Matt Benson Lab 02

1. I suspect the problem occurs with the shapes being used and the use of lighting on those object and having to calculate so many shadows every frame, because they could have some shadows coming off of them. They are using a solid mesh, but it’s actually two meshes stuck together which doesn’t help.
2. The example project works so much better because it is not rendered using solid meshes like the project we made. It uses very simple triangle meshes rendered with

template = MeshE.TetrahedronDraft(.3f);

which makes a shape that is handled very easily. Doing a little digging into the Unity API you can find Mesh.MarkDynamic() which is used in this example project. According to the API:

“Call this before assigning vertices to get better performance when continually updating mesh. Internally this will make the mesh use "dynamic buffers" in the underlying graphics API, which are more efficient when mesh data changes often.”

There are other choices made in this example project that increase efficiency as well, but I feel that these are the most impactful for this project when handling large loads.