**Model Planning:**

The behavior that I intend to model is the mechanism that allows submarines to submerge underwater, surface to the top and to maintain a constant depth underwater. Specifically, the ballast tanks in a submarine allows them to achieve this feat. The ballast tanks are a compartment in a submarine that allows them to change the buoyancy of the submarine by taking in water, this increases the density and the weight of the submarine which gives it a negative buoyancy that lets the submarine sink underwater. To ascend, submarines typically expel the water from the ballast tanks, increasing its buoyancy, and to maintain a certain amount of depth underwater, the submarine must have a neutral buoyancy in which the overall density of the submarine is equal to the amount of displaced water around it. The concept behind this is called Archimedes’ Principle, which basically describes how buoyancy works in terms of the forces and mechanics that allow an object to sink or float on water.

The objective of my simulation is recreate how submarines work when it comes to them control their buoyancy through the use of ballast tanks. Therefore, the simulation would have to take into account the equations for the forces that act upon the submarine that allows it to submerge and ascend such as the equation of the force of buoyancy and the variables for it such as the density of the submarine, the density of water (both seawater and fresh water), weight of the submarine and weight of the displaced water. Because there are many different types of submarines, we would also have to consider the different densities and masses for those submarines as well as the capacity of their ballast tanks. From all these variables and equations, the simulation would then show whether or not the submarine is submerging, ascending or staying at the same depth and also show the change in depth if the buoyancy for it is not zero. Overall, the simulation should be able to accurate simulate this behaviour with a certain degree of realism to it. The simulation will be a continuous, given that the variables and equations that dictate how this simulation will work. For the sake of simplicity, I will eliminate certain variables such as any horizontal movements that submarines may usually have in order to fully demonstrate the mechanics of submerging and surfacing.

The interactivity involved in the simulation will strictly allow whoever is using to adjust the amount of water that the ballast tanks will have. This in turn will allow the user to adjust the density and weight of the submarine, which in turn will adjust the buoyancy of the submarine, thus allowing it to submerge, surface or maintain a certain amount of depth underwater. Since density and weight of water is involved in the simulation, it is possible that the user will also be able to choose whether the simulation is using fresh water or sea water, since they do make a difference in the simulation and for buoyancy. For the submarine, the user might also be able to specify the details on it, such as the weight of the submarine and the ballast tank capacity. If possible, preset values for the submarines could be made available such as preset values from submarines such as something like a midget submarine to bigger ones such as military class submarines.

The simulation loop for this consists of starting the simulation with the initial variables, which will advance the simulation and display the current results. The loop will keep going as is until the simulation is terminated or until the user specifies new values for the variables such as weight and density as well as the amount of water inside the ballast tanks. It is possible that the user can just increment the values as opposed to just inputting a value for simplicity. Afterwards, the simulation will take these new variables and display the current result and repeat until variables are changed or until the simulation is terminated. The input variables used to configure my simulation will be the variables used in the equation to calculate for buoyancy such as mass of the submarine, density of the fluid (in this case, the water), volume of the displaced water, gravity, and force of buoyancy from those other variables. The state variables are the variables that will change every so often, we will obviously have delta T since this is a continuous simulation and we will also have an equation for the change in velocity and acceleration since that will occur when the submarine changes its mass ( , ), change in position too when it has a positive buoyancy or negative buoyancy (). The equation we will use to calculate for the change in force will be:

Finally, the output variables produced by the simulation will be the depth that the submarine is at in meters, the capacity of the ballast tanks given in the mass of it, the time elapsed in the simulation, the buoyancy force applied to the submarine in Newtons, the acceleration and velocity in which the submarine is moving vertically. These output variables would be used in the simulation loop to keep it going for every time step until it is terminated.

Finally, for this simulation, I intend to use Open Source Physics in Java to visualize this data. I intend to have it work very similarly to the bouncing ball program in which it will accept inputs and visualize data accordingly. I intend to interpret the data visually by having a graph show a representation of the submarine as it submerges and goes through the processes it usually does. This will also allow us to see if the data is valid based on whether or not it follows the usual behaviour of submarines.