Департамент образования и науки города Москвы

ГАПОУ Колледж предпринимательства №11

Центр информационно-коммуникационных технологий

**Методический материал по созданию игры “Гоночки”**

|  |  |
| --- | --- |
| Выполнили:  студенты группы ИСиП-31  Мухин Егор  Исламова Лиана  Корсакова Анастасия |  |

Москва, 2021 г.

# **Аннотация**

В данном методическом материале рассматривается создание компьютерной игры «Гоночки». Материал был разобран по видеоурокам «Интенсив need for js». Он предназначен для начинающих программистов и студентов, чтобы они смогли изучить язык программирования JavaScript. После изучения всего материала читатели смогут лучше понимать новый язык программирования, а также начнут разбираться в создании игр. В создании методического материала участвовали студенты группы ИСиП-31: Мухин Егор, Исламова Лиана и Корсакова Анастасия.

# **Пояснительная записка**

Данная тема очень актуальна, так как для большинства студентов интересна тема игр и их создания. И чтобы побудить их изучать новый языка программирования, был создан методический материал, в котором подробно описано написание игры на языке программирования JavaScript.

Цель методического материала: научить созданию игры “Гоночки”.

Ожидаемый результат: готовая рабочая игра “Гоночки”.

Представляемый методический материал намного удобнее, чем видеоуроки, по которым составлен данный материал, так как он удобен в понимании и прочтении.

# **Содержание**

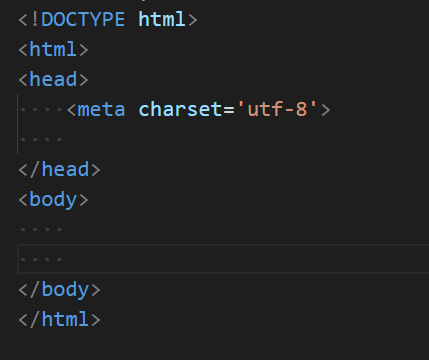
Начнем создание игры «Гоночки», написанной на JavaScript. Для начала скачайте приложенный архив «Материалы для урока.zip», в котором находятся следующие файлы: папка image (в ней имеются картинки: enemy.png, enemy2.png, player.png) и jshintrc. Далее скачайте редактор кода Visual Studio Code. Сайт, на котором можно скачать Visual Studio Code - [https://code.visualstudio.com/](https://code.visualstudio.com/%20). После скачивания редактора кода установите для него плагины. Список необходимых плагинов:

* Auto Complete Tag
* Beautify
* Bracket Pair Colorizer
* Code Runner
* JavaScript Quick Console
* Import Cost
* Open in Browser
* Live Server

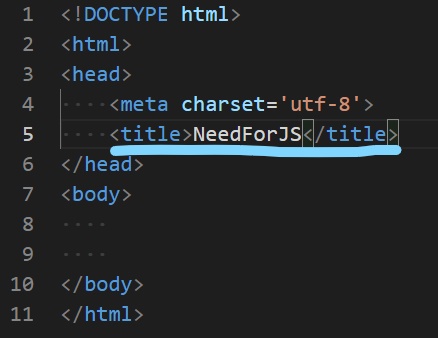
В создании игры нам потребуется изучить инструмент статического анализа кода – JSHint. Чтобы легче понимать и изучать данную технологию, предлагаем вам ознакомиться с сайтом, на котором описаны все опции JSHint. Сайт: <https://jshint.com/docs/options/>.

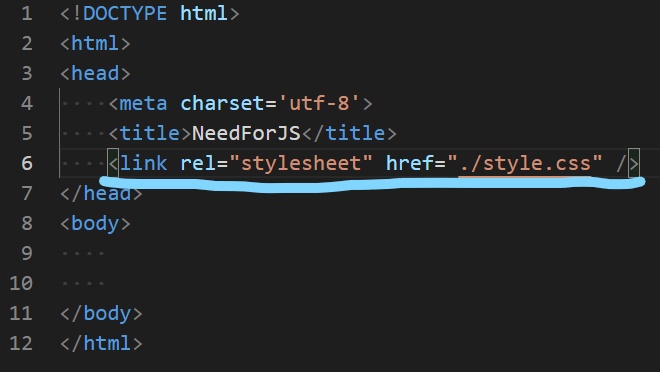
Напишем простую верстку, немного стилей и JavaScript.

