A07 – Third person controller

The Vulkan application whose source code is contained in file `A07.cpp`, requires you to implement a complete Third person controller, that returns both the world matrix of the character, and the view-projection matrix for the camera. In particular, the view matrix should be computed using the *LookAt* technique. All the game logic must be written in file `Logic.hpp`, and implemented in the procedure:

```
void GameLogic(Assignment07 *A, float Ar, glm::mat4 &ViewPrj, glm::mat4 &World) {
	// Input:
	// <Assignment07 *A> Pointer to the current assignment code.
	// <float Ar> Aspect ratio of the current window (for the Projection Matrix)
	// Output:
	// <glm::mat4 &ViewPrj> the view-projection matrix of the camera
	// <glm::mat4 &World> the world matrix of the objectlm::mat4
```

- The *View-Projection Matrix*, has a near plane at *0.1*, and far plane at *100.0*, aspect ratio given in `Ar`, and the FOV-y of the application is 45º. The height of the target for the camera is 0.25, and the target camera distance is 1.5. Pitch should change in the rage -8.7º to 60º.
- The *World Matrix* should be updated using the walk model. The starting angle is 0º (pointing North), and the initial position should be *(-41.5, 0, -19.5)*.
- All constants have been already set up in the code for your convenience.
- The *View-Projection Matrix* and the *World Matrix* should be returned respectively in output variables `ViewPrj` and `World`.
- Axes can be read using the `A->getSixAxis()` method of the `C++` object containing the assignment. Basic code for retrieving the axes in variables `m` and `r` has been already provided for convenience.
- Positions, angles and other persistent variables used by the technique should be stored into `c++ static` variables. An example to store the position in variable `Pos` (initialized to the starting position) has been provided.

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

ESC – quit the application		SPACE BAR – switch between camera and key				
Q: roll CCW	**W**: forward	**E**: roll CCW	**R**: up		↑: look up	
A: left	**S**: backward	**D**: right	F: down	←: look left	↓: look down	→: look right

If everything work, you should be able to have screenshots like the following: