Disease Transmission Simulator

December 15, 2020

By:
John Kulins
Emily Gallagher
James Meurer
Thomas Kohut

Contents

Vision and Business Case

Introduction

The Center for Disease Control and Prevention was established in 1946 and is one of the major components of the Department of Health and Human Services. Since its inception, the agency has strived to protect American citizens from health and safety threats using the highest-quality scientific data available. The CDC also monitors global health concerns and works with global health organizations, such as the World Health Organization, in order to detect and fight off diseases before they can reach America's shores.

Business Goals

Diseases have always been at the forefront of public health risks. Keeping the public informed on how harmful and contagious these diseases are is an important component to preventing them from spreading. When people understand the risks caused by contagious diseases, they are more likely to follow prevention protocols and keep both themselves and other people safe. It is in the CDC's best interest to convey this information to the public in a way that is easily accessible and understandable.

Problem Statement

The recent outbreak of COVID-19 has led to a global pandemic that is causing many lives to be lost and left many survivors with lifelong internal organ damage. Although there have been efforts to create public safety protocols in order to minimize the spread of the virus, misinformation has been spreading rapidly through social media and other sources that undermine these safety protocols. It has become clear that there are many people that still do not understand how diseases spread and why these safety protocols are so important. The following are various problematic beliefs that are currently circulating due to a lack of public understanding:

- The disease is no worse than flu
- The disease is only a threat to the elderly and those who are immunosuppressed
- Stay at home orders and masks are a form of government control or a wider conspiracy
- People believe they will not catch the disease, even if they are also at risk

Business Case

The creation and distribution of the Disease Transmission Simulator will have many positive outcomes for the CDC, public, and even the developers. From the perspective of the agency, the program will show the public that care is put into not only understanding diseases, but also how to educate the public in order to increase their understanding. Taking this initiative shows a willingness to serve the public and benefits the agency through their well-earned good-will and trust.

From the perspective of the developers, there are similar programs that already exist that may be used as reference points for how the users could interact with the program. The developers may build upon the features of these other programs that worked well and also implement new features to complement the current developments.

Business Constraints

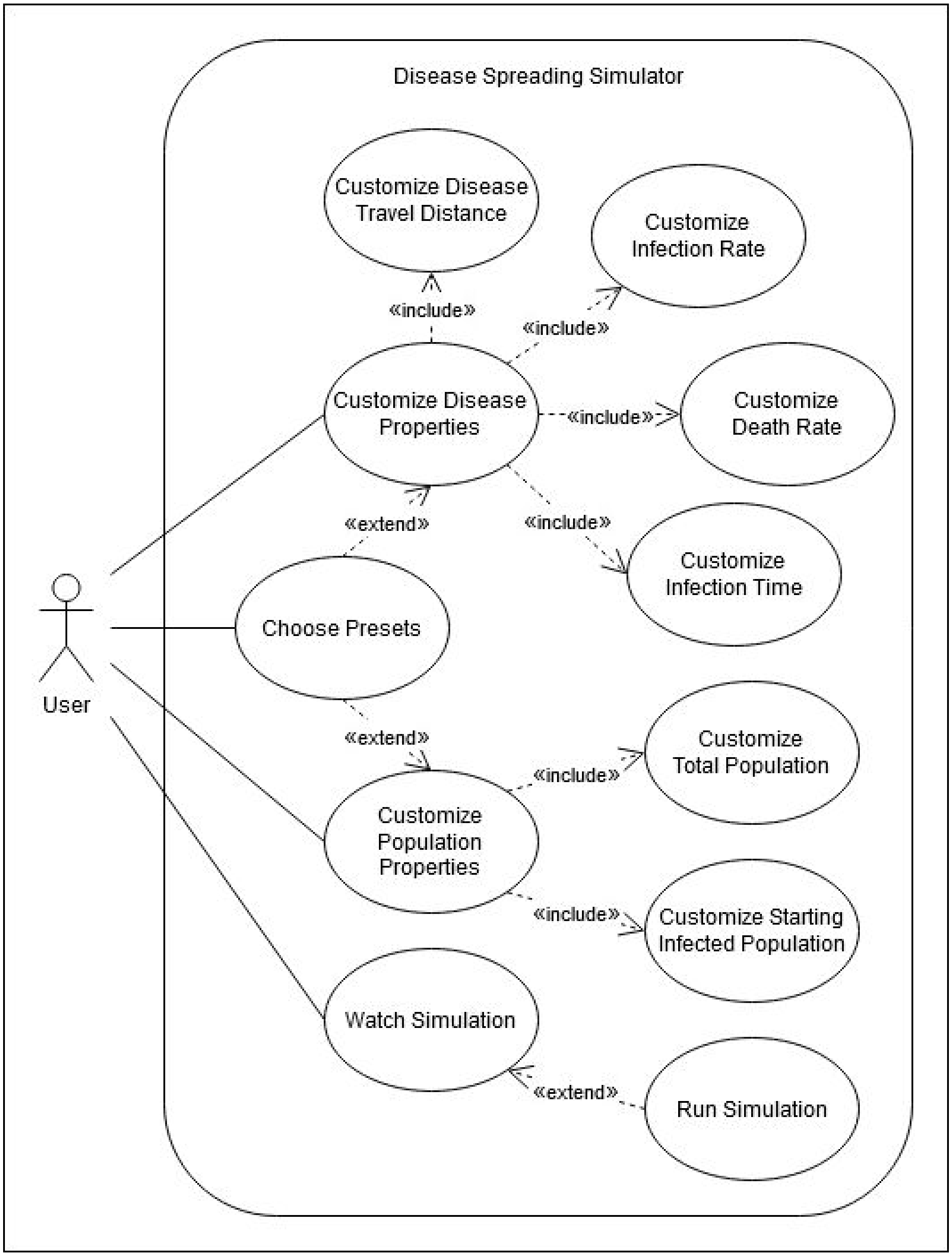
All projects have a variety of unique constraints that both the client and developers must be aware of. For this project, a few notable constraints have been identified as follows:

