Exercise 1 (View-Frustum Culling, 10 Points)

The goal of this exercise is to implement the view frustum culling algorithm from the lecture.

We provide the \textit{VFCulling} framework that loads several objects and provides a method to move the camera along a spline. Please implement \texttt{GLwidget::computeObjectVisibility()}. You will find comments in the source code regarding the inputs and required output.

First of all, you need to calculate the view frustum itself. For that, you will need information from the class variables \texttt{m_flightPath} and the iterator \texttt{m_fpCurPos}. These provide the current camera position, viewing direction, and up-vector. It also contains the parameters: \texttt{m_left, m_right, m_bottom} etc. which define the view frustum.

Please choose the appropriate bounding volume (BV) for each individual mesh. Test the BV against the planes representing the view frustum, as explained in class. Please pay attention to the transformations in \texttt{GLwidget::paintGL()}.

Write the culling result for each mesh to the state vector: \texttt{m_meshStates[i].visible}, where \texttt{i} is the object index.