

Aufgabenstellung - Abschlussübung 'Grundlagen Computergrafik' - TIT17

Im Rahmen der Vorlesung haben Sie Blender zum Rendern von 3D Objekten und virtuellen Szenen kennengelernt und damit bereits erste Erfahrungen gesammelt. In dieser Abschlussübung sollen verschiedene Techniken selbst umgesetzt werden.

Ihre Aufgabe ist es ein virtuelles Gelände automatisch zu generieren und darzustellen. Um Zusatzpunkte zu erhalten können Sie die unten genannten Aufgaben in ein originelles Umfeld einbetten und eine ansprechende Animationen erzeugen.

Hinweis: Sie können sich Modelle und Texturen von entsprechenden Seiten im Internet laden. Bspw.:

<https://www.textures.com/>, <http://www.texturegen.com/>,
<https://sketchfab.com/feed>

Aufgabe 1 – Prozedurales Displacement Mapping: (16 BE)

Aufgabe 1.1 – Diamond-Square: (8 BE)

- Implementieren Sie den [Diamond-Square-Algorithmus](#) in einem Python-Skript um eine „zufällige“ flächendeckende Height-Map-Textur zu erzeugen. Der Diamond-Square-Algorithmus erzeugt nur quadratische Strukturen mit ungeraden Seitenlängen. Ihr Skript soll in der Lage sein beliebige, auch nicht quadratische, Dimensionen zu erzeugen. Die Größe soll über eine globale Variable einstellbar sein. (5 BE)
<https://www.youtube.com/watch?v=h1joDbmxQes>
- Speichern Sie diese Textur in Ihrem Blender-Projekt sowie als separate Datei um diese später in der Script-Node laden zu können. (1 BE)
- Der Wertebereich der in der Textur gespeicherten „Höheninformationen“ soll zwischen 0 und 1 liegen. (1 BE)
- Achten Sie darauf, dass das Skript innerhalb von Blender ausführbar ist. *Hinweis: Nutzen sie die Blender eigene Schnittstelle zum Speichern von Texturen (Siehe Übung 3).* (1 BE)

Aufgabe 1.2 –Displacement-Mapping: (8 BE)

- Fügen Sie ein beliebiges zwei- oder dreidimensionales Objekt (Ebene, Kugel, Würfel, ...) in die Szene ein und nutzen Sie das selbstentwickelte Node-basierte Displacement-Mapping aus Übung 3 um die Oberfläche mittels der in Aufgabe 1.1. erzeugten Height-Map zu verändern. (4 BE)
Hinweis: Das Displacement Mapping sollte in der Lage sein mit beliebigen zwei-oder dreidimensionalen Objekten umgehen zu können. Gehen Sie wie in Übung 3.3 vor.
- Fügen Sie ein Skalierungs- und Verschiebungsparameter ein, die über die „Node-Oberfläche“ einstellbar sind. (2 BE)
- Fügen Sie ein „Flüssigkeits-Schwellwert“ ein der ebenfalls über die „Node-Oberfläche“ einstellbar ist. Bereiche, in denen die Werte unterhalb des Schwellwertes liegen, werden als Flüssigkeit betrachtet und deren Werte auf den Schwellwert gesetzt. (2 BE)

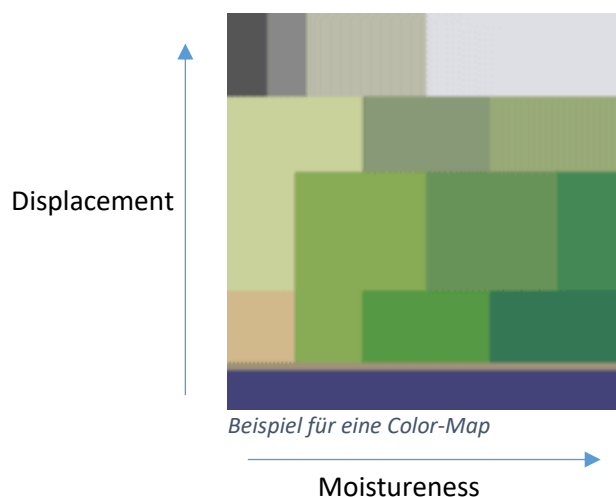
Aufgabe 2 – Prozedurale Farbgebung: (15 BE)

Aufgabe 2.1 – Moisture-Map: (6 BE)

- Erzeugen Sie mittels eines weiteren Python-Skript, das in Blender ausführbar sein muss eine zufällige Moisture-Map mit Wertebereich zwischen 0 und 1 auf Basis von Perlin-Noise. *Hinweis: Die Blender-Python-API biete entsprechende Funktionen. Sie können diese nutzen.* (3 BE)
- Die Dimension der Moisture-Map soll die gleiche Dimension sein wie die der Height-Map und automatisch bestimmt werden. (2 BE)
- Wie die Height-Map, soll die Moisture-Map im Blender-Projekt sowie als separate Datei abgespeichert werden. (1 BE)

Aufgabe 2.2 – Shading: (9 BE)

- Nutzen Sie eine zweidimensionale Color-Map um ihre Oberfläche anhand des Displacements und der Feuchtigkeit einzufärben. (Bsp.: <https://www.redblobgames.com/maps/terrain-from-noise/#biomes>)



Hinweis: Sie können sich entweder selbst eine Map erstellen oder eine vorgefertigte aus dem Internet verwenden. (2 BE)

