

Hochschule Trier
Trier University of Applied Sciences

Dokumentation

Mosaic-Generator

Vorgelegt von:	Maik Thielen
Matrikelnummer:	956422
Studiengang:	Informatik - DMS
Fach:	Tool- und Pluginprogrammierung
Betreuer:	Prof. Dr. Christof Rezk-Salama
Ort, Abgabedatum:	Mettendorf, 31.08.2016

Einleitung und Problemstellung

Der Mosaic-Generator ist eine Software, mit dessen Hilfe Mosaikbilder erzeugt und abgespeichert werden können.

Unter einem Mosaik im Allgemeinen versteht man ein Bild oder ähnliches, welches sich aus kleineren, verschiedenfarbigen Elementen zusammensetzt.

In dieser Software ist ein Mosaik ein Bild, welches aus einer Vielzahl von kleineren Bildern besteht. Im Folgenden und innerhalb der Software werden diese als Tiles (Bildkacheln) betitelt. Das zu generierende Mosaikbildes wird anhand eines zuvor ausgewählten Bildes (Target-Image) erstellt.

Nicht immer führt die Erzeugung eines solchen Mosaikbildes zu einem guten Ergebnis, da einige Faktoren das finale Bild beeinflussen. Die Größe der benutzten Tiles müssen z.B. im Einklang mit dem zu generierendem Bild stehen.

Sind die Tiles zu groß, erkennt man das ursprüngliche Bild nur dann heraus, wenn das Target-Image groß genug ist und somit viele Source-Tiles verarbeitet werden konnten..

Sind die Tiles zu klein, erkennt man die kleinen Teilbilder nicht mehr heraus und das Resultat erinnert eher an einen Schattierungseffekt als an ein Mosaikbild.

Um häufige schlechtere Ergebnisse zu vermeiden, versucht die Software einen guten Mittelweg zwischen Möglichkeiten und Benutzerfreundlichkeit zu finden.

So gibt es z.B. zwei verschiedene Ansammlungen von Tiles. Eine Beispielansammlung, die aus rund 1700 verschiedenen Gamecoverbildern besteht, und eine eigene, die der Endnutzer selber anlegen kann. Die Beispielsammlung ist nicht skalierbar, heißt die Tiles haben zwei verschiedene Größen, mit denen Mosaikbilder generiert werden können. Dies soll verhindern dass die Tiles irreparabel skaliert werden und beispielsweise zu unscharf werden. Die eigene Ansammlung von Tiles ist jedoch jederzeit skalierbar und hier steht es in der Eigenverantwortung des Benutzers inwieweit er diese skaliert.

Oft gilt, desto größer das Source-image, desto mehr tiles können verarbeitet werden und umso besser sieht das Ergebnis aus. Jedoch sollte das Bild auch nicht zu groß sein, da es sonst zu einem Speicherfehler kommen kann.

Entwurf

Um das Mosaikbild generieren zu können, benötigt man zunächst viele verschiedene kleinere Elemente, aus denen sich das Mosaikbild erstellen lässt. Da der Mosaik-Generator mit Bildern arbeitet, wird also eine Vielzahl verschiedener kleiner Bilder benötigt. Für dieses Programm ist es notwendig dass diese allesamt dieselbe Auflösung haben und nicht zu groß oder zu klein sind. Umso mehr verschiedene Tiles zur Verfügung stehen, desto besser sieht idR. das Endergebnis aus. Aus diesen Tiles wird der durchschnittliche RGB-Mittelwert (Arithmetischer) berechnet, um diesen später mit anderen RGB-Mittelwerten vergleichen zu können. Um diese Tiles besser von anderen unterscheiden zu können, werden diese als Source-Tiles bezeichnet.

Als Nächstes braucht man ein Zielbild (Target-Image), welches als Vorlage für das Mosaikbild dient. Hier gilt zu beachten, dass das ausgewählte Target-Image eine hohe Auflösung haben sollte. Umso höher die Auflösung dieses Bildes ist, desto mehr Tiles können in dieses Bild verarbeitet werden und umso besser sieht das Endergebnis aus. Anschließend wird das gewählte Bild in viele kleine Bilder aufgeteilt, damit auch hiervon eine Ansammlung von Tiles erstellt werden kann. Wie bei den anderen Tiles wird auch hier nun jeweils der durchschnittliche Farbmittelwert bestimmt.

