CS 3353 Laboratory Problem Set 3

Due November 5, 2009

These problems are to be done on an individual basis following the Trinity University Academic Integrity Policy. or Trinity University Honor Code.

Academic Integrity and Honor Code

All students are covered by a policy that prohibits dishonesty in academic work. The Academic Integrity Policy (AIP) covers all students who entered Trinity before the Fall of 2004. The Academic Honor Code covers all those who entered the Fall of 2004 or later. The Integrity Policy and the Code share many features: each asserts that the academic community is based on honesty and trust; each contains the same violations; each provides for a procedure to determine if a violation has occurred and what the punishment will be; each provides for an appeal process. The main difference is that the faculty implements the AIP while the Honor Code is implemented by the Academic Honor Council. Under the Academic Integrity Policy, the faculty member determines whether a violation has occurred as well as the punishment for the violation (if any) within certain guidelines. Under the Honor Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to investigate, adjudicate, and assign a punishment within certain guidelines if a violation has been verified. Students who are under the Honor Code are required to pledge all written work that is submitted for a grade: On my honor, I have neither given nor received any unauthorized assistance on this work and heir signature. The pledge may be abbreviated pledged with a signature.

Laboratory problems should be submitted electronically (e-mail to cs3353@ariel.cs.trinity.edu) on or before the due date and should contain a problem write-up, source code to any programs and data sets used in solving the problem. The submitted files should be ASCII text files having Unix end-of-line characters (please convert all Windows and Mac text files to Unix format–I have found that Emacs seems to do a reasonable job of such conversions). If several files need to be submitted, put them in a directory having name *your-last-name-problem-set-number* and create a tar archive of this file system and attach it to your e-mail problem submission.

Object Heirarchy

In this laboratory problem we wish to learn how to link parts of an Open GL model together so that movement of one part causes appropriate movement of other linked parts.

