Simulating the Effects of Social-Distancing Technology on the Spread of a Pandemic

Tilburg University

CSAI

Members/SNR:

Djourdan Gomes-Johnson (2033032), Pietro Garroni (2038045), Ndivhuwo Nyase (2047606)

Word Count: 2119

Problem Statement

At the dawn of a technologically advanced world, it is crucial to contain and manage any global threat to humanity. Therefore it is of the utmost importance to respond to disease outbreaks as effectively and efficiently as possible.

The Covid-19 crisis has surpassed its anniversary, threatening a second wave of the pandemic. During the time of this study, organizations and governments are searching for a tool and/or optimizing their respective methods for reducing the spread of the Coronavirus.

The Netherlands is implementing a smartphone application called CoronaMelder, allowing for the tracking of user locations and the collecting of symptomatic information. The users on this social-network inform their counterparts of their current state of health and then proceed to use the application for suggestions of appropriate actions. Specifically, the Dutch application tracks whether or not users have the virus and then warns the users if they have been in close proximity with other users who are infected with the Coronavirus. At the time of writing, more than 3 million people in the Netherlands have downloaded this application, however it has been estimated that 40-50% of the population should utilize this application for halting the spread of the Coronavirus. This approach to contact tracing is being widely used across the world as many citizens with a smartphone now have the tool to collect such data in real time.

The emergent field of Multi-Agent Systems allow us to collect, visualize and perform analytics on data to improve and properly respond to disease outbreaks. With the use of Netlogo we will investigate the effectiveness of social networking applications for the prevention of Coronavirus spread. We plan on creating an environment with agents that can spread the Coronavirus. Additionally we have different agents, those who have the social network application that warns the agents that they have the virus and should isolate. The other agents will not have the application therefore, they do not know if they have the virus or if they are in close contact with anyone with the virus, and will only isolate through individual recognition of their current state. This is in contrast to the application users that are expected to update their results and take appropriate action (quarantine) once alerted. The experiment has a purpose of testing the following hypothesis: the effectiveness of contact tracing applications and what percentage of the population must adopt the application to stop the spread of the Coronavirus.

We address this investigation through inspiration from the EpiDem Basic and EpiDem Travel and Control models.

Model Description

The construction of this agent-based analytical model initiates with specific design choices. The dynamic model environment consists of programmatic and heterogeneous agents, that are divided into three distinct groups: application users, non-application users, and initially infected agents. These agents own a series of parameters, where five of the nine are true/false values.

The agent's own properties that determine their behaviour in the environment. The list of properties that agents have are:

- Infected
- Cured
- Strategy
- Isolate
- Unisolate
- Isolation-tick
- Infection-length
- Isolation-tendency
- Number

The infected property determines if the agent is infected with the Coronavirus or not. If this property is true for the specific agents, they are able to spread this infected property to agents that are of a patch radius of 0.3, spreading the Coronavirus.

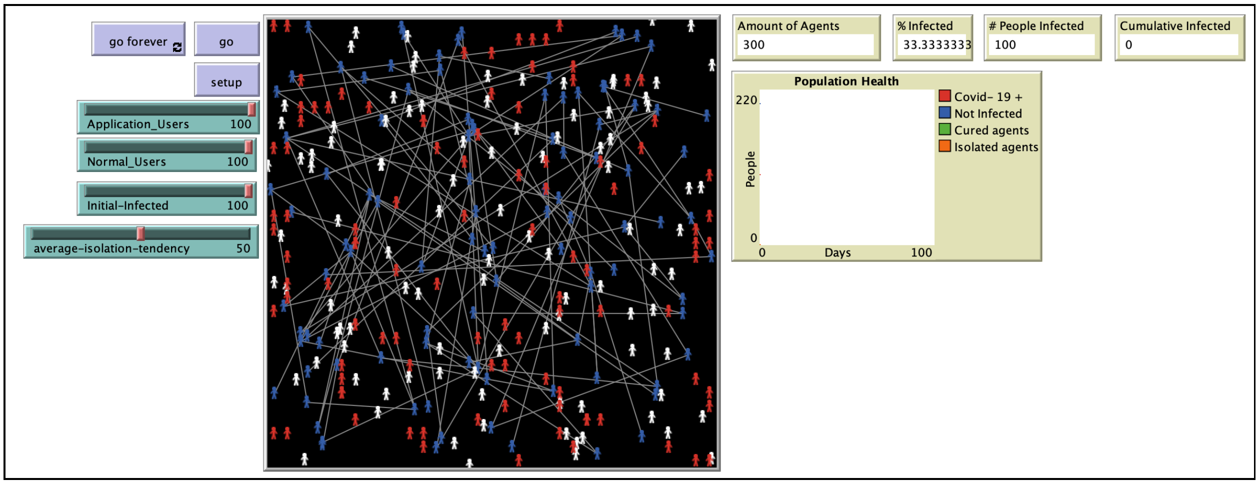
The Cured property refers to the agent state of curity from the Coronavirus or not. This enables the agent to not be infected again. The only way an agent can be cured is to be isolated for a random amount of time, decided by the properties of isolation-tick, infection-length, isolation-tendency, and number.

