

# Aufgabe 4: Würfelglück

Team-ID: ????

Team-Name: 234bcd

Bearbeiter\*innen dieser Aufgabe:  
Michael Köhler

19. September 2021

## Inhaltsverzeichnis

<b>1</b>	<b>Lösungsidee</b>	<b>1</b>
<b>2</b>	<b>Umsetzung</b>	<b>1</b>
<b>3</b>	<b>Beispiele</b>	<b>1</b>
<b>4</b>	<b>Quellcode</b>	<b>2</b>

## 1 Lösungsidee

Für diese Aufgabe muss man wissen, dass Digitalbilder aus vielen Bildpunkten (Pixeln) bestehen, die jeweils eine bestimmte Farbe haben. Wenn man ein Digitalbild ausreichend vergrößert betrachtet, kann man die einzelnen Pixel erkennen. Hier ist ein vergrößerter Ausschnitt aus einem der Beispielfür diese Aufgabe (Kontrast ein wenig erhöht): Bei genauem Hinsehen fällt auf, dass im oberen Bereich des Ausschnittes jedes Pixel eine leicht andere Farbe hat, während unten immer vier Pixel die gleiche Farbe haben. Dies sind die Rhinozelfantenschuppen.

## 2 Umsetzung

Die Lösungsidee wird in Python implementiert. Die Python Imaging Library (PIL) stellt viele Funktionen zur Bildverarbeitung zur Verfügung. Damit funktionieren das Öffnen und Speichern des Bildes und der Zugriff auf die Pixeldaten sehr einfach. Wir importieren dazu das Modul Image der PIL. Mithilfe zweier ineinander geschachtelter For-Schleifen werden alle Pixel einzeln betrachtet. Immer wenn ein Pixel die gleiche Farbe hat wie eines seiner Nachbapixel, färben wir beide Pixel weiß. Da bei dieser Aufgabe alle Pixel weiß gefärbt werden sollen, die zu einem Rhinozelfant gehören könnten, müssen wir aufpassen, dass wir nicht direkt ein Pixel im Bild weiß färben, wenn wir sehen, dass es einen gleichfarbigen Nachbarn gibt. Sonst kann es passieren, dass wir bei den anderen benachbarten Pixeln nicht mehr wissen, welche Farbe das aktuelle Pixel ursprünglich hatte. Dieses Problem wird gelöst, indem nicht die Pixel im Originalbild weiß gefärbt werden, sondern in einer Kopie des Bildes ('ausgabebild'). Dadurch können wir im Originalbild immer alle Pixel in ihrer ursprünglichen Farbe vergleichen. Zu guter Letzt wird das Ausgabebild wieder in eine Datei gespeichert.

## 3 Beispiele

Wir rufen das Programm für zwei der Beispieldateien auf und zeigen jeweils das resultierende Bild in verkleinerter Darstellung:

## 4 Quellcode

Beispiel-Code:

---

```
1 static void Main(string[] args)
2     {
3     //test Comment!
4         List<Player> players = SetPlayers(args);
5         List<MatchUps> matchups = CreateMatchups(players);
6         List<GameResults> gameresults = PlayGames(matchups);
7         RankPlayers(gameresults);
8         Console.WriteLine("Test string");
9     }
10 }
```

---

Program.cs:

---

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4
5 namespace DiceCompare
6 {
7     class Program
8     {
9         static void Main(string[] args)
10        {
11            List<Player> players = SetPlayers(args);
12            List<MatchUp> matchups = CreateMatchups(players);
13            List<GameResult> gameresults = PlayGames(matchups);
14            RankPlayers(gameresults);
15        }
16
17        private static void RankPlayers(List<GameResult> gameresults)
18        {
19            throw new NotImplementedException();
20        }
21
22        private static List<GameResult> PlayGames(List<MatchUp> matchups)
23        {
24            throw new NotImplementedException();
25        }
26
27        private static List<MatchUp> CreateMatchups(List<Player> players)
28        {
29            throw new NotImplementedException();
30        }
31
32        private static List<Player> SetPlayers(string[] args)
33        {
34            throw new NotImplementedException();
35        }
36    }
37 }
38 }
```

---