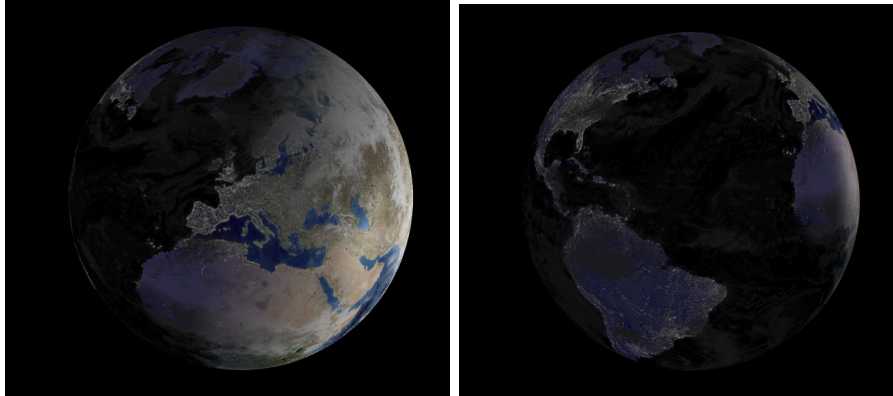


Computergrafik 2 / Aufgabe 3: *Weltanschauung*



In dieser Aufgabe starten Sie mit einer Kugel, einer direktionalen Lichtquelle und einem Phong-Shader. Sie statten die Kugel mit Normalen und Texturkoordinaten aus und modifizieren den Shader so, dass sie einen Erdglobus unter beliebigen Sonnenständen und mit verschiedenen Oberflächeneigenschaften darstellen können. Dabei können Sie sich einiger NASA-Bilder als Texturen bedienen: Topographie, Höheninformation, Wolkendichte etc.

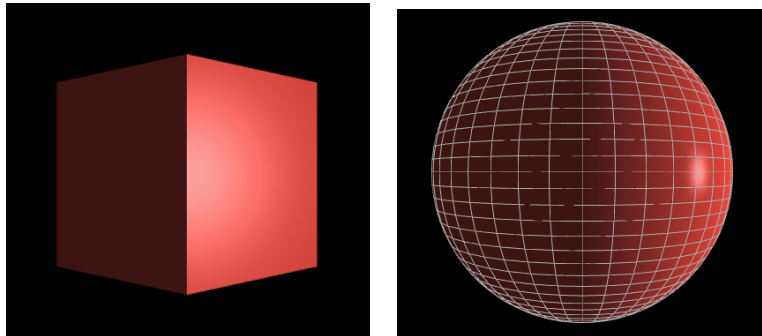
Framework / Vorbereitung

Laden Sie das Framework zu Aufgabe 3 von Moodle herunter und legen Sie es so ab, dass der Ordner auf der gleichen Verzeichnisebene wie das bereit vorhandene Verzeichnis `lib/` liegt. Es handelt sich um eine erweiterte Version des Frameworks von Aufgabe 2 mit Licht, Material, Texturen, Phong-Shading. Der `SceneNode` wurde um die Berechnung der Normalen-Matrix erweitert. In `models/` findet sich die Datei `cube.js`. Dieses Modell stellt die folgenden Vertex-Attribute bereit: Position, Normalenrichtung, Texturkoordinaten. Das Modul `scene_explorer.js` wurde integriert und erlaubt die interaktive Navigation der Szene mittels der Maus (*Drag = Rotation; Shift-Drag = Translation; Alt-Drag oder Wheel = Zoom*). Wenn die Maus hakt, hilf i.d.R. die Escape-Taste.

Der `SceneNode` verwendet nun nicht mehr explizit `Program`, sondern `Material`. D.h. die `draw()`-Funktion eines Modells erhält anstelle eines Programms ein `Material` (siehe `models/cube.js`, `material.js` und Handout). Ein `Material` kapselt ein Programm zusammen mit einem Satz von `uniform`-Werten, die das Programm benötigt. Durch Aufrufe von `Material.apply()` werden die aktuellen Uniform-Werte in dem Programm gesetzt. In `scene.js` können Sie studieren, welche Informationen z.B. das `Material` für den Planet-Shader enthält.

