

CS334: Assignment #1 – Linear Algebra

Out: Jan 25, 2008

Back/Due: Feb 6, 2008

Objective:

This objective of this assignment is to implement a 3D vector class, a 3D point class, a 4x4 matrix class, and operations on these data elements. This assignment will be implemented on top of the previous assignment.

Specifics:

1. 3D vector/point class (remember, they are not the same...)
 - Stores 3 floats.
 - Constructor from 3 floats.
 - Read/write access to elements with square brackets operator.
 - Normalization (for vector)
 - Length (for vector)
 - Dot product (for vector)
 - Cross product (for vector)
 - Multiplication/division with scalar (use operator).
 - Addition/subtraction with another vector/point (use operators).
2. 4x4 matrix class
 - Stores 16 floats
 - Constructor from 16 floats.
 - Read/write access to rows and columns.
 - Matrix transposition.
 - Matrix times column vector.
 - Matrix times column point.
 - Matrix multiplication.
 - Set up as a
 - x,y,z rotation matrix
 - translation matrix
 - uniform scale matrix
 - perspective projection
 - identity (should be default)
 - **Extra credit (10%):** matrix inversion.

Testing:

Show a “point” rotating about a “central axis” and provide GUI controls to change the viewpoint. The GUI (with GLUI) should support providing:

- rotation speed of the point about the central axis
- distance from point to central axis
- x/y/z rotation of the central axis
- x/y/z translation of a base point of the central axis

- scaling of the central axis
- x/y/z rotation of the camera
- x/y/z translation of the camera
- changing the field-of-view of the perspective projection used.
- **Extra credit:** if you did the extra credit, demonstrate it multiplying a matrix by its inverse and show the result.

By default use a field of view of 60 degrees and initialize other parameters to a reasonable value so that both the point and central axis are visible and occupy most of the window. Both the point and central axis should be rendered to the window using only `glDrawPixels` (e.g., you will have to do all computations using –your object classes–). The central axis should be drawn as a simple array of points (e.g., use a for-loop) and the point as a small “crosshair”.

If you have more questions, please see myself or the TA.