Nun existieren zwei verschiedene Ansammlungen von Tiles. Mithilfe eines Rasters, oben links beginnend, werden diese nun miteinander verglichen. Die einzelnen Tiles, die das Target-Image repräsentieren, werden mit allen möglichen Source-Tiles verglichen. Nun wird entweder das bestmögliche Resultat ausgewählt oder vorher eine erst eine Vorauswahl aus potentiellen guten Tiles erstellt, aus der dann letztendlich zufällig das endgültige Tilestück ausgewählt wird. Die ausgewählten Tiles werden den Platz der ursprünglichen Tiles aus dem Target-Image einnehmen.

Als letzter Schritt müssen die ausgewählten Ergebnistiles nur noch zu einem Bild zusammengesetzt werden. Das Resultat ist nun ein Mosaikbild, welches dem ursprünglichem source-image nachempfunden wurde.

Logische Representation

Das Target-Image

Die Klasse *Model* ist für die gesamte Logik des Programmes zuständig.

In dieser findet sich natürlich das oben vermerkte Target-Image wieder, welches mithilfe der Funktion ***changeTargetImage(...)*** eingelesen und abgespeichert wird. Zu Beginn des Programms steht dem Benutzer ein Beispielbild zur Verfügung, welches zu Programmstart in das System eingebunden wird. Zu beachten ist, dass das ausgewählte Bild keine utopische Größenordnung haben sollte. Bei zu Großen Bildern kann es sonst zu **Heap-Speicherproblemen** kommen. Wenn das der Fall ist, sollte das Programm über die beigelegte **Bat-Datei** geöffnet werden. Dort wird über Befehl der zur Verfügung gestellte Heap-Speicher erhöht und somit sollte dieses Problem gelöst sein. Desweiteren kann es dennoch zu Problemen kommen, wenn das ausgewählte Bild den Rahmen der *BufferedImage*-Objekte sprengt.

Die Tiles

Neben Breite und Höher der einzelnen Tiles, werden die oben beschriebenen Tiles erzeugt und in verschiedene Listen abgespeichert. Die Klasse ***Tile*** wird verwendet um Tile-Objekte zu erzeugen und beinhaltet das Bild selbst als *BufferedImage* Objekt und hat die jeweiligen RGB-Durchschnittswerte gespeichert. Außerdem werden einige Daten berechnet: Wie viele Tiles gebraucht werden um eine Reihe des Target-Images auszufüllen, wie viele benötigt werden um eine Spalte auszufüllen und letztendlich wie viele Tiles insgesamt benötigt werden.

Die Tilelisten

Es existieren folgende Listen mit Tiles:

```
// Tile lists
private ArrayList<Tile> tileListSourceExample;
private ArrayList<Tile> tileListSourceOwn;
private ArrayList<Tile> tileListSourceOwnCopy;
private ArrayList<Tile> tileListSourceCurrentUsed;

private ArrayList<Tile> tileListTargetImage;
private ArrayList<Tile> tileListResult;
```

Die **tileListSourceExample** beinhaltet eine Reihe von bereits vorhandenen Source-Tiles, die aus einer Auswahl von ungefähr 1700 verschiedenen Bildern von Spielecovern bestehen. Diese haben eine Auflösung von 35x50 Pixel. Diese Tiles sind nicht skalierbar, um sicherzustellen, dass im Programm immer Tiles vorhanden sind, die ein vernünftiges Mosaikbild generieren können.

Source-Tiles die sich skalieren lassen, finden sich in der Liste **tileListSourceOwn** wieder. Hier werden die Tiles abgespeichert, die sich im Ordner "Tiles" befinden und einer vom User vorausgewählten Größenordnung entsprechen. Es handelt sich also um Tiles, die vom User selbst bereitgestellt werden und alternativ zu den Beispieltiles genutzt werden können.