Furthermore, the strategy property is essentially the focus of this study. This property determines if an agent is on the social networking application. In addition to this, when they are on this network, they are linked with each other, and can access instant testing. Therefore, they know when they become infected. Moreover, once the agents with the strategy become infected they immediately perform the isolate function and move to an empty patch that other agents cannot access therefore they cannot infect any other agents. This is a similar approach to the real world as people who are warned that they have been in close proximity to people with the virus are more likely to isolate for two weeks and get instant testing to determine whether they have the Coronavirus or not.

This model used a series of procedures that simulate real-world actions in a disease-spreading environment.

- Infect: Infects the other agents with the Coronavirus
- Isolate: Isolate the agent, removing possible interactions with other agents.
- Cure: Allow the agent to be immune, restricting further spreading.
- Unisolate: Take the agent out of isolation, returning the patch color from white to black.
- Move: restricting the entering of isolated areas, and allowing the movement of agents.

Figure 1: Model Interface



Application_Users 100

Normal_Users 100

Initial-Infected 100

average-isolation-tendency 50

Amount of Agents 300

% Infected 33.3333333

People Infected 100

Cumulative Infected 0

The Model Interface includes a set of buttons, sliders, monitors, and a plot that aid in the visualization and manipulation of our simulation at an observer's level. The interface addresses the SEIR acronym, displaying the number of Susceptible people, Exposed people, Infected people, and Recovered people. However this model follows a different set of categories (unisolated, infected, cured, and isolated).

Buttons:

- Go forever: runs the simulation until it reaches one of the potential limits set under the go procedure. These include all agents to be infected, all to not be infected, and all cured.
- Go: runs the simulation for one tick, where one tick is equal to the length of one day.
- Setup: this sets-up the simulation, clearing the environment and creating the Application_Users, Normal_Users, and initial-infected users.

Sliders:

- Application_Users: by increments of 1, this slider allows for a minimum of 0 application users, and a maximum of 100.
- Normal_Users: by increments of 1, this slider allows for a minimum of 0 non-application users, and a maximum of 100.
- Initial-Infected: by increments of 1, this slider allows for a minimum of 0 initially infected agents, and a maximum of 100. A limitation to this model is that the initial-infected count must be less than the sum of application users and normal users.
- Average-isolation-tendency: this value impacts the isolation-tendency of the agents in the normal users and initial-infected groups. Isolation-tendency is a randomized procedure for allowing the isolation of agents who are not on the social network.

Monitors:

- Amount of Agents: counts the total agents in the environment.
- %infected: calculated under the to-report %infected code, this monitor allows the observer to immediately recognize the percentage of agents that are infected compared to the entire population.

- # People Infected: this value displays the actual number of infected agents.
- Cumulative Infected: this value sums the entire number of agents that have been infected during the simulation, regardless of a state of curity.

