Санкт-Петербургский политехнический университет Петра Великого

Физико-механический факультет

**Кафедра «Теоретическая механика»**

**КУРСОВОЙ ПРОЕКТ**

**Игра Движение машинок**

по дисциплине «Основы программирования и алгоритмизации»

Работу выполнила

студент гр. 5030103/10001 Сорокопудовой Е. С.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Санкт-Петербург

2022

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| Задачи . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| Описание кода JavaScript. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| Описание кода HTML . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| Результаты . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
| Вывод и заключение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |

**Введение**

JavaScript - язык программирования, разработанный для использования в браузерах. Представленная работа посвящена созданию игры Движение машинок. Целью работы является изучение основных принципов создания простейшей игры, используя полученные в ходе практик знания.

Объект исследования – процесс использования языков программирования для создания игры. Предмет исследования – программные инструменты JavaScript и HTML.

**Задача**

Реализовать и визуализировать игру со следующими правилами и условиями:

На поле в виде дорожной трассы периодически возникают едущие по крайне важным делам машинки. Задача игрока – обогнать как можно больше не особо расторопных соседей, не попав в аварию.

Управление осуществляется посредством нажатия кнопок AWSD

**Описание кода**

**JavaScript**

window.addEventListener("load", main, false);

function main() {

Задаём необходимые переменные

var button = document.getElementById('button');

var button2 = document.getElementById('button2');

var ctx = canvas\_example.getContext('2d');

var w = canvas\_example.width; var h = canvas\_example.height;

var j = 0; var jo = 59; var i = 0; var k = 40; var dx = 0; var dy = 0; var x = 130; var y = 76; var color = 0;

var max = 1; var counter = 0; var count = 0; var time = 0;

Создаём массив, в котором храним все видимые препятствующие машинки.

var car = [];

Вызываем функции для отрисовки стартовой страницы

generateCars();

drawStrokes();

drawCar(x,y);

Добавляем функции события нажатия на кнопки Старт и Стоп

button.addEventListener('click', function(e){

if(button.innerText == 'Стоп'){

button.innerText = 'Продолжить'

display.value= 'Управление: AWDS'

clearInterval(time);

}else{

if(button.innerText == 'Начать'){

timeStart();

display.value= 'Обошёл машинок: 0'

button.innerText = 'Стоп';

}

if(button.innerText == 'Продолжить'){

timeStart();

display.value= 'Обошёл машинок: '+count;

button.innerText = 'Стоп';

}

}

})

button2.addEventListener('click', function(e){

button.innerText = 'Начать'

max = 1;

x = 130;

y = 76;

counter = 0;

count = 0;

car = [];

display.value= 'Управление: AWDS'

generateCars();

drawStrokes();

drawCar(x,y);

clearInterval(time);

})

var display = document.getElementById('display');

Рисуем движущиеся полоски

function drawStrokes(){

ctx.beginPath();

ctx.fillStyle = '#818181';

ctx.fillRect(0, 0, 1200, 420);

ctx.fill();

ctx.fillStyle = '#4E5151';

ctx.beginPath();

ctx.fillRect(0, 0, 1200, 15);

ctx.fill();

ctx.beginPath();

ctx.fillRect(0, 405, 1200, 15);

ctx.fill();

for (var i=0; i<40; i++){

ctx.fillStyle = '#D4D4D4';

ctx.beginPath();

ctx.fillRect(jo+i\*60-50, 137, 24, 12);

ctx.fill();

console.log(jo)

}

for (var i=0; i<40; i++){

ctx.fillStyle = '#D4D4D4';

ctx.beginPath();

ctx.fillRect(jo+i\*60-50, 271, 24, 12);

ctx.fill();

}

}// drawStrokes

В зависимости от положения и цвета вражеской машинки рисуем её.

function Cars(xx,yy,carColor){

ctx.beginPath();

ctx.fillStyle = 'Black';

ctx.fillRect(xx-35, yy-33, 20, 66);

ctx.fillRect(xx+15, yy-33, 20, 66);

ctx.fillStyle = carColor;

ctx.fillRect(xx-50, yy-30, 100, 60);

ctx.fillStyle = '#CBE7E7';

ctx.fillRect(xx+15, yy-25, 15, 50);

ctx.fillRect(xx-35, yy-25, 20, 50);

ctx.fill();

