

## Projekt: Portrait als Baustein-Mosaik

### Projektidee

Eine Firma, die bunte Bausteine für Kinder verkauft, bietet seit einiger Zeit in ihren Geschäften die Möglichkeit, ein Selbstportrait aufzunehmen und dieses als Mosaik aus Bausteinen zusammenzusetzen. Das Foto wird dazu in eine Vorlage aus ca. 4500 Bausteinen in den Farben Weiß, Hellgrau, Dunkelgrau und Schwarz sowie einer Hintergrundfarbe (hier Gelb) umgewandelt. Abbildung 1 zeigt ein Beispiel für eine Bauvorlage.

Die Firma behauptet, das Erstellen der Vorlage geschehe mit ein bisschen Technologie und ganz viel Magie. Mit Ihrem Wissen über Algorithmik und die Codierung von Farben sind Sie jedoch sicher in der Lage, diese Zauberei zu durchschauen und selbst ein Programm zu erstellen, das aus einem Foto eine entsprechende Vorlage erstellt.



Abbildung 1: Vorlage für das Erstellen eines Baustein-Mosaiks

**Aufgabe 1:** Haben Sie schon erste Ideen, wie der Zaubertrick funktioniert? Sammeln Sie algorithmische Ansätze für das Erstellen eines Baustein-Mosaiks aus einem Portraitfoto. Welche Voraussetzungen sollte das Originalbild erfüllen?

## Grundidee

Wir verwenden die Programmierumgebung Snap!<sup>1</sup>, da sie uns geeignete Blöcke zur Verfügung stellt, um auf die Farbwerte der Pixel eines Bildes zuzugreifen. Das Foto, das in ein Mosaik umgewandelt werden soll, verwenden wir als Bühne. Außerdem benötigen wir ein Baustein-Objekt, das über fünf verschiedene Kostüme in den Farben des Mosaiks verfügt. Unser Algorithmus muss dann dafür sorgen, dass das Baustein-Objekt die Bühne systematisch abläuft und jeweils einen Abdruck in der passenden Steinfarbe hinterlässt.

## Vorbereitung

Bevor Sie mit dem Programmieren beginnen können, benötigen Sie ein geeignetes Portrait als Hintergrundbild und Kostüme für das Baustein-Objekt. Erledigen Sie deshalb als erstes mithilfe eines Bildbearbeitungsprogramms die Vorbereitungen in Aufgabe 2.

### Aufgabe 2:

- Nehmen Sie ein Selbstportrait von sich vor einem einfarbigen Hintergrund auf. Überlegen Sie, welche Hintergrundfarbe sich besonders gut eignet.
- Erstellen Sie mithilfe eines Bildbearbeitungsprogramms einen quadratischen Bildausschnitt Ihres Portraits. Wählen Sie geeignete Werte für die Breite und Höhe in Pixeln. Damit die Bearbeitung in Snap! nicht zu lange dauert, sollte das Bild nicht zu groß sein.
- Stellen Sie die Größe der Bühne in Snap! auf die Größe des Bildes ein und importieren Sie Ihr Bild als Hintergrund für die Bühne.
- Wie groß sollen passend dazu die Bausteine sein? Entwerfen Sie mithilfe eines Bildbearbeitungsprogramms fünf gleich große, quadratische Kostüme in den Farben Weiß, Hellgrau, Dunkelgrau und Schwarz sowie der Hintergrundfarbe.
- Importieren Sie die Kostüme für das Baustein-Objekt.

## Algorithmus zum Ablaufen des Bildes

Bevor wir anfangen, die Vorlage zu erstellen, benötigen wir einen Algorithmus für das Baustein-Objekt, so dass es die Bühne einmal vollständig abläuft. In Abbildung 2 stellt das gelbe Quadrat das Baustein-Objekt dar. Dieses muss nun jedes Feld des Gitters einmal besuchen.

### Aufgabe 3:

- Skizzieren Sie in Abbildung 2 mithilfe von Pfeilen, in welcher Reihenfolge Ihr Objekt die Felder ablaufen soll.
- Erstellen Sie ein entsprechendes Skript für das Baustein-Objekt. Hinterlassen Sie dabei an jeder Position einen Abdruck, um besser beobachten zu können, ob die gesamte Bühne abgelaufen wird. Die Skizze aus Abbildung 2 können Sie verwenden, um benötigte Parameter des Baustein-Objektes oder der Bühne zu notieren.