- Awareness of an issue does not always lead to people taking precautions against it. As an example, there are people who will continue to refuse to wear face masks, even after presented with evidence that masks help to combat the spread of airborne diseases.
- There is an increased likelihood that a person who is not technologically adept or familiar with computers may struggle or even refuse to use the system.
- Not all schools nor businesses are using up-to-date technologies and may have trouble accessing the program.
- People have to be aware of the existence of the system before they may use it in the first place. As such, the system will need to be advertised in order to bring awareness to it, which will cost additional resources from the agency.

Executive Summary

The Disease Transmission Simulator is meant to help the agency with spreading information about COVID-19 as well as teaching about safe practices when dealing with the disease . Being able to accomplish these goals will be the agency's top priority and also allow them to be seen as a good provider of information. In order to accomplish this goal, the new system which will be in place will have to go through many changes in iterations as the spread of COVID-19 continues over it’s course. As well as be updated with any new information pertaining to the disease. Another positive of this system would also allow the agency to have a system to reuse for many years to come, as well as allowing us as the developers to have a system to reuse for similar projects in the future.

Use-Case Models



Use Case 1:

Primary Actor: Customer
Precondition: None
Postcondition: The user runs a simulation
Main Success Scenario:

1. The customer selects the disease and hits the play button.

Use Case 2:

Primary Actor: Customer
Precondition: None
Postcondition: The user runs a simulation with optional features added.
Main Success Scenario:

1. The customer selects the disease, disease prevention method, and presses play.

Use Case 3:

Primary Actor: Customer
Precondition: The simulation has successfully started running
Postcondition: The simulation successfully stops.
Main Success Scenario:

1. The customer waits for the point when they stop during the simulation and presses the pause key.

Use Case 4:

Primary Actor: Customer
Precondition: The simulation is paused or has ended.
Postcondition: The user successfully restarts the simulation.
Main Success Scenario:

