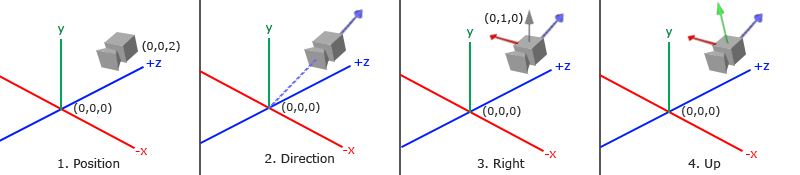
COMP 465 – Warbird Simulator Phase 2 Documentation

1. Camera
   1. To define a camera we need its **position** in world space, the **direction it's looking at**, and the **orientation** of the camera.
      1. Camera Position
         1. The camera position is basically a vector in world space that points to the camera’s position.



* + - 1. Don't forget that the positive z-axis is going through your screen towards you so if we want the camera to move backwards, we move along the positive z-axis.
  1. Camera Direction
     1. We want our camera to look at an object in the world matrix from its position so we need to give the objects position as a 3 dimensional vector.
  2. Camera Orientation
     1. If we want to look at the top of the world space:

glm::vec3 topVector(1.0f, 0.0f, 0.0f);

* + 1. If we want to look up at our object in view space from our camera position: glm::vec3 upVector(0.0f, 1.0f, 0.0f);
  1. How to obtain the coordinates of our models:
     1. In the glmUtils465.hpp located in the includes465 folder there is a method called getPosition that takes in a 4x4 matrix and returns a glm::vec3 variable. We use this method to extract the updated position of our ModelMatrices from their corresponding TransformationMatrix.
  2. Glm::lookAt() function.

glm::mat4 view;

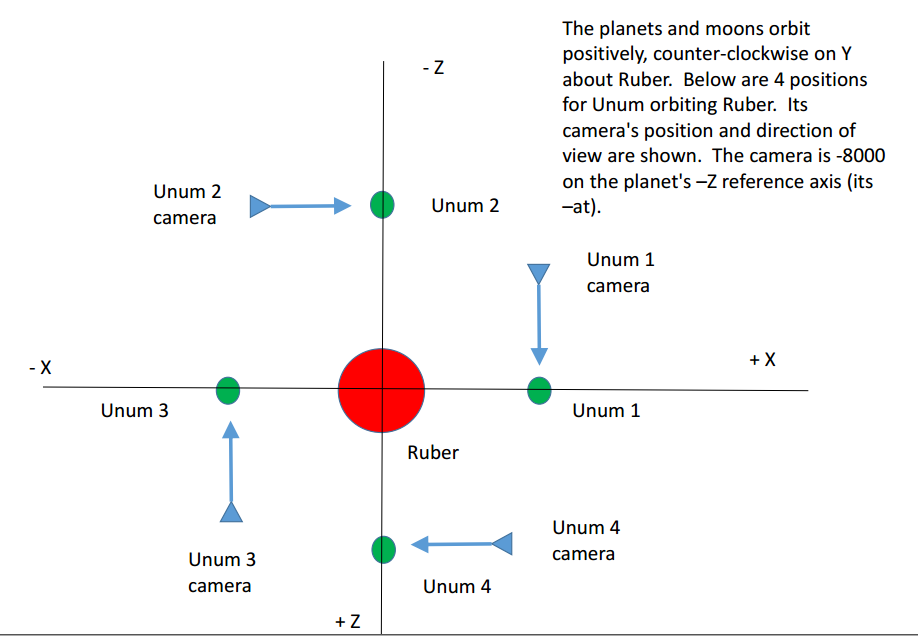
view = glm::lookAt(glm::vec3(0.0f, 0.0f, 3.0f),

glm::vec3(0.0f, 0.0f, 0.0f),

glm::vec3(0.0f, 1.0f, 0.0f));

1. The glm::LookAt function requires a camera position, target position and a vector that represents the up vector in world space.
2. GLM then creates the 4x4 LookAt matrix that we can use as our view matrix.

# The dynamic camera’s follow the professor’s specification:



**Building/Running the Project (beyond the obvious):**

Developed on the Windows Platform.