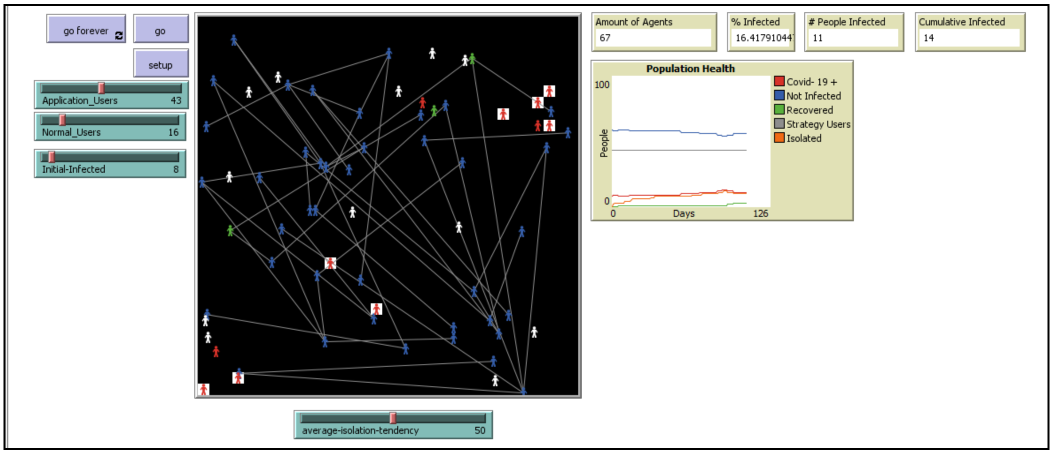
Plot: people to days plotted to show trends of the populations of infected, not infected, cured, and isolated agents.

The decentralized design of the model allows for a destandardization of common knowledge, where information distribution varies between the three distinct groups. This difference highlights the particular goal of this model, providing a higher level of knowledge for users using the application. The 'strategy?' users share information within their network, allowing rapid communication to self-isolate when infected. Understandably, a minimum of two agents must be set for the application users. Without two, the users will not have a higher level of knowledge, causing an error in the code.

The radius of infection for agents was set at 0.2. This number was dictated through observation of infected agents having difficulty spreading the disease, and had to be set to legitimately represent the real-world.

The coloring of agent groups, restrictions of world mapping (horizontally and vertically), symbolizing ticks into days, and randomizing the length of time of curity for each agent, are all implemented in the model to allow this simple environment to symbolize the complex real-world environment of disease spread.

Example:



Application_Users 43
Normal_Users 16
Initial-Infected 8
average-isolation-tendency 50

Amount of Agents 67
% Infected 16.41791044
People Infected 11
Cumulative Infected 14

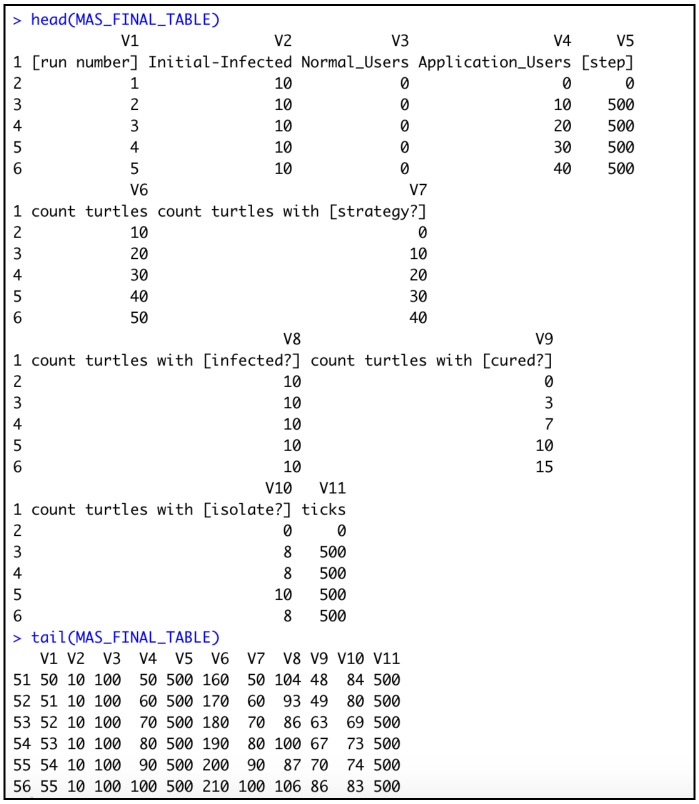
In this specific example, The monitors on the right side of the model show that they are 67 total agents, 16% of those agents are infected, and the cumulative infected agents are 14, thus showing that 3 other agents have been cured. In addition to this the sliders show that there are 43 application users (illustrated by the blue person shape) and there are 16 normal users (illustrated by the white person shape) and the amount of agents that were initially infected is 6. The application users are linked with each other and receive instant testing when infected and isolate themselves after being infected.

