Project Objectives:
In this project, you will review transformations and use them to create interactive animations.

Project Description:
Your second project requires you to implement your own transformation engine and to create your first animated model. Your application will have to read complex models provided in the Fundamentals of Computer Graphics (FCG) file format and manipulate the provided data. You are expected to implement the following components for this assignment.

Part 1
Transformations are fundamentally important in computer graphics and you should be very familiar with how the corresponding matrices are formed. Therefore, you are asked to implement your own transformation engine which replaces all native OpenGL transformations and matrix operations with your own functions (named fcg*)

```c
glTranslatef(x,y,z)   = fcgTranslate(x,y,z)
glRotate(angle, x,y,z) = fcgRotate(angle;x,y,z)
glScale(sx,sy,sz) = fcgScale(sx,sy,sz)
glMultMatrix() = fcgMultMatrix()
glPushMatrix() = fcgPushMatrix()
glPopMatrix() = fcgPopMatrix()
glLoadIdentity() = fcgLoadIdentity()
```

The only OpenGL call to matrix operations that you are allowed to use is glLoadMatrix(), which loads your own matrix into the OpenGL state machine. For this to work you will have to remember that the OpenGL transformations (translate, rotate, scale) internally perform two different tasks, namely (1) create the corresponding transformation matrix and (2) multiply it against the current model view or projection matrix.

Part 2
This part requires you to revisit and sharpen your knowledge in respect to transformations. Specifically, you are asked to animate the porsche1.fcg model.

(a) Using the up and down arrow keys on your keyboard, allow the user to specify a forward or backward rotation of the wheels. I.e. the wheels should keep spinning when one of the two keys is pressed.

(b) Simulate the result of the steering wheel being turned, using the left and right arrow keys. The front wheels should slowly turn left (within certain bounds) or right respectively. Allow the system to perform (a) and (b) simultaneously.

(c) Map (a)-(c) to mouse interactions, i.e. once the user presses the “d” key, the system switches from your normal navigation-mode into drive-mode and mouse movements are mapped to the wheels. Assume that the original position of the mouse, when the trigger key is pressed, represents the car with its parking brake on. When you now push the mouse forward the wheels should spin slowly and accelerate as a function of how far the mouse is moved from the original position. Vice versa, if you move the mouse towards you the rotation should slow down, stop when reaching neutral and eventually reverse. Initially your car can stay in the same location while the wheels are spinning.

HAVE FUN !!!!