

Greetings,

There has been some concern over implementing `Matrix4::BuildTransform`, which takes a `Vector3` for translation, `Matrix3` for rotation, and `Vector3` for scale. Here are a few things to consider:

- The `Vector3`s and `Matrix3` are just compact representations to pass the information to create an affine 4x4 matrix. The `Matrix3` in this case represents the completely rotation matrix you built from your Euler angles representation of rotation. It's the upper 3x3 portion of the 4x4 matrix, since the 4th column is the translation component.
- Using the Zero Engine math library is completely optional; there is a "TODO(student)" inside of that function specifically because it is advised to create the local to world matrix inside of that function, but if you are not using their Math library, then you certainly don't need to do any logic in there. The model matrix must be created somewhere else, in that case.
- If you don't like passing `Vector3`s and a `Matrix3`, feel free to change the types of those parameters. You can submit any code for this you want to, as long as you are the one who wrote it.

Note regarding loading the **Wavefront OBJ** files: bear in mind that face indices begin at ONE instead of ZERO. If you are just taking the integral values of the faces and treating those directly as indices in C++, you are going to notice some strange behavior (undefined, in fact). Several students have run into this issue and, although it's close to the submission deadline, it's being posted in hopes it may help people who are stuck specifically on the object loader.

Finally, it was pointed out there was a mistake in one of the **Main.cpp** comments. It reads "TODO(ben)," but, given the comment and requirements of the assignment, I hope you took that to mean "TODO(student)," which it was supposed to.

If anyone has any additional questions regarding this issue or any other with the framework or assignment, feel free to email either the instructor or the TA, if any.

Thanks,
Pushpak Karnick