

Übungen zu Computergrafik

Sommersemester 2012

Blatt 4

Aufgabe 4.1 (25 Punkte)

Laden Sie sich das Projekt von `/home/cg/2012/Uebung/Blatt4/Aufg/CG12Blatt4.zip` und binden Sie es in Ihre bevorzugte Entwicklungsumgebung ein. Öffnen Sie die Klasse `main.TranslateNGon`. Diese soll folgende Features bereitstellen:

1. Es sollen die Ecken eines regelmäßigen 12-Ecks als Punkte auf dem Bildschirm dargestellt werden.
2. Der Benutzer soll die Möglichkeit haben, das 12-Eck mittels der Tasten *W*, *A*, *S* und *D* zu translätieren.

Punkt 1.) ist bereits implementiert.

1. Erweitern Sie die Klasse um eine Klassenvariable `translation` vom Typ `org.lwjgl.util.vector.Vector3f`, in der die aktuelle Translation des 12-Ecks vermerkt ist.
2. Füllen Sie die Methoden `handleInput()` und `updateUniforms()`, in denen folgendes passieren soll:
 - (a) In `handleInput()` soll geprüft werden, ob die Tasten *W*, *A*, *S* oder *D* gedrückt sind und die Variable `translation` entsprechend verändert werden.
 - (b) In `updateUniforms()` soll aus der Variable `translation` eine Translationsmatrix erzeugt werden und diese als uniform Variable `translate` dem Shaderprogramm übergeben werden.
 - (c) Verändern Sie den Vertexshader `MainT_VS.glsl`, sodass er eine uniform Variable `translate` vom Typ `mat4` bereitstellt und mit dessen Hilfe die Position des Vertex verändert.

Musterlösung vom 23.05.2012:

Die Quellen finden sich hier: `/home/cg/2012/Uebung/Blatt4/Lsg/CG12Blatt4.zip`

Aufgabe 4.2 (30 Punkte)

Kopieren Sie die Klasse `main.TranslateNGon` nach `main.TranslateAndRotatePolygon` und den Vertexshader `MainT_VS.glsl` nach `MainTR_VS.glsl`.

1. Lassen Sie das Programm den Vertexshader `MainTR_VS.glsl` und den Fragmentshader `MainBrightness_FS.glsl` in der Methode `main()` laden.
2. Erweitern Sie die Klasse um eine Klassenvariable `rotation` vom Typ `float`, in der die aktuelle Rotation des 12-Ecks gespeichert wird.
3. Erweitern Sie die Methoden `handleInput()` und `updateUniforms()`, sodass sie folgende Funktionen zusätzlich erfüllen:
 - (a) In `handleInput()` soll nun zusätzlich geprüft werden, ob die Tasten `Q` oder `E` gedrückt sind und damit die Variable `rotation` gegen bzw. mit dem Uhrzeigersinn verändert werden.
 - (b) In `updateUniforms()` soll zusätzlich aus der Variable `rotation` eine Rotationsmatrix um die `z`-Achse erzeugt werden und diese als uniform Variable `rotate` vom Typ `mat4` dem Shaderprogramm übergeben werden.
 - (c) Außerdem soll der konstante Vektor $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ als uniform Variable `inverseLightDir` vom Typ `vec3` dem Shaderprogramm übergeben werden.
 - (d) Verändern Sie den Vertexshader `MainTR_VS.glsl`, sodass er eine uniform Variable `rotate` vom Typ `mat4` bereitstellt und mit dessen Hilfe die Rotation der Vertexposition sowie der -normalen verändert.
 - (e) Außerdem soll die out Variable `brightness` gesetzt werden. Implementieren Sie dazu die Beleuchtungsformel aus dem Exkurs über das Lighting der Vorlesung im Vertexshader.

Musterlösung vom 23.05.2012:

Die Quellen finden sich hier: `/home/cg/2012/Uebung/Blatt4/Lsg/CG12Blatt4.zip`

Aufgabe 4.3 (25 Punkte)

Kopieren Sie die Klasse `main.TranslateAndRotatePolygon` nach `main.TranslateRotateAndScalePolygon` und den Vertexshader `MainTR_VS.glsl` nach `MainTRS_VS.glsl`.

1. Lassen Sie das Programm den Vertexshader `MainTRS_VS.glsl` und den Fragmentshader `MainBrightness_FS.glsl` in der Methode `main()` laden.
2. Erweitern Sie die Klasse um eine Klassenvariable `scalation` vom Typ `org.lwjgl.util.vector.Vector2f`, in der die aktuelle Skalierung in `x`- und `y`-Richtung des 12-Ecks gespeichert wird.
3. Erweitern Sie die Methoden `handleInput()` und `updateUniforms()`, sodass sie folgende Funktionen zusätzlich erfüllen:
 - (a) In `handleInput()` soll nun zusätzlich geprüft werden, ob die Tasten `↑`, `↓`, `→` oder `←` gedrückt sind und damit die Variable `scalation` verändert werden.

- (b) In `updateUniforms()` soll zusätzlich aus der Variable `scalation` eine Skalierungsmatrix erzeugt werden und diese als uniform Variable `scale` vom Typ `mat4` dem Shaderprogramm übergeben werden.
 - (c) Verändern Sie den Vertexshader `MainTRS_VS.glsl`, sodass er eine uniform Variable `scale` vom Typ `mat4` bereitstellt und mit dessen Hilfe die Skalierung der Vertexposition sowie der -normalen verändert. Normalisieren sie die Normale anschließend wieder.
4. **Zusatzaufgabe:** Implementieren Sie, dass man mittels der `NUM4` bzw. `NUM6` die Lichtausrichtung rotieren kann. **Hinweis:** Auch ohne Bearbeitung dieser Aufgabe können Sie 100% der Punkte erreichen.

Musterlösung vom 23.05.2012:

Die Quellen finden sich hier: `/home/cg/2012/Uebung/Blatt4/Lsg/CG12Blatt4.zip`

Aufgabe 4.4 (20 Punkte)

Beantworten Sie Ihrem Tutor Fragen zur Vorlesung.