|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-01-19 | | Соколовский И.М. | |
|  | |  | |
| Принял  *Ассистент* | | Коваленко М.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись руководителя) | |
|  |  | |  | |
|  |  | |  | |

Москва 2020

Оглавление

[**Практическая работа №8 «JavaScript-анимация»** 2](#_Toc59264629)

[**Цель работы** 2](#_Toc59264630)

[**Практическое задание 1: «Использование setInterval»** 2](#_Toc59264631)

[**Практическое задание 2: «Функция «draw»** 4](#_Toc59264632)

[**Практическое задание 3: «Анимированный прыгающий мяч»** 7](#_Toc59264633)

[**Вывод** 10](#_Toc59264634)

**Практическая работа №8 «JavaScript-анимация»**

**Цель работы**

Ознакомиться с возмоностями создания анимации в JavaScript, закрепить прежние навыки работы с JavaScript.

**Практическое задание 1: «Использование setInterval»**

Выбрать изображение любого вида транспорта и реализоватьанимацию (при нажатии на данное изображение) через последовательность кадров, каждый из которых немного меняет HTML/CSS-свойства. Например, изменение style.left от 0px до 100px – двигает элемент. И если это будет выполняться с помощью setInterval, изменяя на 2px с небольшими интервалами времени, например, 50 раз в секунду, тогда изменения будут выглядеть плавными. Принцип такой же, как в кино: 24 кадров в секунду достаточно, чтобы создать эффект плавности.

Для реализации необходимо:

1. Запомнить время начала анимации;
2. Добавить функцию для расчета времени, которое прошло с начала анимации;
3. Закончить анимацию через 2 секунды;
4. В то время как timePassed идёт от 0 до 2000 left изменяет значение от 0px до 400px.

**Описание выполнения задания**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<style>

#car{

position: relative;

cursor: pointer;

}

</style>

</head>

<body>

<img id="car" src="http://pngimg.com/uploads/lotus/lotus\_PNG41.png" width="170" height="140">

<script>

car.onclick = function() {

let start = Date.now();

let timer = setInterval(function() {

let timePassed = Date.now() - start;

car.style.left = timePassed / 5 + 'px';

if (timePassed > 2000)

clearInterval(timer);

}, 20);

}

</script>

</body>

</html>

Результат работы кода:



Рисунок 1

**Практическое задание 2: «Функция «draw»**

Создать анимацию набирающегося текста в элементе формы textarea при нажатии на кнопку. Добавить в <head> следующую функцию для анимации (дописав функции для textArea и расчёта времени bounce):

function animate(options) {

var start = performance.now();

requestAnimationFrame(function animate(time) {

// timeFraction от 0 до 1

var timeFraction = (time - start) / options.duration;

if (timeFraction > 1) timeFraction = 1;

// текущее состояние анимации

var progress = options.timing(timeFraction)

options.draw(progress);

if (timeFraction < 1) {

requestAnimationFrame(animate);

}

});

}

**Описание выполнения задания**

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css">

<script>function animate(options) {

let start = performance.now();

requestAnimationFrame(function animate(time) {

// timeFraction от 0 до 1

let timeFraction = (time - start) / options.duration;

if (timeFraction > 1) timeFraction = 1;

// текущее состояние анимации

let progress = options.timing(timeFraction)

options.draw(progress);

if (timeFraction < 1) {

requestAnimationFrame(animate);

}

});

}</script>

</head>

<body>

<textarea id="textExample" rows="3" cols="45">Поздравленья с Днём Рожденья принимай! Веселись до посиненья, пей, гуляй!

</textarea>

<button onclick="animateText(textExample)">Старт</button>

<script>

function animateText(textArea) {

let text = textArea.value;

let to = text.length,

from = 0;

animate({

duration: 10000,

timing: bounce,

draw: function(progress) {

let result = (to - from) \* progress + from;

textArea.value = text.substr(0, Math.ceil(result))

}

});

}

function bounce(timeFraction) {

for (let a = 0, b = 1, result; 1; a += b, b /= 2) {

if (timeFraction >= (7 - 4 \* a) / 11) {

return -Math.pow((11 - 6 \* a - 11 \* timeFraction) / 4, 2) + Math.pow(b, 2)

}

}

}

</script>

</body>

</html>

Результат работы кода:

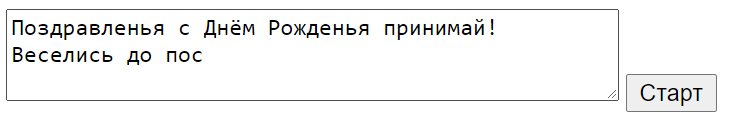


Рисунок 2

**Практическое задание 3: «Анимированный прыгающий мяч»**

Создать анимацию с прыгающим мячом. При нажатии на мяч он должен падать вниз до созданной горизонтальной линии и несколько раз подскакивать от этой поверхности. Чтобы получить эффект «скачущего» мяча, можно использовать функцию расчёта времени bounce в режиме easeOut.

**Описание выполнения задания**

<!DOCTYPE HTML>

<html>

<head>

<script>

function animate(options) {

var start = performance.now();

requestAnimationFrame(function animate(time) {

var timeFraction = (time - start) / options.duration;

if (timeFraction > 1) timeFraction = 1;

var progress = options.timing(timeFraction)

options.draw(progress);

if (timeFraction < 1) {

requestAnimationFrame(animate);}

});

}</script>

<style>

#field

{

height: 200px;

border-bottom: 1px yellow groove;

position: relative;

}

#ball {

position: absolute;

cursor: pointer;

}

</style>

</head>

<body>

<div id="field">

<img src="ball.png" width="40" height="40" id="ball">

</div>

<script>

function makeEaseOut(timing) {

return function(timeFraction) {

return 1 - timing(1 - timeFraction);

}

}

function bounce(timeFraction) {

for (let a = 0, b = 1, result; 1; a += b, b /= 2) {

if (timeFraction >= (7 - 4 \* a) / 11) {

return -Math.pow((11 - 6 \* a - 11 \* timeFraction) / 4, 2) + Math.pow(b, 2)

}

}

}

ball.onclick = function() {

let to = field.clientHeight - ball.clientHeight;

animate({

duration: 3000,

timing: makeEaseOut(bounce),

draw(progress) {

ball.style.top = to \* progress + 'px'

}

});

};

</script>

</body></html>

Результат работы кода:

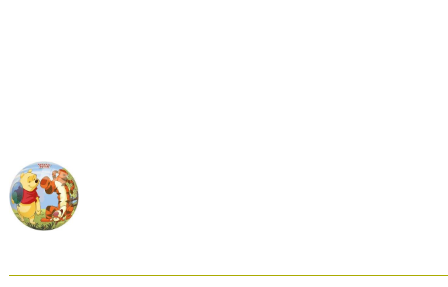


Рисунок 3

**Вывод**

В результате выполнения данной практической работы, был получен навык задания анимации в JavaScript, а также были закреплены навыки работы с DOM.