



Aufgabenblatt 4

13.07.2018

Aufgabe 8: Transformationen: Skalierung, Verschiebung, Rotationen

Hinweis: Ausgabe ist so verändert, dass die Transformationen auf das gerade selektierte Objekt im Szenegraphen angewendet werden sollen!

In dieser Aufgabe sollen Transformationen in homogenen Koordinaten für ein einzelnes gerade selektiertes Objekt realisiert werden. Es ist nicht nötig die Transformation selbst auf die Punkte anzuwenden, sondern es soll die jeweils anzuwendende Transformationsmatrix berechnet werden, der Renderer führt die Transformation dann aus. Diese Transformationsmatrix wird dann im Szenegraph gespeichert und in `CgSceneControl::renderObjects()` mit der Matrix-Stack Funktionalität beim Render-Aufruf übergeben. Initial ist das gemäß Aufgabe 7 eine Einheitsmatrix, diese sollen Sie entsprechend ersetzen/verändern. Generell sind die Berechnungen der Transformationsmatrizen eine Funktionalität die vom `Control` übernommen und z.B. in einer statischen Hilfsklasse implementiert werden können, das Objekt in lokalen Koordinaten wird dabei gar nicht verändert.

- Implementieren Sie eine Funktionalität die per Tastendruck ein Objekt im Szenegraph selektiert. Dies könnte z.B. so realisiert sein, dass ein Tastendruck immer zum nächsten Objekt weiterschaltet (Reihenfolge beliebig). Ändern Sie die Farbe des gerade selektierten Objektes, so dass die Selektion visuell überprüft werden kann. Key-Events sind für das Observer-Pattern schon implementiert und können in der `HandleEvent()` - Methode direkt abgefragt werden.
- Implementieren Sie eine Skalierungs-Funktionalität, diese soll über Tastendruck (z.B. mit +/-) gesteuert werden, Die Skalierung soll in jeder Koordinaten-Richtung gleich sein, aber prinzipiell soll beliebige Skalierung in jeder Koordinatenrichtung vorgesehen werden (Argument ist also ein `vec3`). Erstellen Sie die entsprechende Matrix und speichern Sie diese in das gerade selektierte Objekt.
- Analog soll per Tastendruck (z.B. x,y,z- Taste) eine Rotation um die lokale x-, y- oder z-Achse des Objekts ausgeführt werden können. Um diese Funktionalität optisch besser einschätzen zu können zeichnen Sie die lokalen Koordinatenachsen des jeweils gerade selektierten Objektes als 3D- Objekte in 3 verschiedenen Farben mit Hilfe der Funktionalität für Zylinder aus der vorherigen Aufgabe.
- Ergänzen Sie eine Funktionalität zur Verschiebung des selektierten Objektes, der Verschiebungsvektor soll dabei über das GUI gewählt werden, die Verschiebung über ein entsprechendes Event über das Observer-Pattern ausgelöst werden.
- Erweitern Sie ihre Implementierung derart, dass die Kombination von Transformationen korrekt abgearbeitet wird, also mehrere Transformationen in die lokale Transformationsmatrix aufmultipliziert werden können. Dazu ist es nötig z.B. das „Wandern“ beim Skalieren eines Objektes durch eine entsprechende Translation zu vermeiden etc...

Aufgabe 9: Rotation um eine beliebige Achse

Hinweis: Ausgabe ist so verändert, dass die Rotation auf das selektierte Objekt im Szenegraphen angewendet werden soll! Unit-Test fällt weg.

Implementieren Sie die Rotation um eine beliebige Achse gemäß der Herleitung aus der Vorlesung.

- a) Erstellen Sie die notwendigen Einzel-Matrizen (wieder in statischer Hilfsklasse) und multiplizieren Sie diese miteinander um die Gesamt-Rotationsmatrix zu erhalten. Sie sollen dies zum Verständnis der Funktionalität tatsächlich Schritt für Schritt von Hand machen, obwohl es eine fertige Rotationsmatrix in der glm-Library gibt. Die Rotationsachse und der Winkel soll per GUI einstellbar sein. Wenden Sie dann die Rotationsmatrix auf die Transformationsmatrix des gerade selektierten Objektes an.