}

Рисуем машинку. Которой управляет игрок

function drawCar(xx,yy){

ctx.beginPath();

ctx.fillStyle = 'Black';

ctx.fillRect(xx-35, yy-33, 20, 66);

ctx.fillRect(xx+15, yy-33, 20, 66);

ctx.fillStyle = 'Red';

ctx.fillRect(xx-50, yy-30, 100, 60);

ctx.fillStyle = '#CBE7E7';

ctx.fillRect(xx+15, yy-25, 15, 50);

ctx.fillRect(xx-35, yy-25, 20, 50);

ctx.fill();

}

Меняем положение машинок, не выпуская управляемую за пределы трассы.

function goOn(){

if(jo <= -1){jo= 59}

jo-=4;

x += dx;

y += dy;

if(x<50){

x = 50;

}

if (x>1150){

x = 1150;

}

if(y<48){

y = 48;

}

if (y>372){

y = 372;

}

drawStrokes();

for(var i=0; i<max; i++){

if(car[i].x < -60){

count++;

car.splice(i, 1);

console.log(car)

display.value = 'Обошёл машинок: ' + count;

generateCars();

}

car[i].x-= 10;

}

Проверяем на аварию

for(var i=0; i <max; i++){ if( ((y-30 < car[i].y - 30 )&&(y+30 > car[i].y - 30)||(y-30 < car[i].y + 30 )&&(y+30 > car[i].y + 30))&&((x-50 < car[i].x - 50 )&&(x+50 > car[i].x - 50)||(x-50 < car[i].x + 50 )&&(x+50 > car[i].x + 50))){

ctx.beginPath();

ctx.fillStyle = 'Black';

ctx.fillRect(0, 0, w, h);

ctx.fill();

drawCar(x,y);

Cars(car[i].x,car[i].y, car[i].col);

clearInterval(time);

dx = 0;

dy = 0;

display.value = 'Обошёл машинок: ' + 0;

setTimeout(function() { alert('Обошёл машинок: ' + count); }, k+10);

return;

}

}

make();

for(var i=0; i<max-1; i++){

for(var j=i+1; j<max; j++){

if ((Math.abs(car[i].x - car[j].x) < 120)&&(car[i].y = car[j].y)){

car.splice(j, 1);

max--;

}

}

}

for(var i =0; i<max; i++){

Cars(car[i].x,car[i].y,car[i].col);

}

drawCar(x,y);

}//goOn

function make(){

counter++;

if(counter == '50'){

counter = 0;

max++;

generateCars();

max++

generateCars();

console.log(max);

}

}

Создаём вражескую машинку посредством рандомного появления на одной из трёх полос.

function generateCars(){

var red = Math.trunc(Math.random()\*256);

var green = Math.trunc(Math.random()\*256);

var blue = Math.trunc(Math.random()\*256);

color = 'rgb(' + red + ',' + green + ',' + blue + ')';

car.push({x:1350, y:0, col: color});

var a = Math.floor(Math.random() \* (3 - 1 + 1)) + 1;

if(a == '1'){car[max-1].y = 76;}

if(a == '2'){car[max-1].y = 210;}

if(a == '3'){car[max-1].y = 344;}

}

setInterval – метод, вызывающий функцию каждый определённый интервал времени (в миллисекундах)

function timeStart(){

time = setInterval(goOn, k);

}

Функция события. Вызывается посредством нажатия на одну из кнопок с соответствующим кодом

document.addEventListener('keydown', function(event) {

if (event.code == 'KeyA'){ // влево

dx = -10;

}

if(event.code == 'KeyW'){ // вверх

dy = -10;

}

if(event.code == 'KeyD'){ // вправо

dx = 10;

}

if(event.code == 'KeyS'){ // вниз

dy = 10;

}

});

document.addEventListener('keyup', function(event) {

if (event.code == 'KeyA'){ // влево

dx = 0;

}

if(event.code == 'KeyW'){ // вверх

dy = 0;

}

if(event.code == 'KeyD'){ // вправо

dx = 0;

}

if(event.code == 'KeyS'){ // вниз

dy = 0;