Das eigentliche Shader-Paar, welches Sie als Ausgangsbasis verwenden sollen, ist ein Phong-Shader in den Dateien `shaders/planet.vs` und `shaders/planet.fs`.

Aufgabe 3.1: Phong-Shader, Würfel und die Erdkugel



Wenn Sie die `index.html` im Browser anschauen, sehen Sie einen roten Würfel, der von einer **direktionalen Lichtquelle** (von rechts) angeleuchtet wird. Starten Sie die Animation, so dreht sich die Richtung der Lichtquelle so, als würde sich eine Sonne um den Würfel herum bewegen. Große Teile des Würfels liegen dabei die meiste Zeit im Schatten (gut, dass die Erde kein Würfel ist!). Mit den Tasten `x` und `y` sowie mit der Maus (s.o.) können Sie die Szene leicht von allen Seiten erforschen. Mit den Tasten `S` und `Shift-S` können Sie die Sonne manuell steuern.

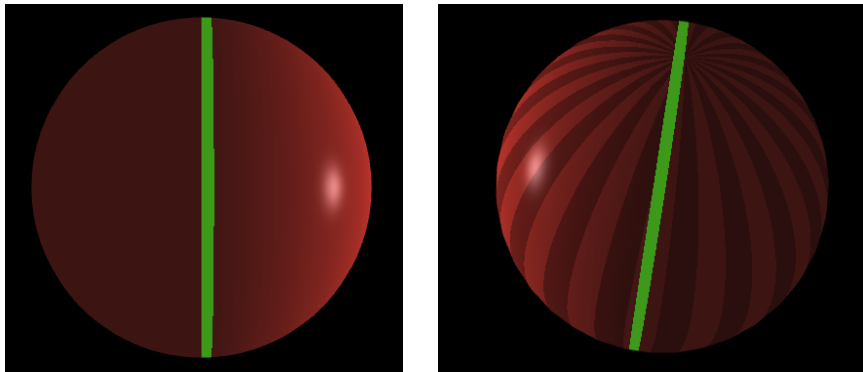
Portieren Sie nun zuerst Ihre `ParametricSurface` aus Aufgabe 2 in diese Aufgabe, und ersetzen Sie den Würfel durch eine Kugel.

- Verwenden Sie dafür folgende Kugelformel:

```
var positionFunc = function(u,v) {  
    return [ r * Math.sin(u) * Math.cos(v),  
            r * Math.sin(u) * Math.sin(v),  
            r * Math.cos(u) ];  
};
```
- Der Aufruf von `ParametricSurface.draw()` muss nun anstelle eines Program ein `Material` entgegennehmen; Sie können dann mit `Material.getProgram()` auf das in dem `Material` gekapselte Programm zugreifen (siehe Handout und `models/cube.js`).
- Ihre Kugel muss zusätzlich zu Vertex-Position auch eine Normale pro Vertex angeben, welche vom Kugelmittelpunkt nach außen zeigt und die Länge 1 hat. Sie können der `ParametricSurface` z.B. neben der `positionFunc` auch noch eine `normalFunc` übergeben, welche für ein gegebenes `(u,v)` die Normale berechnet. *Position und Normale sind sich bei der Kugel sehr ähnlich.*
- Die Normalen werden vom Phong-Shader unter dem Namen `vertexNormal` erwartet, siehe `cube.js` und `shaders/planet.vs`

Wenn Sie Ihre Kugel mit Normalen ausgestattet haben, sollten Sie beobachten können, wie bei eingeschalteter Animation das Licht gleichmäßig um die Kugel herum wandert. Fügen Sie der Szene eine `drawOption` "Show Grid" hinzu, mittels derer auch das Wireframe der Kugel angezeigt wird (verwenden Sie dazu ein anders Material). Rotieren Sie die Kugel so, dass die Pole der Kugel in der Y-Achse zu liegen kommen, also aus Sicht der initialen Kamera oben und unten liegen.

Aufgabe 3.2: Tag-Nachtgrenze und Texturkoordinaten



Um die Arbeitsweise den Phong-Shaders zu verstehen, visualisieren Sie den Übergang zwischen Tag und Nacht als grünen Streifen auf der Kugel.

