Assignment 1

Implement the following Force Generators. The logic and pseudo code for the first 2 can be found in the book as denoted by the chapter and section. For the 3rd one implement as you see fit. Each simulation also needs to have a level of user control explained under each section. All the simulations should have the default FPS controls enabled.

Bungee chord (Chap 6, 6.2.3) (25 %)

This is essentially a one directional spring - it pulls but doesn't push. Two objects can be connected viz. a bungee chord, so you should follow our implementation of PairedSprings. The simulation needs to start with one object tied to a fixed point with a bungee chord.

User control - At any point during the simulation I should be able press a button (say space) and this should spawn a new object at the current (camera or camera + forward) location which is attached to the previous object using a bungee chord as well!

Buoyancy (Chap 6, 6.2.4) (40 %)

True buoyancy can only be determined by calculating actual volume of liquid displaced. You don't have to do that - you can simply make it proportional to the depth of the object as explained in the book.
For this simulation you need a cubical volume which is the only region in which entities with a "BuoyancyComponent" can experience buoyancy force. Please have some visual to represent this volume (a wireframe box / cube will suffice).

User Control -

- At any point during the simulation I should be able press a button (say space) and this should spawn a new object with a "BuoyancyComponent" at the current (camera or camera + forward) location. I will use this to test the bounds of your bouyancy volume.
- I should be able to increase the density of the liquid by pressing 'up arrow' and reduce it using 'down arrow' (within some predefined limits, density can obviously no go below zero). Have some visual to represent the current density (either through color or text.

N-Body (35 %)

This is a simulation of multiple objects under the effect of mutual gravity (ONLY). For this simulation turn off the default gravity force generator. The gravitational force between two objects are given by

where 'm1' and 'm2' are the masses and 'r' is the distance between them. Each object experiences the force in the direction of the other object. In reality the value of G is $6.674\times10^{-11}\ m^3\cdot kg^{-1}\cdot s^{-2}$
However that will not give very realistic results unless the masses are huge, so set the value of G to whatever looks good.

I do not expect you to do any sort of spatial partitioning, so you can simple brute force using a double for loop like we did for the sphere contact demo.

The simulation should start with 10 spheres each with a different mass and non zero velocity. This simulation needs no user controls beyond the normal FPS controls,

This is sort of what it should look like, obviously with a lot less objects:
https://www.youtube.com/watch?v=j6AOA5_5Hyo

NOTE 1: I only care about the final result of your simulation, that means while I recommend using the ECS based engine we have been using in class, if you so choose you can build everything from scratch, however that means using no external physics library and no plagiarism (obviuosly).

NOTE 2: You will need to show me the assignment in class after reading week.