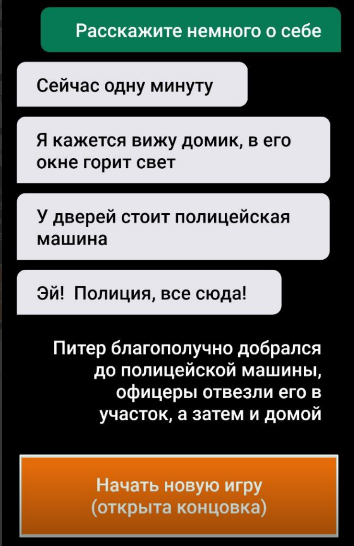
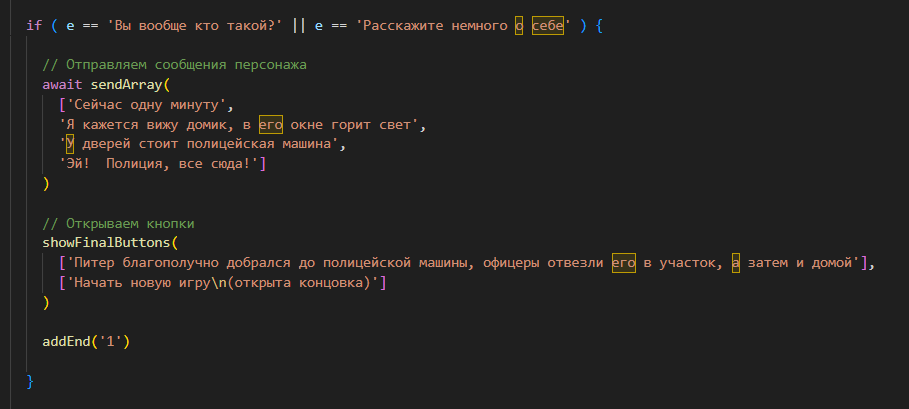
1. **Стандартный сюжетный модуль**



1. **Сюжетный модуль для завершения сюжета(поражение)**

****

1. **Сюжетный модуль для завершения сюжета(открыта концовка)**

****