**Hinweis 1:** Wenn Sie mit festen Werten arbeiten, reichen die dunkelblauen Blöcke aus dem Bereich *Bewegung* aus. Möchten Sie Ihr Skript flexibel gestalten, so dass es auch für andere

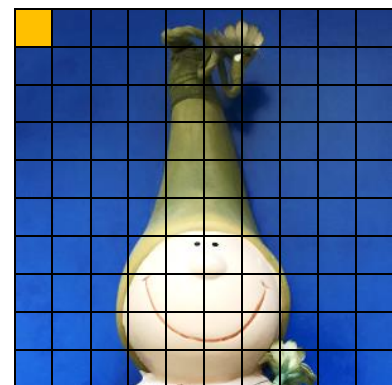


Abbildung 2: Raster für die Zerlegung eines Bildes in Bausteine

<sup>1</sup> Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

Größen der Bühne oder der Bausteinkostüme funktioniert, finden Sie im Bereich *Fühlen* weitere hilfreiche Blöcke. Einige davon zeigt Tabelle 1.











Blöcke	Mögliche Einstellungen	Bedeutung
	 	Mit diesem Block lassen sich Höhe und Breite eines Objekts herausfinden
	     	Mit diesem Block lassen sich die Koordinaten der Bühnenränder sowie Höhe und Breite der Bühne abfragen.

Tabelle 1: Blöcke zum Abfragen von Eigenschaften eines Objektes bzw. der Bühne

**Hilfe:** Wenn Sie nicht weiterkommen, finden Sie in den Dateien *ABO\_Aufg3\_Hilfe1*, *ABO\_Aufg3\_Hilfe2* und *ABO\_Aufg3\_Hilfe3* schrittweise Hilfen für einen möglichen Ansatz.

**Hinweis 2:** Um einen Abdruck eines Kostüms zu hinterlassen, kann der Block *stemple* aus dem Bereich *Stift* verwendet werden.

## Baustein-Mosaik erstellen

Anstatt das Original des Bühnenbildes zu verändern, übermalen wir es einfach mit Abdrücken eines Kostüms des Baustein-Objektes in der jeweils passenden Farbe<sup>2</sup>. Das so entstandene Bild können wir dann mit einem Rechtsklick auf die Bühne abspeichern.

Das Skript aus Aufgabe 3 muss also so angepasst werden, dass an jeder Position ein zum Foto passendes Kostüm ausgewählt wird, bevor ein Abdruck hinterlassen wird. Dazu ist Ihr Wissen über die Codierung und Darstellung von Farben im RGB-Modell gefragt.

Im Bereich *Fühlen* steht der Block *Farbton bei ...* zur Verfügung (s. Abbildung 4). Dieser kann so eingestellt werden, dass er eine Liste mit dem Rot-, dem Grün und dem Blauwert des Bühnenbildes an der aktuellen Position des aufrufenden Objektes zurückgibt.





Abbildung 4: Block *Farbton bei* (oben) mit passender Einstellung zur Abfrage des aktuellen RGB-Wertes (unten).

### Aufgabe 4:

- Überlegen Sie Kriterien, nach denen Sie abhängig vom Rot-, Grün- und Blauwert das passende Kostüm für das Baustein-Objekt auswählen können.

<sup>2</sup> Beim Hinterlassen eines Abdrucks werden die äußersten Pixel des Kostüms teilweise heller dargestellt als der Rest. Dadurch ergeben sich die hellen Streifen in Abbildung 1. Da wir die einzelnen Bausteine in der Vorlage dadurch sogar besser unterscheiden können, ist das aber unproblematisch.

- b) Erweitern Sie Ihr Skript so, dass beim Durchlaufen der Bühne an jeder Position das passende Kostüm ausgewählt und ein Abdruck hinterlassen wird.

**Aufgabe 5:** Um das Mosaik aus Bausteinen zusammenzusetzen, benötigt man entsprechend viele Bausteine der jeweiligen Farbe. Erweitern Sie Ihr Programm so, dass es beim Erstellen der Vorlage für jede Farbe zählt, wie viele Bausteine benötigt werden.