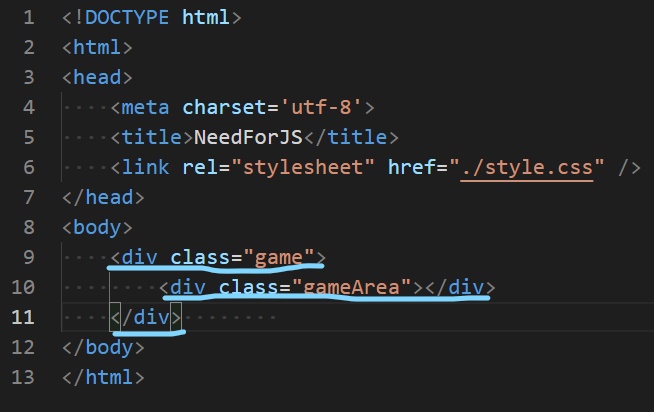
1. Для начала создадим файл index.html для того чтобы прописать в нем верстку игры.
2. Напишем структуру верстки. Тег <html> внутри него теги <head> и <body>.



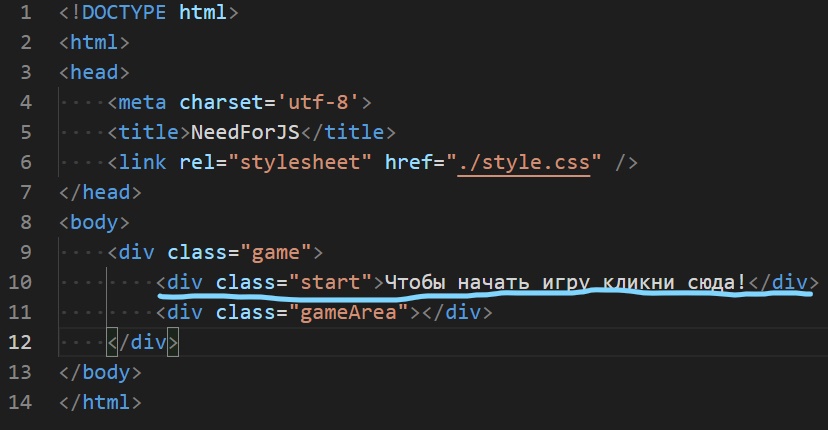
1. Внутри тега <head> напишем заголовок <title> “NeedForJS”.



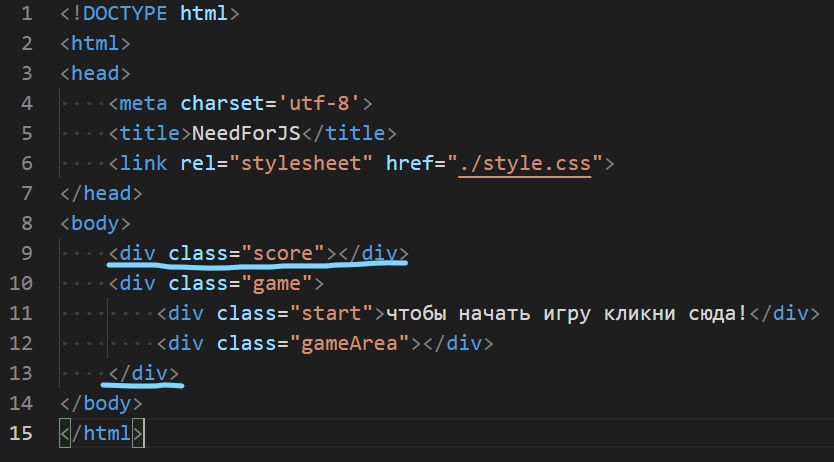
1. Добавим в проект файл со стилями (style.css) и подключим его c помощью тега <link>.
2. В <body> необходимо создать игровое поле. Создаем <div> с классом “game”. В нем создаем <div> с классом “gameArea”



