311170030 Introduction to Computer Graphics

Project 2: Interactive Hierarchical Modeling

Due: 4 Jan (Thursday), 2024 12:00pm

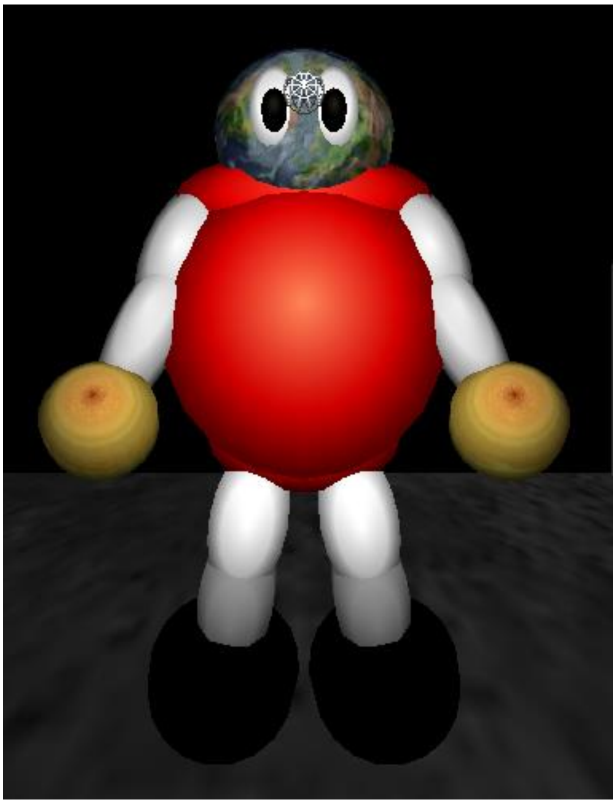
10 % per day late penalty, **Fail the course if you copy**

Topics

- Hierarchical model and data structures
- Matrix stacks
- Color and Lighting
- Display lists
- Handling keyboard events
- Textures mapping
- Familiar with animation

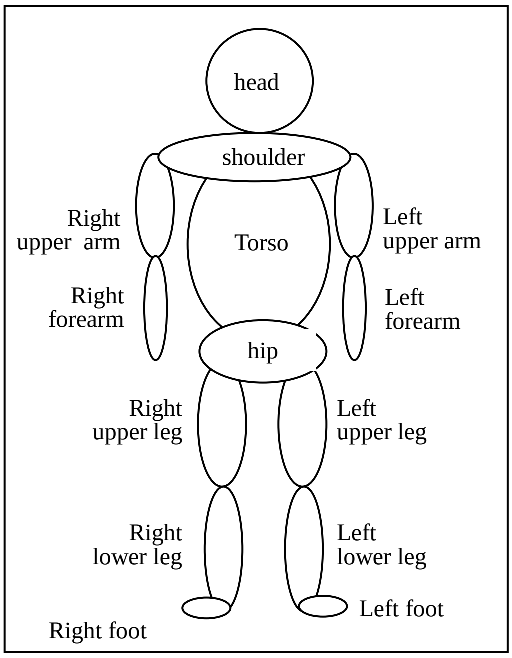
Introduction

This assignment requires you to create a "puppet" in a hierarchical fashion from a number of basic primitives such as cube, sphere, cone, torus, etc. The puppet will be rendered and lighted interactively using OpenGL by programming in C++ language. The final puppet should be drawn using a suitable selection on light position and materials so that the 3D structure of the puppet is obvious. The figure below shows you an example of such a puppet. In this assignment, you also need to make a walking animation, make your puppet walk properly, just as the Demo.