### Optimierung des Ergebnisses

Das Ergebnis ist nicht nur abhängig von Ihrem Algorithmus, sondern auch von Ihrem Ausgangsbild. Möglicherweise ist es daher sinnvoll, das Ausgangsbild zuvor mit einem Bildbearbeitungsprogramm vorzubereiten. Neben dem Verändern von Helligkeit und Kontrast kann es auch hilfreich sein, den Hintergrund einheitlich zu färben. Wenn Sie beispielsweise einen blauen Hintergrund gewählt haben und die Person blaue Augen hat, so ist es schwer anhand der RGB-Werte zwischen dem Hintergrund und der Augenfarbe zu unterscheiden, um das passende Kostüm auszuwählen. Ein fotografiertes Hintergrund weist außerdem immer Schattierungen auf, so dass die RGB-Werte von Position zu Position leicht variieren. Wenn wir den Hintergrund in einer Farbe färben, die ansonsten nicht im Bild vorkommt, können wir in unserem Algorithmus anhand dieses einen RGB-Wertes einen Bildpunkt eindeutig dem Hintergrund zuordnen.

Des Weiteren können Sie überprüfen, ob Sie eine passende Zuordnung der Helligkeitswerte zu den Bausteinfarben gewählt haben und ob die Größe der Bausteine in einem guten Verhältnis zur Größe des Bildes steht.

Um die einzelnen Bausteine in der Vorlage besser erkennen und tatsächlich danach bauen zu können, sind Bausteinkostüme mit einer Umrandung hilfreich. In Abbildung 5 wurden beispielsweise grün umrandete Bausteinkostüme verwendet.

Wenn Sie schon einmal einen Eindruck von dem Mosaik aus Bausteinen erhalten möchten, können Sie statt der einfarbigen Kostüme auch Fotos von Bausteinen in den verschiedenen Farben verwenden. Ein Beispiel zeigt Abbildung 6.

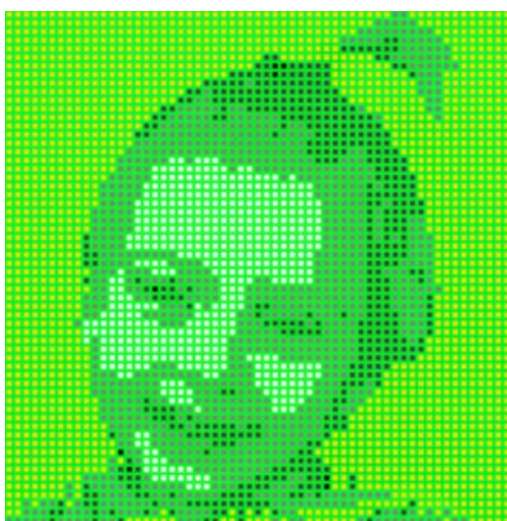


Abbildung 5: Umrandung der Bausteine zur besseren Unterscheidung der Bausteine.

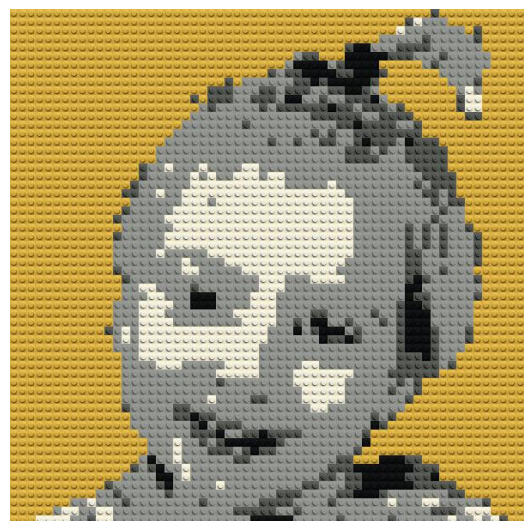
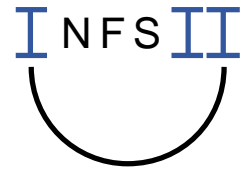


Abbildung 6: Mosaik aus Fotos von Bausteinen als Vorschau

**Aufgabe 3:** Erweitern und optimieren Sie Ihr Programm. Dazu können Sie die obigen Anregungen nutzen. Sicher haben Sie aber auch noch eigene Ideen.



Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Von der Lizenz ausgenommen ist das InfSII-Logo.

**Bildnachweis:** Die Abbildungen wurden von der Autorin selbst erstellt. Die abgebildeten Blöcke wurden der Entwicklungsumgebung Snap! in der Version 7.2.5 entnommen. Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

Für die korrekte Ausführbarkeit der beiliegenden Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.