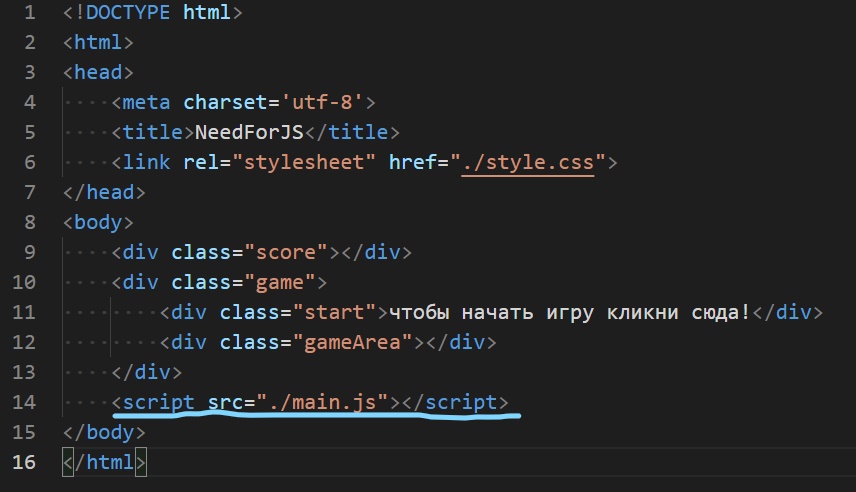
1. Также нам создать блок кликнув по которому мы будем начинать игру. В <div> с классом “game” создадим еще один <div> с классом “start”.



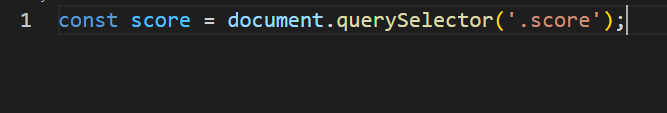
1. Чтобы в игре присутствовал какой-то соревновательный момент добавим блок с очками. Добавим <div> с классом “score”.



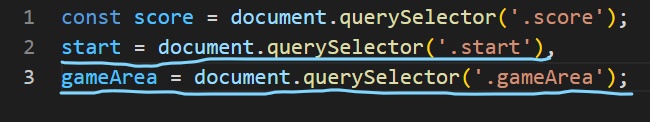
1. Надо создать файл со скриптом (main.js) и подключить в <body> с помощью тега <script>.



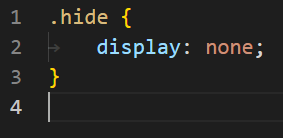
1. В файле main.js объявляем константу score с элементом страницы, в данном случае с классом “score” для дальнейшей работы с элементами страницы.

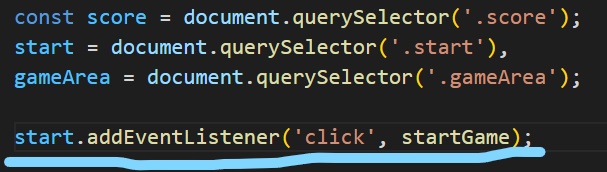


1. Задаем следующие константы start и gameArea

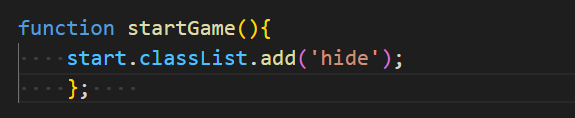


1. Создаем обработчик события, при клике на элемент. В файле style.css создаем класс hide и добавляем свойство display со значением none. Далее в файле main.js добавляем обработчик событий для элемента start, где при клике на него добавляется класс hide и скрывает кнопку старта. Обратимся при этом к функции startGame, которую пропишем позже.

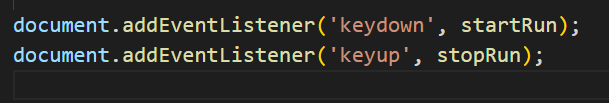




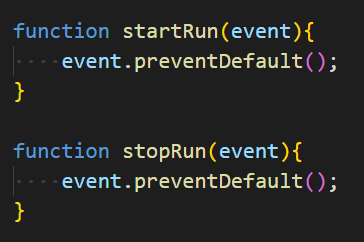
1. Объявим функцию startGame.