- Fügen Sie der Szene eine `drawOption "Debug"` hinzu, und reichen Sie den Bool-Wert über eine geeignete `uniform` an die `planet`- Shader weiter.
- Visualisieren Sie auf dem Planeten den Bereich, in welchem der Übergang zwischen Tag und Nacht stattfindet. Wenn der "Debug-Schalter" an ist, malen Sie im `planet`-Shader diejenigen Fragmente grün, bei denen der Winkel zwischen Sonnenlicht und Oberfläche zwischen 0° und 3° beträgt. Alle für die Berechnung benötigten Vektoren liegen bereits im Phong-Shader vor.
- Vergewissern Sie sich, dass die Tag-Nacht-Linie wie gewünscht mit der Sonne wandert, wenn die Animation eingeschaltet ist.

Damit wir die Erde mit schönen Texturen ausstatten können, soll Ihre Kugel nun noch 2D-Texturkoordinaten pro Vertex zu Verfügung stellen. Letztere sollen nun im Fragment-Shader verwendet und zunächst visuell überprüft werden.

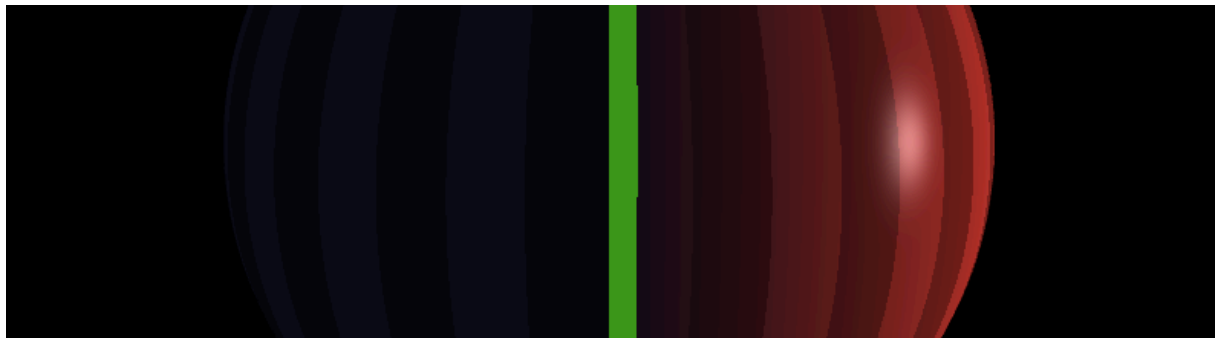
- Erweitern Sie Ihre `ParametricSurface` / Kugel um ein weiteres Vertex-Attribut, welches aus zwei Koordinaten pro Vertex besteht und unter dem Namen `vertexTexCoords` im Shader verwendet werden soll. Nehmen Sie initial einfach die Werte von `u` und `v` als Texturkoordinaten.
- Sorgen Sie dafür, dass die Texturkoordinaten, die Kugel unter dem Namen `vertexTexCoords` bereitstellt, vom Vertex-Shader entgegengenommen und an den Fragment-Shader unter einem Namen wie z.B. `texCoords` weitergereicht werden (vgl. Handout).
- Visualisieren Sie den Verlauf der ersten Texturkoordinate (z.B. `texCoords.s`) im Debug-Modus, indem Sie immer abwechselnd einen Streifen von Fragmenten dunkler machen und einen Streifen unangetastet lassen. Am einfachsten multiplizieren Sie den ambienten und den diffusen Term jeweils immer mit einem "Abdunkel-Faktor", der abhängig von der Texturkoordinate gesetzt wird.
- Tipp: Modulo ist Ihr Freund und heißt in der Shading Language `mod()`.
- Ändern Sie dann die Funktion so, dass die Texturkoordinaten zwischen 0 und 1 liegen, und dass `texCoords.s` im Debug-Modus vertikale Streifen produziert.
- Vergewissern Sie sich, dass bei Einschalten der Animation die Tag-Nacht-Linie wandert, aber die Texturstreifen stehen bleiben

Aufgabe 3.3: Tag- und Nachttextur



Nun soll das Erscheinungsbild des Planeten mit Texturen aufgewertet werden. Siehe auch Beispiel-Bilder auf der nächsten Seite.