Results

After success in model design, a Behaviour Space experiment was run for a total of 55 runs. Each run was repeated one time, measuring total agents, total strategy users, total infected agents, total cured agents, and total isolated agents. The Initial-infected was standardized to 10 per trial, Normal Users ranged from 0 to 100 with increments of 25, and Application Users ranged from 0 to 100 as well, but with increments of 10. A time limit of 500 was set to roughly represent the time Corona has been spreading. This data was organized under a table and imported into R Studio for further analysis.

Below Figure 2, R Studio generated the first six and last six trial runs. This provided our initial visualization of our Behavior Space experiment.

Figure 2: Head and Tail Summaries of the Data Collected (using R)

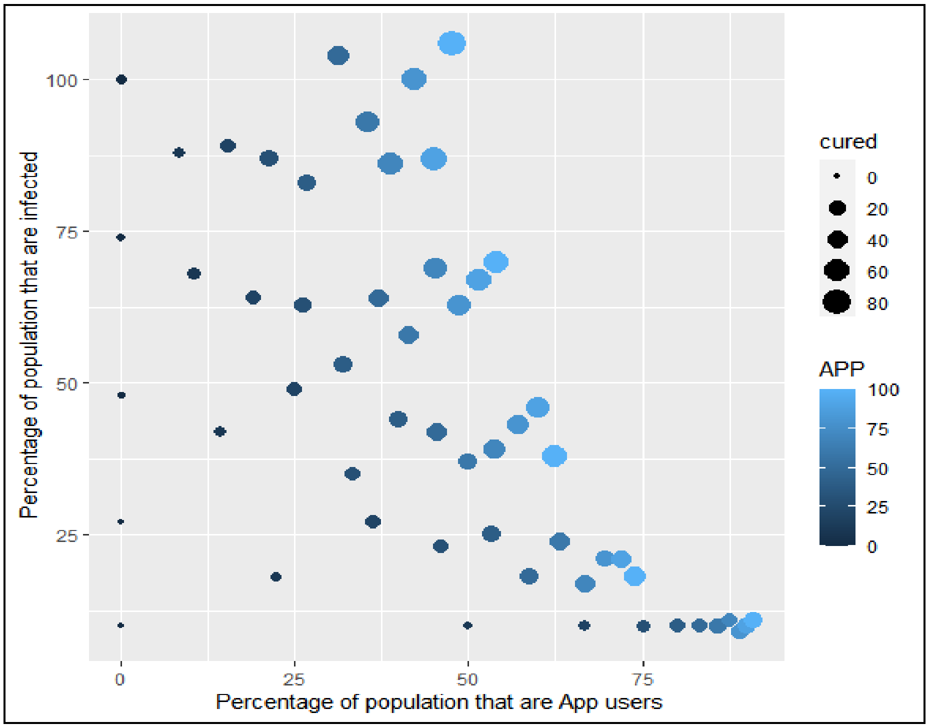


```
> head(MAS_FINAL_TABLE)
            V1               V2           V3                V4     V5
1 [run number] Initial-Infected Normal_Users Application_Users [step]
2            1               10            0                 0      0
3            2               10            0                10    500
4            3               10            0                20    500
5            4               10            0                30    500
6            5               10            0                40    500
             V6                             V7
1 count turtles count turtles with [strategy?]
2            10                              0
3            20                             10
4            30                             20
5            40                             30
6            50                             40
                              V8                          V9
1 count turtles with [infected?] count turtles with [cured?]
2                             10                           0
3                             10                           3
4                             10                           7
5                             10                          10
6                             10                          15
                            V10   V11
1 count turtles with [isolate?] ticks
2                             0     0
3                             8   500
4                             8   500
5                            10   500
6                             8   500
> tail(MAS_FINAL_TABLE)
   V1 V2  V3  V4  V5  V6  V7  V8 V9 V10 V11
51 50 10 100  50 500 160  50 104 48  84 500
52 51 10 100  60 500 170  60  93 49  80 500
53 52 10 100  70 500 180  70  86 63  69 500
54 53 10 100  80 500 190  80 100 67  73 500
55 54 10 100  90 500 200  90  87 70  74 500
56 55 10 100 100 500 210 100 106 86  83 500
```

Figure 3 (below) illustrates the relationship between the percentage of the population who are infected compared to the percentage of application users. The results of this experiment indicates that when the percentage of application agents is below 50% the infectious rate amongst people can rise to 100%. However the graph does show a downwards trend as the percentage of application users is above 50% as more users get the application, the less users get infected. In addition to this, if the percentage of users increases to above 75%, only a small percentage of users (less than 5%).