1. The customer presses the start button.
2. The customer enters new desired inputs and presses the start button

Risk Management Plan

There are multiple risks involved in the development of the software. One such risk is a group member falling behind without communicating as such. This could become a hindrance to other group members if a group member falls behind on a certain part, it could prevent the implementation of other features that require it to function correctly. In the worst case scenario, the project may fall back irreparably to the point where cuts to the project will have to be. Another risk is miscommunication between group members. This would hinder coordination between group members and make it difficult to coordinate development of larger parts of the project, such as connecting front and back ends of the project. There is also the risk of the information being unavailable or not accurate, particularly with how certain preventative measures affect diseases. This may not be the case with COVID-19, but this is more likely with common or older diseases, such as the common cold or the plague. This may result in some guesswork on our part. To follow up on this risk, there is also the risk that the mathematical formulas are off, resulting in inaccurate simulation results. This can be handled through multiple group members reviewing the data and checking to see if it matches up to the formulas.

Iteration Plan

The iteration plan is an incremental plan with each segment based on the length of the sprint. This will allow our group to work on the project over time on an easily trackable timescale. During these increments, each group member will review and complete each objective set out for them at the beginning of the sprint. If one member is having difficulty or falling behind, another will assist.

Software Development Plan

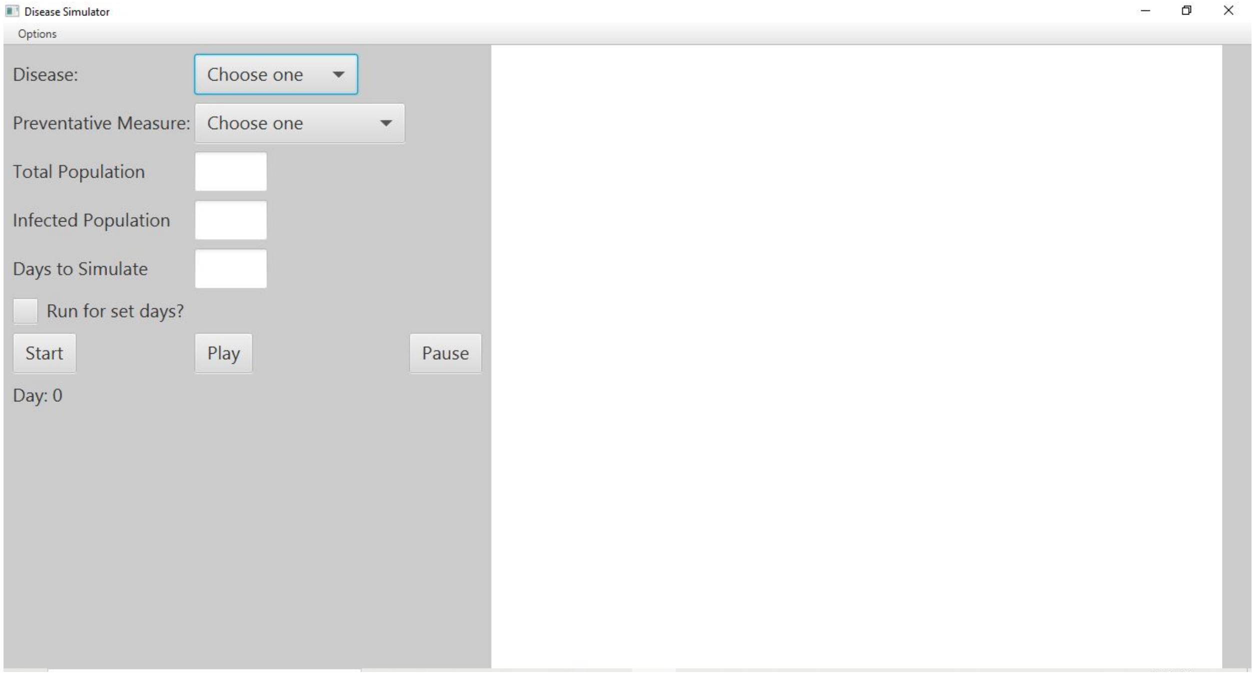
For sprint one we discussed the design and overall idea for the project. During this time we figured out how we were going to create the final product and also what it should look like in terms of design and aesthetics. For sprint two we decided to work on coding the user interface and beginning on coding the classes that will be used to determine the data. Additionally, during this time we would be researching for relevant data to use in the program. This coding process will be carried into the next sprint. Then, for sprint three we will still be working on the coding aspect but given that more of us can work on it it should be moving by a bit faster. By the end of sprint three we should be done with the main code and for sprint four we would be working on the finishing touches. This means the final colors and making sure that it works as intended.

