

**Problem Set 1**  
**CMSC 427 – Computer Graphics**  
**Assigned September 11, 2007**  
**Due September 25, 2007**

**Programming Exercise: Car 101**

You will program a simple vehicle that can move around a 3D scene. Think of it as a car that is driving on a plane that also contains some buildings. You will add buildings, and the ability to steer the car and change its speed. You'll also add a special somersault feature. We are including an executable showing my version of the finished project. Your version need not be identical in every detail. For example, you can vary the speed at which the car moves, how rapidly it steers, or the size and position of the buildings. However, keep the interface the same, using the keyboard functions we specify, so that we will be able to easily test the result.

We are giving you skeleton code that will display a rectanguloid on the screen. You will enhance this code to add more rectanguloids, to change the type of projection, and to allow yourself to navigate about the scene. Note that included in the code that we give you is a file `vector3.cpp` that contains many useful functions for vectors. There are skeleton functions for 3D rotation, which you might find it useful to write and use. We are also including code for the class "cube", which you may use in drawing cubes (actually, rectanguloids).

1. **10 points** Add more buildings. For this part, enrich the scene that you are looking at by adding at least four more buildings on the ground plane.
2. **20 points** Move forward at fixed speed. Now enhance the car so that it can move. The car should always move forward, in the direction in which it is looking. You should use `glutTimerFunc` so that you can control the speed in a machine independent way.
3. **20 points** Use mouse to change direction. Now add a feature allowing you to steer the car. Pressing on the left mouse button should turn you to the left. The right mouse button should steer you to the right.
4. **10 points** Change speed. Now allow keyboard input that will change the speed of the car. Pressing "s" should slow the car down, while pressing "f" should make it go faster.
5. **10 points** Somersault. The car also has a special trick. Pressing "m" causes it to do a somersault. While the car is somersaulting, it continues to move forward, and cannot be steered to the left or right. During a somersault, the car rotates about a direction perpendicular to the direction it is going in that is also perpendicular to the "up" direction.

6. **Challenge Problem (up to 20 points):** For extra credit, you can add additional features. You should avoid the topics of future problem sets (mapping textures onto the cubes, calculating collisions so that you don't run into the cubes, and getting objects other than the car to move). Some possible ideas: On the side or in another window, show the scene from overhead; add a curved bridge and have the car's position follow the bridge, so that as you go over it you are pointing up and then down (ie, you are not always pointing parallel to the ground plane as you drive); or add a roller coaster that the car can drive onto.

**Pencil and Paper exercises:** For all these problems, you might compute a matrix by taking the product of several matrices. If you do that, it is not necessary to multiply these out. It is fine to express your answer as the product of matrices.

1. **2D Viewing Position - 10 points.** As we move and change our viewing direction, we want to consider the world using a coordinate system that moves with us. In this problem, we consider a 2D world. We will describe locations using the original, world coordinate system, but we will also define a local coordinate system, in which the viewer's position is the origin, and the direction the viewer is looking in is the  $x$  axis.
  - a. Suppose we are positioned at  $(6,3)$ , in world coordinates. We have not rotated at all, so our viewing direction is  $(1,0)$ , the original  $x$  axis. Consider a point,  $p$ , with world coordinates  $(15,12)$ . What are  $p$ 's coordinates in our local coordinate system?
  - b. Give a matrix that will transform any point from its world coordinates to the local coordinate system in (a).
  - c. Now suppose we move to  $(8,2)$ , and are looking in the direction  $(0,1)$ . What are the coordinates of  $p$  in the new local coordinate system?
  - d. Give a matrix that will transform any point from its world coordinates to the local coordinate system in (c).
  - e. Suppose we are located at  $(4,2)$ , and a point with world coordinates  $(8,5)$  has coordinates  $(0,5)$  in the local coordinate system. What are the world coordinates of the point that has local coordinates  $(5,5)$ ?
  
2. **3D Viewing Position – 10 points.** Now we consider a 3D world. Assume that we again use our location as the origin of a local coordinate system, and that now our viewing direction is used as the  $z$  axis, and the “up” direction defines the  $y$  axis.
  - a. Suppose we are positioned at  $(2, 4, 3)$ . We have not rotated at all, so our viewing direction is the original  $z$  axis, and our up direction is the original  $y$  axis. Consider a point,  $p$ , with world coordinates  $(16, 6, 9)$ . What are  $p$ 's coordinates in our local coordinate system?
  - b. Give a matrix that will transform any point from its world coordinates to the local coordinate system in (a).

- c. Now suppose we are still positioned at  $(2,4,3)$ , but we are looking directly at a point located at  $(3,5,4)$ . Up is in the direction of the point  $(1,3,5)$ . Now what are  $p$ 's coordinates in our local coordinate system?
- d. Give a matrix that will transform any point from its world coordinates to the local coordinate system in (c).

**3. Rotation about an axis – 10 points.**

- a. Give a matrix that rotates things by 60 degrees about the  $x$  axis.
- b. Give a matrix that rotates things around an axis in the direction:  $\left(\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}\right)$ . So, for example, this rotation will not move the point at  $(1,0,1)$ . The rotation may be by any amount.
- c. Give a matrix that rotates about this axis by 60 degrees.

4. **Challenge Problem – 10 points:** One way to create a 3D rotation is to first rotate about the  $x$  axis, then rotate about the  $y$  axis, then rotate about the  $z$  axis. Suppose that  $R$  is a rotation created like this. Prove that  $R$  consists of a rotation about a fixed axis. That is, prove that there must be a line through the origin, so that any point on that line does not move when we perform  $R$ . (The rotation, then, will be about this fixed axis). You will probably need to recall some linear algebra for this.

Note: it is not enough to just say:  $R$  is a rotation, so it has a fixed axis. You have to prove that when we combine rotations, each of which has a fixed axis, we do, in fact wind up with a new rotation that also has a fixed axis.