Befüllen der Listen

Um die beiden Source-Listen zu erzeugen, wird die Methode **generate_tile_list_source(...)** genutzt. Genauer sorgt die Methode dafür, dass aus einem übergebenem Pfad Tiles der übergebenen Größe (wenn vorhanden) erzeugt und einer übergebenen Liste hinzugefügt werden.

TileListSourceOwnCopy dient dazu eine Sicherheitskopie von der eigenen Source-Tile-Liste sicherzustellen. Diese wird genutzt um beim Skalieren der eigenen Source-Tiles auf die Basisgröße der Ursprungstiles zugreifen zu können. Dadurch soll ein Qualitätsverlust durch häufiges Skalieren vermieden werden.

Um im Programm vermerken zu können welche Tile-Source-Liste aktuell aktiv ist, wird die aktuell benutzte Liste an die Liste ***tileListSourceCurrentUsed*** referenziert.

In den letzten beiden Listen ***tileListTargetImage*** sowie ***tileListResult*** landen, wie der Name schon vermuten lässt, einmal die Tiles, die aus dem Target-Image generiert wurden und einmal jene, die das finale Mosaikbild zusammensetzen.

Um die ***tileListTargetImage*** zu befüllen, wird die Methode ***generate_tile_list_targetImage()*** verwendet. Diese erstellt, abhängig der Tile-Größe, aus dem Target-Image so viele Tiles wie nötig und speichert diese dann in der Liste ab.

Generate_tile_list_resultImage() sorgt für die Befüllung der ***tileListResult***, indem die ***tileListTargetImage*** mit der ***tileListSourceCurrentUsed*** verglichen wird. Nun wird, abhängig von der Einstellung, entweder die Methode ***compare_tile_with_list_and_returns_best_match(...)*** oder die Methode ***compare_tile_with_list_and_returns_random_good_match(...)*** eingesetzt. Es wird jedes Tile der ***tileListTargetImage*** mit allen Tiles der ***tileListSourceCurrentUsed*** verglichen. Das Tile, aus der ***tileListTargetImage***, welches dem Vergleichstile aus der anderen Liste am ähnlichsten ist (Abhängig der RGB-Farbwerte), wird als bester Treffer in die ***tileListResult*** hinzugefügt. Alternativ wird über die andere Methode stattdessen zuerst eine Vielzahl von guten Treffern in eine Zwischenliste gespeichert, von denen dann eines zufällig ausgewählt wird. Hier gibt es einen Tolleranzwert ***maxRGBDifference*** der Information darüber gibt, wie hoch der farbliche Durchschnittswert maximal sein darf. Standardmäßig ist dieser Wert auf 100 gesetzt, er lässt sich jedoch nachträglich zwischen 50 bis 150 einstellen.

Die ***tileListResult*** enthält also die Tiles, die ausgewählt wurden, um das Target-Image nachbauen zu können.

Mithilfe der Methode ***generate_mosaic()*** wird das ganze ins Rollen gebracht. Zunächst wird, wenn nötig, das Target-Image skaliert. Das passiert dann, wenn das Ursprungsbild nicht durch die Dimension der Tiles teilbar ist (Damit kein Rest besteht). Anschließend werden dann die beiden Methoden ***generate_tile_list_targetImage()*** und ***generate_tile_list_resultImage()*** ausgeführt. Zum Schluss wird ein neues BufferedImage-Objekt erzeugt und die Bilder der Tiles aus der ***tileListResult*** werden in dieses Bild gemalt. Dadurch entsteht das fertige Mosaikbild, welches nun abgespeichert werden kann.

Bedienung der Software

Bei der Entwicklung des Mosaic-Generators war mir wichtig, dass die Software leicht zu bedienen ist. Bei der Arbeit mit Bildern und Tiles kann es schnell zur Verwirrung kommen, vor allem dann, wenn eine Software viele Einstellungsmöglichkeiten bereithält. Der Mosaic-Generator soll jedoch schnell und einfach dabei abhilfe schaffen ein Mosaikbild zu generieren.