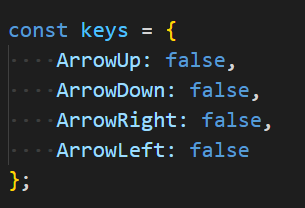
1. Пропишем два обработчика события, которые будут использоваться во всем документе.



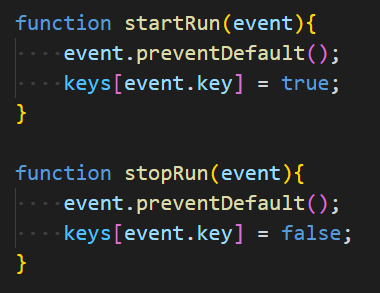
1. Опишем функции startRun и stopRun



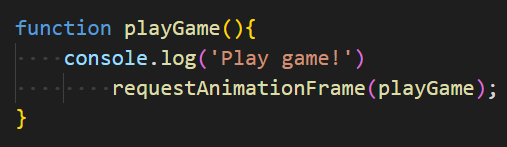
1. Создадим объект keys с названиями клавиш, который необходим для управления автомобилем



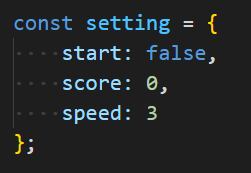
1. Дополним функции startRun и stopRun

**

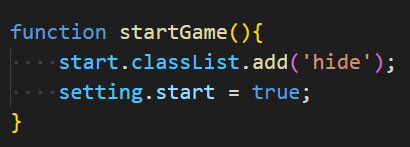
1. Создадим новую функцию playGame для того, чтобы анимация при движении машины была плавной.



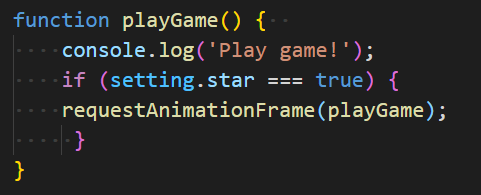
1. Создадим объект setting с первоначальными данными. Он будет содержать количество очков, скорость игры и статус игры.



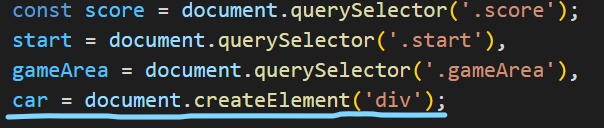
1. Дополним функцию startGame



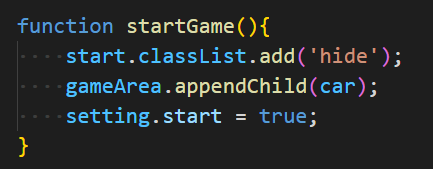
1. Пропишем условие в функции playGame



1. Зададим новую константу car, для того чтобы создать машину.

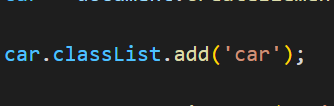


1. Добавляем в функцию startGame элемент в игровое поле используя метод appendChild

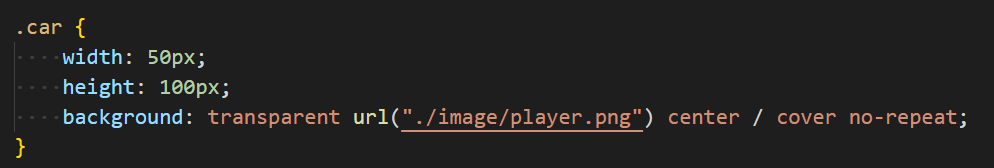


1. Добавим класс для машины (main.js) и пропишем стили в стилевом файле (style.css)

