A06 – View matrices

The Vulkan application whose source code is contained in file `Assignment06.cpp`, shows a model of a space station where a robot can move in. Both first and third person view are implemented using the functions contained `view.cpp`. Pressing the SPACE key on the keyboard, the view cycles between four view modes:

1. First person view
2. Third person view
3. Top view
4. Impostor view

The first two modes needs to be implemented for the assignment, while modes 3 and 4 are there to help you debug your application. In particular:

Mode 1: First Person View
The application calls procedure `LookInDirMat(…)` which must create and return a `glm::mat4` view matrix. It receives as input a `glm::vec3` vector called `Pos`, which contains the location of the camera. The direction where it is aiming is instead contained in `glm::vec3` `Angs`. In particular:

- Element `Angs.x` contains the direction of the camera α.
- Element `Angs.y` contains the elevation of the camera β.
- Element `Angs.z` contains the rollo of the camera ρ.

Mode 2: Third Person View
The application calls procedure `LookAtMat(…)` which must create and return a `glm::mat4` view matrix. It receives as input a `glm::vec3` vector called `Pos`, which contains the location of the camer, and a `glm::vec3` vector called `aim` that contains the target. The up vector, which is not passed to the procedure, is $\boldsymbol{u} = |0,1,0|$. The camera roll is passed as an extra floating point parameter `Roll`. Although roll is not generally included in the Look-at view matrix model, it can be implement in one of two different ways:

1. Changing the direction of the up vector, according to the roll angle.
2. Placing a rotation along the z-axis of $-\rho$ after transformation of the view matrix created with the technique shown during the lessons.

You can move the view using the same keys as in *Assigment0*:

ESC – quit the application		SPACE BAR – move to the next view | **X**: shows the walls in wireframe				
Q: roll CCW	**W**: forward	**E**: roll CW			↑: look up	
A: left	**S**: backward	**D**: right		←: look left	↓: look down	→: look right