# CS 3353 Laboratory Problem Set 1

Due September 29, 2009

These problems are to be done on an individual basis following the Trinity University Academic Integrity Policy or Trinity University Honor Code.

## Academic Integrity and Honor Code

All students are covered by a policy that prohibits dishonesty in academic work. The Academic Integrity Policy (AIP) covers all students who entered Trinity before the Fall of 2004. The Academic Honor Code covers all those who entered the Fall of 2004 or later. The Integrity Policy and the Code share many features: each asserts that the academic community is based on honesty and trust; each contains the same violations; each provides for a procedure to determine if a violation has occurred and what the punishment will be; each provides for an appeal process. The main difference is that the faculty implements the AIP while the Honor Code is implemented by the Academic Honor Council. Under the Academic Integrity Policy, the faculty member determines whether a violation has occurred as well as the punishment for the violation (if any) within certain guidelines. Under the Honor Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to investigate, adjudicate, and assign a punishment within certain guidelines if a violation has been verified. Students who are under the Honor Code are required to pledge all written work that is submitted for a grade: On my honor, I have neither given nor received any unauthorized assistance on this work and heir signature. The pledge may be abbreviated pledged with a signature.

Laboratory problems should be submitted electronically (e-mail to cs3353@ariel.cs.trinity.edu) on or before class time on the due date and should contain a problem write-up, source code to any programs and data sets used in solving the problem. The submitted files should be ASCII text files having Unix end-of-line characters (please convert all Windows and Mac text files to Unix format–I have found that Emacs seems to do a reasonable job of such conversions). If several files need to be submitted, put them in a directory having name *your-last-name-problem-set-number* and create a tar archive of this file system and attach it to your e-mail problem submission.

## Using the 2D Viewing Transformation

In this laboratory problem we wish to learn to use the 2D viewing transformation as well as functions which will draw pictures made from 2D objects given a representation for pictures which is stored in a picture file. One task is to design and implement a suitable data structure for objects. A second task is to design and implement a data structure for pictures. A third task is to use the 2D viewing transformation. Several suggestions are given for the data structure and organization of the functions. These are just guidelines. Feel free to innovate, but keep in mind that the functionality described below should be implemented.

### Data Structures for Objects and Pictures

We assume that an object may be drawn as a collection of line segments. The geometry of a simple object might be described by a a matrix of coordinates as shown in Figure 1.

Each row of the matrix would represent a coordinate in the object. When an object is drawn, a move-to the first point is performed followed by line-to's for each of the succeeding points. It may be necessarry to

$$[X\ Y]_{N \times 2}$$

Figure 1: A Matrix of Coordinates

model complex objects as collections of these simple objects. It may also be appropriate to add an object transformation matrix to the definition of a complex object to facilitate a change of coordinate system for the object.

A data structure for a picture might include a geometric description of each object in the picture. Such a description might consist of a pointer to the object and a transformation matrix which positions that object in the picture. This transformation maps the local coordinate system of the object into its position (in world coordinates) in the picture. During the drawing operation, this transformation must be multiplied by the object's local coordinate system transformation.

Use the functions provided in OpenGL and any others that you need to define to produce a program which will draw a picture which is similar to the office floor plan shown in Figure 2 below. The program should set up the window and viewport and a picture structure which describes the office floor plan. The program should then draw that picture.

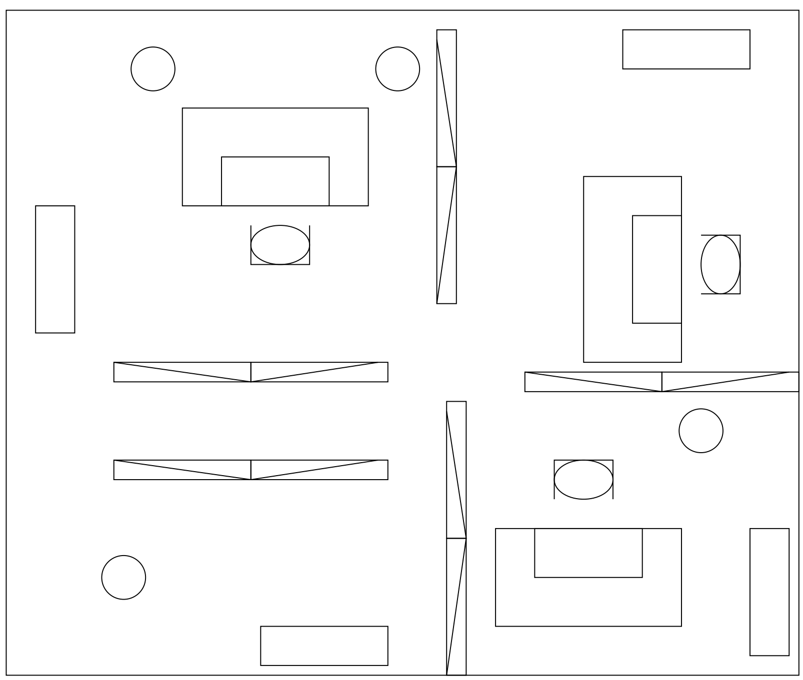

Figure 2: An Office Floor Plan

Problem Set 1 Solution [ HTML ] [ PS ] [ PDF ]