The course web page class-files link contains a program, robot.c, which is shown below:

```
/*
 * Copyright (c) 1993-1997, Silicon Graphics, Inc.
 * ALL RIGHTS RESERVED
 * Permission to use, copy, modify, and distribute this software for
 * any purpose and without fee is hereby granted, provided that the above
 * copyright notice appear in all copies and that both the copyright notice
 * and this permission notice appear in supporting documentation, and that
 * the name of Silicon Graphics, Inc. not be used in advertising
 * or publicity pertaining to distribution of the software without specific,
 * written prior permission.
 *
 * THE MATERIAL EMBODIED ON THIS SOFTWARE IS PROVIDED TO YOU "AS-IS"
 * AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE,
 * INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR
```

```c
 * FITNESS FOR A PARTICULAR PURPOSE.  IN NO EVENT SHALL SILICON
 * GRAPHICS, INC.  BE LIABLE TO YOU OR ANYONE ELSE FOR ANY DIRECT,
 * SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY
 * KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING WITHOUT LIMITATION,
 * LOSS OF PROFIT, LOSS OF USE, SAVINGS OR REVENUE, OR THE CLAIMS OF
 * THIRD PARTIES, WHETHER OR NOT SILICON GRAPHICS, INC.  HAS BEEN
 * ADVISED OF THE POSSIBILITY OF SUCH LOSS, HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE
 * POSSESSION, USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * US Government Users Restricted Rights
 * Use, duplication, or disclosure by the Government is subject to
 * restrictions set forth in FAR 52.227.19(c)(2) or subparagraph
 * (c)(1)(ii) of the Rights in Technical Data and Computer Software
 * clause at DFARS 252.227-7013 and/or in similar or successor
 * clauses in the FAR or the DOD or NASA FAR Supplement.
 * Unpublished-- rights reserved under the copyright laws of the
 * United States.  Contractor/manufacturer is Silicon Graphics,
 * Inc., 2011 N.  Shoreline Blvd., Mountain View, CA 94039-7311.
 *
 * OpenGL(R) is a registered trademark of Silicon Graphics, Inc.
 */

/*
 * robot.c
 * This program shows how to composite modeling transformations
 * to draw translated and rotated hierarchical models.
 * Interaction:  pressing the s and e keys (shoulder and elbow)
 * alters the rotation of the robot arm.
 */
#include <GL/glut.h>
#include <stdlib.h>

static int shoulder = 0, elbow = 0;

void init(void)
{
  glClearColor (0.0, 0.0, 0.0, 0.0);
  glShadeModel (GL_FLAT);
}

void display(void)
{
   glClear (GL_COLOR_BUFFER_BIT);
   glPushMatrix();
   glTranslatef (-1.0, 0.0, 0.0);
   glRotatef ((GLfloat) shoulder, 0.0, 0.0, 1.0);
   glTranslatef (1.0, 0.0, 0.0);
   glPushMatrix();
   glScalef (2.0, 0.4, 1.0);
   glutWireCube (1.0);
   glPopMatrix();

   glTranslatef (1.0, 0.0, 0.0);
   glRotatef ((GLfloat) elbow, 0.0, 0.0, 1.0);
   glTranslatef (1.0, 0.0, 0.0);
   glPushMatrix();
   glScalef (2.0, 0.4, 1.0);
   glutWireCube (1.0);
   glPopMatrix();

   glPopMatrix();
   glutSwapBuffers();
}

void reshape (int w, int h)
{
   glViewport (0, 0, (GLsizei) w, (GLsizei) h);
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity ();
   gluPerspective(65.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glTranslatef (0.0, 0.0, -5.0);
}

/* ARGSUSED1 */
void keyboard (unsigned char key, int x, int y)
{
   switch (key) {
      case 's':
```

```
         shoulder = (shoulder + 5) % 360;
         glutPostRedisplay();
         break;
      case 'S':
         shoulder = (shoulder - 5) % 360;
         glutPostRedisplay();
         break;
      case 'e':
         elbow = (elbow + 5) % 360;
         glutPostRedisplay();
         break;
      case 'E':
         elbow = (elbow - 5) % 360;
         glutPostRedisplay();
         break;
      case 'b':
         elbow = (elbow - 5) % 360;
         shoulder = (shoulder - 5) % 360;
         glutPostRedisplay();
         break;
      case 27:
         exit(0);
         break;
      default:
         break;
   }
}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
   glutInitWindowSize (500, 500);
   glutInitWindowPosition (100, 100);
   glutCreateWindow (argv[0]);
   init ();
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutKeyboardFunc(keyboard);
   glutMainLoop();
   return 0;
}
```

Your task in this laboratory problem is to extend the robot program in the following ways:

- *Hand*: You should add a hand to the robot forearm which consists of at least 2 jointed fingers and a jointed thumb. The hand should be attached to the forearm with a wrist. The wrist should be movable at the wrist joint and the fingers should be able to grip and release.
- *Torso*: The robot arm should attach to a torso at a shoulder joint.
- *Surfaces and Lighting*: The robot arm should be displayable in either wire frame or shaded surface form. It should be possible to display the robot arm using just ambient light or lighted by a single point light source.
- *Modeling Robot Arm Parts*: You should use simple rectangular solids to model all arm parts (torso, upper arm, forearm, wrist and jointed fingers). The fingers and thumb need only have 2 joints, that is, one joint which attaches the finger to the wrist and a second joint in the middle of the finger.
- *Interaction*: You should define keystroke inputs to control each joint movement as well as surface shading and lighting control.
- *Other Fingers and Arms(Optional)*: You may include additional fingers on the robot hand and/or you may add a second arm with hand and fingers to the robot.
- *Limiting Joint Movement (Optional)*: If you have time, you might experiment with constraints on the movement of certain joints using the limits of a human arm as a model.

Problem Set 3 Solution [HTML] [PS] [PDF]