- Nutzen Sie das selbstentwickelte Lambert-Shading um den Geländeflächen eine diffuse Oberfläche zu geben. (2 BE)
- Nutzen Sie das selbstentwickelte Phong-Shading um der Flüssigkeit eine spekulare Oberfläche zu geben. (2 BE)
- Setzen Sie dies mittels einer oder mehreren Script-Nodes innerhalb eines Blender-Materials um. Die Nodes sollen als zusätzliche Input die Height-Map, Moisture-Map und Color-Map erhalten. (3 BE)

Aufgabe 3 – Wellenanimation: (4 BE)

- Setzen Sie Normal-Mapping um, um eine Wellenanimation zu schaffen und die Flüssigkeit realistischer aussehen zu lassen. Greifen Sie hierfür auf Texture- und Normal-Map-Node von Blender zurück. (2 BE)

- Nutzen Sie die Animationsfunktionalität von Blender und die in Blender vorhandenen Nodes um die Wellenbewegung zu animieren. (2 BE)
<https://watersimulation.tumblr.com/post/115928250077/scrolling-normal-maps>
 - *Hinweis: Greifen Sie auf selbsterstellte oder aus dem Internet bezogene Normalenkarten zurück. Diese können Sie sich auch mittels entsprechenden Tools (bspw. <https://github.com/Theverat/NormalmapGenerator>) aus Bumpmaps erzeugen. Legen Sie für die einzelnen Key-Frames in Ihrer Animation den Offset fest mit dem Sie auf die Normalenkarte zugreifen. Damit können Sie eine „Verschiebung“ Karten erzielen.*
-

Originalität:

Neben den oben genannten Punkten vergeben wir noch weitere 4 Punkte für Originalität und zusätzlichen Arbeiten. Dies erlaubt uns hervorragende Arbeiten hervorzuheben. Möglichkeiten um in diesem Bereich zu Punkten sind folgende:

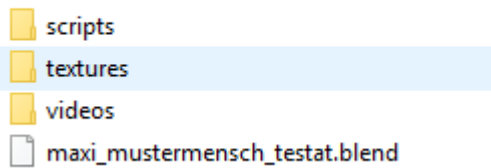
- **Originelles Setting**
- **Einbau von Physik**
- **Verwendung einer Environment Map**
- **Nutzung von Kamerabewegungen**

Bearbeitung und Notenvergabe:

- Der Programmentwurf soll in kleinen Gruppen von 3 Studenten bearbeitet werden.
- Nutzen Sie Blender 2.80
- Erstellen Sie zur Abgabe ein Projekt auf GitHub und fügen Sie uns dem Projekt hinzu:
 - <https://github.com/boitumeloruf>
 - <https://github.com/thomasgolda>
- Schicken Sie uns einen Link zum Commit der Ihre Lösung beinhaltet. Achten Sie darauf einzelne, kleinere Commits zu machen, damit wir den Fortschritt nachvollziehen können. Der reguläre Zeitstempel der Email gilt als Abgabezeitpunkt. Sämtliche Commits, die nach dem uns mitgeteilten Commit bzw. dem Abgabzeitpunkt erstellt wurden, werden nicht berücksichtigt.
- Entwickeln Sie Code, der gut lesbar ist (Kommentare, Zeilenlänge nicht größer als 100 Spalten).
- Erzeugen Sie eine finale flüssige Videosequenz von 20-30 Sekunden Länge. (Auflösung mindestens 1280x720) die Ihre Ergebnisse zeigt. Diese stellt auch die Basis der Originalitätsbewertung dar.

- Legen Sie der Abgabe eine Text-Datei mit den Namen der Studenten, die an dem Programmmentwurf gearbeitet haben bei.
- Das Blender-Projekt und alle Skripte müssen ausführbar sein. Sollten irgendwelche Abhängigkeiten bestehen müssen diese erwähnt und erläutert werden.
- Ihre Lösung sollte in einer Blender-Projektdatei enthalten sein. Als zusätzliche Sicherheit speichern Sie bitte sämtliche Shading-Skripte als .osl und Python-Skripte als .py Dateien in den Ordner „scripts“ ab. Texturen die Sie erzeugt und genutzt haben legen Sie bitte zusätzlich in den Ordner „textures“ ab. Erzeugte Video-Sequenzen landen im Ordner „videos“.

Name



- Sollten Sie Aufgaben nicht vollständig lösen können, dann legen Sie eine Notiz dazu bis wohin Sie gekommen sind. Wir können auch Teilpunkte vergeben, falls ersichtlich.
- Nutzen Sie zur Hilfestellung die in den Vorlesungsfolien referenzierte Literatur und die verfügbaren Beispiele, sowie die Dokumentation von Blender.
- Sollte Fragen zur Aufgabenstellung haben oder vor Problemen stehen, die Sie durch eigene Recherche nicht lösen können, so zögern Sie bitte nicht uns zu kontaktieren.
- Es darf kein Code direkt dupliziert und in Ihrem oder einem anderen Programmmentwurf eingefügt werden.

Die Gesamtbenotung wie folgt:

- | | |
|--|------------------|
| ○ Aufgabe 1 – Prozedurales Displacement Mapping: | 16 Punkte |
| ○ Aufgabe 2 – Prozedurale Farbgebung: | 15 Punkte |
| ○ Aufgabe 3 – Wellenanimation: | 4 Punkte |
| ○ Originalität: | 4 Punkte |
| ○ Abgabe nach vorgegebener Ordnerstruktur | 1 Punkt |

Insgesamt: 40 Punkte

Für 4,0: 20 Punkte

Für 1,0: 37 Punkte

VIEL ERFOLG!

Abgabe bis Mittwoch, 11. Dezember 2019 um 23:59 Uhr