A07 – World matrices

The Vulkan application whose source code is contained in file Assignment07.cpp, shows seven Tetris-like pieces in the outer space. The pieces need to be placed to match their wireframe counterparts using world matrices, created either using Euler angles or Quaternions.
You have to implement the functions for creating the world matrices in worldMat.cpp.
Pressing the SPACE key on the keyboard, the view cycles between one in which objects are positioned using rotations expressed with Euler angles, and one where quaternions are used.

Euler Angles

The application calls procedure `MakeWorldMatrixEuler(…)` which must create and return a `glm::mat4` world matrix. It receives as input a `glm::vec3` vector called `pos`, which contains the location of the object. The rotation of the object is instead contained in `glm::vec3 YPR`. In particular:

- Element `YPR.x` contains the yaw.
- Element `YPT.y` contains the pitch.
- Element `YPT.z` contains the roll.

Scaling factors along the three main directions are contained in `glm::vec3 size`.

Quaternions

The application calls procedure `MakeWorldMatrixQuat(…)` which must create and return a `glm::mat4` world matrix. It receives as input a `glm::vec3` vector called `pos`, which contains the location of the object. The rotation of the object is instead contained in quaternion `glm::quat rQ`. Scaling factors along the three main directions are contained in `glm::vec3 size`.

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

ESC – quit the application		SPACE BAR – move to the next rotation method				
Q: roll CCW	**W**: forward	**E**: roll CW	**R**: up		↑: look up	
A: left	**S**: backward	**D**: right	**F**: down	←: look left	↓: look down	→: look right