

Aufgabenstellung - Abschlussübung 'Grundlagen Computergrafik' - TIT18/TIM18

Im Rahmen der Vorlesung haben Sie Unity 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 Userstory erzeugen.

Hinweis: Sie können sich Modelle und Texturen von entsprechenden Seiten im Internet laden. Bspw.: <https://www.textures.com/>, <http://www.texturegen.com/>, <https://www.assetstore.unity3d.com/>

Aufgabe 1 – Prozedurales Displacement Mapping: (15 BE)

Aufgabe 1.1 – Diamond-Square: (7 BE)

- Implementieren Sie den [Diamond-Square-Algorithmus](#) in einem 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>
- Der Wertebereich der in der Textur gespeicherten „Höheninformationen“ soll zwischen 0 und 1 liegen. (1 BE)
- Schreiben Sie ein Skript, dass beim Starten die gewünschte Textur erzeugt und einem Material bereitstellt. Achten Sie darauf, dass Sie das Skript einem Material zugeordnet ist. *Hinweis: Nutzen sie die Unity eigene Schnittstelle zum Erzeugen von Texturen (Siehe: <https://answers.unity.com/questions/9919/how-do-i-create-a-texture-dynamically-in-unity.html>).* (1 BE)

Aufgabe 1.2 –Displacement-Mapping: (8 BE)

- Fügen Sie eine Kugel in die Szene ein und nutzen Sie das selbstentwickelte Displacement-Mapping aus Übung 3.3 um die Oberfläche mittels der in Aufgabe 1.1. erzeugten Height-Map zu verändern. Achten Sie darauf, dass gewählte Objekt eine hohe Vertexdichte hat. Sie können dazu entweder die mitgelieferte Kugel (sphere.fbx) nutzen, oder mittels Blender oder einem anderen Tool selbst eine erzeugen. (4 BE)
Hinweis: Das Displacement Mapping sollte in der Lage sein mit beliebigen zwei-oder dreidimensionalen Objekten umgehen zu können.
- Fügen Sie ein Skalierungs- und Verschiebungsparameter ein, die über die GUI-Oberfläche einstellbar sind. (2 BE)
- Fügen Sie ein „Flüssigkeits-Schwellwert“ ein der ebenfalls über die GUI-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)

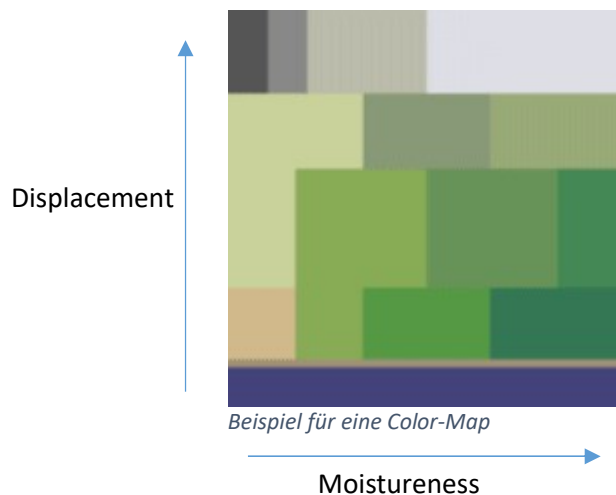
Aufgabe 2 – Prozedurale Farbgebung: (15 BE)

Aufgabe 2.1 – Moisture-Map: (6 BE)

- Erzeugen Sie eine weitere Textur, die eine zufällige Moisture-Map mit Wertebereich zwischen 0 und 1 auf Basis von Perlin-Noise darstellt und weisen Sie die ebenfalls dem Material zu. *Hinweis: Die Unity-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 Unity-Projekt genutzt werden können. (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 einem Unity Shader um. Der Shader soll dabei auf die Height-Map, Moisture-Map und Color-Map zurückgreifen können. (3 BE)

Aufgabe 3 – Wellenanimation (4 Punkte):

- Implementieren Sie ein Normal-Mapping um eine Wellenanimation zu schaffen und das Wasser realistisch aussehen zu lassen. (4 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.*

Bonusaufgabe – Verbesserte Höhenkartengenerierung (2 BE)

- Der Diamond-Square-Algorithmus aus Aufgabe 1 wird bei Anwendung auf einem dreidimensionalen Objekt zu „unschönem“ Verhalten an der Verbindungskante führen. Dieser wird sich als harte Kante zeigen.
- Überlegen Sie sich eine Möglichkeit diese harte Kante zu vermeiden und setzen Sie diese zusätzlich zu Aufgabe 1 um.

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 3-4 Studenten bearbeitet werden.
- Nutzen Sie die zu Beginn der Vorlesung angegebene Version von Unity (v2019.4)
- 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 Abgabezeitpunkt erstellt wurden, werden nicht berücksichtigt.

- Entwickeln Sie Code, der gut lesbar ist (Kommentare, Zeilenlänge nicht größer als 100 Spalten).
- **Geben Sie sowohl das Unity-Projekt als auch ein kompiliertes und ausführbares Spiel für Windows ab.**
- Legen Sie der Abgabe eine Text-Datei mit den Namen der Studenten, die an dem Programmentwurf gearbeitet haben bei.
- Das Unity-Projekt und alle Skripte müssen ausführbar sein. Sollten irgendwelche Abhängigkeiten bestehen müssen diese erwähnt und erläutert werden.
- Nutzen sie die Ordnerstrukturen aus den Übungen um Ihre Assets (Materialien, Texturen, Shaders, ...) ordentlich zu sortieren.
- 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 Unity.
- 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 Programmentwurf eingefügt werden.

Die Gesamtbenotung wie folgt:

- | | |
|--|------------------|
| ○ Aufgabe 1 – Prozedurales Displacement Mapping: | 15 Punkte |
| ○ Aufgabe 2 – Prozedurale Farbgebung: | 15 Punkte |
| ○ Aufgabe 3 – Wellenanimation: | 4 Punkte |
| ○ Bonusaufgabe | 2 Punkte |
| ○ Originalität: | 4 Punkte |
| ○ Abgabe nach vorgegebener Ordnerstruktur und inkl. vorkompilierter „Executable“ | 1 Punkt |

Insgesamt: 41 Punkte

Für 4,0: 20 Punkte

Für 1,0: 38 Punkte

VIEL ERFOLG!

Abgabe bis Dienstag, 29. Dezember 2020 um 23:59 Uhr