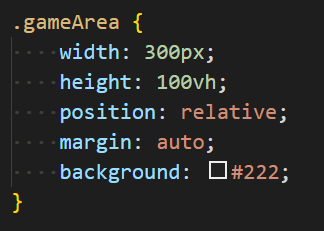
*//В файле main.js*

**

*//В файле style.css*

**

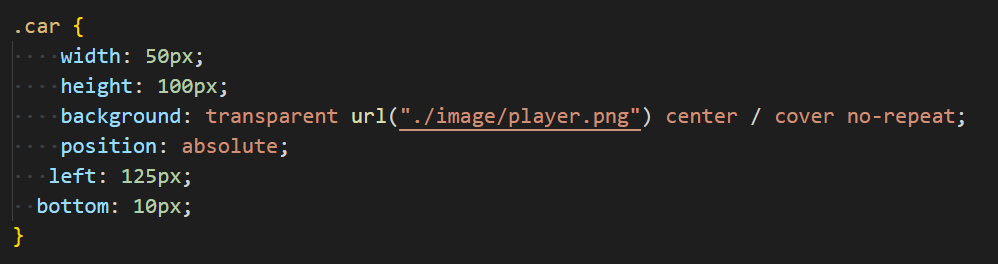
1. Создадим дорогу в стилевом файле (style.css).

**

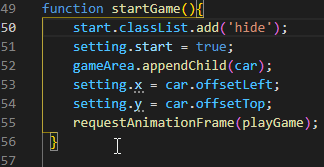
1. Расположим автомобиль наверху по центру дороги (style.css).

**

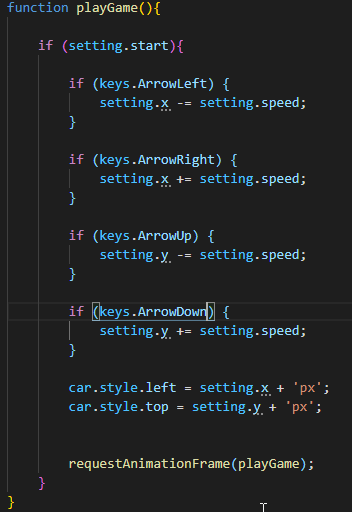
1. Расположим автомобиль внизу по центру дороги (style.css).

**

1. Пишем движение автомобиля (main.js).



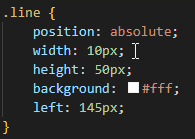
1. Напишем условие если игра запущена (main.js)



1. Чтобы машина не уходила за пределы дороги добавим условия (main.js)



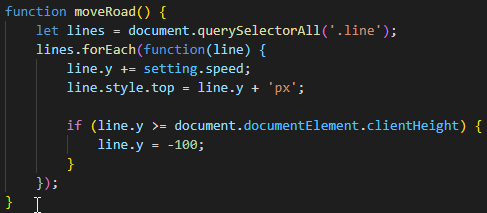
1. Реализуем анимацию движения автомобиля (style.css)



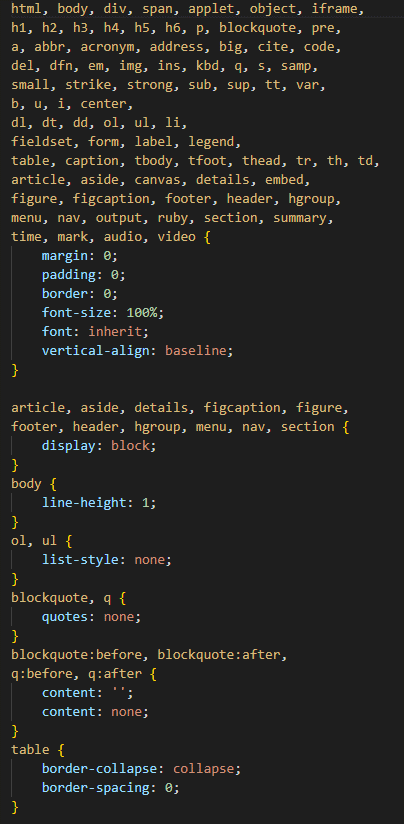
1. Пишем цикл для добавления линий на дороге (main.js)



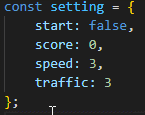
1. Добавим функцию движения дороги (main.js)