Implementation Model

The following class we have provided is the current database class that we are using in our system. This class is very important to our system as it is storing the data necessary for our simulation to run, and contains the data necessary for the simulation to run accurately.

```
package com.sim;

/**
 * Static class that contains all {@code Disease} and {@code Precaution} object presets.
 *
 * @author Emily Gallagher
 * @version 1.1
 * @since 0.2
 */
public final class Databases
{
   //----- FIELDS -----
   /** Contains all Disease preset information. */
   private static final Disease[] DISEASES_;

   /** Contains all Precaution preset information. */
   private static final Precaution[] PRECAUTIONS_;

   //----- CONSTRUCTORS -----
   // Initializes static fields.
   static
   {
      DISEASES_ = new Disease[]
      {
         // Initialize all Disease presets here.
         new Disease("Covid-19", 100, .037f, .1f,.018f , 14, 8),
         new Disease("Influenza", 100, .055f, .1f,.0009f , 2, 10)

      };

      PRECAUTIONS_ = new Precaution[]
      {
         // Initialize all precautionary measure presets here.
        new Precaution("None", 1.0f),
        new Precaution("Mask", 0.3f),
        new Precaution("Distance", 0.2f),
        new Precaution("Masks and Distance", 0.1f)
      };
   }

   /**
    * Sole Private Constructor:<br />
    * Prevents the static class from being instantiated.
    */
   private Databases() { }

   //----- METHODS -----
   // Get
   /**
    * Returns a copy of the Disease found at the specified index of the Database's disease array.
    *
    * @param index The index of the Disease.
    * @return A copy of the Disease found at the specified index.
    * @throws IndexOutOfBoundsException Thrown if index is less than 0 or greater than the
length
    *                   of the disease array minus 1.
    */
   public static Disease getDiseaseCopy(int index) throws IndexOutOfBoundsException
   {
      return new Disease(DISEASES_[index]);
   }

   /**
    * Returns a copy of the first Disease in the Database's disease array with the specified name.
    *
    * @param name The name of the Disease.
    * @return A copy of the first Disease found with the specified name or null if none is found.
    */
   public static Disease getDiseaseCopy(String name)
   {
      for (Disease disease : DISEASES_)
      {
         if (disease.getName() == name)
         {
            return new Disease(disease);
         }
      }

      return null;
   }

   /**
    * Returns the number of Disease objects stored in the Database.
    *
    * @return The number of Disease objects stored.
    */
   public static int getDiseasesLength()
   {
      return DISEASES_.length;
   }

   /**
    * Returns a copy of the Precaution found at the specified index of the Database's
    * precautionary measure array.
    *
    * @param index The index of the Precaution.
    * @return A copy of the Precaution found at the specified index.
    * @throws IndexOutOfBoundsException Thrown if index is less than 0 or greater than the
length
    *                             of the precautionary measure array minus 1.
    */
   public static Precaution getPrecautionCopy(int index) throws IndexOutOfBoundsException
   {
      return new Precaution(PRECAUTIONS_[index]);
   }

   /**
    * Returns a copy of the first Precaution in the Database's precautionary measures array with
    * the specified name.
    *
    * @param name The name of the Precaution.
    * @return A copy of the first Precaution found with the specified name or null if none is
    *        found.
    */
    public static Precaution getPrecautionCopy(String name)
    {
        for (Precaution precaution : PRECAUTIONS_)
        {
            if (precaution.getName() == name)
            {
                return new Precaution(precaution);
            }
        }

        return null;
    }

    /**
     * Returns the number of Precaution objects stored in the Database.
     *
     * @return The number of Precaution objects stored.
     */
    public static int getPrecautionsLength()
    {
        return PRECAUTIONS_.length;
    }
}
```