Hauptansicht

Aus diesem Grund ist die Hauptansicht minimal gehalten und enthält nur das Wesentliche: Das aktuelle Bild und einen Button zum generieren des Mosaikbildes. Da dank Beispiel immer ein Target-Bild vorhanden ist, kann so jederzeit über einen Klick auf den Button ein Mosaik generiert werden.

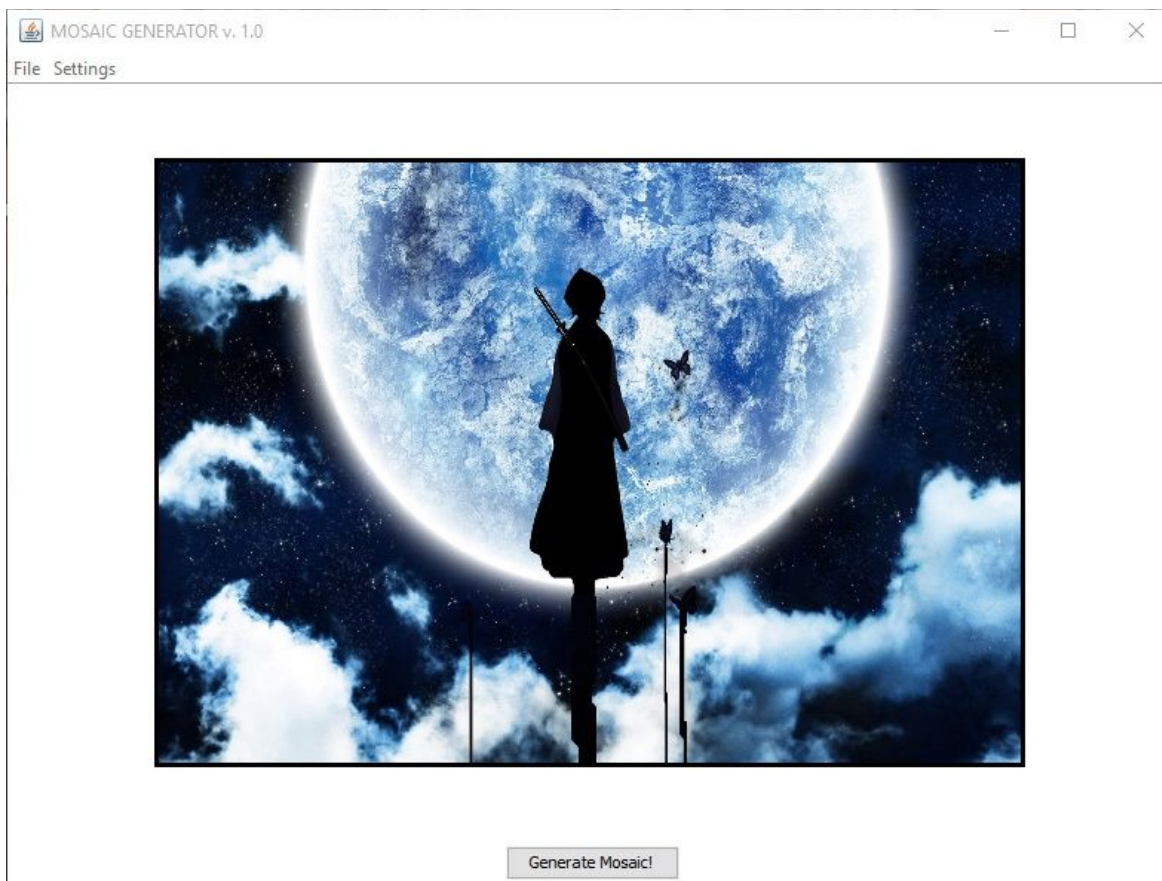


Bild: Hauptansicht mit Beispiel Target-Image

Diese Vorgehensweise wurde bewusst gewählt um das Bild möglichst groß darstellen zu können, ohne auf Platz aufgrund von weiteren Buttons oder anderen Einstellungswerkzeugen wie Slider oder Checkboxes verzichten zu müssen.

