Zhihua Wang G48459695

The code contain two part.

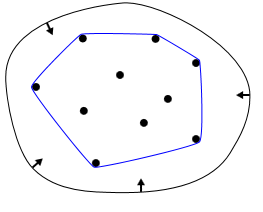
One part is under main directory, its function is to use classes such as Matrix class, Collision class to build animation. This part contain two files, SimpleGLUT.cpp, which is the entry of the program and the main code to build animation, and StdAfx.cpp, which includes stdafx.h file to include some precompiled files to accelerate program.

The other part contains 7 files, which are matrix.h, operation.h, vec3.h, vexhull.h, parameter.h, collision.h and model.h.

matrix.h is the code of my own Matrix class. The routines of my Matrix class include set value of a matrix, add two matrixes and multiply two matrixes.

Most code of vec3.h comes from project CG1\_LAB1. It defines vertex class. I overload one vec3’s operator \*, to define vec3 \* Matrix.

vexhull.h is the code of my Vexhull class, which’s function is to find the convex hull of any set of points. A set of points is defined to be convex if it contains the line segments connecting each pair of its points. The convex hull of a given set X is defined as the (unique) minimal convex set containing X. Here is an example.



I use the convex hull of an item to detect collision.

operation.h contains operations needed during program. Xproduct2d is to calculate the cross product of two vectors. Distance2d is to calculate the distance between two points. quicksort is the routine using quick sort algorithm to sort arrays. quicksort routine needs cmp as a parameter. In my program, quicksort is used to sort item’s points in polar coordinate in order to calculate the convex hull of the item.

parameter.h contains global parameters, such as number of items, locations of files containing items’ points, directions of items and so on.

collision.h is the code of my Collision class and Collisionitem class. Collision class is used to detect collision. When one item has collision with others, it rotates 0.05 and detects collision again. If there is still a collision, it rotates 0.05 again. Repeat this action until there is no collision and the item can move again. The angle of every rotation can be change if we need to do more precisely. We use this theory to detect two items collision. If two polygons’ projections do overlap in any axes, there is collision between two polygons, otherwise, there is no collision. And other theory tell us that we only need to detect the axes of two polygons’ every edges. Collisionitem class serves Collision class.

Most code of model.h comes from project CG1\_LAB1. It models items in the world. We use these models to do translation in the computer. I rewrite some function and fill some functions using my own class. For example, I fill the rotate routine using Matrix class.

Using w, s, a, d we can control car’s position.

Using up, down arrow we can control car’s scale.

Using left, right arrow we can control car’s direction.

We store model’s matrix in Model class. Whenever redraw pictures, we first replace current GL\_MODEL\_VIEW matrix of this matrix, then draw pictures. Whenever we apply a transformation on an item, we use a matrix recording this transformation and multiply this matrix with this model’s current matrix and replace current matrix of the result.