



Laboratory Three: Hierarchical Modelling

Introduction

This laboratory is based on the stickman example from class. The code for this example is available on Blackboard in the modeling part of the course. The stickman model is a hierarchical model and we stated that it is easy to animate its motion. In this laboratory we will start by adding a walking animation.

Walking Animation

As noted in class we can produce a walking motion by a rotation about the x axis. The rotations for the two legs should be in the opposite direction. Assuming that the variable walk contains the angle of rotation this can be done using the following code:

```
/* draw right leg */
matrixStack.push(model);
model = glm::rotate(model, walk, glm::vec3(1.0, 0.0, 0.0));
model = glm::rotate(model, -135.0f, glm::vec3(0.0, 1.0, 0.0));
glUniformMatrix4fv(modelLoc, 1, 0, glm::value_ptr(model));
glBindVertexArray(leg->vao);
glBindBuffer(GL_ARRAY_BUFFER, leg->vbuffer);
glDrawElements(GL_TRIANGLES, leg->indices, GL_UNSIGNED_SHORT, NULL);
model = matrixStack.top();
matrixStack.pop();

/* draw left leg */
matrixStack.push(model);
model = glm::rotate(model, -walk, glm::vec3(1.0, 0.0, 0.0));
model = glm::rotate(model, 135.0f, glm::vec3(0.0, 1.0, 0.0));
glUniformMatrix4fv(modelLoc, 1, 0, glm::value_ptr(model));
glBindVertexArray(leg->vao);
glBindBuffer(GL_ARRAY_BUFFER, leg->vbuffer);
glDrawElements(GL_TRIANGLES, leg->indices, GL_UNSIGNED_SHORT, NULL);
model = matrixStack.top();
matrixStack.pop();
```

Note that we only need to add one line for each leg to do the rotation.

Now we need to correctly vary the rotation angle on each update. We assume that the leg can move from -45 degrees to +45 degrees in a walking cycle. We can use an idle procedure to perform this computation. The code for this procedure is:

```
void idleFunc() {
    static float rate = 0.1;
    static float dwalk = 0.1;

    glutSetWindow(window);
    walk += dwalk;
    if(walk > 45.0)
        dwalk = -rate;
    if(walk < -45.0)
        dwalk = rate;
    glutPostRedisplay();
}
```

Note that we have used two static variables to maintain the walking state instead of exposing them as global variables.

Bonus activities (optional)

If you finish early here are some things to try:

- Add another animation to the stickman, such as waving his hand or shaking his head.
- Introduce a drawInstance procedure that takes in a cylinder object and the model matrix as parameters to draw a cylinder instance to enable code reuse
- Observe the effects of increasing the number of sides when drawing a cylinder instance. How is performance of the code affected? How is the smoothness of the surface of cylinder affected?
- Try to make the polygon edges in the cylinder visible as a black wireframe on top of the cylinder (hint: use GL_LINES)

Laboratory Report

Submit your source code and a screenshot of your final result in the Blackboard dropbox.