User Interface



Disease Simulator

Options

Disease: Choose one

Preventative Measure: Choose one

Total Population

Infected Population

Days to Simulate

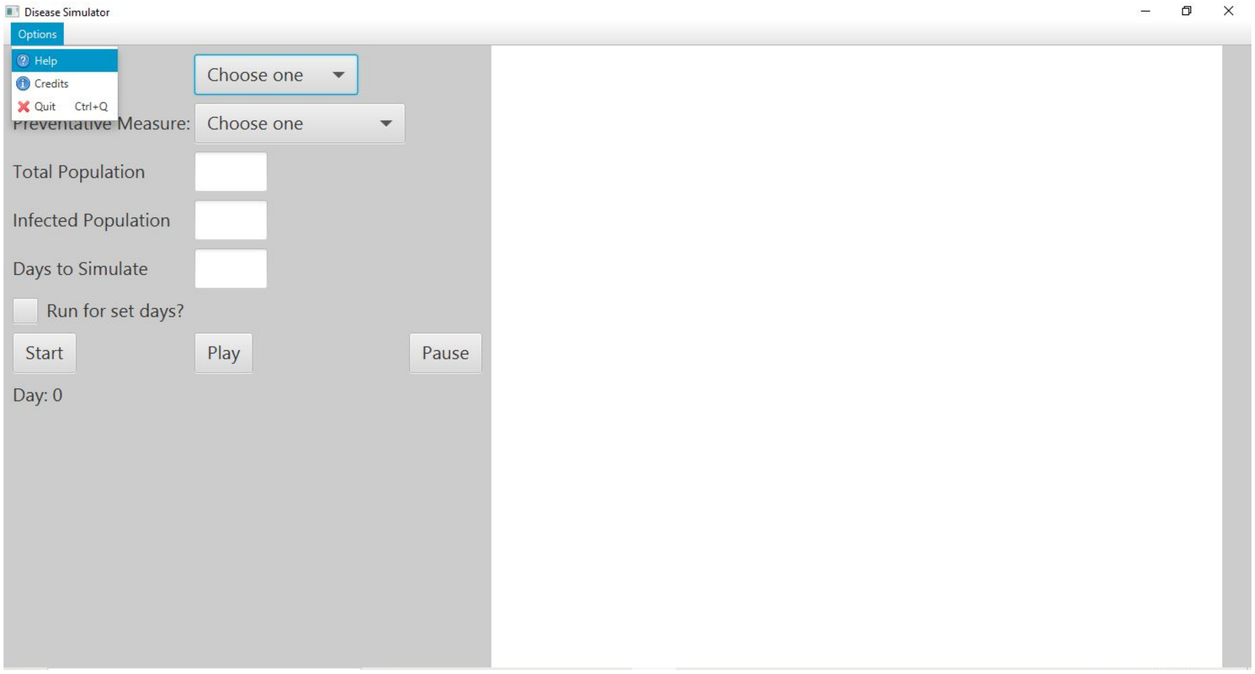
□ Run for set days?

