REGULATIONS

Due date: 19.11.2009 Thursday 23:59

Late Submission: Late submission is not allowed.

Submission: Use the COW system to submit your homework file. The homework should be done and submitted individually. You will submit a single tar.gz file named “hw1.tar.gz”, which includes your source code for the assignment and a makefile. You will also submit a hardcopy assignment report to the assistants at a defined time and place which will be announced in the newsgroup. Make sure that your code can be compiled by a single “make” command at the shell prompt. The makefile should produce an executable named “hw1” which can be executed directly by typing “./hw1” at the command line.

Evaluation: Your homework will be evaluated in a Linux environment. Before submission, you have to test your homework at one of the department’s computers using Linux to have a common platform.

Cheating: In case of cheating, all parts involved (source(s) and receiver(s)) get zero.

Newsgroup: You must follow the newsgroup for discussions and possible updates on a daily basis.

MAIN THEME - TIME FOR HYPNOSIS

In your first assignment, you are going to implement a pendulum by just using two types of objects, cylinders and spheres. Your implementation should be able to animate two types of motions in pendulum, one ball motion and all balls motion. The purpose of this assignment is to make you understand the geometric transformations and introduce you to the computer animation basics.
SPECIFICATIONS

- In this assignment, you are going to implement a 3D pendulum which is capable of performing two motions.

- Since you are going to implement a 3D object, you have to adjust your camera and light in order to be able to see nice looking objects. The camera positions and lighting are not in the scope of this assignment. For that reason, you can initialize OpenGL with the below code. You can make changes on this initialization code if you think that it is necessary. The details of these initialization statements will be defined at the recitation. This initialization code may be put in a function that is called once before OpenGL enters the rendering loop.

```c
glEnable(GL_DEPTH_TEST);
glEnable(GL_COLOR_MATERIAL);
glEnable(GL_NORMALIZE);

glShadeModel(GL_SMOOTH);

// SET LIGHTING
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
GLfloat position[] = {1000.0, 1000.0 , 1000.0, 0.0};
glLightfv(GL_LIGHT0, GL_POSITION , position);
glLightfv(GL_LIGHT0 , GL_AMBIENT , blackColor);
glLightfv(GL_LIGHT0 , GL_DIFFUSE , whiteColor);
glLightfv(GL_LIGHT0 , GL_SPECULAR , whiteColor);

// SET CAMERA
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(90, 1 , 1 , 1000000);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(2000,2000,-2000,0,0,0,(2)/sqrt(3),
        sqrt(2) -(2/sqrt(3)),2/sqrt(3));
```

- Don’t forget to define depth buffer while defining the glutInitDisplayMode. This is required since you are performing depth test for your 3D objects. Also remember to clear the GL_DEPTH_BUFFER_BIT at the beginning of your display callback.

- You must use the timer callback of GLUT in order to have a smooth motion. 25 milliseconds is an acceptable value for the frequency of the callbacks to the timer callback function.
• You are just allowed to use the two functions below for creating the components of your pendulum. No other objects can be drawn with either GLU functions or by defining other geometry in OpenGL. You can call the below functions as much as you want in your implementation. If you draw some objects without calling these functions, your assignment will be penalized %50.

```c
// Gives the unit sphere
void drawSphere(void)
{
    GLUquadric *quadric = gluNewQuadric();
    gluSphere(quadric , 20 , 100 , 100);
}

// Gives the unit cylinder
void drawCylinder(void)
{
    GLUquadric *quadric = gluNewQuadric();
    gluCylinder(quadric, 10, 10, 40 , 100, 100);
}
```

• Your pendulum should look like the pendulum image given in the first page. You are free to adopt a different design if you want as long as it satisfies the following conditions:
  o Each pendulum ball should be connected to holder with at least two ropes formed by cylinders.
  o Each pendulum ball should be a sphere.
  o The pendulum holder should be designed by considering the physics rules. It should be formed such that the pendulum can be put on top of a table.

**FUNCTIONALITY**

• There will be three motion modes in your implementation.

• **Still Mode:** In this mode, there will be no animation. The pendulum will be shown without any motions. When your program is started, this is the mode that you have to render. At any time, pressing the key “1” will make the current motion mode still mode.

• **One ball motion:** In this mode, you will animate the turn based motion of the balls on the left most and right most sides of the pendulum. These two balls will move in turns going up and down. In real life, this motion can be created by lifting one ball and then releasing it. This motion will go on forever until the motion mode is changed. At any time, pressing the key “2” will make the current motion one ball motion. One ball motion is illustrated in Figure 1.

• **All balls motion:** In this mode, you will animate the motion of all balls left and right at the same frequency. In real life, this motion can be created by lifting all the balls on one side and releasing. This motion will go on forever until the motion mode is changed. At
any time, pressing the key “3” will make the current motion all balls motion. All balls motion is illustrated in Figure 2.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Figure 1 : One Ball Motion Illustration**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Figure 2 : All Balls Motion Illustration**

**REPORT QUESTIONS**

After implementing the assignment according to the above specifications, you will answer the questions in the next pages and hand the hardcopies of the next two pages.
CENG477 – INTRODUCTION TO COMPUTER GRAPHICS

ASSIGNMENT 1 REPORT

The questions should be answered in a clear hand-writing format. Computer print-outs will not be accepted. Please don’t put the hardcopies into a plastic folder. Use a staple if you have more than one page. Please try to fit your answers in the areas provided below. You can use other sheets if it doesn’t fit. Make sure that you write your number also on that extra sheet and give references to that sheet in the related question.

1. What is the transformation matrix that you apply on the ball in the middle during the motion performed on all balls motion? Clearly show how you derive that transformation matrix.

2. What is the purpose of using glPushMatrix( ) and glPopMatrix( ) functions in your implementation? Why do you think these two functions are named with the term “matrix”?
3. Why do you think you have to apply the transformations at the reverse order? In other words, why do you write the transformations which should be applied primarily after you write the transformations that should be applied later on?

4. Find the place you call the drawSphere( ) function. Change the order of the glTranslate and glScale you use for this sphere. What is the result? What happens as a result of this change? Why do you think this change of order produced that result?

5. Find a place that you use glRotate with glTranslate. Change the order of these transformations. What is the result? What happens as a result of this change? Why do you think this change of order produced that result?