- Bedienen Sie sich im Konstruktor der Szene der Funktion `texture.Texture2D`, um `textures/earth_month04.jpg` als Textur zu laden.
- Binden Sie die Textur in Ihrem `planet`-Programm mittels der Methode `setTexture()` an die Textureinheit 0 und z.B. an den `uniform`-Bezeichner `"daylightTexture"`. Diesen Schritt sollten Sie nur dann ausführen, wenn die Textur auch wirklich geladen wurde (`onAllTexturesLoaded()`, siehe SU-Folien).
- Greifen Sie nun im Planet-Fragment-Shader auf die Textur zu. Erweitern Sie den Shader so, dass Sie den Wert der Textur anstelle des diffusen Material-Koeffizienten verwenden. Ersetzen Sie nicht den gesamten diffusen Term! Trotz der Werte aus der Textur soll die Abdunkelung zum Abend immer noch gut zu erkennen sein. Hellen Sie die Werte aus der Textur ggf. nach Geschmack auf (z.B. mittels `*` und/oder `pow()`).
- Führen Sie eine Checkbox "Daytime Texture" ein, welche es ermöglicht, zwischen der texturierten Darstellung der Tagseite und der roten Phong-Darstellung zu wechseln.
- Da die Textur nur in den diffusen Term eingeht, wie sie automatisch auf der „Tagseite“ der Erde angezeigt, während aufgrund des Einfallswinkels bei Phong die Nachtseite der Erdkugel nur mit der ambienten Farbe dargestellt wird. Verändern Sie die Beleuchtungsfunktion so, dass auf der Nachtseite der Erde die Textur `earth_at_night_2048.jpg` angezeigt wird. Interpolieren Sie dazu auf geeignete Weise zwischen Tag- und Nachttextur.
 - Jenseits der Tag-Nachtgrenze sollten nur die Lichter zu sehen sein (nur ambienter Term); dort wo die Sonne steil einfällt (mittags) sollen die Lichter nicht zu sehen sein (ambienter Term = 0).
 - In der Dämmerung nahe der Tag-Nacht-Linie sollte man sehen, dass die Menschen am Anfang/Ende des Tages das Licht einschalten. Der Übergang von Tag und Nacht sollte glatt ohne sichtbaren "Bruch" erfolgen.
- Führen Sie eine Checkbox "Night Lights" ein, welche es ermöglicht, die zusätzliche Nachttextur ein- und auszuschalten. Wenn Sie ausgeschaltet ist, sollte die "Rückseite" der Erde schwarz oder zumindest sehr dunkel sein. Stellen Sie sicher, dass in der Mittags-Zeit auf der Erde keine Lichter angehen.
- Trotz der nachfolgenden Veränderungen soll der Debugging-Modus weiterhin die Tag-Nacht-Linie visualisieren, um die Überblendung prüfen zu können.



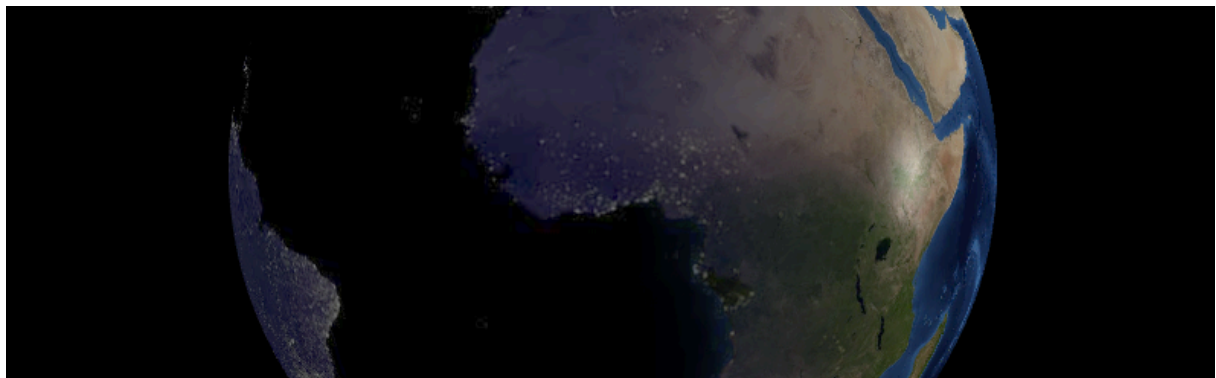
ohne Texturen, Tag-Nacht-Grenze im Debug-Modus



nur Tag-Textur (diffuser Term)

