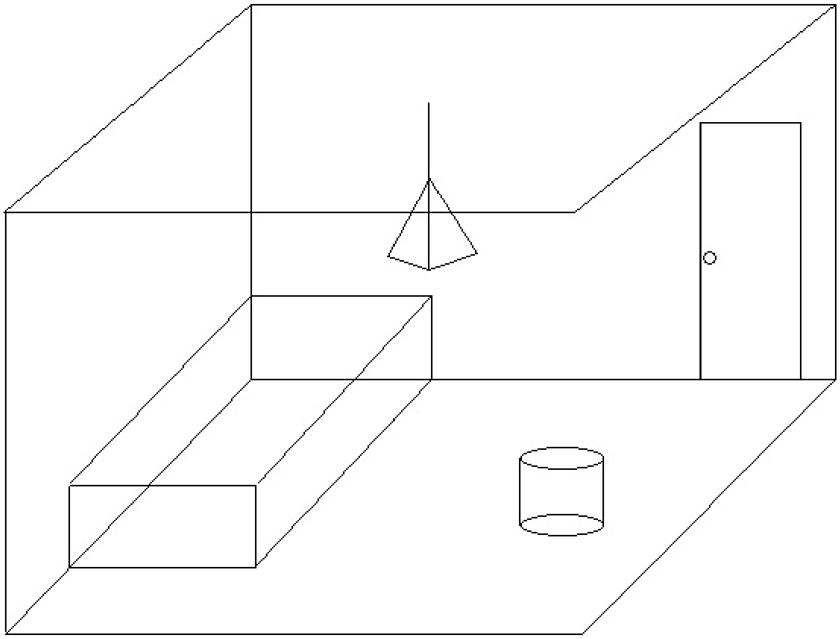
CS 488 – Fall 2013

Project 2 – Design a Bedroom

Due Date: Tuesday, 10/29/13 11:59pm



Summary:

The goal of this project is to design a simple bedroom. Your constructed room has a back and side walls, ceiling and floor. The furniture you are to use for this room is a bed next to the side wall, a door on the back wall, a cylinder object as a round stool and an extravagant chandelier (a pyramid object) attached to the ceiling hanging down from a wire.

Your room view should be shown clearly on an 800×800 display, but you need to design your canonical viewing volume to only see a portion of the room when rendering your scene. Your program should allow the user to use the ‘r’ key to rotate the room about the y axis; the ‘t’ key to toggle the camera view to a front view, side view, top view, and user defined perspective view; the ‘i’ key to zoom in and the ‘o’ key to zoom out along the direction of the COP to the center of the room.

Input:

You are to use the provided XML format for your database file. This file contains the specifications for your 3d objects which are the bed, stool and chandelier, and the 2d drawings which are the door and string holding the chandelier. In addition, you are to specify the VRP, VUP, and COP, Umin, Umax, Vmin, Vmax, and front and back plane clipping plane parameters, where your final viewing volume must be clipped. An example XML file will be provided on Piazza soon.

The world coordinate origin is located on bottom corner of the back and side wall with the y axis increasing up the wall and x axis runs parallel toward the back wall. A right handed coordinate system is used.

Requirements:

1. The implementation MUST be coherent with algorithms and formulas as explained in the book.

2. You can only use the primitives: GL_LINES and GL_LINE_LOOP to draw the object.

3. The program must be able to read all the specified inputs from the given XML format. I will provide a sample program that will be able to parse through the XML specified database and extract the project specified data. Feel free to use this code.

4. You must write your own code for the matrix operations and not use any GL function to perform perspective transformations. Each transformation of the vertex points on the object must be calculated through your own series of 4×4 matrices i.e. translation, rotation, shear, and scale matrices (these series of transformation matrices can be calculated into one matrix)

5. The final projection must be an orthographic projection, since you implemented perspective with your own transformations; therefore, you will need to use gluOrtho2D, you cannot use any of the 3D viewing volumes, such as glFrustum, glOrtho or gluPerspective.

Suggestions:

Suggested steps to build the program are as follows:

1. Start out building your own matrix arithmetic-library.
2. Matrix operations: initialize a zero or an identity matrix; add and multiply two matrices.
3. Vector operations: compare the normalized form of a vector; cross product of two vectors.
4. Mixed operations: multiply a scalar by a vector; multiply a matrix by a vector.
5. Transformations: rotation matrix given an angle about the x, y or z axis; rotation matrix given the Rx, Ry and Rz components; scale; translate; shear; projection matrix transforming a perspective into a parallel view volume
6. Transformations from Cartesian to Homogeneous coordinates and vice versa.
7. Using your matrix library, carefully implement the sequence of required transformations.
8. Separately implement and test a homogenous clipping algorithm; integrate it into the main program

Compilation/Programming-Style

The program must be written using C/C++ using the GLUT library. You can program using any operating system (Windows, Linux and Mac), and you can use any IDE (Visual Studio, Eclipse, Xcode and so on). However, your final product must compile on the CS Computer Labs computers in room 2254 SEL. You can use your CS account on these machines to confirm that this requirement is fulfilled. Programming style is essential; you should document your code so that I can understand the important steps/algorithms in your code. Use block comments to clarify a section of code and line comments to explain a technical detail on the implementation. You should also format and organize your code in a readable fashion. Without documentation or ReadMe file, you will lose 10% of your final grade as penalty.

Submission

You will submit your project on Piazza. Please submit only ONE archive file (zip or tar) which includes the source code, possible header files, ReadMe file.

ReadMe

You MUST include a readme file which summarized what you did. In this file you should also specify the algorithm(s) you used and document any other information that is relevant to my understanding your algorithm. The algorithms you explain in the readme file should coordinate with your comments within the program.

Grading

This programming assignment is worth 5% of the final grade (5 points). The maximum value possible for an assignment will be reduced if handed-in late. If the assignment is turned in up to 24 hours late this maximum value is reduced by a 50% penalty (2.5 points). If an assignment is turned in up to 48 hours late this maximum value will be reduced by a 100% penalty (0 points); this also means that a program not 100% correct can receive a value less than zero but not less than -5 points on this assignment. An assignment turned in more than 48 hours late or not turned in at all will result in student losing an additional 100% loss (that means you get a -5 points on this assignment).

Grading is based on the stated requirements including comments, readme and programming style. You get 3 points for constructing the play room with the furniture, and 2 points for user interactions (rotation, view changing and zoom in/out). A 100% working program handed-in on time does not necessarily receive full credit. A non-working program handed-in on time does not necessarily receive 0 points. Failing to provide sufficient cogent comments and/or a poor programming style will result in a penalization.

Extra Credit

Use the ‘a’ key to rotate your object about a user specified axis of rotation for 1 point, and add a sphere to represent a beach ball into the room for 0.5 point.