Menüleiste

Datei und Datenverwaltung wurde auf eine Übersichtliche Menübar ausgelagert. Hier sind zwei Menüpunkte vorhanden: **File** und **Settings**.

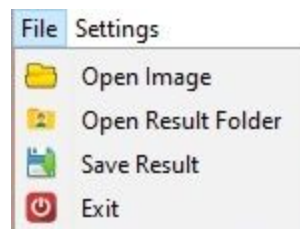


Bild: Menüpunkt File

Die Unterpunkte aus **File** sind alle Selbsterklärend. Über **Open Image** lässt sich über einen geführten Dialog ein neues Target-Image auswählen und in das Programm einbinden. **Open Result Folder** öffnet den Ordner indem die generierten Mosaikbilder aufzufinden sind (Results). **Save Result** öffnet einen Dialog zum Absichern des letzten generierten Mosaikbildes. Ein Klick auf den Unterpunkt **Exit** beendet das Programm.

Im Menüpunkt **Settings** sind die gesamten Einstellungsmöglichkeiten untergebracht.

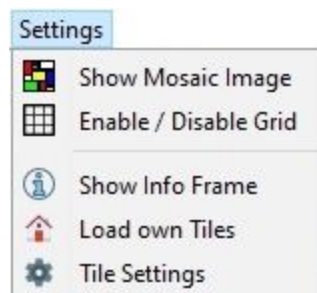


Bild: Menüpunkt Settings

Es existieren zwei Optionen zum Ein- oder ausschalten. **Show Mosaic Image** sowie **Enable / Disable Grid**. Diese sind über Checkboxes dargestellt, so lassen sie sich schnell aktivieren oder wieder deaktivieren.

Um Platz zu sparen, wird das generierte Mosaikbild nicht extra angezeigt, da dies nur unnötig Platz im Hauptfenster einnehmen würde. Es wird entweder das Target-Image oder das zu letzt generierte Mosaikbild angezeigt. Möchte man sich das Mosaikbild anzeigen lassen, muss man oben bei der Menüliste die oben erwähnte Checkbox **Show Mosaic Image** auswählen.

Ein Raster kann über die zweite Checkbox **Enable / Disable Grid** ein- oder wieder ausgeblendet werden. Hier gilt jedoch zu beachten, dass aufgrund von Rundungsfehler das angezeigte Grid nicht immer perfekt ist. Das kommt daher dass das Grid über Ganze Zahlen dargestellt ist, während das Bild an sich auch über Fließkommazahlen skaliert werden kann. So sollte es lediglich als grobe Orientierung genutzt werden, wie das Bild in verschiedene Tiles aufgeteilt werden könnte.

Die unteren drei Menüpunkte **Show Info Frame**, **Load own Tiles** und **Tile Settings** sind normale Menüitems, heißt diese sind nicht aktivierbar oder deaktivierbar, sondern sind sie normal wie Buttons bedienbar.

Ein Klick auf **Show Info Frame** öffnet ein weiteres Fenster mit Informationen rund um das Target-Image und den Source-Tiles.

