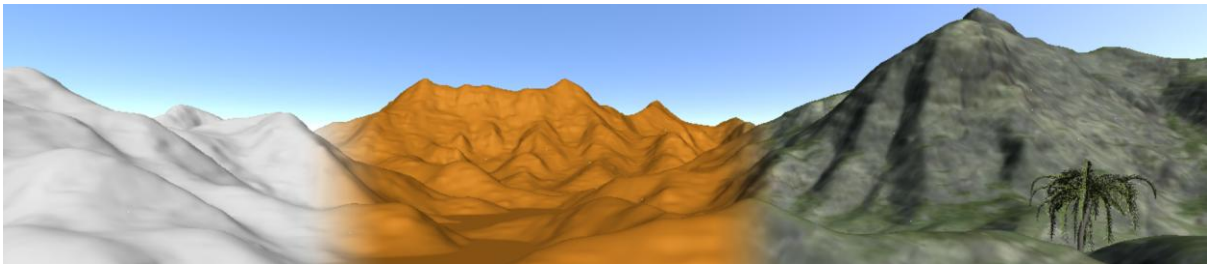


Aufgabe 2: Interaktive Darstellung mit OpenGL



Gegeben ist ein OpenGL Programm, das ein Geländemodell darstellt, welches geeignet erweitert werden soll. Die Teilaufgaben lauten wie folgt:

1. Parametrisieren Sie eine feste (vom Betrachter unabhängige) Lichtquelle, welche die Lichtverhältnisse an einem sonnigen Tag nachbildet.
2. Legen Sie geeignete Materialeigenschaften (Ambient-, Diffuse-, Spekular-Koeffizienten) für das Gelände (Methode `RenderTerrain()`) fest. Beachte: `glColor` kann nur zwischen `glBegin` und `glEnd` gesetzt werden und ist daher hier nicht geeignet.
3. Mittels Tastendruck (Alt+K) soll die integrierte Kamera (`GetFirstPersonCamera()`) ausgeschaltet werden können. Anschließend soll mittels der Tasten 1 bis 3 jeweils eine interessante Kameraposition manuell gesetzt werden können. Die Methode `EvaluateActionKey()` kann genutzt werden, um auf Tastatureingaben zu reagieren.
4. Fügen Sie dem Landschaftsmodell mindestens 4 Palmenmodelle hinzu. Hierfür wird bereits eine Methode (`RenderPalm()`) zur Verfügung gestellt. Jede Palme soll eine andere Skalierung und Position besitzen (Matrix Stack benutzen!).
5. Das Programm verwendet den Shader `shader.glsl`. Dieser Shader hat bereits Zugriff auf drei Texturen: Eine Normalen-Map (Textur mit Oberflächennormalen für jeden Punkt im Gelände), eine Ambient Occlusion Map (Textur mit Informationen zur Selbstverdeckung im Gelände) und eine Oberflächentextur, welches Steine und Gras im Gelände abbildet. Die Texturen haben bereits die richtigen Koordinaten und sind im Fragment Shader unter den Variablen `normalTexel`, `aoTexel` und `terrainTexel` abgreifbar.
Verwenden Sie die Texturen um den ambienten (ambient occlusion map) und diffusen (normal map + Lichtrichtungsvektor) Lichtterm des phongschen Beleuchtungsmodells unabhängig vom festgelegten Material zu berechnen. Der spekulare Term kann vernachlässigt werden, da im Gelände im Allgemeinen keine Spiegelungen auftreten. Verwenden Sie außerdem die Terrain Textur, um das Gelände einzufärben.
6. **Zusatzaufgabe (optional):** Integrieren Sie in die Beleuchtungsberechnung Cell-Shading. Beim Cell-Shading wird der Lambertsche Term „ $\max(0, n \cdot l)$ “ quantisiert. Im Normalfall werden 4-8 Cluster verwendet. Schalten Sie hierfür die Textur aus, um einen besseren Effekt zu erzielen.

Beachte: Das Programm besteht aus verschiedenen Teilprojekten (lib/dll). Das Startprojekt (*.exe) ist das *Application-Projekt*. Alle Teilaufgaben können durch das Hinzufügen von Programmcode im

Modul: *OpenGLWindow.cpp (*.h)* in diesem Projekt *Application* und des zugehörigen Shaders *shader.glsl* (Ordner Shader) gelöst werden.

Das Programm erfordert eine OpenGL 2.0 kompatible Grafikkarte mit mindestens 256MB VRAM. Sollte Ihnen solch ein System nicht zur Verfügung stehen, können die Rechner im Computergrafiklabor für die Umsetzung genutzt werden. Das Projekt benötigt mindestens Visual Studio 2010

Abgabetermin der Lösung: 11. Juli 2014!

Abgabeform : Mail mit Zip-Datei im Anhand in der Form „Name_Vorname.zip“. Die Zip-Datei muss den Quelltext (*OpenGLWindow.cpp / .h*) und den modifizierten Shader *shader.glsl* enthalten. Sollten zusätzliche Module verwendet werden, sind diese mitzuliefern. **Bitte nicht die von uns bereitgestellten Include-Dateien und Texturen hinzufügen.**

An: steve.duebel@uni-rostock.de, Betreff: OpenGL Aufgabe