Question 2: Visualizing camera intrinsics (25 points)

The projection parameters are usually referred to as “camera intrinsics”: for perspective projection they would be the vertical field of view, aspect ratio of near plane, distance of near and far planes from the camera. A good way to visualize these settings is to draw the view volume created by these parameters. The view volume for perspective projection would be a frustum and that for orthographic projection would be a box whose sides are aligned with the coordinate planes of the view coordinate system.

Similar to Assignment 4, you have written a program that supports two cameras: a global stationary camera and a keyboard-controlled camera (that may or may not be tied to a moving object). You wish to visualize the camera intrinsics of the keyboard-controlled camera when you view the scene from the global stationary camera. For example, if your keyboard-controlled camera uses perspective projection then when seen from the global stationary camera you want to see:

1. Four lines starting from the position of the keyboard-controlled camera
2. The near and far plane drawn in their respective sizes as rectangles (in wireframe)
3. The four lines in step 1 should end at the four vertices of the far plane
4. The frustum thus drawn should point correctly in the direction where the keyboard controlled camera is looking.

Obviously you want to see this view volume move as you control the camera from your keyboard.

Explain point-wise how exactly you will create this visualization. You do not have to include code that actually creates the above wireframe ‘mesh’: you may simply mention that you have a method that creates it. But depending on how you are drawing this, the vertex positions for the view volume may change so explain how you will specify them.

Your answer should include (a) an explanation of how you will create the wireframe model (b) how you will draw this in your draw function (when you will draw this, what will be the transformations, etc.)

Note: pseudo-code/code snippet is recommended and would probably be the best way to explain your technique).