HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Konstanz, 11.09.2017

Assignment 2

„Computer graphics“

Deadline 28.11.2017, F033.

Programing frame-work:
Download the programing frame-work from the web-page of the lecture. The zip-file contains three files `main.cpp`, `vec.h` and `mat.h`:

- The file `main.cpp` contains the programming frame-work, which supports the usage of the timer-callbacks, the so-called double-buffer and several display functions.
- The files `vec.h` and `mat.h` provide implementations of vector and matrix classes. Some of their methods are already implemented. If necessary, extend the classes by further methods, following the examples in the existing code.

For the presentation of the program, implement a key-press event to switch between the visualizations of exercise 3 and 4! The functionality of both exercises will be evaluated by demonstration and source code inspection!

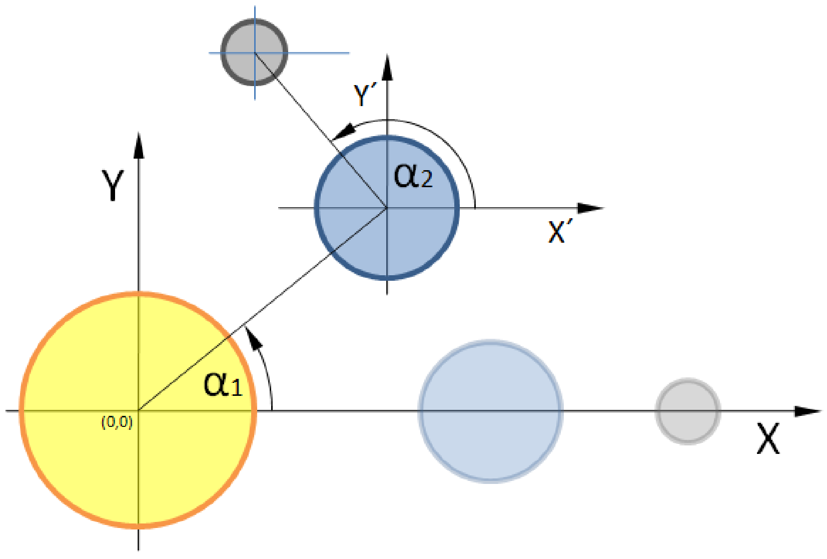

Figure 1 Setup of the solar system in exercises 3 and 4.

HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Exercise 3 (Transformations in 2d) 5 points

Implement a visualization of an animated solar system. The solar system contains a central sun, a planet and a moon as in Figure 1. The planet rotates around the sun while the moon rotates around the planet. Both rotations happen simultaneously.
You can use for the implementation the classes `CVec2*` and `CMat2*`, i.e. here use exclusively 2d-objects. The classes `CVec2i` and `CVec2f` cannot be used simultaneously. However, to allow for integer data for the function `glVertex2i`, use an explicit type-cast.

Approach:

- Adapt your implementation of the Bresenham algorithm from assignment 1 to work with pixels. To this end, the OpenGL methods `glBegin(GL_POINTS), glEnd()` in combination with `glVertex2i` to draw an individual pixel.
- Define three positions for the sun, the planet and the moon. Initially the sun is in the origin and the planet and moon on the positive x-axis. Define two angles and angle increments for the planet and the moon. Draw the three celestial bodies on the screen using the Bresenham algorithm.
- Implement a function to rotate an arbitrary point around the origin. Test your function rotating the planet around the sun.
- Implement a function to rotate an arbitrary point around another arbitrary. Test your function rotating the moon around the planet.
- Combine both functions to rotate the planet around the sun and simultaneously rotate the moon around the planet. Realize these functions in the display function `display1`.

Exercise 4 (Homogenous coordinates) 5 points

Implement the transformations from exercise in homogenous coordinates and combine them in the display function `display2`.
Here, use exclusively 3x3 matrices and 3d vectors (homogenous coordinates). In particular, implement the rotation around an arbitrary point as a single matrix (product of several matrices)!

Deadline 28.11.2017, F033.