Carter Meyer, 9/13/2022, Data Structures, Environment Simulation

<u>Pitch</u> It will be a simulation of the spread of a disease. Objects will be placed in a 2d vector like a grid. The user will set the population of each demographic. A button will appear allowing the user to proceed one day. At the end of each day a grid will appear showing each person's location, then on the side stats will appear showing how many are infected from each demographic and the entire population as a whole, and out of those infected it will show how many are asymptomatic, symptomatic, or dead because of the disease. Each day the objects on the grid will move by one grid space. An object has a chance of getting infected by another object if it is 1-2 grid spaces away (the infectionChance will decrease with distance.)

<u>Outline:</u>

Main: Would ask the user for the number of each demographic they would like (3-10 for each demographic). Those numbers will then be put into a Simulation constructor

Human class. There will be an "infected" boolean. If tru it says infect. If "infected" is true then a "level" int could be set to 1/2/3. Setters and getters abound. Have an infectable int: that just states the rate that the object could be infected. There will also be an age int. 1 is child, 2 is adult, 3 is elder.

levelIncrease function:
- if level is set to 1(Incubation period) 25% chance it will turn to 2(asymptomatic) 25% chance it will turn to 3(symptomatic),
- if level is 2 or 3: 75% chance it will turn to 0 (not infected) and turn infected to false, 25% chance it will turn to 4(death)

setInfect():

Simulation
Simulation constructor would spawn with 3-10 children, 3-10 adults, and 3-10 edlerly objects in a 2D vector at random locations. w/ setting the population int to 9-30. Also have it start with a certain number of the population set as ill. Day counter. Will include instances of Human via a

2d vector of "Humans
childDied variable: keeps track of every dead child
adultDied variable: keeps track of every dead adult
elderDied variable: keeps track of every dead elder
Day counter

movePeople function: Go through vector of objects(people): if there is an object detected 50% chance it moves life, right, up, or down in the vector if there is space.

calculateInfectionRate: First, set infection rate to 0. Now, If the object is 1 space away from an infected person, add 50% to their infectable int, if an object is 2 spaces away, add 30% to their infectable int. If this is triggered then add on 40% if they are elderly, 30% if they are a child, and 20% if they are an adult.

proceedInfection function: goes through the vector and activates levelIncrease function for all objects with infection as true, otherwise activates the infect function.

readStats: Go through each object: for each demographic have a counter for how many are dead, alive, at a certain stage of infection, or uninfected (these are printed out right away instead of being fields in the class because they change often and may go from infected to dead, or infected to uninfected). Then if their level is 4: add one to their demographics death counter and delete the object

dayForward function:
- Day counter +1;
- Call movePeople
- Call resetInfectionRate Function
- Call calculateInfectionRate function
- Call proceedInfection function

List of tests: List of tests you plan to conduct after your code is done and the purpose for testing them
- Test different amounts of humans
- Test if the percentage rates are accurate and work
- End to end testing of 100 days
- See how many days go by before everybody dies