

Airplane Epidemiology

New Mexico
Supercomputing Challenge
Final Report

April 6, 2010

Team 119
Silver High School

Team Members

Forest Brown, Freshman

Teacher

Mrs. Peggy Larisch

Project Mentor

Dr. Camacho

Acknowledgements

The author wishes to acknowledge the following persons for their help and expertise in the development of the project, as well as the direction they have supplied.

- Mrs. Peggy Larisch – Teacher, Silver High School, Advanced Computer Studies
- Mr. Camacho – Teacher, Western New Mexico University, Microbiology
- Mr. Patrick Fischer – Student, Silver High School

The author gives a special thanks to Ms. Heidi Zimmerman for her advice in the creation of the project and her support through the finish.

Table of Contents

Executive Summary	E-1
Introduction	1
Project Proposal	2
Analytical Methodology	3
Results	6
Conclusions.....	10
References.....	11
Appendix A: Boeing 757	12
Appendix B: Charts	13
Appendix C: Graphs.....	14
Appendix D: Screenshot.....	16
Appendix E: Code	17

Executive Summary

The purpose of this project is to create a model of a Norwalk virus spreading within the confines of an airplane. Norwalk viruses are extremely infectious and are responsible for a majority of gastrointestinal infections around the globe. In order to model this, a program will be written that uses real-life variables and statistics to determine the number of people infected over a set flight time.

1.0 Introduction

1.1 Purpose

There have been very few studies performed to discover how a virus spreads on an airplane by physical contact. With air travel being the second most common method of transportation in the United States [1], it is important to know how disease travels inside an airplane. The goal of the project is to create a model of a virus spreading among the passengers in an airplane. This is to be done by creating a computer program that uses variables occurring in real life to model the virus on an airplane.

The virus to be modeled will be a Norwalk, or Norway, virus. This virus is commonly called Stomach Influenza, but that is a misconception. Norwalk viruses are not a form of influenza, and do not usually spread by airborne particles; they instead spread by the fecal-oral route. This happens when an infected person gets the virus on their body, the virus contacts someone else, and that person ingests the virus. Norwalk viruses are usually associated with large groups of people because that is where they spread most efficiently.

1.2 Scope

There are numerous variables the program could use, and the project could become very large. To keep the project manageable, only key variables will be included. Currently the program utilizes the variables most universal in airplanes, such as lavatory use and flight attendant activity. The program leaves highly erratic variables – for instance, level of passenger contact – unobserved. Unless passengers are in close physical interaction, the level of transmission due to contact can be deemed insignificant.

1.3 Computer Program

The program will be created using NetLogo. NetLogo allows various visual outputs to be utilized while still retaining mathematical integrity. Since the program is agent-based, NetLogo is the logical program of choice.

2.0 Proposal

This project develops a computer program that models the spread of a Norwalk virus within the confines of an airplane over a set time period. Fifty percent of all gastrointestinal outbreaks are due to Norwalk viruses [3], and that makes it important to know how the virus travels; to date, there have been no studies on Norwalk viruses spreading on an airplane. This project is therefore a novel undertaking. In order to correctly model the project, the program shall take several variables into account, such as lavatory use, contact between a flight attendant and passengers, probability of hand washing, and probability of infection. The computer will then generate a two dimensional model of a Boeing 757 [4] and, using a compressed time scale, output all passenger movement and the status of the virus. For the layout of a Boeing 757, refer to Appendix A.

3.0 Analytical Methodology

3.1 Mathematical Bases

Program Description

The program is a network of possible outcomes of scenarios randomly generated from input values. Therefore, the only math applicable is probability. For instance: a passenger is in the lavatory. If the lavatory is infected with the Norwalk virus, there is a possibility the passenger will become infected. If not, after returning to his or her seat, will the passenger be infected by the virus if transported by the flight attendant? And so on. Consequently, the results of tests with the same input values using the program will never be exactly the same. Only an average can be constructed.

In each scenario, there is only one passenger symptomatic of the virus; a full complement of passengers; and one flight attendant.

Statistics

A Norwalk Virus is extremely infectious [2] compared to other viruses. The probability of infection is dependent on the strain; therefore, the program allows a range from 1% to 7%. This is based on the minimum number of viral bodies required for a successful infection [3]. Since one viral particle has an infection probability from 0.1% to 0.7% and it requires ten to successfully infect a person, the program allows the user to set the probability of infection between 1% and 7%.