Puppet Structure

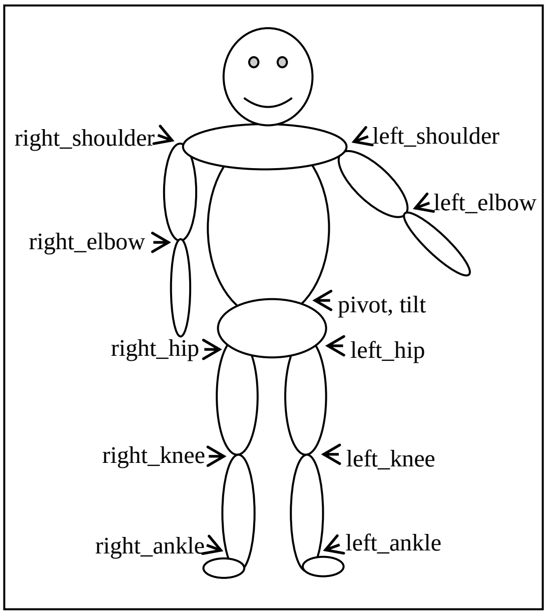
You are encouraged to be creative with your model, as long as it has at least the same parts and the same degrees of freedom as shown below.



Your puppet should at least include the following 14 parts:

1) head
2) torso
3) shoulder
4) hip
5) left and right upper arms
6) left and right forearms
7) left and right upper legs
8) left and right lower legs
9) left and right foot

The definition of your puppet is to include **12** free angles (12 degrees of freedom), two for each arm, three for each leg, and two for waist. These angles will be used in keyboard handling and animating the puppet. To implement the puppet walking under the animation subroutine, you **MUST** define the joint variables of your puppet same as what we have defined, otherwise your puppet would not perform walking properly. Here is a graph and a list of those variables and their meanings:

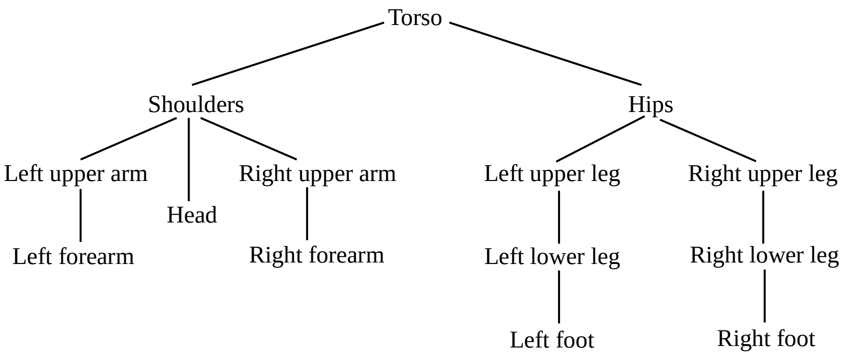


left_shoulder, right_shoulder
– rotation of the upper arm forward and
backward about the shoulder;
left_elbow, right_elbow
– rotation of the forearm forward and
backward about the elbow
left_hip, right_hip
– rotation of the upper leg forward and
backward about the hip
left_knee, right knee
– rotation of the lower leg upward and downward about the knee
left_ankle, right_ankle
– rotation of the foot forward and backward

All angles should be bounded by their corresponding minimum and maximum values. For example, it should not be possible to rotate a knee or an elbow the “wrong way”, or to rotate a torso or foot more than 90 degrees from their neutral position.

To create the puppet model, a primitive should be appropriately transformed and instanced to create the torso (with its center as the origin of the puppet coordinate system). Instances for shoulder and hip should be positioned relative to this system. Their centers form the origins for two secondary systems. With respect to the shoulder system, a head and two upper arms form the origins of three subsidiary coordinate system origins. Each primitive for forearm should be positioned relative to the upper arm’s center. The legs, consists of upper legs, lower legs and feet, will be constructed similarly, with respect to the hip coordinate system.