These results suggest that application users have to exceed 50% of the population for the application to be effective at halting the spread of the Coronavirus.

Figure 3: plot of infected to percentage of app users



Furthermore, the opposite effect can be observed when investigating normal users. As described in the graph below, The overall picture shows that as the percentage of normal users increases, the more the population becomes infected. In addition to this, we see that the results have a direct linear proportion. Once all agents are normal users, we observe that almost 100% of the population gets the Coronavirus. Thus suggesting the need for the contact tracing application to stop the spread of the Coronavirus.

Figure 4: plot of infected to percentage of non-app users

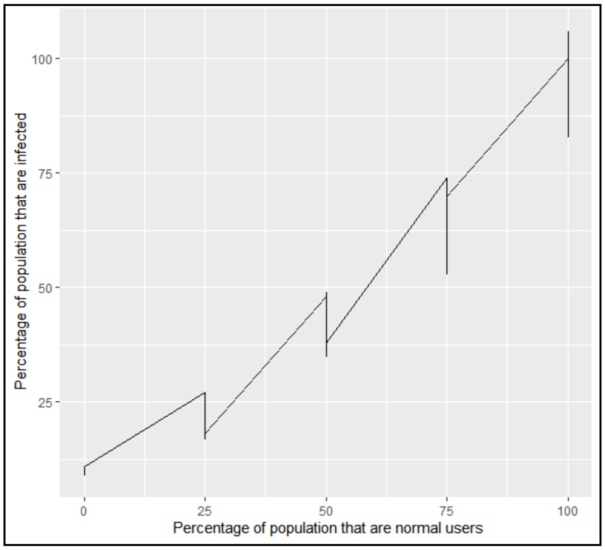
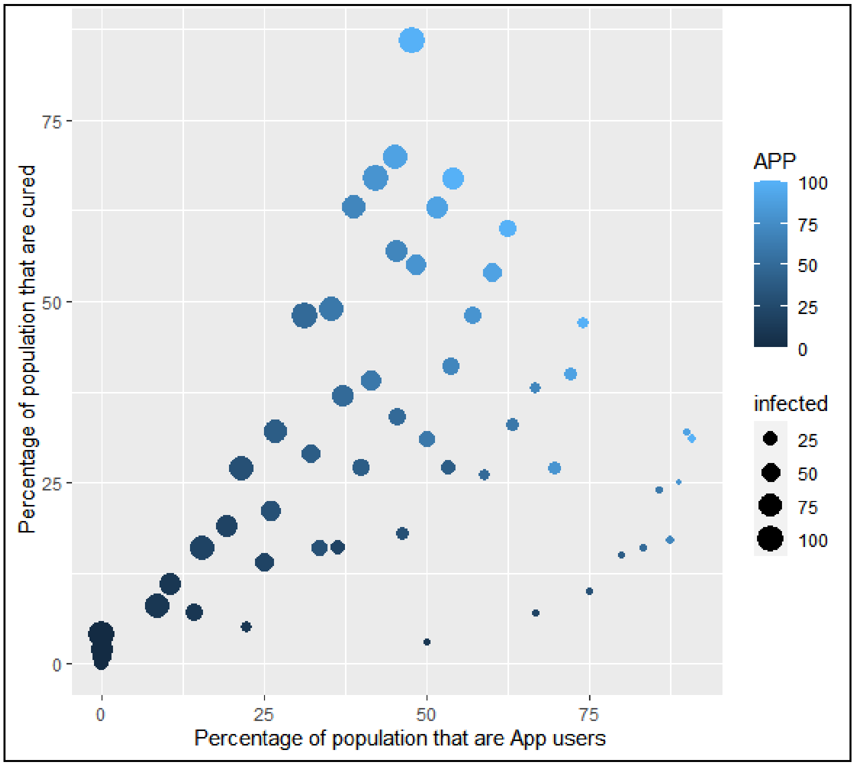


Figure 5: graph showing the relationship between the percentage of population that is cured and the percentage of population that are using the technology.