Start Play Pause

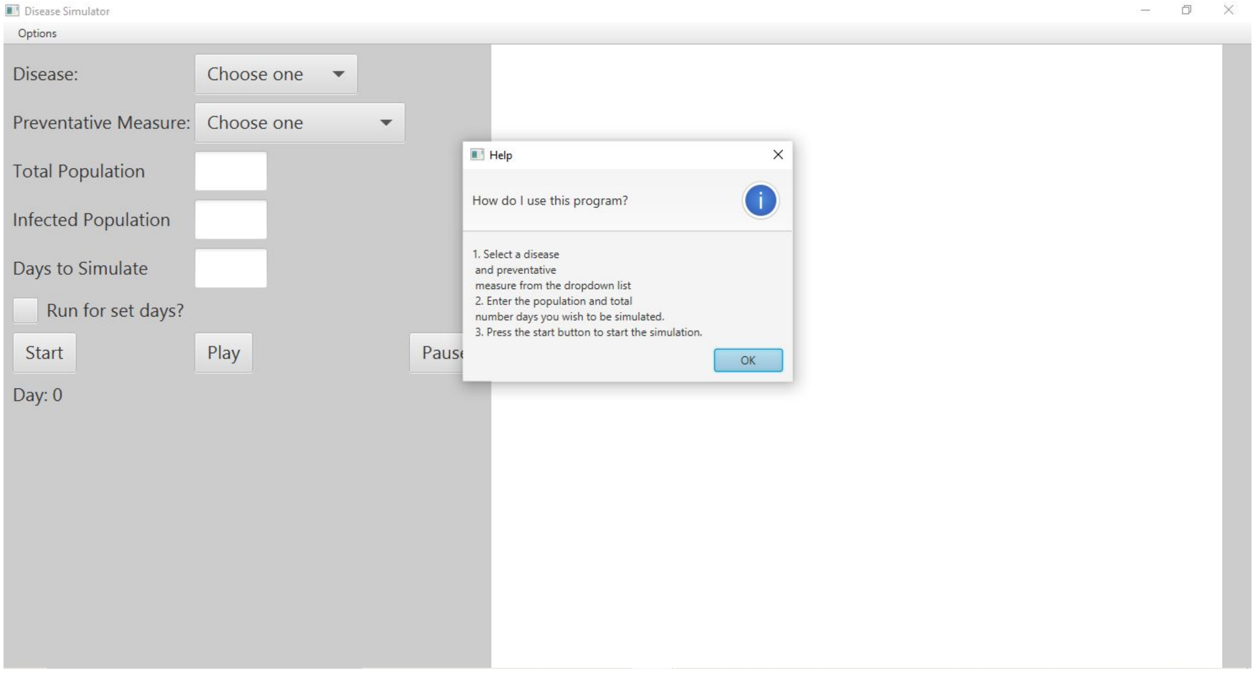
Day: 0

First Start Up

As soon as users start up the program they are presented with what looks to be an elaborate screen. However it is very simple to use. The screen asks users to choose from a drop down list of diseases so far only two have been implemented COVID-19 and Influenza. It also asks the user to choose from a list of preventative measures such as wearing a mask, practicing social distancing regulations, both, or even none of them. From there they are also asked for a total population to put in, an infected population amount and the amount of days to stimulate (this option comes into play when the “run for set days?” checkbox is checked off. From there users will hit start to run the simulation, pause to be able to pause in between moments of the simulation and then finally play in order to resume the simulation playback



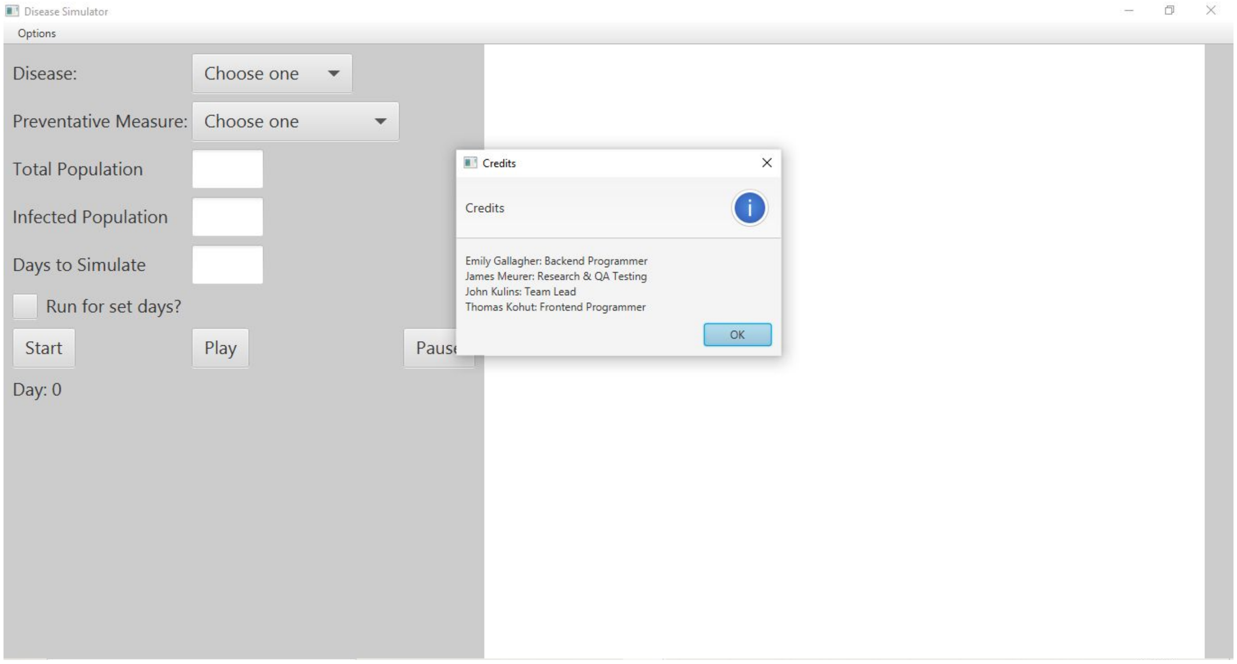
Clicking the options tab: When a user clicks on the Options tab. Three options come up the first being a help option which explains to the user how to use the program which can be seen below.



