

Lab Assignment #2, Part B  
Due Oct. 1, 2019  
To be graded in class before 5:20pm

Transformation matrices

The goal of this lab is to experiment with transformation matrices, and interact with them using ImGui. Consult the class notes and the glm documentation for details about declaring vectors and matrices, and calling functions to create matrices, such as `glm::translate`, `glm::lookAt`, and `glm::perspective`.

This assignment will be graded in class, but also submit your code to Blackboard before the end of class.

- Mesh scaling. [5 pts]
  - The GUI demo template code currently supports uniform scaling transformations. Modify the code to support nonuniform scaling.
  - Create an ImGui SliderFloat3 to control the value of vec3 scale components. Allow the slider to give scale values between -2.0 and 2.0
  - In the draw function find the declaration of `glm::mat4 M`, and the `glm::scale(...)` function call. Modify that function call to implement nonuniform scaling.
    - Note the scale factor `mesh_data.mScaleFactor` fits the mesh to the screen
- Mesh translation. [10 pts]
  - Declare a new `glm::vec3` variable named 'pos' below 'scale' and initialize it to (0.0f, 0.0f, 0.0f).
  - Create an ImGui SliderFloat3 to control the value of pos. Allow each component of the vector to go from -1.0 to +1.0
  - In the draw function, declare a matrix `glm::mat4 T` and initialize it to a translation matrix, using the pos variable as the argument to `glm::translate(...)`.
  - Make sure you declare T before M is declared. When initializing M multiply on the left with T.
    - The declaration of M should now look like: `glm::mat4 M = T*glm::rotate(...`
- Camera translation [10 pts]
  - Create a slider which changes the camera position. Hint: Modify the `glm::LookAt(...)` function call in the declaration of the camera view matrix, V.
- Camera projection matrix [15 pts]
  - Create sliders to change each of these parameters in the `glm::perspective(...)` function call
    - camera field-of-view angle
    - camera near clip distance
    - camera far clip distance
- Change M multiplication order [5 pts]

- Create a checkbox which changes the order of the matrix multiplication used to define  $M$ .
- Currently the order is  $T * R * S$  where  $T$  is translation,  $R$  is rotation and  $S$  is scale.
- When the checkbox is checked make the order be  $R * T * S$
- Based on the behavior you see when manipulating the checkbox and the rotation and translation sliders, does matrix multiplication commute? In other words, is it always true that  $T * R = R * T$  ?