

# MMP WS2016 / 2017 - Video Übung 1

---

- Robin Mehlitz (857946)
- Tom Oberhauser (859851)

## Änderungen

---

### Aufgabe 1

#### Aufgabe 1a

Änderungen der Videoquellen in `color.html` von

```
<source src="../../assets/v2b.webm" type="video/webm" />
<source src="../../assets/v2b.mp4" type="video/mp4" />
```

zu

```
<source src="../../assets/TV-100s-2110.podm.h264.webm" type="video/webm" />
<source src="../../assets/TV-100s-2110.podm.h264.mp4" type="video/mp4" />
```

#### Aufgabe 1b

Einbinden von zusätzlichen Reglern in `color.html`

```
Brightness R: <input id="color_offset_r" type="range" min="-100" max="100"
onchange="processor.change_color_offset()"/>
Brightness G: <input id="color_offset_g" type="range" min="-100" max="100"
onchange="processor.change_color_offset()"/>
Brightness B: <input id="color_offset_b" type="range" min="-100" max="100"
onchange="processor.change_color_offset()"/>
```

und Erweiterung der `processor.js`

```
...
var config = {
  color_offset: 0,
  color_offset_r: 0,
  color_offset_g: 0,
  color_offset_b: 0,
  grayscale: false,
  sepia: false
};
```

```

var processor = {

    // "brightness" effect: change_color_offset: handled by html <input id="color_offset" ...
    change_color_offset: function() {
        // get color config value
        config.color_offset = document.getElementById("color_offset").value;
        this.log("color offset = "+config.color_offset);
        config.color_offset_r = document.getElementById("color_offset_r").value;
        this.log("color offset_r = "+config.color_offset_r);
        config.color_offset_g = document.getElementById("color_offset_g").value;
        this.log("color offset_g = "+config.color_offset_g);
        config.color_offset_b = document.getElementById("color_offset_b").value;
        this.log("color offset_b = "+config.color_offset_b);
    },
    ...
    computeFrame: function() {
        ...
        offset_r = parseInt(config.color_offset_r);
        offset_g = parseInt(config.color_offset_g);
        offset_b = parseInt(config.color_offset_b);
        ...
        r = r + offset + offset_r;
        g = g + offset + offset_g;
        b = b + offset + offset_b;
    }
}

```

## Aufgabe 2

Regler in `color.html` einbinden

```

b/w: <input id="grayscale" type="checkbox" onchange="processor.change_color_filters()"/> <br/>
sepia: <input id="sepia" type="checkbox" onchange="processor.change_color_filters()"/><br/>

```

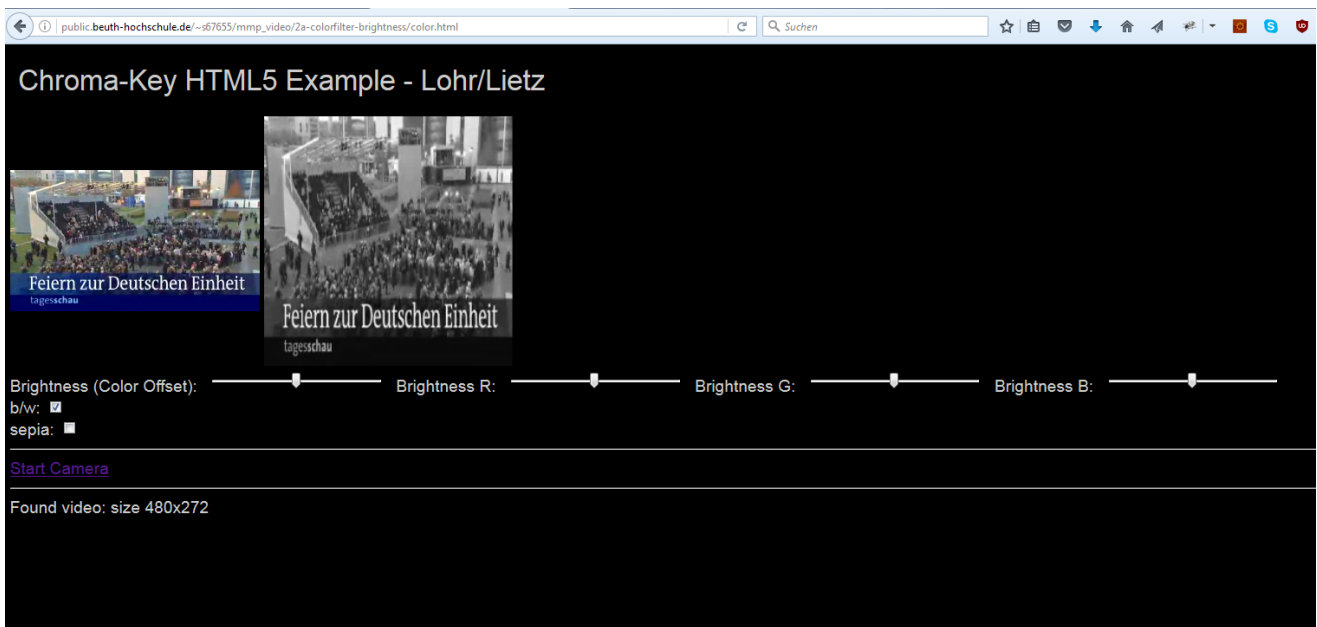
## Aufgabe 2a

Graustufe durch gewichtete RGB Summe ermitteln

```

if (config.grayscale) {
    var Y = 0.3 * r + 0.59 * g + 0.11 * b;
    r = Y;
    g = Y;
    b = Y;
}

