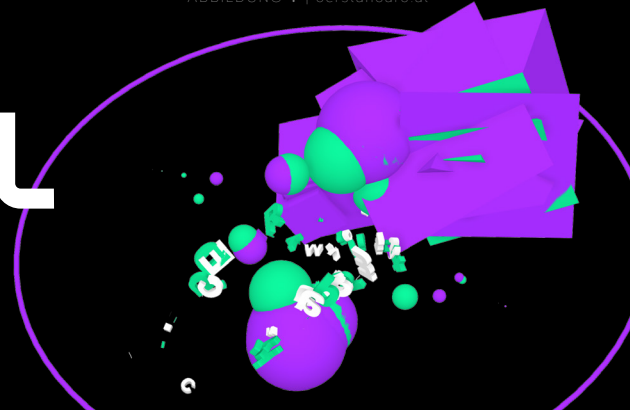


PROCEDURAL ANIMATION

Patrick Wagesreiter | S10628010



Den Ausgangspunkt und die Inspiration meines Konzepts bilden abstrakte Collagen von primitiven geometrischen Objekten wie in Abbildung 2, die mithilfe einer harmonischen Farbgebung zu einem stimmigen Bild verschmelzen.



ABBILDUNG 2 | Mood

Welche Eigenschaften der Daten wurden betrachtet?

Die Grundlage meines Algorithmus besteht hauptsächlich aus der Betrachtung der Analogie der vorliegenden Daten. Aufeinanderfolgende idente Datensätze werden für die Darstellung ignoriert, um möglichst viele unterschiedliche geometrische Objekte generieren zu können.

Was kann aus dem Bild gelesen werden?

Insgesamt setzt sich der Algorithmus aus fünf verschiedenen Elementen zusammen. Je monotoner die Eingangsdaten sind, desto weniger unterschiedliche Elementtypen finden sich schlussendlich in der grafischen Repräsentation wieder. Analog dazu ist die fertige Collage deutlich abwechslungsreicher, je vielfältiger die Eingabe ist.

Auf welchen Regelsätzen basiert die Darstellung?

Wie bereits erwähnt, werden in erster Instanz aufeinanderfolgende idente Daten ignoriert. Im nächsten Schritt werden die Bytes in Integer-Werte und anschließend in ASCII Werte konvertiert. Danach findet die eigentlich Analyse der Werte statt, indem die ASCII Werte anhand ihres Typs in Buchstaben, Zahlen (Werte ≤ 5 und Werte >5), Leer- und Sonderzeichen (inkl. Steuerzeichen etc.) getrennt und in Geometrie umgewandelt werden. Die Skalierung der Objekte ist meistens (indirekt) von der Anzahl der Schleifendurchläufe abhängig, sodass auch gleiche Werte unterschiedliche Ergebnisse liefern. Die Rotation und die Translation basieren auf der Berechnung des Volumens eines Kreises, welche als Variable auch wieder die Anzahl der Schleifendurchläufe benutzt.

Die Einfärbung der Elemente unterliegt prinzipiell einer ähnlichen Systematik wie die der Erzeugung der Geometrie. Zusätzlich wird der Integer Wert der Bytes Modulo 2 gerechnet, sodass die zwei Primärfarben (Türkis & Lila) abwechselnd vergeben werden. Weiß wird in Kombination mit Türkis nur für Buchstaben verwendet, damit in Summe ein möglichst ausgeglichener und harmonischer Gesamteindruck entsteht.

Welche Rückschlüsse erlaubt es uns auf die dargestellten Daten?

Befinden sich etwa nur Pyramiden und Tori im fertigen Bild, so kann davon ausgegangen werden, dass die Eingangsdaten nur Zahlen beinhalten. Eine Mehrheit von Buchstaben in der Szene lässt hingegen auf einen Text als Ausgangsmaterial schließen. Interessant ist auch die Verteilung der Objekte im Raum. Die Kreiszahl pi (Abbildung 3) produziert zum

