* **User hits 'W' (Draw your wireframe)**at a suitable initial position from the viewer.

**Finish**

* **User hits 'N' (Draw next wireframe)**Organize the PLY files in a list going from 1-43. Hitting N should load and draw the next wireframe model to the current one in your list of PLY files. You can hardcode filenames if you want. The PLY files may not all be of the same size. So to properly set up the viewing position using LookAt, you may have to calculate the bounding box of the mesh and then set your view distance to a suitable multiple of the bounding box

**Finish**

* **User hits 'P' (Draw previous wireframe)**Organize the PLY files in a list going from 1-43. Hitting P should load and draw the previous wireframe model to the current one in your list of PLY files.

Finish

* **User hits 'X' (Translate your wireframe in the +ve X direction)**Continously move your wireframe some small units along the +ve X axis and redraw it. Use the idle function to animate this. The ply file should continue to slide along the +ve X axis till the user hits 'X' again. Essentially, the 'X' key acts as a toggle key. If the ply file is stationary and the user hits the 'X' key, the ply file should continue to slide along the +ve X axis until the user hits 'X' again. Camera position remains fixed for this translation and all other translations below. The exact amount to move the ply file before redrawing will affect how much and how much your translation is apparent depends on how far you positioned your wireframe from the viewer. So, it's left to you as a design choice to pick an appropriate distance to translate the wireframe along the +ve X axis each time the user hits 'X'.

Finish

* **User hits 'x' (Translate your wireframe in the -ve X direction)**Use the idle function to continuously move your wireframe some units along the -ve X axis. The number of units to translate your wireframe each time the user hits 'x' is left to you as a design choice.

Finish

* **User hits 'Y' (Translate your wireframe in the +ve Y direction)**Use the idle function to continuously move your wireframe some units along the +ve Y axis. The number of units to translate your wireframe each time the user hits 'Y' is left to you as a design choice.

Finish

* **User hits 'y' (Translate your wireframe in the -ve y direction)**Use the idle function to continuously move your wireframe some units along the -ve Y axis. The number of units to translate your wireframe each time the user hits 'y' is left to you as a design choice.
* **User hits 'Z' (Translate your wireframe in the +ve Z direction)**Use the idle function to continuously move your wireframe some units along the +ve Z axis. The number of units to translate your wireframe each time the user hits 'Z' is left to you as a design choice.

 Finish

* **User hits 'z' (Translate your wireframe in the -ve Z direction)**Use the idle function to continuously move your wireframe some units along the -ve Z axis. The number of units to translate your wireframe each time the user hits 'z' is left to you as a design choice.

Finish

* **User hits 'R' (Rotate your wireframe in an X-roll about it's CURRENT position)**Just like a rotisserie chicken, rotate your wireframe smoothly 360 degrees at a moderate speed (X-roll) about its CURRENT position (not about the center of the scene) This rotation is NOT the same as moving the wireframe in a wide arc. The rotation should be about the X axis and the wireframe should not translate while rotating. Hint: Use double buffering (glutSwapBuffers( )) to make the rotation smooth. You can continously update the new wireframe positions and redisplay the meshes in the glutIdleFunc function.

Finish

* **User hits Key 'B':**Toggle pulsing meshes ON/OFF. When ON, the mesh faces pulse back and forth continuously as described above. When OFF the meshes do not pulse. Hint: Use double buffering (glutSwapBuffers( )) to make the breathing smooth. You can continously update the new vertex positions and redisplay the meshes in the glutIdleFunc function.

Finish

* **User hits Key 'm':**Toggle a drawing of each face normal (ON/OFF). When ON, the mesh normals of all mesh faces are drawn. When off, the face normals are not drawn. Each face normal is drawn as a short line starting from the middle of each face and extending outwards. When you draw all normals, it will look like the mesh has pins sticking out of each face

Finish

* **User hits Key 'h':**Increment the amount of shearing of the wireframe along the Y axis by a small amount. Repeatedly hitting the 'h' key should shear the wireframe by a bit more and more. Note that after you shear the mesh, performing a transform (e.g. rotation, scale or translate) should transform the sheared mesh.

Finish

* **User hits Key 'H':**Decrease the amount of shearing of the wireframe along the Y axis by a small amount. Repeatedly hitting the 'H' key should shear the wireframe by a bit less and less. Note that after you shear the mesh, performing a transform (e.g. rotation, scale or translate) should transform the sheared mesh.

Finish(these two functions,’h’,’H’, I don’t know if it’s 100% right, because now, after shearing, the X-coord move will be affected)