Conclusion

The experiment conducted showed results that currently match the empirical studies done on testing the effectiveness of a contact tracing application in a population by Oxford University. As the results suggested that if more than half the population use this application with the required social distancing interventions, it has the potential to stop the spread of the Coronavirus. Our results show that in a model environment of a few hundred agents with not so specific epidemiology configurations to use for slowing down the spread of the virus. The model shows that we could stop the spread of the Coronavirus and the epidemic if above 50 to 60 percent of the population utilize the application and the suggestions it gives.

Limitations:

Although our findings suggest that above 50% of the population need to install and use the application which does coincide with empirical evidence done recently about the contact tracing and tracking applications. Our model does have limitations. It is very difficult and in many cases impossible to simulate the complex system of the Coronavirus and how it spreads through an environment. Our model does well in describing the macroevolutionary effects of spreading the Coronavirus in a population that has effective contact tracing with the use of smartphone networking applications, however is limited in explaining the detailed spread of the Coronavirus with the use of the application at a micro-level. In addition to this, the model makes assumptions of the behaviour of the agents it represents as they are a simplistic model of human behaviour. Assumptions such as agents will always have the application operating all the time, which is relatively unknown for the human population as many people close the application, do not take their smartphones with them, and forget to turn on their GPS or bluetooth location sensor. These limitations do impact how realistic our findings are.

Bibliography

BBC News (2020, May 3) *Coronavirus: trial of mobile app to track infections - BBC News* [video]. YouTube. https://www.youtube.com/watch?v=f-AVfZTg6f0&t=141s

Bob Trenwith (2019, December 22) *Agent Based Modeling- 2.3.2 - Properties* [Video]. YouTube https://www.youtube.com/watch?v=znkQzn5FXf4

Complexity explorer (2019, March 7) *Agent Based Modeling: Patches and Links* [Video]. YouTube. https://www.youtube.com/watch?v=BtckMRbN9cw

Deane Rease (2014, November 16) *plots in Netlogo* [Video]. YouTube. https://www.youtube.com/watch?v=97GSQrUKW20

Complexity Explorer (2019, March 7) *Agent Based Modeling: Plotting the Data* [Video]. YouTube. https://www.youtube.com/watch?v=QbIvQNGkLKw

Gabriel Wurzer (2009, october 27) *NetLogo :: Lesson 16* [Video]. YouTube. https://www.youtube.com/watch?v=LYmA--eUfdE&t=294s

Guimera R, Uzzi B, Spiro R, Amaral L (2005) Team Assembly Mechanisms Determine Collaboration Network Structure and Team Performance, Science. 308(5722). P. 697-702. https://davidhales.name/abm-netlogo/lab-networks/netlogo-abm-networks-with-v2.pptx.pdf

Peter Brooks (2012, July 21) *Turtles and Patched and Links* [Video]. YouTube. https://www.youtube.com/watch?v=FWuI5siPJ2E

Stonedahl, F. and Wilensky, U. (2008). NetLogo Virus on a Network model. http://ccl.northwestern.edu/netlogo/models/VirusonaNetwork. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

The Verge (2020, april 16) *How your phone will track coronavirus* [Video]. YouTube. https://www.youtube.com/watch?v=b7dxCRm5t5g

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Yang, C. and Wilensky, U. (2011). NetLogo epiDEM Basic model. http://ccl.northwestern.edu/netlogo/models/epiDEMBasic. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

深度碎片 (2018, September 16) *06 how to plot an histogram and multiple lines* [Video]. YouTube. https://www.youtube.com/watch?v=bJW8g69znFQ