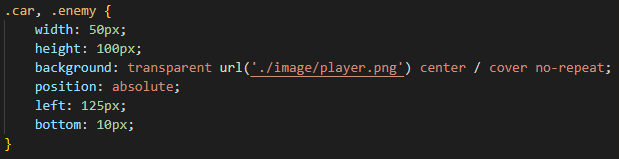
1. Обнуляем css стили, добавляем в начало документа. (style.css)



1. Добавляем новое свойство в константу setting (main.js)

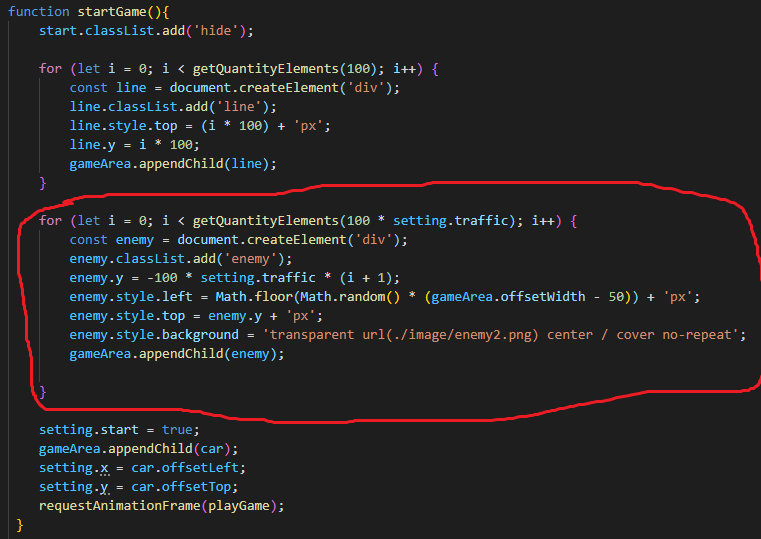


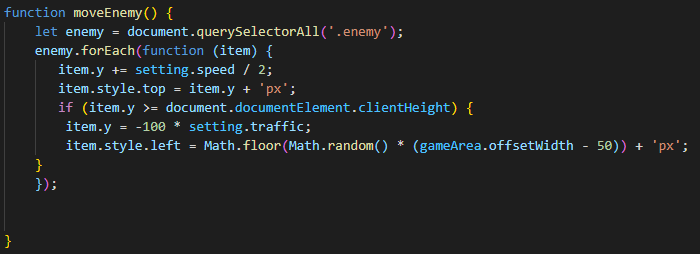
1. Добавляем к стилю элемента car элемент enemy, так как у них одинаковые стили. (style.css)



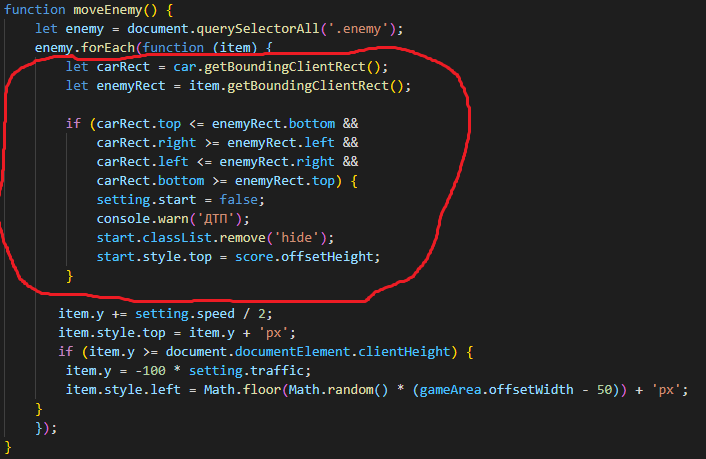
1. Пропишем функцию, которая определяет какое кол-во элементов нужно для заполнения страницы (main.js)



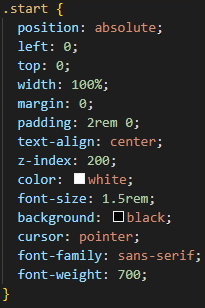
1. Напишем код, который будет создавать траффик на дороге (main.js)
2. Напишем функцию, которая заставит траффик двигаться по дороге (main.js)