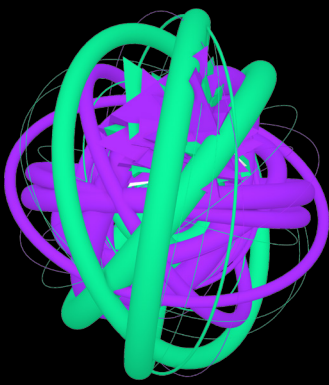


ABBILDUNG 3 | pi.txt

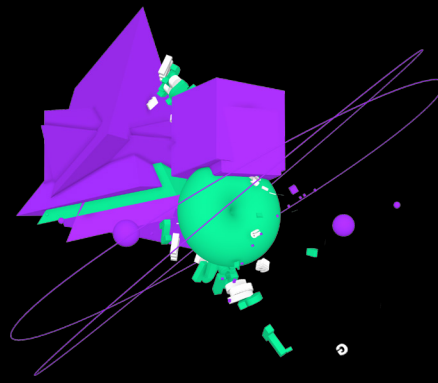


ABBILDUNG 4 | text.txt

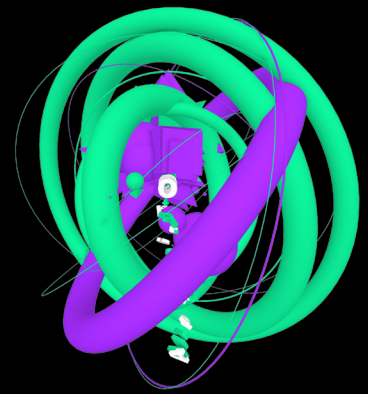


ABBILDUNG 5 | DreamOn.mp3

Beispiel ein sehr homogenes visuelles Resultat, bei dem die Objekte nicht sehr weit im Raum verteilt sind, weil sich auch die Ausgangsdaten immer wieder wiederholen.

Wie wurden die Daten verfremdet?

Die Verfremdung der Daten erfolgt in Form einer Transformation von ASCII Daten in bunte geometrische Grundformen, die – bis auf die Ausgabe der Buchstaben – nahezu keinen Hinweis mehr auf den ursprünglichen Inhalt der Daten beinhalten.

Wie wird das Script aufgerufen und verwendet?

Das Script wurde für Autodesk Maya (Version 2013 +) konzipiert und lässt sich mit dem Aufruf folgender Zeilen in einem leeren Python Tab im Script Editor starten:

```
import Wagesreiter_Visualizer
reload(Wagesreiter_Visualizer)
paths = [
    r'>projectPath/text.txt>,
    r>projectPath/image.jpg>,
    r>projectPath/pi.txt>,
    r>projectPath/website.txt>,
    r>projectPath/DreamOn.mp3>,
]
Wagesreiter_Visualizer.Visualizer(paths[1], 150)
```

Die ersten zwei Zeilen dienen zum Import und Reload der Python Datei, welche den Algorithmus beinhaltet. In den nächsten Zeilen werden die Pfade zu den gewünschten Eingangsdaten in der paths Liste definiert. Der Funktionsaufruf in der letzten Zeile besteht aus den Parametern paths[x] und einer Zahl,

wobei x die Auswahl der Datei aus der Pfadliste (der Wert 0 würde text.txt selektieren) ist und der zweite Parameter die Anzahl der Bytes, die zur Analyse verwendet werden sollen, bestimmt. Für das Generieren der Beispielbilder wurden etwa 150 Bytes pro Datei ausgewertet.

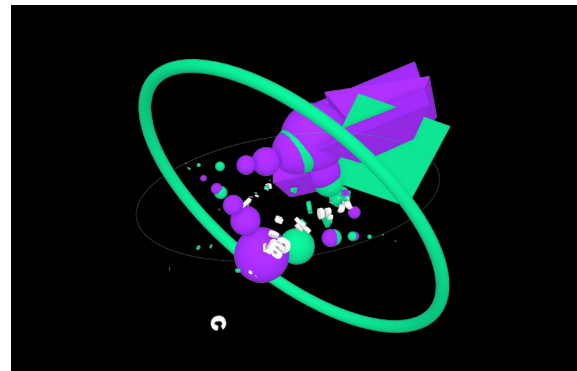


ABBILDUNG 6 | image.jpg

Am Ende des Scripts passt sich die Kamera noch an die generierte Szene an, sodass alle Elemente von der Kamera erfasst werden. Außerdem wird in den Viewport 2.0 gewechselt und Smooth Shading für die Szene aktiviert. Zum Rendern wurde der Hardware 2.0 Renderer mit aktiviertem Anti-Aliasing und Screen-space Ambient Occlusion verwendet.