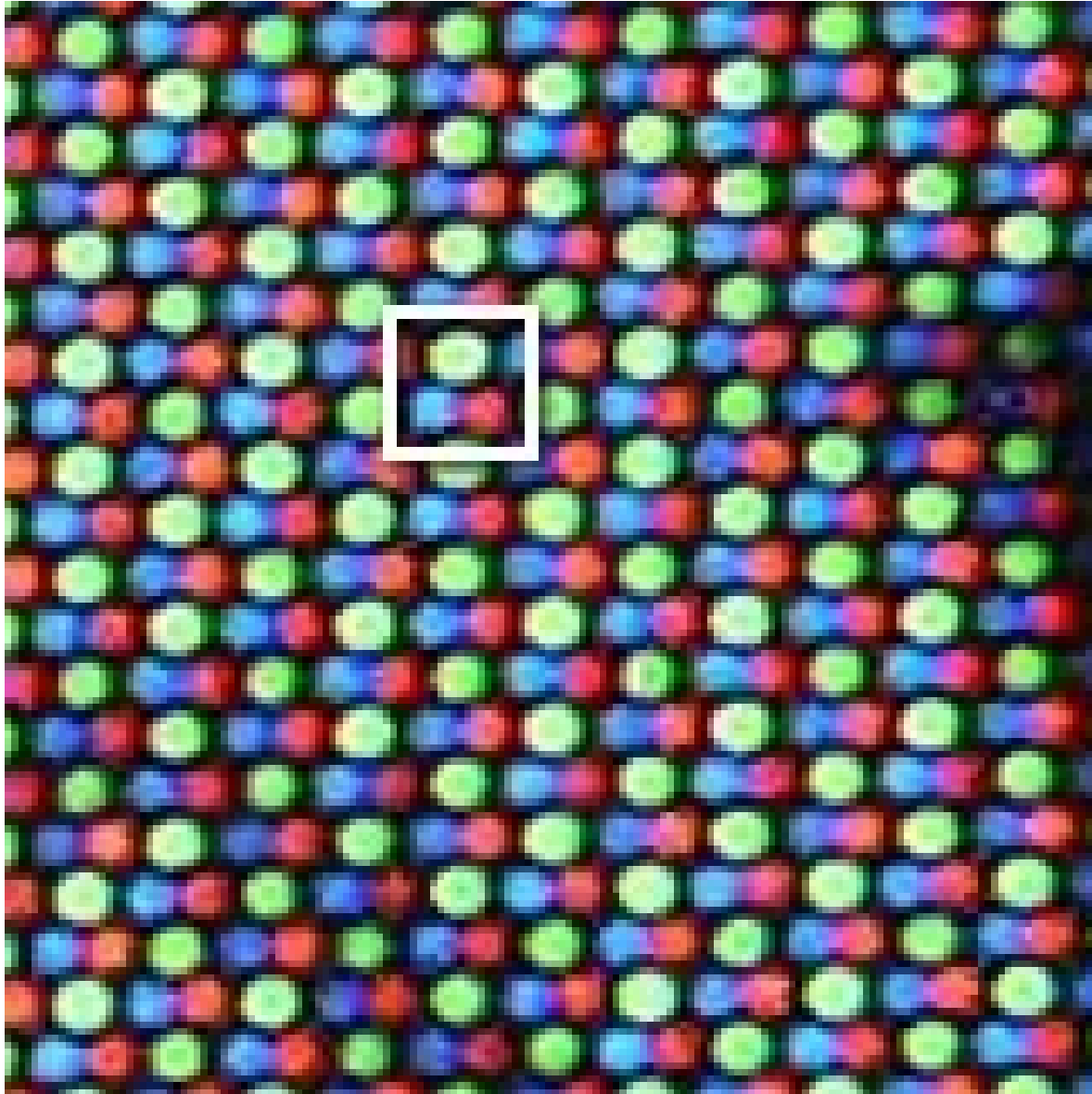


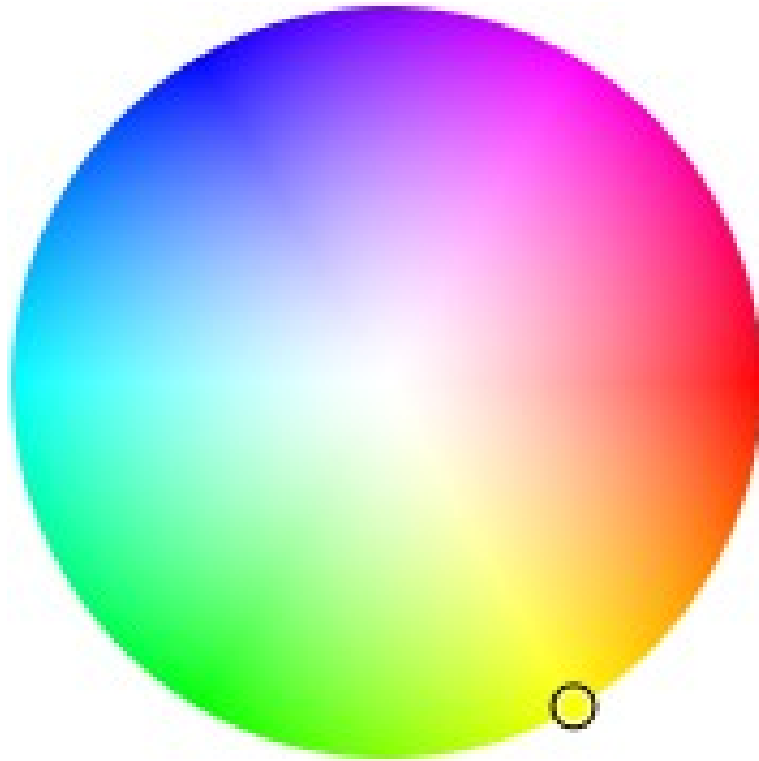
OOP mit Java

While-Schleife (2)
mit
Variable

Pixel auf dem Bildschirm



RGB-Farben



RGB

R:	<input type="range" value="255"/>	<input type="text" value="255"/>
G:	<input type="range" value="255"/>	<input type="text" value="255"/>
B:	<input type="range" value="0"/>	<input type="text" value="0"/>

Jede am Bildschirm darstellbare Farbe hat einen **Rot-**, **Grün-** und **Blau-**Anteil:

0 (dunkel) → 255 (hell)

Bsp. Gelb: Rot = 255, Grün = 255, Blau = 0

Farben in Java

```
/**  
 * Färbt das GameWindow gelb  
 */  
void main()  
{  
    Color farbe;  
    farbe = new Color(1f, 1f, 0f);  
  
    window.clear(farbe);  
    window.paintFrame();  
}
```

Die Klasse Color

Notwendig: Import des **AWT**-Package
(Advanced Window Toolkit)

```
import java.awt.*;
```

Wir verwenden diesen **Konstruktor**:

```
Color(float r, float g, float b)
```

r, g, b sind Dezimalzahlen („float“)

0.0 = dunkel, 1.0 = hell

Beispiel mit while-Schleife

```
void main()  
