|  |  |
| --- | --- |
| **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | |
| **Задание №6**  **по теме «Анализ изображения при использовании инструментов фреймворка*Vue.js*»**  **по дисциплине «Web-программирование»** | |
|  | Выполнил:  Студент группы ТКИ-542  Волочинский И.О.  Ефремов Д.Ю.  Проверил:  Доцент кафедры УиЗИ, к.т.н., с.н.с  Сафронов Ф.И. |
| Москва 2024 | |

Оглавление

[1. ЦЕЛЬ РАБОТЫ 3](#_gjdgxs)

[2. ФОРМУЛИРОВАКА ЗАДАЧИ 3](#_1fob9te)

[3. ТАБЛИЦА СООТВЕТСВИЯ ПЕРЕМЕННЫХ 3](#_3znysh7)

[4. WEB-СТРАНИЦА 4](#_2et92p0)

[4.1. Код web-приложения 4](#_tyjcwt)

[4.2. Результат отображения в браузере 4](#_3dy6vkm)

[4.3. Сети Петри 4](#_1t3h5sf)

[5. ВЫВОД ПО РАБОТЕ 5](#_4d34og8)

1. **Цель работы**

Провести анализ изображения, применить навыки создания простых локальных одностраничных web-приложений под управлением Фреймворка Vue.js на языке JavaScript.

1. **ФОРМУЛИРОВАКА ЗАДАЧИ**

Средствами *SVG* или *Canvas* под управлением фреймворка *Vue.js* построить программное, браузерное *web*-обеспечение, реализующее анализ подгруженного изображения (*\*.jpeg*) по варианту.

1. **таблица соответсвия переменных**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Цветовая модель** | **Объект графики на web-странице** | **Реализация управления гистограммой** | **Режим вывода загруженного изображения** | **Режим построения гистограммы** | **Вариант** |
| **3** | *RGB* | *Canvas* | Суммарная / Все каналы на одном | Прямой (оригинал) | Инверсный (справа-налево) | **17** |

1. **Web-страница**
   1. **Код ImageAnalyzer.vue**

<style>

body {

*font-family*: Arial, **sans-serif**;

}

.card {

*margin*: 15**px** 5**px**;

}

canvas {

*border*: **solid** 1**px** #999;

}

p {

*margin*: 5**px** 0 3**px**;

}

</style>

<template>

<div class**="image-analyzer"**>

<div class**="card"**>

<p>**Выберите изображение из файла**</p>

<input type**="file"** @change**="**handleFileChange**"**>

<br>

<img :src**="**imageSrc**"** @load**="**getImageData**"** v-if**="**imageSrc**"**>

</div>

<div class**="card"**>

<p>**Гистограмма**</p>

<div>

<input type**="radio"** value**="value"** v-model**="**histogramMode**"** @change**="**drawHistogram**"**>

<label>**Значение**</label>

<input type**="radio"** value**="color"** v-model**="**histogramMode**"** @change**="**drawHistogram**"**>

<label>**Цвет**</label>

</div>

<canvas ref**="histogramCanvas"** width**="256"** height**="150"**></canvas>

</div>

</div>

</template>

<script>

export default {

data() {

return {

imageSrc: null,

imageData: null,

histogramMode: 'value',

histogramData: {

cyan: new *Array*(256).fill(0),

magenta: new *Array*(256).fill(0),

yellow: new *Array*(256).fill(0),

black: new *Array*(256).fill(0),

},

};

},

methods: {

handleFileChange(*event*) {

const file = *event*.target.files[0];

if (file) {

this.imageSrc = *URL*.createObjectURL(file);

}

},

getImageData(*event*) {

const canvas = *document*.createElement('canvas');

const ctx = canvas.getContext('2d');

canvas.width = *event*.target.width;

canvas.height = *event*.target.height;

ctx.drawImage(*event*.target, 0, 0);

this.imageData = ctx.getImageData(0, 0, canvas.width, canvas.height).data;

this.calculateHistogram();

this.drawHistogram();

},

calculateHistogram() {

this.histogramData.red = new *Array*(256).fill(0);

this.histogramData.green = new *Array*(256).fill(0);

this.histogramData.blue = new *Array*(256).fill(0);

*console*.log('RED ' + (this.imageData[0] & 0xFF).toString(10))

*console*.log('GREEN ' + ((this.imageData[0] >> 8) & 0xFF).toString(10))

*console*.log('BLUE ' + ((this.imageData[0] >> 16) & 0xFF).toString(10))

for (let i = 0; i < this.imageData.length; i += 4) {

const r = this.imageData[i];

const g = this.imageData[i + 1];

const b = this.imageData[i + 2];

this.histogramData.red[r]++;

this.histogramData.green[g]++;

this.histogramData.blue[b]++;

}

},

drawHistogram() {

const canvas = this.$refs.histogramCanvas;

const ctx = canvas.getContext('2d');

ctx.clearRect(0, 0, canvas.width, canvas.height);

const maxValue = *Math*.max(

...this.histogramData.red,

...this.histogramData.green,

...this.histogramData.blue

);

const drawChannel = (*channel*, *color*) => {

ctx.fillStyle = *color*;

for (let i = 0; i < 256; i++) {

const value = this.histogramData[*channel*][i];

const height = (value / maxValue) \* canvas.height;

ctx.fillRect(i, canvas.height - height, 1, height);

}

};

if (this.histogramMode === 'value') {

drawChannel('red', 'red');

drawChannel('green', 'green');

drawChannel('blue', 'blue');

} else {

const combined = new *Array*(256).fill(0);

for (let i = 0; i < 256; i++) {

combined[i] =

this.histogramData.red[i] +

this.histogramData.green[i] +

this.histogramData.blue[i];

}

ctx.fillStyle = 'gray';

for (let i = 0; i < 256; i++) {

const value = combined[i];

const height = (value / maxValue) \* canvas.height;

ctx.fillRect(i, canvas.height - height, 1, height);

}

}

},

},

};

</script>

* 1. **Код App.vue**

<script setup>

import Clients from './components/ImageAnalizer.vue'

import TheWelcome from './components/TheWelcome.vue'

</script>

<template>

<main>

<Clients />

</main>

</template>

<style scoped>

header {

*line-height*: 1.5;

}

.logo {

*display*: **block**;

*margin*: 0 **auto** 2**rem**;

}

@media (*min-width*: 1024**px**) {

header {

*display*: **flex**;

*place-items*: **center**;

*padding-right*: calc(var(--section-gap) / 2);

}

.logo {

*margin*: 0 2**rem** 0 0;

}

header .wrapper {

*display*: **flex**;

*place-items*: **flex-start**;

*flex-wrap*: **wrap**;

}

}

</style>

* 1. **Результат отображения в браузере**

|  |
| --- |
|  |
| 1. – Результат запуска |
|  |
| 1. – Начальная страница |
|  |
| 1. – Результат отображения гистограммы значения |
|  |
| 1. – Результат отображения гистограммы для цвета |

* 1. **Сети Петри**

|  |
| --- |
|  |
| 1. – Сети Петри |

1. **вывод по работе**

Мы получили навыки по созданию локальных одностраничных web-приложение под управлением фреймворка Vue.js на языке JavaScript.