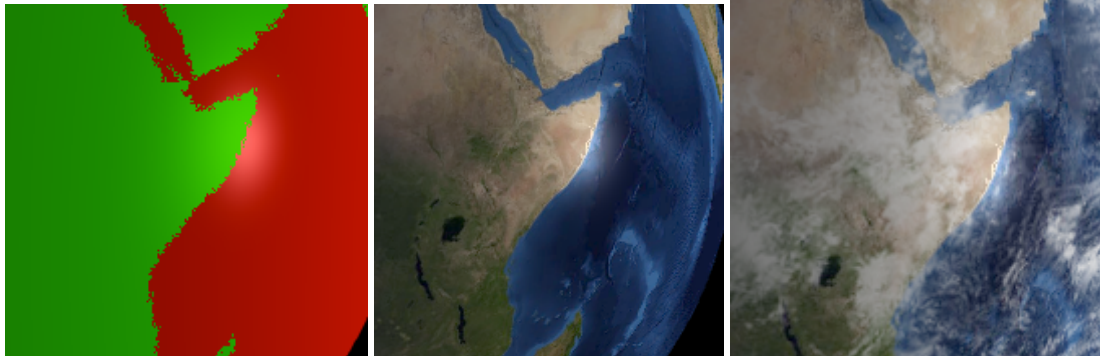


Nur Nacht-Textur, man sieht die Überblendung auf der Tagseite



Tag-und Nachttextur mit Überblendung

Aufgabe 3.4 (optional, nur für eine sehr gute Note): Wasser und Wolken



Durch die Natur des Phong-Shaders sieht die Erdkugel noch eher wie eine Plastik-Kugel aus, da Erde und Wasser gleichermaßen spekulare reflektieren. Es erscheint jedoch realistischer, wenn Wasser stärker spiegelt als die Landflächen.

- Führen Sie zunächst einen Schalter "Red/Green" ein, der bewirkt, dass sowohl auf Tag- als auch auf Nachtseite Landfragmente grün und Wasserfragmente rot gezeichnet werden.
- Verwenden Sie dazu z.B. die Information aus der topografischen oder bathymetrischen NASA-Textur, die die Höhen- bzw. Tiefeninformation kodiert. Ermitteln Sie, welche Texturwerte welcher Höhe entsprechen, und setzen Sie im Shader abhängig von dieser Information die Farbe.
- Fügen Sie einen Schalter "Gloss Map" hinzu, der bewirkt, dass sich der spekulare Term Ihrer Beleuchtungsfunktion abhängig von Wasser vs. Land verändert. Manipulieren Sie wahlweise den spekularen Materialkoeffizienten, den Phong-Exponenten, oder beides.
- Die bisher verwendeten Texturen der Erde sind wolkenfrei. Fügen Sie der Erde abhängig von einer Checkbox "Clouds" eine Wolkenschicht auf Basis der Textur `earth_clouds_2048.jpg` hinzu. Der Rot-Kanal der Textur kann wie eine "Wolkendichte" interpretiert werden. Je dichter die Wolken sind, desto mehr sollte die Wolkenfarbe die Erdfarbe überdecken (Interpolation). Die Wolken sollten auch eine sinnvolle Auswirkung auf die Sichtbarkeit der Lichter in der Nacht haben (Abdunklung).

Abgabe

Die Abgabe soll via Moodle bis zu dem dort angegebenen Termin erfolgen. Verspätete Abgaben werden wie in den Handouts beschrieben mit einem Abschlag von 2/3-Note je angefangener Woche Verspätung belegt. Geben Sie bitte pro Gruppe jeweils nur eine einzige .zip-Datei mit den Quellen Ihrer Lösung ab (das Texturverzeichnis können Sie dabei auslassen).

Demonstrieren und erläutern Sie dem Übungsleiter Ihre Lösung *in der nächsten Übung nach dem Abgabetag*. Die Qualität Ihrer Demonstration ist, neben dem abgegebenen Code, ausschlaggebend für die Bewertung! Es wird erwartet, dass alle Mitglieder einer Gruppe anwesend sind und Fragen beantworten können.