Per Button-Klick soll das gerade selektierte Objekt zur Veranschaulichung entsprechend rotiert werden. Die Rotationsachse soll mit Hilfe der vorhandenen Funktionalität für Zylinder als 3D Objekt in den lokalen Koordinaten des selektierten Objektes gezeichnet werden.

Aufgabe 10: Implementierung einer Szene

Testen Sie ihre Implementierung des Szenegraphen, indem Sie die in der Vorlesung vorstellte Szenerie mit dem Schachspieler nach-konstruieren. Zur Umsetzung soll sowohl ihr Szenegraph mit entsprechenden Objekt-Hierarchien als auch ihre selbst implementierten Transformationen genutzt werden. Einige Objekte sollen selbst erzeugt werden (Tisch, Stuhl und die Bauern-Schachfiguren), alle anderen Objekte (z.B. sitzende Person, die anderen Schachfiguren) können als Netz geladen werden. Entsprechende obj-Dateien sind im Datenverzeichnis CgData vorhanden.

- Insgesamt soll ein Tisch mit Schachbrett und Figuren darauf, einer Kiste mit weiteren Schachfiguren, sowie ein Stuhl und eine sitzende Person erzeugt werden. Dabei sollen so wenige geometrische Objekte wie möglich verwendet werden (also das was den Szenegraphen mit Referenzierung der Geometrie-Objekte ausmacht). Dazu gehen sie wie folgt vor



- Der Stuhl auf dem die Person sitzt soll aus Zylindern (Stuhlbeine, Rahmen) und Würfeln (Sitzfläche, Lehne) konstruiert werden. Die notwendigen Transformationen und Geometrie-Objekte sollen mit passender Hierarchie im Szenegraphen erzeugt werden. Im Idealfall gibt es also *einen* Zylinder, *einen* Würfel, etc. der jeweils an unterschiedlichen Positionen in unterschiedlich transformierter Form benutzt werden.

- Die sitzende Person kann als Netz aus einer Datei geladen werden. Sie sollte als Kindknoten an den Stuhl angefügt werden. Wird über den Selektionsmechanismus der Stuhl gewählt und transformiert, soll dieselbe Transformation auch auf die sitzende Person angewendet werden.
- Der Tisch soll aus Zylindern oder Würfeln (Gestell) und Würfeln (für die Bretter die, die Tischfläche repräsentieren) erstellt werden. Entsprechende Transformationen sind im Szenegraphen abzulegen.
- Platzieren Sie einen entsprechend transformierten Würfel als Schachbrett auf dem Tisch. Als Kind-Knoten sollen einige rotationssymmetrische Schachfiguren als Bauern (konstruiert mit der Funktionalität aus Aufgabe 5) auf das Schachbrett positioniert werden (nur ein geometrisches Objekt für alle Bauern!). Zusätzlich soll eine König liegend auf dem Schachbrett realisiert werden (Laden aus obj Datei)
- Die weiteren Figuren können ebenfalls aus obj-Dateien eingelesen werden. Achten Sie auch hier darauf, dass Sie jeweils nur eine einzige geometrische Instanz für die 4 Springer, 4 Läufer, 4 Türme, 2 Damen verwenden. Weisen Sie 2 den Objekten 2 verschiedene Farben (hell/dunkel) zu um eine typische Schach-Szene zu erzeugen.
- Erzeugen Sie eine Kiste (Seitenwände aus transformierten Würfeln) und platzieren Sie einige Figuren darin („wild“ rotiert unter Benutzung ihrer eigenen Rotationsmatrix aus Aufgabe 9) .
- Für die Kiste und das Schachbrett gilt wieder: wird das Eltern-Objekt im Szenegraphen selektiert, soll die anzuwendende Transformation auch auf die Kind-angewendet werden. Dies soll anhand der Verschiebung und Rotation der jeweiligen Objekte und ihrer Kinder demonstriert werden können (über Tastendruck x,y,z für die Rotation und 3 weitere Tasten für die Translation).