How do I use this program?

1. Select a disease
and preventative
measure from the dropdown list
2. Enter the population and total
number days you wish to be simulated.
3. Press the start button to start the simulation.

Clicking on the credits option will bring up the credits which will list those who had created the program as shown below.



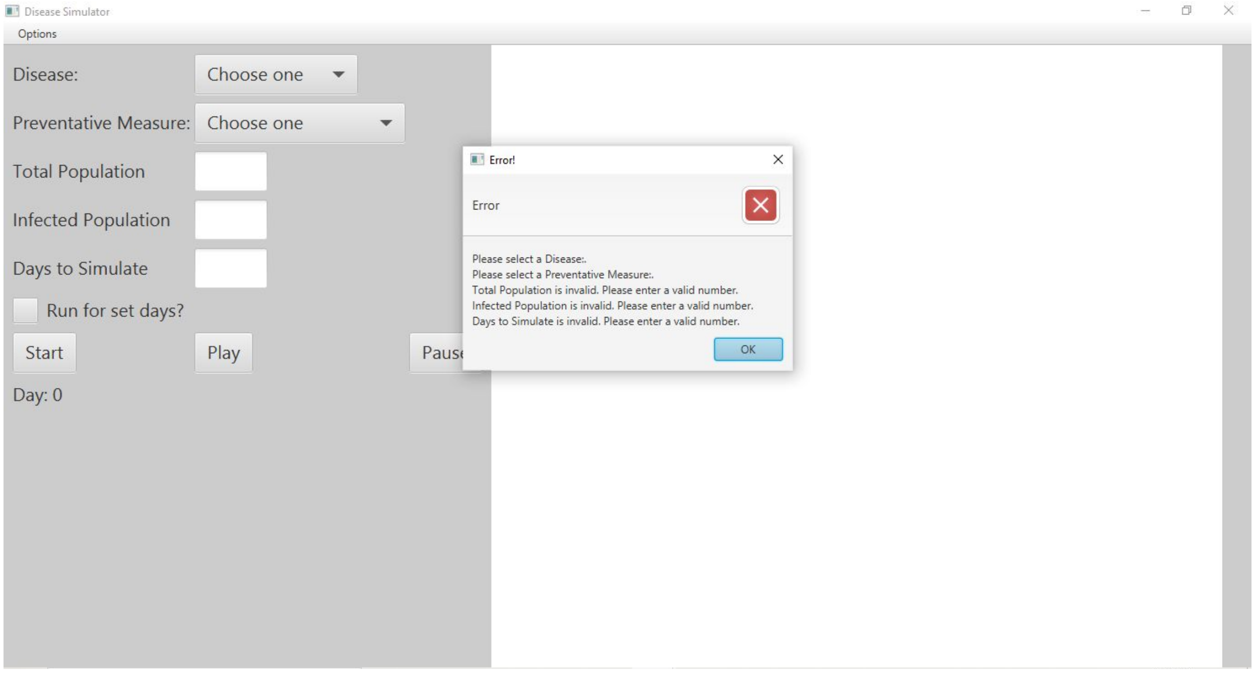
Credits

Credits

Emily Gallagher: Backend Programmer
James Meurer: Research & QA Testing
John Kulins: Team Lead
Thomas Kohut: Frontend Programmer

And clicking close will close out of the program itself.

