CS 373 2014 Assignment 1

Due date: Monday 14th April 2014 12.00 pm (Noon)

This assignment is worth 8.33 % of your final mark. Submit all the answers compressed in a single ZIP archive to the assignment drop box on the web.

This assignment builds on your self-learning skills in C++, the homogeneous transform, geometry and colour lectures content of this course.

The hand-written solutions Q1 must be submitted as PDF, first on written on paper and bundled with Q2, Q3 and Q4 code for submission. You should submit your work as a zip archive. Name the zip or 7z archive using your upi (say lsky001.7z for Luke Skywalker). Each archive should only contain all .cpp, .h files to run your code (e.g. q1.cpp, q1.h, q2.cpp, q2.h, etc…).
Make sure your code compiles in the CS lab's machine.

Question 1: Dot, Cross Product, vector and matrix multiplication (10 marks)

Given the generic vectors a = $(a1, a2, a3)^T$ and b = $(b1, b2, b3)^T$ compute and **hand-write** the full solution (SHOW YOUR WORKING) for the following operations:

a) Produce the magnitude of vector a×b (cross product of vector a and b)
b) Produce the matrix ab^T
c) Produce the scalar a^Tb
d) Produce |a| and |b|
e) Produce the a.(a×b) (i.e. the dot product of vector a by the cross product of vector a by b)

Note that if you do not show intermediate calculations you will not receive marks for the relevant Q1 questions

Question 2: C++ implementation for the area and height of a parallelogram. (20 marks)

Using all the functions provided in matrix.cpp and matrixops.cpp and using the skeleton code provided in Q2.cpp, produce the additional code to compute the area and height of a parallelogram defined by vectors a and b (as per lecture 4 slide 4).

Question 3: C++ implementation for the normal of a triangle given the vertices a, b and c coordinates (20 marks)

Using your implementation of Question 2, all the functions provided in matrix.cpp and matrixops.cpp and the skeleton code provided in Q3.cpp provide the code to compute the normal (of magnitude 1) of the triangle defined by vertices a, b and c. Make sure to provide the right direction for the normal as exemplified in lecture 3 slide 7.

Question 4: C++ implementation for the normal of a polygon given the vertices' coordinates (25 marks)

Using your implementation of Question 3, all the functions provided in matrix.cpp and matrixops.cpp and the skeleton code provided in Q4.cpp provide the code to compute the normal (of magnitude 1) of a polygon defined by its vertices. Make sure to provide the right direction for the normal as exemplified in lecture 3 slide 7. The core of this question is to order the vertices so they are listed in counterclockwise order in accordance with the material presented in lecture 3. To try to tackle this problem, first place yourself in the ideal case where all vertices are on the same plane. You need to:

1. Order the vertices in counter clockwise way
2. Compute the robust normal as per L3 slide 7

For sake of simplicity, in this question you will limit your program to handling polygons with 5 vertices.

Question 5: Find the weakest link!! (25 marks)

Using your implementation of Question 4, all the functions provided in matrix.cpp and matrixops.cpp and the skeleton code provided in Q4.cpp provide the code to find the weakest vertex of the polygon. This vertex is defined as providing the normal to the polygon which deviates the most from the robust normal to the polygon as computed in Q4. You need to:

1. Order the vertices in counter clockwise way
2. Compute the normal to the polygon at each vertex (L3 slide 7)
3. Compute the robust normal as per Q4
4. Find the weakest vertex
5. Remove the weakest vertex from the polygon
6. Order the vertices of the new polygon
7. Recompute the robust normal to the new polygon

Make sure to provide the right direction for the normal as exemplified in lecture 3 slide 7. **For sake of simplicity, in this question you will limit your program to handling polygons with 5 vertices**.