|  |  |
| --- | --- |
| Картинки по запросу самгту новый логотип | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  ВЫСШЕГО ОБРАЗОВАНИЯ  «САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» |

Кафедра «Вычислительная техника»

Лабораторная работа № 1

по дисциплине «Программное обеспечение обработки визуальных данных»

«Формирование цветовой палитры для отображения 14-битных кодов пикселей»

Выполнили:

студенты 3-ИАИТ- 9

*Котляров А.А., Ефимушкин Н.А.*

Приняли:

доцент Мартемьянов Б.В.,

Лапир В.Д.

Самара 2019

**Цель работы:** ознакомление с форматом представления кадров изображений, генерируемых камерой малого космического аппарата дистанционного зондирования Земли «Аист-2D» в инфракрасном диапазоне электромагнитного излучения, и разработка программы формирования цветовой палитры для отображения 14-битных кодов яркостей пикселей таких кадров.

**Задание**

Разработать программу формирования цветовой палитры для отображения 14-битных кодов яркостей пикселей кадров, изображений, генерируемых камерой малого космического аппарата дистанционного зондирования Земли «Аист-2D» в инфракрасном диапазоне электромагнитного излучения.

**Выполнение и результаты:**

Формируемые камерой наборы видеоданных содержат 14-битные коды пикселей. То есть, яркость пиксела может достигать значения 214 - 1= 16383 условных единицы. Стандартная видеосистема компьютера формирует изображение на экране монитора в формате 8 бит на яркость каждого из трех основных цветов формата RGB. Такая видеосистема способна отображать

224 > 16 млн цветов. Но для адекватного восприятия распределения поля температур в пределах ИК кадра необходимо использовать цветовую палитру, в которой понятна последовательность смены цветов при увеличении значения кода пиксела.

Для формирования указанной палитры воспользуемся RGB кубом, но не единичным, а со стороной длиной 255 единиц. 255 – это максимальное значение кода длиной 1 байт.

На рисунке 1 показан цветовой куб и стрелками указана последовательность обхода его ребер, которую следует использовать для формирования цветовой палитры. Очередной цвет палитры следует фиксировать на форме при каждом передвижении вдоль ребра на 1 единицу. В результате в сформированной палитре будет 1276 цветов: 256×5-4.

Рисунок 1 – Цветовой куб и обход его ребер   
для формирования цветовой палитры

0

R

G

B

255

255

255

0

R

G

B

255

255

255

Для создания палитры и вывода информации о яркости пикселей используется окружение браузера, а также HTML, CSS и язык программирования JavaScript.

**Код проекта:**

1. файл index.html

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta http-equiv="X-UA-Compatible" content="ie=edge" />

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

<title></title>

</head>

<body>

<div class="container" id="cont">

<div style="background-color: rgb(0,0,0); width: 6px;"></div>

</div>

<script>

cont.onmouseover = function(e) {

let target = e.target;

let field = document.querySelector(".field");

let maxNum = 16383;

let position = e.clientX;

let percent = position / 999;

let num = +parseInt(percent \* maxNum).toString(2);

if (String(num).length < 14) {

let newStr = "";

for (let i = 0; i <= 14 - String(num).length - 1; i++) {

newStr += 0;

}

num = num.toString();

num = newStr + num;

}

field.innerHTML = `Цвет: ${

target.style.backgroundColor

} (Яркость пикселя: ${(percent \* maxNum).toFixed(

0

)}), 14-битный код: ${num}`;

};

</script>

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

</body>

</html>

1. файл style.css

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body {

  display: flex;

  justify-content: space-between;

}

.container {

  background-color: black;

  height: 200px;

  width: 1000px;

  margin-top: 100px;

  display: flex;

}

.innerdiv {

  width: 20px;

  min-height: 200px;

}

.field {

  align-self: center;

  border: 3px solid black;

  border-radius: 20px;

  margin-top: 100px;

  margin-right: 20px;

  padding: 10px;

  width: 320px;

  height: 200px;

  font-size: 20px;

}

1. файл script.js

function componentToHex(*c*) {

  var hex = c.toString(16);

  return hex.length == 1 ? "0" + hex : hex;

}

function rgbToHex(*r*, *g*, *b*) {

  return "#" + componentToHex(r) + componentToHex(g) + componentToHex(b);

}

let divs = [];

for (let k = 0; k < 5; k++) {

  for (let i = 0; i <= 255; i++) {

    if (k === 0) {

      let div = document.createElement("div");

      div.style.backgroundColor = rgbToHex(0, 0, i);

      div.classList.add("innerdiv");

      divs.push(div);

      continue;

    } else if (k === 1) {

      let div = document.createElement("div");

      div.style.backgroundColor = rgbToHex(0, i, 255);

      div.classList.add("innerdiv");

      divs.push(div);

      continue;

    } else if (k === 2) {

      let div = document.createElement("div");

      div.style.backgroundColor = rgbToHex(0, 255, 255 - i);

      div.classList.add("innerdiv");

      divs.push(div);

      continue;

    } else if (k === 3) {

      let div = document.createElement("div");

      div.style.backgroundColor = rgbToHex(i, 255, 0);

      div.classList.add("innerdiv");

      divs.push(div);

      continue;

    } else if (k === 4) {

      let div = document.createElement("div");

      div.style.backgroundColor = rgbToHex(255, 255 - i, 0);

      div.classList.add("innerdiv");

      divs.push(div);

      continue;

    }

  }

}

for (let i = 0; i < divs.length; i++) {

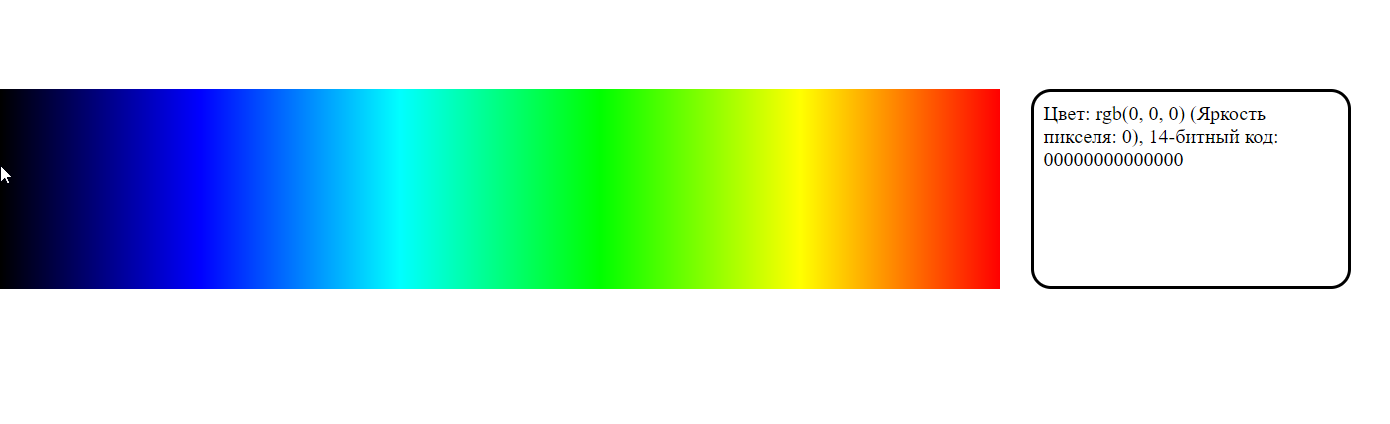
  document.querySelector(".container").appendChild(divs[i]);

}

let field = document.createElement("div");

field.classList.add("field");

document.body.appendChild(field);

**Результаты:**

