

## Introduction to Augmented Reality

### Exercise 11 (P) Getting to know OpenGL and GLFW

Now that our tracker is working, we will shift our focus to rendering. For cross-platform 3D graphics, the Open Graphics Library (OpenGL), originally developed by SGI, is widely used. As OpenGL doesn't contain code for window handling, user interaction etc., GLFW will be used, too.

- (a) Start a new program which should use GLFW to open a rendering window. Use the functions `glfwInit`, `glfwCreateWindow`, and `glfwMakeContextCurrent`. Your window should support double buffering (by default), RGBA colors, and a depth buffer. In your `main` function, create a rendering loop with `glfwWindowShouldClose`.
- (b) Your new rendering window will so far not contain anything (not even black). Implement your own rendering function which clears the window using `glClear`, and call it in the rendering loop. As you should be using a double-buffered window, call `glfwSwapBuffers` at the end of the loop to exchange the front and back buffers. After swapping the buffer make sure to call `glfwPollEvents` to process all pending events;
- (c) In order to allow resizing the window, you also need to register a `resize` callback via `glfwSetFramebufferSizeCallback`. This callback is passed the new window dimensions by GLFW. Call `glViewport` to set the rendering area to the new size.
- (d) For three-dimensional rendering, you need to set two matrices, the modelview and the projection matrix. The active matrix is set by `glMatrixMode`. Clear the modelview matrix at the beginning of your display callback by using the `glLoadIdentity()` function and set the projection matrix using `gluPerspective( 30, ((double)width / (double)height), 0.01, 100 )` in your resize callback.
- (e) Finally, it's time to render something. Keep in mind that the camera is by default located at the origin and looks down the negative z axis, so use `glTranslatef` to move your world in negative z direction and, at long last, draw a solid sphere using `drawSphere` we provide.