```



chroma key bw

## Aufgabe 2b

```
if (config.sepia) {
// https://www.cs.utexas.edu/~scottm/cs324e/Assignments/A5_Images.htm
var R = (r * 0.393) + (g * 0.769) + (b * 0.189)
var G = (r * 0.349) + (g * 0.686) + (b * 0.168)
var B = (r * 0.272) + (g * 0.534) + (b * 0.131)
r = R;
g = G;
b = B;
}
```



chroma key sepia

## Aufgabe 3

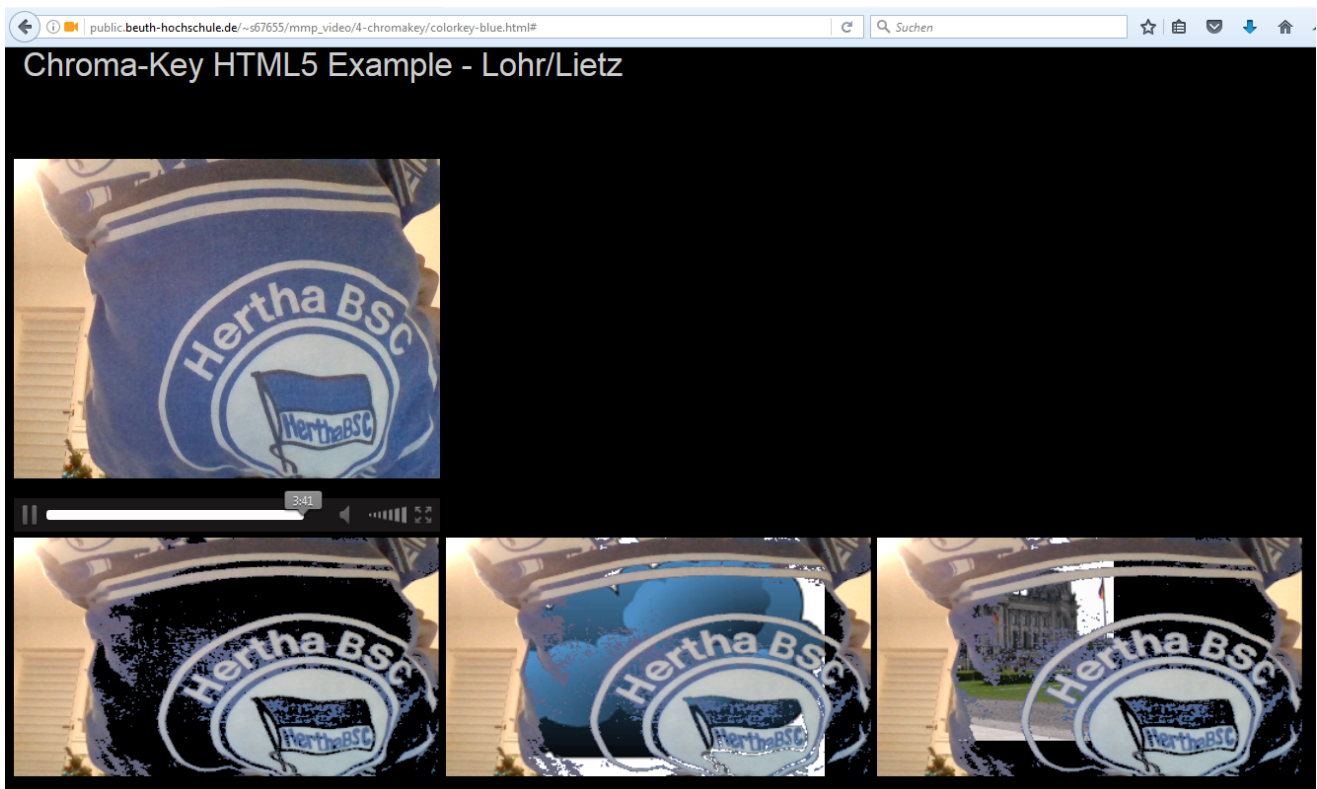
- Kopie der colorkey-green.html zu colorkey-blue.html

- Kopie der `processor.js` zu `processor-blue.js`
- Anpassung des gesuchten Farbwertes in `processor-blue.js`

```
if (r < 100 && g < 100 && b > 100) {
  // set "alpha" value to (0)
  frame.data[i * 4 + 3] = 0;
}
```

## Aufgabe 4

Hierfür musste nur noch die `camera.js` ins Projekt `2a-colorfilter-brightness` kopiert werden. Der Rest war bereits implementiert.



color key blue livecam