EECS 351-1: Introduction to Computer Graphics
Instructor: Jack Tumblin

ProjectA: Shape

Winter 2012

Your mission in Project A is to a) learn how to draw moving, turning possibly jointed colored shapes with openGL's basic drawing primitives (lines, triangles, etc) using arrays of 3D vertex coordinates, b) to use the GL_MODELVIEW matrix stack to transform them interactively, c) ensure that parts of your image move continuously without user input (animation) and d) make these parts and possibly other parts move in response to keyboard and mouse inputs.

You may choose to draw ANYTHING for your project that meets the basic requirements. **our assignment is to use openGL to make an *interactive* drawing that shows something *you* find interesting and compelling.** An octopus? Fractal trees that grow, wave in the wind? An N-legged walking creature whose legs consist of M segments (M>2)? A stick-figure model of a human hand that opens and closes? A bicycle? A car? A pterodactyl? A helicopter?

Requirements:

In-Class Demo: on the Project's due date (Thurs Jan 19) you will demonstrate your program to the class, and two other students will evaluate it by filling out 'Grading Sheets'. Professor Tumblin or TA Xiang Huan will also fill out a grading sheet, and use all three combined with your written report to determine your grade for the project.

You must submit your project to Blackboard/CMS by Friday, Jan 20 before midnight.
Submit just one single ZIP file that contains: a) your written project report as a PDF file, and b) just one directory that holds all your source code and your CodeBlocks files (.cbp etc). Do not include executables, object files, etc; keep it compact!
(How? see the purple menu on left-hand side of the webpage: 'Assignments').

Project A consists of:
1)—Report: A short written, illustrated report, submitted as a printable PDF file. Length: >1 page, and typically <5 pages, but you should decide how much is sufficient.
A minimal report consists of 4 sections:

a)--your name, netID, and a descriptive title for your project
(e.g. Project A: Planetary Gear Transmission, not just Project A)

b)—a brief 'User's Guide' that starts with a paragraph that explains your goals; instructions on how to make your on-screen image run/stop (animation) and change (e.g. A,a,F,f keys rotate outer ring forwards/backwards; S,s,D,d keys rotate inner ring forwards/backwards; arrow UP/DN keys speed up/slow down the electric motor, left/right keys act as the car's accelerator, text shows velocity in kilometers/hour.) Your classmates should be able to read ONLY this report and easily run and understand your project without your help!

c)—a brief 'Code Guide' that explains your classes/data structures/files, and how they're accessed from GLUT callbacks. You can refer readers to well-commented code rather than re-explaining it, but explain enough for your work to be sensible and extensible by your classmates.

d)—a brief, illustrated 'Results' section showing at least 4 still pictures of your program in action (use screen captures; no need for video capture), with text explanation of what the pictures are showing us, and how the project met your goals.

2)—User Instructions: When your program starts or user presses the F1 (help) key, print a brief set of user instruction in the console or graphics window e.g. 'arrow keys steer the skateboard, drag left mouse key to make robot crouch/stand; right mouse drag to aim the flame-thrower'.

2)—OpenGL: your program must use OpenGL calls to draw something interesting and colorful on-screen in 2D. You **MUST use openGL drawing primitives** (e.g. points, lines, triangles) and you MAY use GLUT primitives (e.g. http://resumbrae.com/ub/dms423_f04/08/05.html; every wire-frame shape has a solid version), but openGL primitives are REQUIRED. You may *NOT* use glVertex() commands, as these are deprecated; use vertex arrays or vertex buffer objects (see starter code).

3)—GLUT callbacks: your program should make proper use of registered callbacks for **keyboard, display, and mouse**. As demonstrated in the 'starter code', callbacks let your program respond to the mouse, to changes in the display window size, to keyboard inputs, and more. You may wish to try the simple pop-up menu features of GLUT / freeGLUT as well.

4)—Transformations: Your code must use openGL's GL_MODELVIEW matrix to transform the points of your object so that it moves and/or does something interesting on-screen. Use OpenGL calls only to translate, scale, rotate, shear, etc. all the points in an object: I will not accept hard-coded expressions, such as "x = x + xtrans; y = y + ytrans;"

5)--Motion! Like all projects in this course, your program must show a picture that moves and changes, both by itself (animation) or in response to user inputs (interaction) from mouse or keyboard. Users must be able to move, grab, animate and pause objects they see on screen.

6)—At least two (2) or more Different *Kinds* of jointed, moving objects. A key goal of this project is to familiarize you with matrices and transformations (later expanded and called 'scene graphs') to make jointed objects and nested coordinate systems in openGL. Your project MUST show at least two different *kinds* jointed objects, each with at least 3 sequentially-connected segments. For example, a one-legged hopping robot with 3 segments might have body, thigh, and shin. You'll need several calls to 'glPushMatrix()' and to 'glPopMatrix()' to draw your objects. Each and every joint in your jointed objects must rotate, yet the joints must stay pinned together as if connected by hinges or sliding joints; joints should not separate as they move.

Sources: You are welcome to use any, all, or none of the 'starter code' I supply for you. I wrote the starter code myself so I pledge it will always work, so you can begin with that and gradually modify it step-by-step You may also use the 'starter code' from the book, or from openGL.org, or any other useful resource you can find. Be sure to give proper credit to the people/sites that help you—list the URLs, share them with others (use the CMS/Blackboard discussion board, etc.); make this YOUR code, not a tweaked version of a web-poster's work.

Have fun with this!