INTRODUCTION

This program is an OpenGL implementation of a 3D 24- degree of freedom (DOF) penguin. The 9 Each degree of freedom can be controlled via the spinners in the control bar. Transformation hierarchies and a transformation stack were used to simplify the drawing of each limb in the correct location. Lighting effects were enabled for the penguin. The penguin can be animated, in which case the magnitude of articulation for each limb is controlled via Catmull-Rom interpolation of keyframes. The basic components of the implementation can be divided into: drawing, lighting, and keyframe animation.

DRAW

The penguin consists of multiple quadrilaterals positioned to appear solids. The following primitive volumes were used in drawing and scaled/rotated accordingly to fit the penguin: cube, frustum, triangular prism, and a trapezoidal prism. Matrix transformations of the frame of reference and a drawing hierarchy are used to simplify the drawing.

The drawing hierarchy is as follows:

```
Body ---> Left Arm ---> Left Forearm
   \---> Right Arm ---> Right Forearm
   \---> Left Leg ---> Left Foot
   \---> Right Leg ---> Right Foot
   \---> Head ---> Bottom Beak ---> Top Beak
```

During drawing, normals were specified once for each vertex for use in lighting calculations. These normals are scaled to unit length by enabling the GL_RESCALE_NORMAL flag.

The penguin can be rendered in five different modes: wireframe, solid, solid with outline, matte, and metallic. The wireframe and solid rendering were achieved by enabling the appropriate GL flags. Depth testing was enabled for the solid with outline rendering style. This style required two passes of the penguin rendering, once as solid, and once as wireframe with glPolygonOffset set to allow the wireframe depth test to appear in front of the solid.

LIGHTING

In addition to the three rendering modes mentioned above, it is also possible to render the penguin using a matte or metallic material. To achieve this, a point lighting source was placed on the positive z-axis at a large radius so as to not interfere with the penguin's motion. This point source rotates in a large radius in the xy-plane. Appropriate values for glMaterial were set to simulate the materials. In general, the metallic material had a larger shininess and specular component than that of the matte material. The matte material held a larger diffuse component than the metallic material.

The use of the GL_SMOOTH flag allows the lighting to appear as a gradient between vertices, realistically portraying the different materials.

KEYFRAME ANIMATION

Keyframes were used to animate the penguin. Each keyframe is provided a time, keyframe number, and magnitude values for each of the 24 DOFs. The saving and loading of these keyframes to a file was managed by the starter code.

When the animation is started, the animation timer is set to *0*. At time *t* for each keyframe, the values for each DOF is precisely what is specified in the keyframe. Between keyframes, Catmull-Rom interpolation is used to determine values for the DOFs.

To use Catmull-Rom interpolation, four vectors are required: the DOF vector for frame *i, i+1* and the tangent vectors at frame *i, i+1*. At the boundaries i.e. *i = 0* and *i = N*, where *N* is the number of keyframes, the tangents are determined by subtracting keyframe *1* from keyframe *0* and keyframe *N, N-1*. Within the center, the tangent vector of keyframe *i* is calculated as keyframe *i+1 -* keyframe *i-1*. A Catmull-Rom scale of 0.5 was arbitrarily chosen for the calculation of the tangent vectors.

Two sample keyframe files are provided.

The first keyframe file demonstrates the flapping of the penguin's arms in combination with a rotation of the camera.

The second keyframe file demonstrates the effect of lighting as the viewing position is rotated in an orbit around the penguin.