<Assignment 4 > Software Design Document

CS2300 Section 1 Spring 2022

<Daniel Frates>

Project Description

Describe the problem you are addressing in your own words.

For part a I have to take an eye location, a light direction and the vertices to triangles and determine whether to cull them. Then I have to determine the intensity

For part b I have to find the parallel projection of each point and the perspective projection of each point

For part c I have to find the distance from the point to the plane and then find where the point intersects with a triangle.

Approach

Describe how you solved the problem, providing an outline of the program logic.

For part a I read in the input file and took the eye location, light direction and the vertices of triangles and I found the centroid of each triangle. I then performed culling on the triangles. For intensity I found the light intensity based on the light direction and the normal.

For part b I read in the point on the plane, the normal to the plane, and the projection direction. From this I performed parallel projection of each point. Then I performed a perspective projection of each point.

For part c I read in the point on plane, the normal, and a point. I then computed the distance from that point to the plane. Then I read in two points and then found if they intersected with any of the triangles in the matrix.

Detailed Design

Programming Language

Describe the programming language (and version, if relevant). If there are unique features of the language that are important to your approach, describe them.

I used C# for this assignment because of my familiarity with the language. I have also worked with 2d arrays in C# and felt more comfortable using it then using any other language when it came to matrices. A unique feature of C# is the file I/O, it is in my opinion easier to read and write files with C#.

Modules

Provide a brief description of each module in your program, including the inputs, the outputs, and the actions that the module takes on the data.

Part A:
PartA(): Runs Part A
Culling(): Performs culling on a triangle
Intensity(): Finds the intensity from each normal
Normal(): Gets normal from 2 vectors
VectorDot(): Performs vector dot product
Shave(): Shaves matrix to column elements
Cross(): Performs cross poduct on vectors
Vector(): Creates vector from 2 points
Magnitude(): Gets the magnitude of the given vector
FindCentroid(): Finds the centroid of the given triangle
format of 3x1 matrix
EyeLocation(): Gets eye location from first row of matrix
format of 3x1 matrix
LightDirection(): Gets light direction from first row of matrix
format of 3x1 matrix

Part B:
PartB(): Runs part b from triangles
ParallelProjection(): Option A Parallel Projection of each point
PerspectiveProjection(): Option B Perspective Projection of each point
ToMatrix(): Takes a list of 3x1 matrices and creates a matrix
LoadPoints(): Loads points from input matrix
option 'A' loads points from row 1-n
option 'B' loads points from row 0 at 7-9 to row n
points are in a list of 3x1 matrices
VectorSubtract(): Performs vector subtraction on 2 vectors
PointOnPlane(): Gets point on plane from a row of matrix
format of 3x1 matrix
ProjectionDirection(): Gets projection direction from first row of matrix
format of 3x1 matrix

Part C:

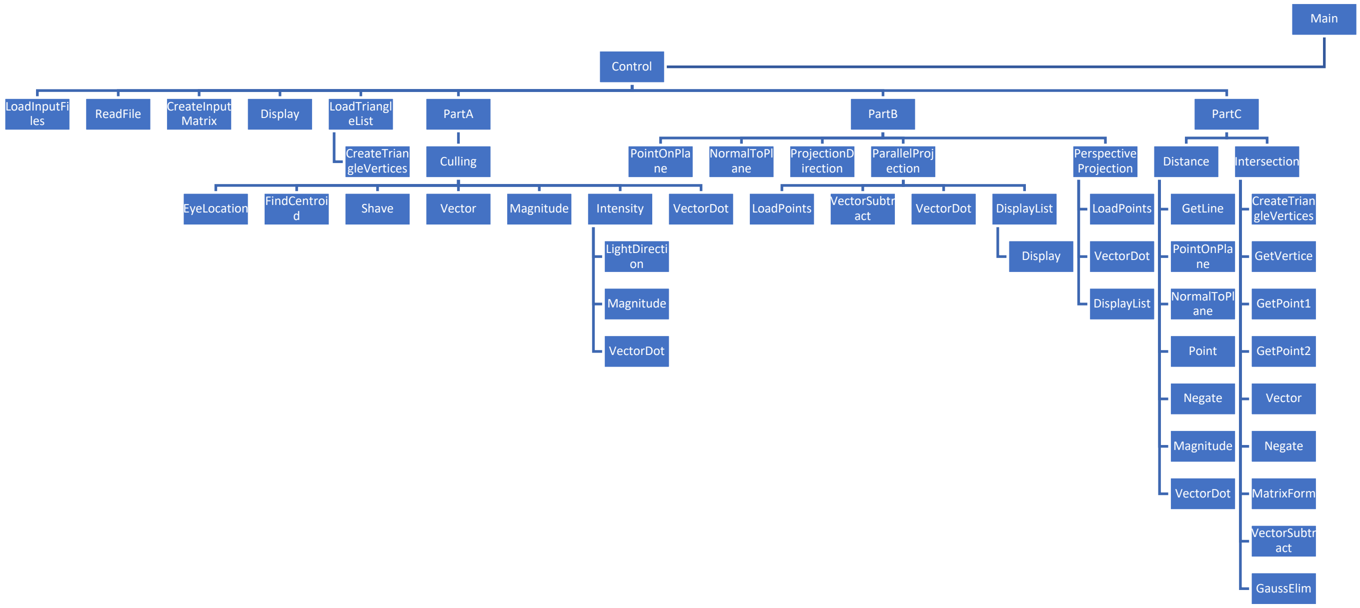
PartC(): Runs part C
Distance(): Finds the distance from the point to the plane
Intersection(): Finds if point intersects with triangle
MatrixForm(): Puts the vectors into matrix form as column vectors
GaussElim(): Performs gaussian elimination
format of 3x1 matrix
CreateEquation(): Creates the equation from the specified row of matrix and vector
format of nx1 matrix
AddVector(): Performs vector addition on 2 vectors
format of nx1 matrix
MatrixDot(): Performs matrix dot product on 2 matrices
GetPoint1(): Gets the point from column 0-2
format of 3x1 matrix
GetPoint2(): Gets the point from column 3-6
format of 3x1 matrix
GetVertice(): Gets the vertice of a specified triangle and a row
from the triangle. format of 3x1 matrix
Negate(): Turns all elements of vector to their negative value
GetLine(): Gets a row from matrix
format of 9x1 matrix
NormalToPlane(): Gets the normal to the plane from a specified row
format of 3x1 matrix
Point(): Gets the point from the specified row
format of 3x1 matrix

Helper Methods:

LoadTriangleList(): Creates a list of triangles from the vertices
format of 3x3 matrix
CreateTriangleVertices(): Creates the triangle vertice points from the matrix and a
specified index. format of 3x1 matrix
CreateInputMatrix(): Creates the input matrix from the 1d array
LoadInputFiles(): loads input files and stores them in an array of strings
ReadFile(): reads file and loads values into input matrix
DisplayList(): Displays a list of 2d arrays
Display(): displays matrix

Flowcharts

Provide a flowchart describing how the modules interact. If you have only one module, then use the flowchart to describe its main logic.



Key Data Structures

Describe the data structures that are important to your approach.

The data structures that were important to my approach were the 2D arrays used as matrices and lists to hold 2d arrays.

Test Description

Describe the input files used for testing your code, explaining why you're using each one for testing and how it shows your program is working correctly.

I used my own input file to make sure I could read in the file correctly and I used the input file given to use to test my operations.