

Hintergrundtausch



Die Methode „hintergrundtausch“ ist etwas anspruchsvoller als die Methoden, die ihr bisher implementiert habt. Das macht sie besonders geeignet dafür, etwas dabei zu lernen. (Arraygrenzen, Sinnvolles Testen, Toleranzen bei Farbwerten usw.) Damit alle im Team die Gelegenheit haben, ihre Fähigkeiten zu verbessern, ist vorgesehen, dass jede Zweiergruppe im Team diese Methode implementiert.

Hintergrundtausch



1. Habt ihr euch schon einmal gefragt, wie in Filmen Schauspieler in alle möglichen, vielleicht sogar computergenerierten Welten kommen? Das Prinzip ist einfach, was irgendwie zum Nachmachen anregt. Fotos von euch in einer anderen Welt oder in einem anderen Jahrhundert? Kein Problem. Implementiert einfach eine Methode, die den Hintergrund eines Bildes austauscht.

Der Methodenrumpf ist auch hierfür bereits vorhanden:

```
public static Pixel[][] hintergrundtausch(Pixel[][] pixelmap1,  
                                           Pixel[][] pixelmap2) {  
  
    return pixelmap1;  
}
```

Hintergrundtausch

Beachtet folgende Hinweise:

- a) Ihr braucht ein Foto von euch vor dem Greenscreen, um diese Methode zu testen.
 - b) Das Array `pixelmap1` ist das Bild, bei dem der Hintergrund ausgetauscht werden soll. `Pixelmap2` ist der neue Hintergrund.
 - c) Ihr braucht einen Farbwert, den ihr als Hintergrundfarbe definiert, nehmt einfach den Pixel in der (0,0) Ecke, also links oben. Speichert die Farbwerte als erstes in 3 `int`-Variablen ab.
 - d) Bilder können unterschiedlich groß sein. Damit das Programm nicht abstürzt, sollten die Schleifen nur bis zum Ende des kleineren Arrays laufen. Das Minimum der beiden Längen von 2 Arrays erhaltet ihr auf diese Weise:

```
int minBreite = Math.min(pixelmap1.length, pixelmap2.length);
```

Dasselbe müsst ihr für die Höhe der Bilder durchführen.
Bevor ihr weitermacht: Stürzt euer Programm ab, wenn ihr die Schleifen ohne Inhalt laufen lasst? Testet das erst mal!
 - e) Vergleicht jetzt in der Schleife jeweils den Farbwert des aktuellen Pixels mit dem Referenzwert, den ihr gespeichert habt. Stimmen die Werte überein, dann überschreibt die Farbwerte des aktuellen Pixels.
 - f) Die Farbwerte der einzelnen Pixel des Bluescreens weichen voneinander ab. Erzeugt eine Integer Variable „toleranz“. Farbwerte des `pixelarray1` sollen ausgetauscht werden, wenn alle 3 Farbkanäle nicht mehr als „toleranz“ von dem Referenzfarbwert abweichen.
- Tipp:** Den Abstand zwischen zwei Zahlen (a, b), erhält man mit `Math.abs(a, b)`
- g) Auch diese Methode solltet ihr ausgiebig testen.

Hier könnt ihr euch etwas notieren, Skizzen davon machen, wie die Pixel umsortiert werden sollen oder Aufzeichnen, welche Probleme bei Bildern unterschiedlicher Größe auftreten können.

Ab jetzt werdet ihr erst einmal ohne Arbeitsblätter weiter machen. Ihr könnt selbst entscheiden, wie ihr weiter macht um Material für euer Video zu erstellen. Überlegt euch ab jetzt eigene Bildbearbeitungen, implementiert diese und erzeugt Bilder für euer Video. In etwa einer Stunde geht es dann weiter mit der Bearbeitung von Sounds.