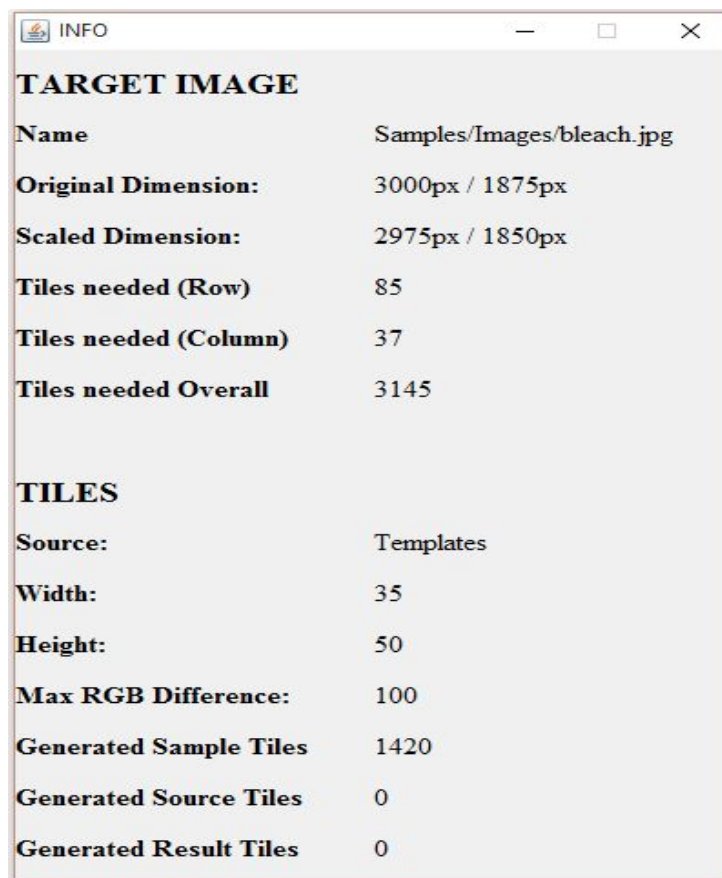


Bild: Infofenster

Möchte man anstelle der bereitgestellten Beispiels-Source-Tiles lieber Eigene verwenden, so lässt sich das über den Menüpunkt **Load own tiles** einrichten. Es öffnet sich zunächst ein kleines Fenster, in dem man die Größe der eigenen Tiles angeben muss. Dies ist ein wichtiger Schritt, da das System nach Bildern sucht, die dieser Größe entsprechen. Sollten Bilder im Ordner Tiles enthalten sein, die sich von der Größe unterscheiden, werden diese ignoriert und nicht auf die gewünschte Größe skaliert. Das wurde bewusst nicht getan, da die Tiles sonst einen unerwünschten Qualitätsverlust erleiden könnten. Der Benutzer hat selbst Sorge zu tragen, dass die Tiles im Ordner die gewünschte Größe enthalten.

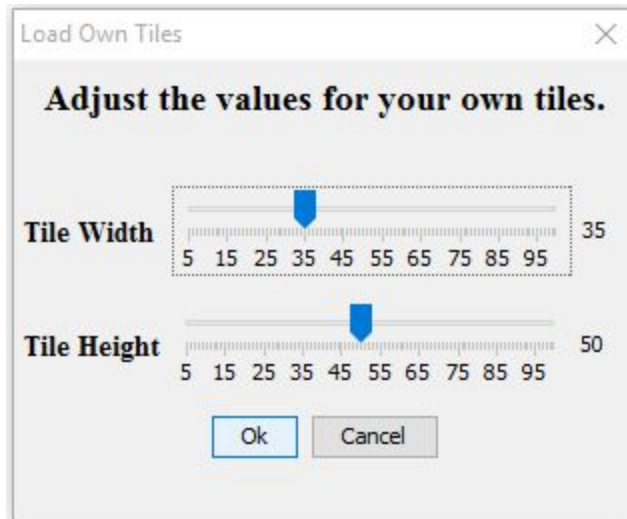


Bild: Einrichtungsfenster eigene Tiles

Das letzte und wohl wichtigste Menüitem ist wohl **Tile Settings**. Hier können Einstellungen vorgenommen werden, die die Tiles an sich manipulieren. Veränderungen der Tiles führen zu anderen Ergebnissen.

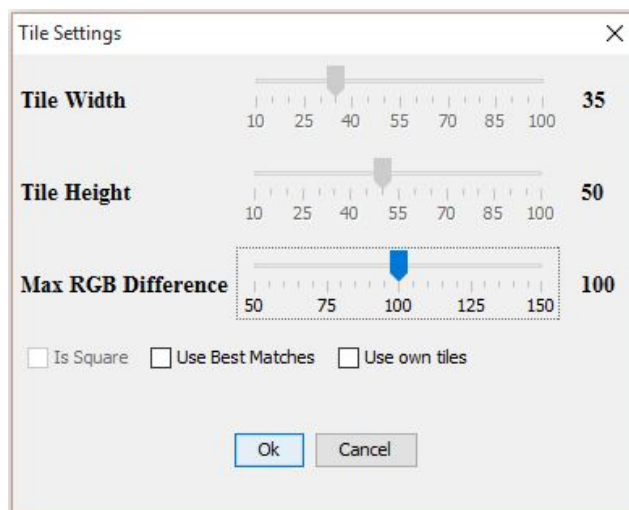


Bild: Bearbeitungsfenster Tiles

Mit den Slidern **Tile Width** und **Tile Height** kann man die Größe der eigenen Source-Tiles nachträglich verändern.