The program also assumes that some people on the airplane wash their hands. Although surveys indicate that approximately 80% of people wash their hands, they acknowledge the fact that many people lie about this. In order to incorporate this, the program sets 12% of the passengers as hand-washers. This may seem harsh, but with the amount of disease spread by people who don't wash hands (which is substantial), it was fit to reduce the number of people who wash hands to this amount.

3.2 Computer Applications

Input

The program takes these variables into account: the contact the symptom-showing passenger has with the rest of the plane; lavatory use; whether or not a passenger washes his or her hands; contact between passengers and the flight attendant; likeliness of contracting the virus; and flight time. It uses these variables to model the dynamics of the program.

The program iterates for a user-defined time period. This time period is scaled to approximate various flight times of the aircraft. NetLogo uses "ticks" to measure time, which is based on the computer's internal clock. This program is scaled so that 600 ticks are equal to one-hour of real time.

The computer cannot direct events towards an objective. From the beginning of the run-time of the model to the end, all variables are different for individuals in the model. As it runs, the program continually updates circumstances in the program.

Circumstances

- Infecting the Lavatory

If the lavatory is being used by the symptom-showing passenger, it is “infected” because it is assumed this particular passenger does not wash his or her hands. The project has only one such passenger.

As a side note, the virus travels on contact alone in the model, therefore eliminating the insignificant possibility of an airborne infection [3].

- Infection from the Lavatory

First, the program takes into account whether the lavatory is “infected”, or contains the virus. If this is true, the program then determines whether a person washes his or her hands. If the lavatory is not infected, however, the program does no further calculation of this instance.

If the person does not wash his or her hands, then the computer exercises the third variable: the likeliness of infection. This is achieved by creating a random number between 1 and 100; if it is within the range set by the user, the person is now infected. However, if the person washes his or her hands, then it runs the calculation twice; this lowers the chance of infection, but there is always a chance of infection when there is contact with the virus.

- Infection from the Flight Attendant

If the flight attendant carries the virus, all passengers he or she serves after he or she contacts the virus are at risk for infection. While the flight attendant is serving the passengers, if he or she is infected, the program monitors the passengers currently being served. For each passenger served, the program determines whether that passenger is infected or not based on the probability of infection. All people are contacted by the flight attendant on an hourly basis.

Once the flight attendant finishes serving the passengers, he or she “washes” his or her “hands”, thus removing the virus from himself or herself until he or she contacts the infected person again. In the model, the flight attendant can only receive the virus from the symptomatic passenger because, otherwise, undue complications would be included in the program.

- Lavatory in Use

Although seemingly obvious, the program must tell a passenger to use a different lavatory if the one the passenger is attempting to enter is occupied. The probability of

this happening is very low, but it was included in the program in order to keep the project accurate.

Output

The program is expected to output the following data: a graph depicting the total number of infected passengers; a monitor reporting the total number of infected passengers; and a two-dimensional representation of a Boeing 757 showing the status of all passengers and the progress of the virus while the program runs. This is the main reason the project was programmed in NetLogo; it has many methods of relaying data. In this project, NetLogo generates a running representation of the model while the program operates. While it is working, NetLogo outputs the current number of infected persons and creates a graph of this.

In order to simplify the program, some assumptions have been made: there is no take-off or landing delays of passenger activity; there is a relatively constant flow of traffic to or from the lavatory; and, lastly, all seats on the airplane are filled. All assumptions were made in order to simplify the program.

4.0 Results

4.1 Computer Calculations

The program performs well and its results are sound. Since the flight attendant checks the passengers once every hour, the number of infected people rises dramatically during these periods. Less constant is the infection rate among the people that used the lavatory and were “exposed” to the virus in that way; as such, the people infected in this manner were very few.

NetLogo proved somewhat limited for this project. At times, NetLogo would not execute certain commands. Syntax was checked and proved to be correct. A solution was reached by splitting commands into subroutines that executed independently. The subroutines were then simply called by the program when needed. Overall, this solution has proven to be effective.

The results given by the program appear to be quite sound. Although there are no published studies to directly compare to, output data is within the parameters set by the user and the program. This validates the program’s function. The numerical data fits expected trends, and there are no outlier points that may affect the overall accuracy of the program.

4.2 Explanation of Graphs and Charts

Several graphs and charts were created in order to accommodate the multiple variables in this model. Refer to Appendix 2 for all charts in this section, and Appendix 3 for all graphs.