The coordinate system of each body is shown as:



Puppet Manipulation

Your puppet model should have 12 degrees of freedom as indicated in the demo program *puppet* through keyboard events handling. The requirements are summarized as the following:

"q","Q": rotate left upper arm backward, forward about the shoulder
"w","W": rotate right upper arm backward, forward about the shoulder
"e","E": rotate left forearm downward, upward about the elbow
"r","R": rotate right forearm downward, upward about the elbow
"y","Y": rotate torso left about the waist
"u","U": rotate torso forward and backward about the waist
"a","A": rotate left upper leg backward, forward about the hip
"s","S": rotate right upper leg backward, forward about the hip
"d","D": rotate left lower leg downward, upward about the knee
"f","F": rotate right lower leg downward, upward about the knee
"g","G": rotate left foot backward, forward about the ankle
"h","H": rotate right foot backward, forward about the ankle

Viewing (Camera) Parameter, Lighting and Material Properties

Basic view parameters are provided in *submit.c*, but you may need to make some modifications on these parameters according to the size and the coordinates of your model. The places where you need to make such modifications are commented in *submit.c*. Four special keyboard events handling are provided:

Key_UP: rotate the camera upward   Key_DOWN: rotate the camera downward
Key_LEFT: rotate the camera left   Key_RIGHT: rotate the camera right

A spot lighting is needed in this assignment. And you need to design the keyboard events handling on this lighting as shown in demo program *puppet*:

"k","l": rotate the light about X axis in clockwise and counterclockwise directions
"o","p": rotate the light about Y axis in clockwise and counterclockwise directions

Similarly, you need to modify its position according to your model size and position.
Five material properties are provided in *submit.c*, you can use them directly, or design new ones by yourselves.

Animation:

You need to simulate the robot walking as shown in demo program *puppet*, or you can make the legs and arms harmonized like a human walking, go wild with your imagination.

Implementation Details

Every piece of your puppet should be modeled into a function and store in a display list like this:

```
void DrawPart_name(void)
{
        glNewList(part_name,GL_COMPILE);
          /* GL Commands to model */
        glEndList();
}
```

The most important function is *DrawPuppet( )* that draws all parts of the puppet and with GL commands glPushMatrix() and glPopMatrix() it establishes the relationships between objects.

In this assignment, you have to implement the following 12 subroutines. Modify the source file *submit.c*. The detail description of each subroutine is printed as comments in the source file.

void DrawTorso( );
void DrawHip( );
void DrawHead( );
void DrawShoulder( );
void DrawUpperArm( );
void DrawForeArm( );
void DrawUpperLeg( );
void DrawLowerLeg( );
void DrawFoot( );
void DrawPuppet(void);
void LoadGLTextures();
void keyboard (unsigned char key, int x, int y);

In the implementation, you may have to make use of the following OpenGL commands: glRotatef, glTranslatef, glScalef, glPushMatrix, glPopMatrix, glNewList, glEndList … etc. together with a set of GLUT commands such as glutSphere, glutCube, glutCone … etc.

Grading Scheme

Your assignment will be graded by the following marking scheme:

●	Draw Puppet in a hierarchical fashion	25%
●	Puppet Manipulation (Keyboard Events Handling)	20%
●	Lighting, Spotlight(Keyboard Events Handling) and Material	15%
●	Walking animation	15%
●	Add at least two textures (A spherical texture (head) and a cube texture (floor))	15%
●	Creativity (Additional Feature)	10%
Total		**100%**

Guidelines to submit programming assignments to graphics account

1) You are suggested to write your programs on Windows, since there will be enough technical support. If you developed the program in other platforms, ***make sure your program can be compiled and executed on Windows as the program will only be tested on this platform***.

2) Modify the provided *submit.c*, and provide all your code in this file. No more additional .c or .h files are allowed. Type your full name and student ID in *submit.c*. ***Missing such essential information will lead to mark deduction.***

3) Zip the source code file (i.e. *submit.c*), the executable file (i.e., *submit.exe*), and the readme file (i.e., *readme.txt*) in a .zip or .rar file. Name it with your own student ID (e.g. 2014333333.zip). That is, there should be exactly **three** files in your submitted package.

4) Mail the zip file to the address: **https://send2me.cn/yxC9vs-a/QT2B3-cNrVAWRw**

5) An acknowledgement email will be sent to you once your assignment is received. Otherwise, resubmit your assignment since you have submitted a null email.

6) In case of multiple submissions, only the latest one will be considered.

7) ***Fail the course if you copy.***