Team Status Update #2:

Week 4:

Goals:

- Chase: work on getting the AnisoMPM domos to Compile
- Alex: Finish getting the toolbar add--on installed
- Addison: Implement a toolbar or other accessible interface within blender to add-on
- Thomas: Work on getting the AnisoMPM code to compile on Windows

Accomplishments:

- Chase:try to reinstall ubuntu.I'm unsure as to why
- Alex: Figured out installation of mods and implementation of various add-on components, including toolbar addition and hotkey presses
- Addison: Alex basically beat me to it! Figured out a good implementation for add-on components we can build off of
- Thomas: Did not end up getting the code to compile. Since Chase is also working on this, I started working on my next goal, which is to create a python script which generates header files similar to the ones they use in their examples. I get the code basically working, but have yet to add a lot of customization.

Setbacks:

- Chase:ubuntu errors with finding the files
- Alex: Several attempts at creating mods failed, due to difficulty in importing multi-file mods into Blender. All one-file mods worked splendidly, may have to keep mods in one file.
- Addison:
- Thomas: Could not get anything to compile. I was trying to get gcc working from the Windows command line. That did seem to work, but still had trouble compiling the AnisoMPM code.

Week-5:

Goals:

- Chase~~:run the demos~~ working on ubuntu.
- Alex: Start working on a pop-up box that can be called via the button on the previously mentioned toolbar.
- Addison: Work on importing .mesh files via the add-on template. If possible, use the meshes directly from working demo
- Thomas: Update my python script to have more customizability. Find all the variables that need to be changed between each header.

Accomplishments:

- Chase:try to reinstall ubuntu into the files for the demos but ubuntu is not Not recognizing the connection or any files on my computer. Meeting with our

Advisor to see if she could help me figure out what is wrong with my computer or ubuntu. Not much.

- Alex: No real progress on the mods this week. Have located API documentation, but midterms have eaten a lot of time.
- Addison: Although I wasn't able to find a method to import .mesh files, I spoke with the authors and found a project that, if we can get it to work, could convert between .mesh and .obj file structures which will work nicely with Blender
- Thomas: Made the header generator more customizable, but not completely. Can now specify the .mesh file to be used along with some other basic parameters (framerate, animation length, etc.).

Setbacks:

- Chase:ubuntu not being able to communicate or locate files on my personal computer and not entirely sure why
- Alex: Time issues and dense documentation loads have not come together to create an environment conducive to the next development phase.
- Addison: Turns out .mesh files are very complicated, and the ones specifically generated by the SIGGRAPH project use a somewhat unique form, but after contact with them I was directed to a couple projects by the same authors, TetWild and PyMesh, that use these types of meshes. I will focus on building/utilizing these to work with the project's meshes
-
- Thomas: Could not find everything that the header files need to change. There's a lot of variables and I've been trying to figure out what they all do. Made some progress with it, but it's not completely finished.

Week-6:

Goals:

- Chase:~~work on figuring out what in the demos can be changed and altered for use in our project~~. Ubuntu.
- Alex: ~~Get the pop-up box to accept user input, some text and some radio buttons, to allow us to set settings as needed.~~
- Addison: ~~figure out adjustable variables in the original code (elasticity, etc.) and see if we can modify them without recompiling the entire project~~ Build PyMesh and/or TetWild and convert a demo .mesh to a .obj
- Thomas: Fine tune the code I wrote the previous week. Figure out what user input it needs to customize the header files properly.

Accomplishments:

- Chase:none
- Alex: Still no real code implementation, but great strides have been made in reading and understanding the API framework.

- Addison: I managed to get the base project of TetWild installed, however there is a large array of dependencies that I am not able to get working and PyMesh is not installing properly
- Thomas: Did not get the code completely done, but I think I found the patterns to how the headers work. There's seems to be a handful of variables that you can easily specify, and then there's chunks of code that give more information about the object (inner material, outer material, other colliding objects, etc.).

Setbacks:

- Chase:ubuntu is not working at the moment, unable to figure out why, on my personal and I think it is my computer and some settings that I have overlooked, but I don't know at this point .
- Alex: More scheduling and dense API documentation woes.
- Addison: Had to push the original goal back to work on the TetWild and PyMesh works found last week. I've been focused on building and running the projects using Docker, which the authors recommended, but after still not seeing success I will try working on them using CMake in the upcoming weeks
- Thomas: Seems there's more to customizing the headers than just changing a few variables. Need to generate different chunks of code that perform different tasks.

Goals/ Plan for Next 3 Weeks:

Chase: getting ubuntu fully recognize my computer and the files on it, this has now become a multi-week endeavor,If not working with Thomas on his approach on getting them to compile

Week 7:
Alex: Implement a functional pop-up box.
Addison: Build PyMesh and/or TetWild using CMake and convert a demo .mesh file
Thomas: Finish my python script for generating AnisoMPM headers. Should get it to generate different chunks of code specifying different parts of the object and colliding objects.

Week 8:
Alex: Get functional buttons/options on the popup box.
Addison: Use our python add-on to call PyMesh/TetWild and import an object from a .mesh
Thomas: Work on getting Blender to call the Python script to generate headers.

Week 9:
Alex: Get the pop-up box to accept user input, some text and some radio buttons, to allow us to set settings as needed.

Addison: figure out adjustable variables in the original code (elasticity, etc.) and see if we can modify them without recompiling the entire project [original goal that has been pushed back]
Thomas: Get Blender to call GenerateHeaders.py with as little effort from the end user as possible. They shouldn't have to know every detail of what input is required. Maybe they could adjust a few sliders for some variables and the rest of the input could be figured out automatically.