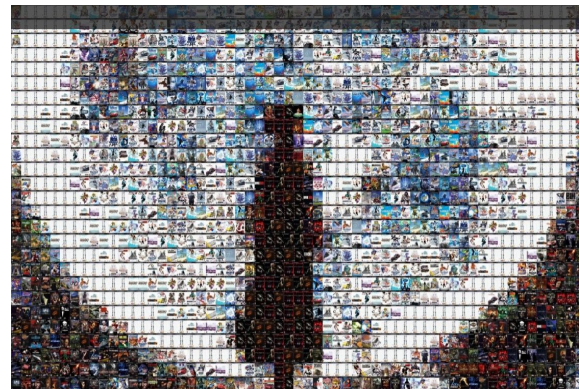
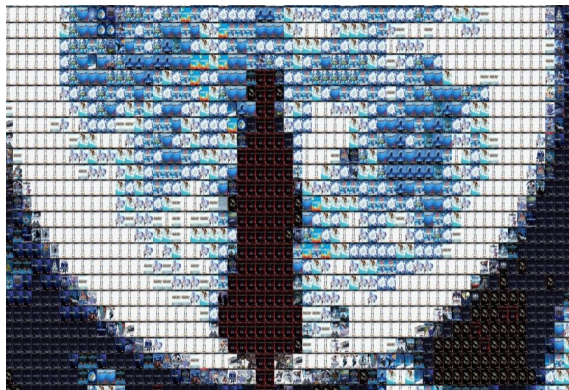
Die Checkbox **is Square** sorgt dafür, dass die Slider **Tile Width** und **Tile Height** immer denselben Wert haben und sich gegenseitig anpassen.

Aktiviert man die Option **Use Best Matches** wird die Toleranz der **maximalen RGB Difference** deaktiviert. Das heißt dass bei der Suche nach dem Source-Tiles, welches das aktuelle Tile des Source-Images ersetzen soll, immer das Resultat ausgewählt wird, mit der geringsten Differenz vom durchschnittlichem RGB-Wert. Der Vorteil ist dass das Ergebnis nun mathematisch betrachtet dem Target-Image am ähnlichsten sieht. Der Nachteil jedoch, dass sich über diesen Weg viele Source-Tiles wiederholen werden. Gerade bei größeren Flächen derselben Farbe erkennt man dann häufig eine Wiederholung der einzelnen Source-Tiles.

Deaktiviert man diese Option, wird nicht immer das beste Tile ausgewählt, sondern ein zufälliges Tiles, welches sich vom farblichen Unterschied her noch im Rahmen der Toleranz bewegt.

Über den Slider **Max RGB Difference** kann man diesen Toleranzwert verändern. Wie oben schon erwähnt liegt dieser Standardmäßig bei 100. Das heißt dass der durchschnittliche RGB-Mittelwert eines potentiellen Source-Tiles sich maximal um 100 unterscheiden darf zu dem Vergleichstile aus dem Target-Image.

Zum Abschluss noch eine kurze Gegenüberstellung, um den Unterschied zwischen aktiviertem und deaktiviertem Best Match Modus zu verdeutlichen.



Links deutlich zu erkennen, wie die Person, die vor dem Mond steht, aus denselben Tiles besteht, da die Person im Schutze der Nacht nur schwarz wirkt. Rechts im Bild sieht man das Ergebnis mit einer Toleranz von 100. Die Person besteht nun nicht mehr aus nur denselben Tiles. Zum Vergleich das originale Bild zu sehen auf Seite 6 dieser Dokumentation.