{  
    Color farbe;           // Variablen  
    float r, g, b;  
    int zähler;  
  
    r = 0.0f;              // Anfangswerte  
    g = 0.0f;              // (float-Werte  
    b = 0.0f;              // mit f am Ende)  
  
    zähler = 0;  
    ...
```

Beispiel mit while-Schleife

```
...  
while (zähler < 100)  
{  
    farbe = new Color(r, g, b);  
    window.clear(farbe);  
    window.paintFrame();  
  
    r = r + 0.01f;  
    g = g + 0.01f;  
    b = b + 0.01f;  
    zähler = zähler + 1;  
}
```

Vereinfachende Syntax

```
...  
while (zähler < 100)  
{  
    farbe = new Color(r, g, b);  
    window.clear(farbe);  
    window.paintFrame();  
  
    r += 0.01f;  
    g += 0.01f;  
    b += 0.01f;  
    zähler++;  
}
```


Vereinfachende Syntax

Operator +=

`r += 0.01f;` erhöht r um 0,1

Operator ++

`zähler++;` erhöht zähler um 1
(nur für Integer)

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
Freigegeben unter CC BY-NC-SA 4.0, März 2021



Grafiken:

- Vergrößerte Pixel:
de.wikipedia.org, Lizenz CC BY-SA 3.0