CS348K – Visual Computing Systems

Final Project Proposal

Name & SuNET ID: Hannah Norman (hnorman)

Project Title: *DyDef: A Visualization Tool for Dynamic Deformations on Shell Objects*

Summary

I am going to develop a GUI that allows users to apply, simulate, and visualize dynamic deformations on shell objects. The project will build on existing computational models for shell deformation and will include features for both simulation control and visualization. I will use DeformFX—a software library for nonlinear finite element simulation—as the foundation of this application. This GUI will primarily serve as a tool for understanding the physical behavior of shell structures under various constraints, but it will also provide a foundation for future extensions such as training AI agents in immersive worlds.

Inputs and Outputs

Inputs:

- geometry of shell object
    - simulation mesh
    - visualization mesh
- user-specified force/boundary input

Outputs:

- visual representation of shell object in GUI
- graphical representation of displacements/stresses/strains

The major constraint of this application will be machine performance. Real-time visualizations (i.e., 30fps) are simply impossible given the current, existing state of the DFX software; a more realistic performance threshold should max out around 3fps. Even so, certain shortcuts and behind-the-scenes cheats could be implemented to give the illusion of real-time rendering at certain stages—for example, pre-rendering several deformation approximations and allowing the user to adjust force/boundary input while viewing these hypothetical deformations in real time.

Task List

1. Get application (DFXHelloShell) built and running.
2. Create and/or find several basic shell-object meshes for development and testing.
3. Modify app to handle mouse input for describing the applied force (or different boundary condition). An extension includes expanding allowed input mediums.
4. Add stress visualization / dynamic deformation to linear simulation.

Nice-to-haves:

- alternative visualization options (i.e., add support for nonlinear deformations)

Expected Deliverable

I intend to create an application demo that shows visualization and graphical representation outputs changing in response to force and boundary value inputs, ideally with reasonable performance. I want my application to be interactive and responsive to user inputs in pseudo-real time. By this, I mean that the base application currently is expected to run at around 3fps, but for certain aspects of the demonstration, I would like to suggest real-time response-rate, even if some shortcuts were taken behind the scenes to produce this illusion. I provided an example under the *Inputs/Outputs* section above of a possible approach for this goal, wherein I implement a pipeline that pre-renders several deformation approximations and allows the user to adjust force/boundary input while viewing these hypothetical deformations in real time.

Of course, as I develop my project and come to understand the limits of what I can implement in the app and under the time constraints of this quarter, these goals and approaches will likely change. But as it stands, I plan to create a visually-pleasing, interactive demo application that is intuitive to use and, at the very least, has features that simulate aspects of real-time visualizations of dynamic deformations.

Biggest Risks

- *cannot get DFXHelloShell application compiled and running*
    - I can eliminate this risk by seeking help from Justin (my upcoming summer internship boss), who has experience with the software.
- *overzealous goals given my inexperience in this subject area*
    - I can reduce this risk by tackling the project incrementally, going one step at a time with patience and a willingness to put in extra legwork on self-study.

What do I need help with?

If there are any academic papers relevant to this topic that I should be aware of, I'd love to be pointed in their direction, lest I miss them during my own background reading. Also, I would appreciate any thoughts on the scope of this project. I am willing to put in a lot of time, but this is my first time completing a term project on my own, and I am worried I may have bitten off a bit more than I can chew for a 5–6-week timeline.