

Übungsblatt 2

„Computergrafik“

Besprechung und Abgabe spätestens am 29.11.2016, F033.

Programmgerüst für die Übung:

Laden Sie sich die zip-Datei zur Übung mit den drei Dateien `main.cpp`, `vec.h` und `mat.h` von der web-Seite der Vorlesung:

- Die Datei `main.cpp` enthält ein Programmgerüst, das die Benutzung eines Timer-Callbacks, den so genannten Doublebuffer und mehrere Displayfunktionen enthält.
- Die Dateien `vec.h` und `mat.h` stellen Implementierungen einer Vektor- und einer Matrixklasse dar. Einige Funktionen sind bereits exemplarisch implementiert. Implementieren Sie bei Bedarf die fehlenden Funktionen. Orientieren Sie sich dabei an dem existierenden Code.

Für die Abgabe soll es möglich sein, zwischen den Visualisierungen aus den Aufgaben 3 und 4 per Tastatur umzuschalten! Die Funktionalität aus beiden Aufgaben wird anhand des Sourcecodes überprüft!

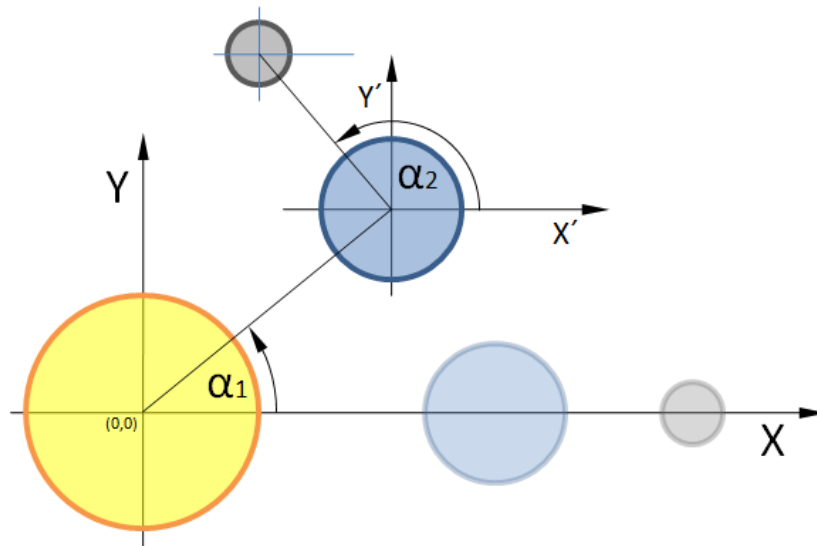


Abbildung 1 Aufbau des Sonnensystems für Aufgaben 3 und 4.



Aufgabe 3 (Transformationen in 2D)

5 Punkte

Implementieren Sie die Visualisierung eines animierten Sonnensystems. Das Sonnensystem soll aus einer Sonne, einem Planeten und einem Mond bestehen, wie in Abbildung 1. Der Planet soll um die Sonne rotieren und der Mond um den Planeten. Beide Rotationen sollen gleichzeitig stattfinden.

Für diese Aufgabe stehen Ihnen nur die Klassen `CVec2*` und `CMat2*` zur Verfügung, d.h. ausschließlich 2d-Objekte. Sie können die Klassen `CVec2i` und `CVec2f` nicht zusammen benutzen. Um dennoch ganzzahlige Daten an die Funktion `glVertex2i` übergeben zu können, nutzen Sie einen expliziten Type-Cast.

Herangehensweise:

- Schreiben Sie zuerst Ihre Bresenham-Algorithmen aus Übung 1 um, damit sie auf Pixelbasis funktionieren. Benutzen sie zum Setzen der Pixel die `glBegin(GL_POINTS), glEnd()` Funktionalität zusammen mit `glVertex2i`.
- Definieren Sie die drei Positionen der Himmelskörper als `CVec2f`-Typen. Definieren Sie weiter zwei Winkel und die zugehörigen Winkelinkremente (für Planet und Mond). Zeichnen Sie die drei Körper mit Hilfe der Bresenham-Funktionen auf dem Bildschirm. Die Sonne sollte sich dabei im Ursprung befinden. Planet und Mond sollen initial entlang der positiven X-Achse (horizontal) angeordnet sein.
- Implementieren Sie eine Funktion, um einen beliebigen Punkt um den Ursprung zu rotieren. Testen Sie diese Funktion, indem Sie den Planeten um die Sonne rotieren lassen.
- Implementieren Sie eine Funktion, um einen beliebigen Punkt um einen beliebigen anderen Punkt rotieren zu lassen. Testen Sie diese Funktion, indem Sie den Mond um den Planeten rotieren lassen.
- Fügen Sie alle Funktionen zusammen, so dass der Planet um die Sonne und gleichzeitig der Mond um den Planeten rotiert. Realisieren Sie diese Funktionalitäten mit einer Display-Funktion `display1`.

Aufgabe 4 (Homogene Koordinaten)

5 Punkte

Implementieren Sie die in Aufgabe 3 bereits erstellten Transformationen nun zusätzlich in homogenen Koordinaten und fassen Sie die Funktionalität in einer Funktion `display2` zusammen.

Sie dürfen dazu lediglich 3x3 Matrizen und 3d Vektoren (homogene Koordinaten) verwenden. Stellen Sie dabei insbesondere die Rotation um einen beliebigen Punkt als eine einzige Matrix (Produkt aus mehreren Matrizen) dar!

Besprechung und Abgabe spätestens am 29.11.2016, F033.