}

});

}

**HTML**

В файле HTML создаём холст, необходимые для запуска игры кнопки и экраны.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<p align ="center">

<canvas style="border: 1px solid black;" id="canvas\_example" width="1200" height="420"></canvas><br>

<input disabled type="display" value='Управление: AWDS' id='display'>

<button type='button' id='button' style="width:100px; background: red"> Начать </button>

<button type='button' id='button2' style="width:100px; background: red" > Заново </button>

</p>

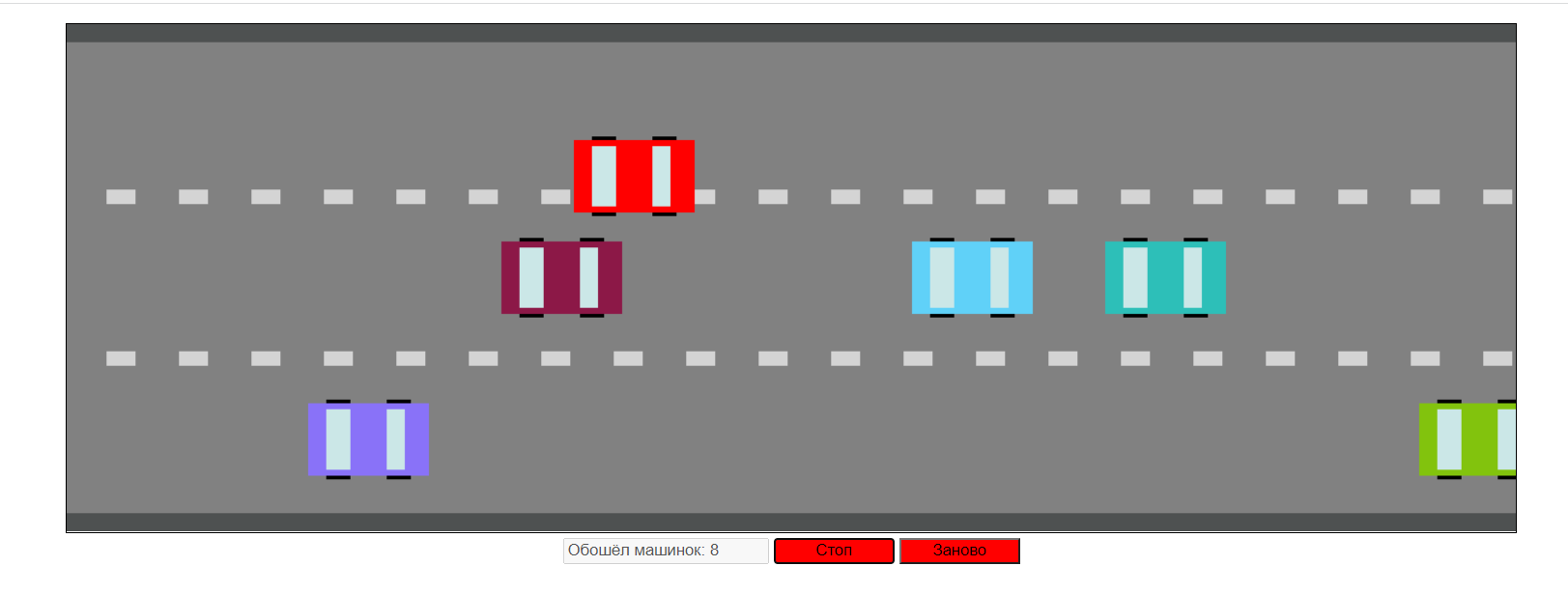
<script src="cars.js"></script>

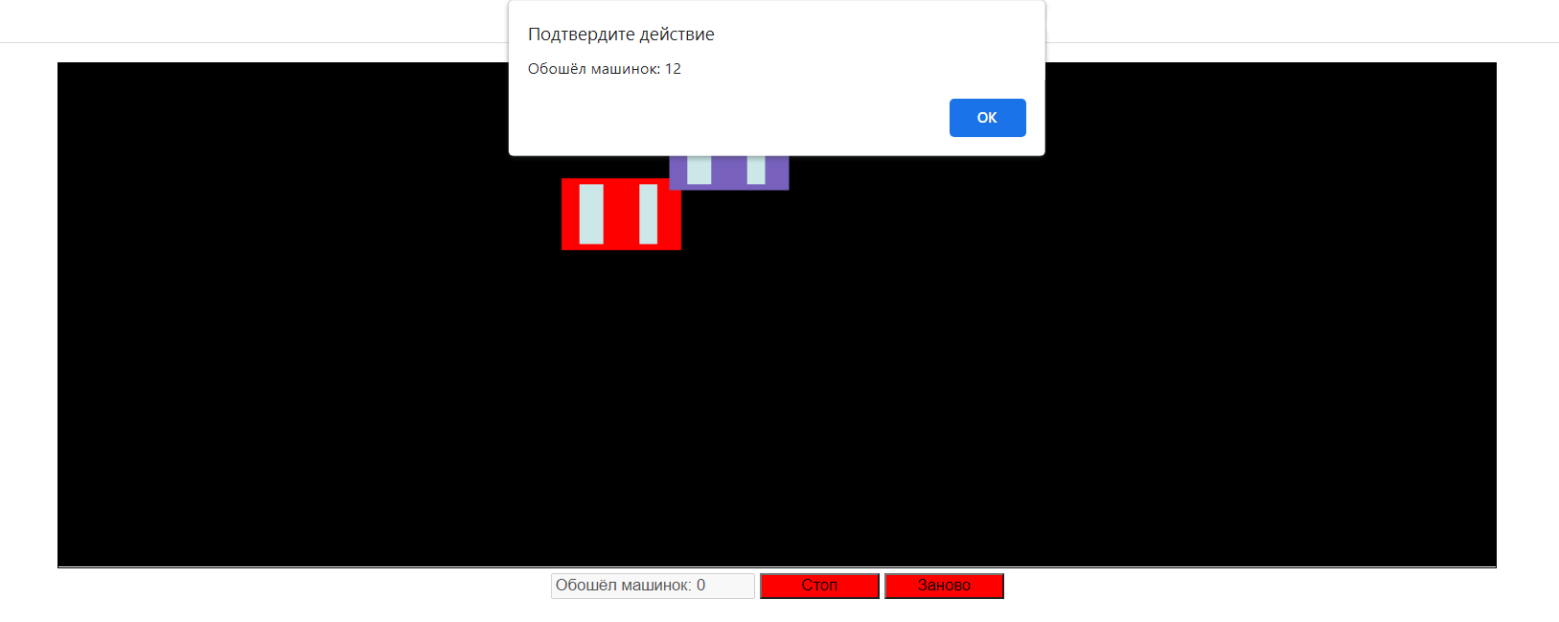
</body>

</html>

**Результаты**

Мы создали игру Движение машинок, используя JavaScript. Ниже представлена визуализация описанного проекта.





**Вывод**

С помощью методов объектно-ориентированного программирования был составлен код простейшей компьютерной игры Движение машинок. Поставленные в ходе работы задачи выполнены, мы получили полезный опыт в работе над созданием несложных проектов и составлением курсовых работ.

**Заключение**

Безусловно, существует возможность добавления новых механик к игре в качестве последующей доработки программы и улучшения динамической и визуальной составляющих игры, а также добавление более сложных условий и правил. В заключении хотелось бы сказать, что JavaScript – это функциональный и актуальный язык, позволяющий выполнять большой спектр задач, включающий и разработку простейших браузерных игр.

**ФОРМА ЗАДАНИЯ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

Санкт-Петербургский политехнический университет Петра Великого

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

студенту группы 5030103/10001 Сорокопудовой Е. С. и 10004 Мекрюковой П. И.

***1. Тема проекта:*** игра Движение машинок

***2. Срок сдачи студентом законченного проекта***

***3. Исходные данные к проекту:*** курс лекций по основам программирования и алгоритмизации.

***4. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения.

Примерный объём пояснительной записки - страниц печатного текста.

***5. Перечень графического материала*** (с указанием обязательных чертежей и плакатов): отсутствует

***6. Консультанты***

***7. Дата получения задания:*** «\_\_\_».\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) (инициалы, фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_ *(дата)*