Not Entering the Data



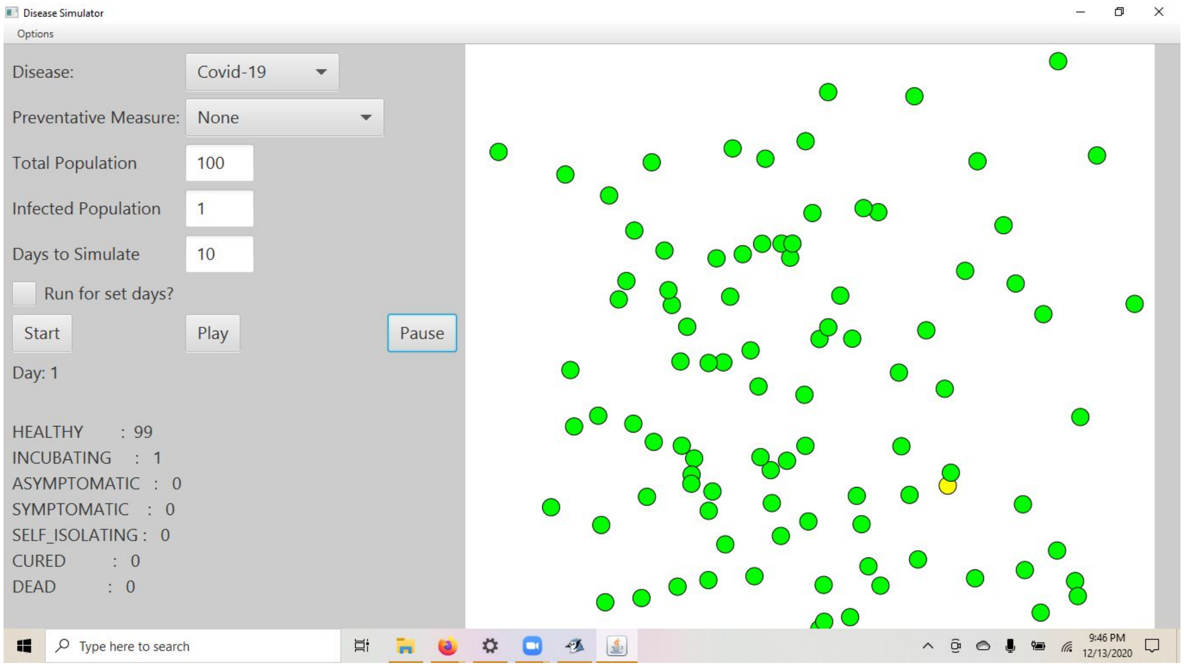
Error!

Error

Please select a Disease:.
Please select a Preventative Measure:.
Total Population is invalid. Please enter a valid number.
Infected Population is invalid. Please enter a valid number.
Days to Simulate is invalid. Please enter a valid number.

Not entering any data into the program will pop up one of a few errors. Some of these errors will include the user not selecting a Disease, a preventative measure, a total population that is not valid, and days to simulate not being valid.

Running the Program



Disease: Covid-19

Preventative Measure: None

Total Population 100

Infected Population 1

Days to Simulate 10

□ Run for set days?

Day: 1

```
HEALTHY       : 99
INCUBATING    :  1
ASYMPTOMATIC  :  0
SYMPTOMATIC   :  0
SELF_ISOLATING :  0
CURED         :  0
DEAD          :  0
```

After users have entered their information and hit start, the simulation will run and display their results. Green are healthy individuals, Yellow are those who are incubating the infection. Red shows thoses who are symptomatic, blue shows those who are cured and red shows those who are dead.

Glossary

Term	Definition	Aliases
Center for Disease Control and Prevention	Division of HHS. Monitors and manages illnesses and global health concerns.	CDC
COVID-19	Disease discovered in 2019. Caused a global pandemic.	Coronavirus
Department of Health and Human Services	A cabinet-level executive branch department of the U.S. federal government with the goal of protecting the health of all Americans and providing essential human services.	HHS
World Health Organization	A specialized agency of the United Nations responsible for international public health.	WHO