1. Напишем код для реализации столкновений с траффиком (main.js)

****

1. Оформим кнопку начинающую игру (style.css)



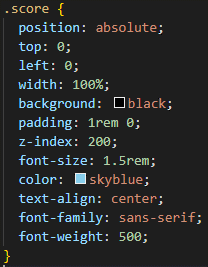
1. Сделаем подсчет очков. Для начала пропишем строчку кода в функции startGame для обнуления количества очков (main.js)



1. В функции playGame пропишем код, который будет считать количество очков и выводить их на экран (main.js)



1. Пропишем стили для строки очков (style.css)



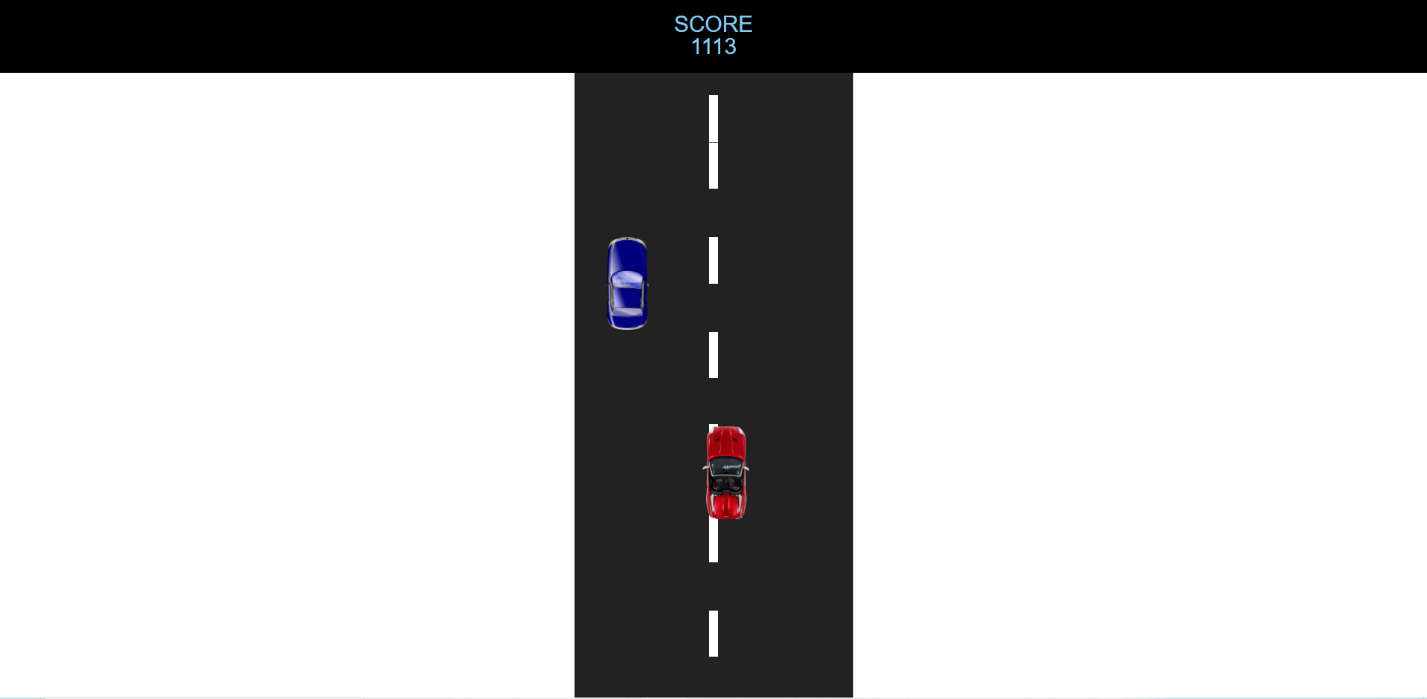
1. В функции startGame пропишем позицию для старта машины после аварии. (main.js)



Не забудьте удалить выделенные свойства из (style.css)



1. На этом мы закончили создание нашей игры. Вот так выглядит итоговый результат.



**Список рекомендуемой литературы**

[**https://learn.javascript.ru/**](https://learn.javascript.ru/)