<More in 3D> Software Design Document

CS2300 Section 4 Fall 2021

<Brodie Crawford>

Project Description

This project is about lines, vectors, triangles, points and planes. I will complete three different parts, all of which show a different aspect of programming 3D data. In Part A, I will be performing culling on several triangles inputted from a file to optimize rendering speed. The first line will give me the eye location and the light direction, and all lines afterwards will give me triangular planar facets. After that, I will compute the light intensity depending on which direction I’m looking from and the vector normal to the triangle. For Part B, I will perform parallel and perspective projections of points onto planes. The first line will make a plane and projection direction (used if performing a parallel projection), and all lines afterwards will be points to project onto the given plane. Lastly, for Part C, I will use two different ways of taking input to first find the distance of a point from a plane and then find the intersection point between a line and a triangle, if it exists. The first way of taking input will give me a plane and a point for each line of the file, and I will output the distance of the given point to the given plane. The next subpart requires me to take the first six numbers in the file as two points making a line, and all subsequent lines afterwards will give me a triangle (one per line), with which I can test it against the line to see if the line intersects and where.

Approach

To complete this project, I will have three different executable files, each corresponding to one part (A, B, or C). For all parts, I will have 4 sections, with sections to initialize input and output files, create scanners and writers to operate on them, a section with variables to hold input and output data, and a formatter that will print my output uniformly to the output file. For each part, I will scan numbers for the first line before I use a while loop that will scan each line for me and transfer the data to different double arrays I have set up to receive it. Then, I will transfer the data over to my modules that will do the calculations for me and give me the results. After that, I will write my data to a file with my formatter, sometimes with an if statement to test the data before I write it. My modules will not be exclusively called by main, because some of my modules will be called by other modules for code optimization.

Detailed Design

Programming Language Used: Java

Editor: Eclipse

**Programming Language Description:**

Java makes it easy to read and write to files with large amounts of data, because no memory allocation is needed like for any of the C languages, and it has predefined structues like Scanner, BufferedWriter, and ArrayList that can easily be used.

**Bcrawford\_partA modules:**

public static void main: reads an eye location, a light direction, and a series of triangle vertices to different variables

public static double[] findCentroid: gives me the centroid of the triangle made up by the three points passed as parameters

public static double[] viewVector: returns the view vector given the eye location and the centroid

public static double[] normalVector: gives me the vector normal to the triangle made up by the three points passed as parameters

public static double calculateIntensity: returns the light intensity given the light direction and the normal vector of a triangle

public static double[] crossProduct: returns the cross product of the two passed vectors

public static double dotProduct: returns the dot product of the two passed vectors

public static double[] subtractVectors: returns the vector corresponding to the subtraction of the second passed vector from the first passed vector

public static double vectorLength: returns the length of the passed vector

**Bcrawford\_partB modules:**

public static void main: reads a point on the plane, a normal vector, a projection direction, and a series of points into different variables

public static double[] parallelProjection: returns the new location of the point passed using the point on the plane, the normal vector, the projection direction, and the point to be projected

public static double[] perspectiveProjection: returns the new location of the point passed using the point on the plane, the normal vector, and the point to be projected

public static double dotProduct: returns the dot product of the two passed vectors

**Bcrawford\_partC modules:**

public static void main: First reading: reads points on a plane, vectors normal to the plane, and points off the plane. Second reading: reads two points making a line and then triangle vertices

public static double distPointToPlane: calculates the distance from the given point to the given plane and returns it

public static String lineAndTriangleIntersection: returns a string saying whether the given line intersects with the given triangle

public static double dotProduct: returns the dot product of the two passed vectors

public static double[] subtractVectors: returns the vector corresponding to the subtraction of the second passed vector from the first passed vector

public static double[][] inverseMatrix3x3: returns the inverse of the given 3 x 3 matrix

**Module Flowchart:**

Diagram

Description automatically generated

Diagram, schematic

Description automatically generated

**Key Data Structures:**

Double – stored lots of return values for me

Double[] – gave me a way to store many points and vectors

Double[][] – helped me store a matrix in one case

ArrayList<Double> and ArrayList<Integer> – helped me store culling and light intensity data temporarily in Part A

DecimalFormat – made all my output look nice

String – helped me print out some statements as a result of 3D operations to files