Michelle Daniels’ Robot’s Usage of OpenGL, GLU, and GLUT

Definitions adapted from: [http://cs.uccs.edu/~semwal/indexGLTutorial.html](http://cs.uccs.edu/~semwal/indexGLTutorial.html) OpenGL
and [http://openglut.sourceforge.net/group_api.html](http://openglut.sourceforge.net/group_api.html)

**OpenGL**

Functions:
- `glCallList( )` // Execute a display list.
- `glClear( )` // Clear buffers to preset values.
- `glClearColor( )` // Specify color to be used when color buffers are cleared.
- `glColor3f( )` // Specify new red, green, blue values for current color.
- `glEnable( )` // Enable GL capabilities (used for robot scene lighting and depth testing).
- `glEndList( )` // Signal end of a list of a display list; matches `glNewList( )`.
- `glGenLists( )` // Generate a contiguous set of empty display lists.
- `glLightfv( )` // Set light source parameters.
- `glLoadIdentity( )` // Replace current matrix with identity matrix.
- `glMaterialfv( )` // Specify material parameters for lighting model. GLfloat version.
- `glMatrixMode( )` // Specify which type of matrix is current matrix.
- `glNewList( )` // Initiate a new display list, which is a group of GL commands stored for subsequent execution; matches `glEndList( )`.
- `glOrtho( )` // Multiply current matrix by an orthographic matrix.
- `glPopAttrib( )` // Restore values of state variables saved with last `glPushAttrib( )` call.
- `glPopMatrix( )` // Pop current matrix stack down by one, duplicating the current matrix; matches `glPushMatrix( )`.
- `glRotatef( )` // Compute a matrix that performs a counterclockwise rotation of angle degrees about the vector from the origin through point (x,y,z). angle, (x,y,z) are arguments. Current matrix is multiplied by this rotation matrix, with product replacing the current matrix. GLfloat version.
- `glScalef( )` // Multiply current matrix by scaling matrix, with arguments specifying scale factors along x, y, z axes. GLfloat version.
- `glShadeModel( )` // Select flat or smooth shading.
- `glTranslatef( )` // Multiply current matrix by translation matrix, with arguments specifying translations in x, y, z directions. GLfloat version.
- `glViewport( )` // Set the viewport by specifying its rectangle’s lower left corner in pixels and also its width and height.

Types:
- GLfloat // GL floating-point
- GLuint // GL unsigned 32-bit integer

GLU
- gluLookAt( ) // Define a viewing transformation by specifying position of eye point, reference point, and up vector.

GLUT

3D Objects

Functions:
- glutSolidCube( ) // Draw a solid-shaded cube, centered at origin.
- glutSolidSphere( ) // Draw a shaded sphere, centered at origin.
- glutWireCube( ) // Draw a wire-frame cube, centered at origin.

Windowing

Functions:
- glutCreateWindow( ) // Send a request for a window to be constructed.
- glutDisplayFunc( ) // Set display callback for current window.
- glutInit( ) // Initialize GLUT library.
- glutInitDisplayMode( ) // Set the window creation display mode.
  GLUT_DOUBLE argument signals double-buffering (see glutSwapBuffers( ) ).
- glutInitWindowPosition( ) // Requests future windows to open at a given position.
- glutInitWindowSize( ) // Requests future windows to open using a given width, height.
- glutKeyboardFunc( ) // Set keyboard callback for current window.
- glutMainLoop( ) // Initiate event loop.
- glutPostRedisplay( ) // Mark current window as needing a redisplay.
- glutReshapeFunc( ) // Set reshape callback for current window; called when window is reshaped.
- glutSwapBuffers( ) // Swap buffers for current window. Signals that you’re done drawing to current window. If double-buffered, swap front and back buffers.