Appendix

Note: the highlighted code is referred from other models (cited in bibliography)

```
globals
[
  cumulative_infected   ;;The number of total infections, regardless of curity.
  r0                     ;; The reproduction rate of the agents.
]

turtles-own
[
  infected?              ;; If true, the agent is infectious.
  cured?                 ;; If true, the agent has been cured and cannot be infected again.
  strategy?              ;; If true, the agent is using the social-distancing technology.
  isolate?               ;; If true, the agent will self-quarantine
  unisolate?             ;;If true, the agent is unisolated.

  isolation-tick         ;; The agent will cure after 14 days of isolation.
  infection-length       ;; How long the person has been infected

  isolation-tendency     ;; Chance that Normal Users or Initial Infected agents will self-quarantine
during any hour being infected.

  number                 ;;Variable for randomizing the potential curing and unisolating.
]

;;setup the simulation.
to set up
  clear-all
  setup-people
  reset-ticks
end

;;creates 3 groups of agents: those using the strategy, those not using strategy, and those
initially infected.
to setup-people
  create-turtles Application_Users
  [set shape "person"
    set size 1
    set color blue
    setxy random-xcor random-ycor
    set cured? false
    set infected? false
    set strategy? true
    set isolate? false
    set unisolate? false
    create-link-with one-of other turtles
  ]

  create-turtles Normal_Users
  [set shape "person"
    set size 1
    set color white
    setxy random-xcor random-ycor
    set cured? false
    set infected? false
    set strategy? false
    set isolate? false
    set unisolate? false

    assign-tendency
  ]

  create-turtles Initial-Infected [
    setxy random-pxcor random-pycor
    set infected? true
    set cured? false
    set strategy? false
    set isolate? false
    set unisolate? false
    set  shape "person"
    set color red

    assign-tendency
  ]
end

;assign-tendency
to assign-tendency
  set isolation-tendency random-normal average-isolation-tendency
  average-isolation-tendency / 4

  if isolation-tendency > average-isolation-tendency * 2 [ set isolation-tendency
average-isolation-tendency * 2 ]
  if isolation-tendency < 0 [ set isolation-tendency 0 ]
end
```

```
;;assign color for infected and cured agents.
to assign-color  ;; turtle procedure
  if infected?
    [ set color red ]
  if cured?
    [ set color green ]
end

;;go procedure for agents
to go

  ;;limit set if all agents are infected.
  if all? turtles [infected?]
    [ stop ]

  ;;limit set if all agents are cured.
  if all? turtles [cured?]
    [ stop ]

  ;;limit set if no agents are infected (but not cured).
  if all? turtles [not infected?]
    [ stop ]

  ;;procedure for agents not isolated.
  ask turtles
  [if not isolate?
    [ move ]]

  ;;procedure for infected agents using strategy.
  ask turtles
    [if infected? and strategy?
      [isolate]]

  ;;procedure for infected agents using strategy.
  ask turtles
    [ if infected? and not isolate?
         [ infect ] ]

  ;;procedure for cured agents to move out of isolation and back into world.
  ask turtles
   [if cured?
      [move]]
  ;; procedure to allow Normal Users a chance to isolate.
  ask turtles
       [ if not isolate? and infected? and not strategy? and not cured? and (random 100 <
isolation-tendency)
        [ isolate ] ]

  ask turtles
  [if cured?
    [unisolate]]

  ;;procedure for infected agents.
  ask turtles
   [ let infected-neighbors count other turtles with [infected?] in-radius 0.2
    if infected-neighbors > 0 [
    set infected? true
    assign-color
     set infection-length 24 + ( random-normal infection-length 5.0 ) ;; mean of infection length and
standard-deviation multiplied by 24 hours
    set cumulative_infected cumulative_infected + 1
    ]
  ]
  tick
end

;;infect procedure
to infect
  set infected? true
end

;;isolate procedure.
to isolate
  set isolate? true
  set isolation-tick isolation-tick + 1
  move-to patch-here
  ask (patch-at 0 0) [ set pcolor white ]
  set number random 50
  if isolation-tick >= number [cure]
end

;;cure procedure
to cure
  set cured? true
  set infected? false
  set isolate? false
  set color green
  if cured? [unisolate]
end

;;unisolate the agent.
to unisolate
  set isolate? false
  set unisolate? true

  ;;return the patch color to black.
  ask (patch-here) [ set pcolor black ]
  if unisolate? [move]
end

;;move procedure
to move

  ;;restricting the entering of isolated areas.
  let front-patches patches in-cone 3 60
  if front-patches = white [set heading heading - 180]

  rt random-float 360
  fd 1
end

;;reporting percentage infected.
to-report %infected
  ifelse any? turtles
    [ report (count turtles with [infected?] / count turtles) * 100 ]
    [ report 0 ]
end
```