Charts will be discussed first. The first item is Chart 1 (below). This chart depicts the relationship between the length of the flight and the number of people infected based on the likeliness of infection. The project resulted in a significant difference in the number of people infected due to the placement of the symptom-showing person as well as the likeliness of infection; consequently, this chart only relates the results obtained when this person is in first class.

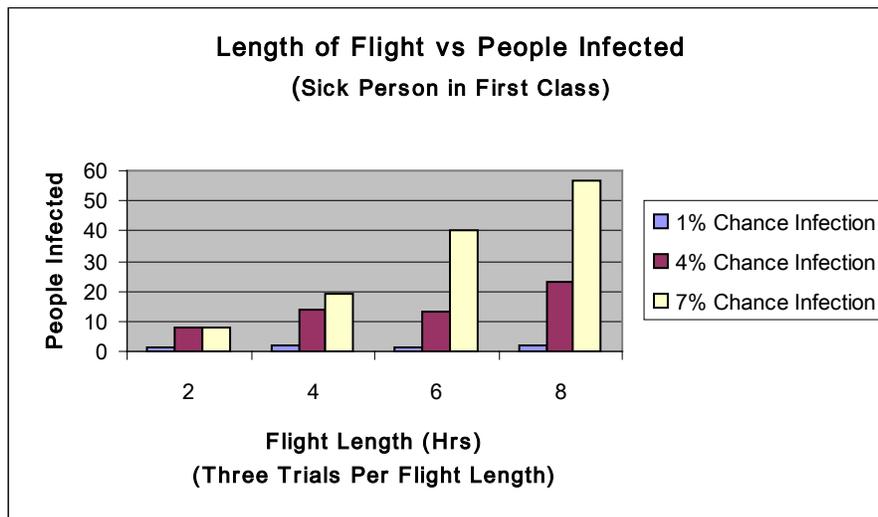


Figure 1: Chart 1

To explain: the flight attendant serves the first class first, and then serves the second class. After serving the second class, the attendant goes back to his or her station and washes his or her hands; therefore, if the flight attendant gets the virus on his or her “hands” in first class, second class is at greater risk for infection. On the other hand, if the flight attendant contacts the virus in the second class, the first class will not be at risk for infection.

Continuing: Chart 1 shows that, regardless of the length of the flight, the number of people infected when the likeliness of infection is 1% is insignificant. When the likeliness is 4%, the results are of more consequence, but there is no clearly defined rise in the number of infected people. At 7%, however, the chart shows an exponential increase in the number of people infected.

Chart 2 (below) displays the results obtained from the circumstances in Chart 1 *aside from the placement of the symptom-showing person*. In this case, the symptom-showing person is in second class; the overall number of people infected decreases. This is a direct result of the program’s routines concerning the flight attendant. As in Chart 1, when the infection chance is 1%, the results are insignificant, but now a 4% chance is similar. A 7% chance of infection retains its importance, but it loses the steady climb seen in Chart 1.

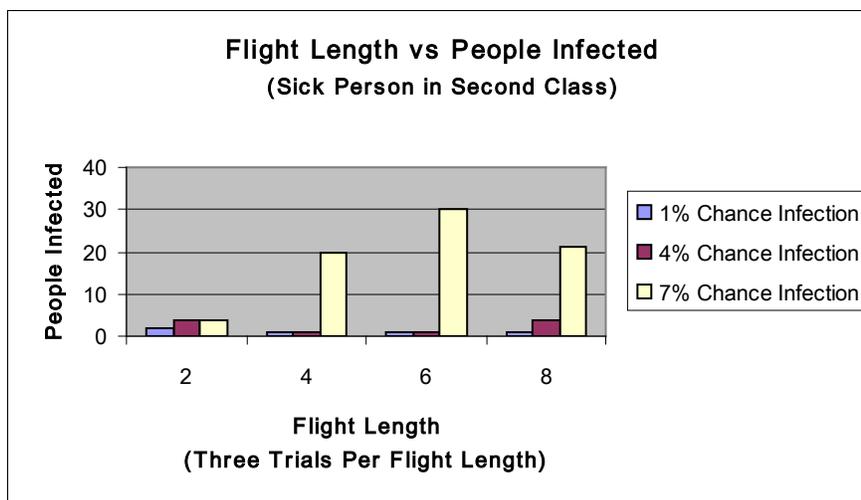


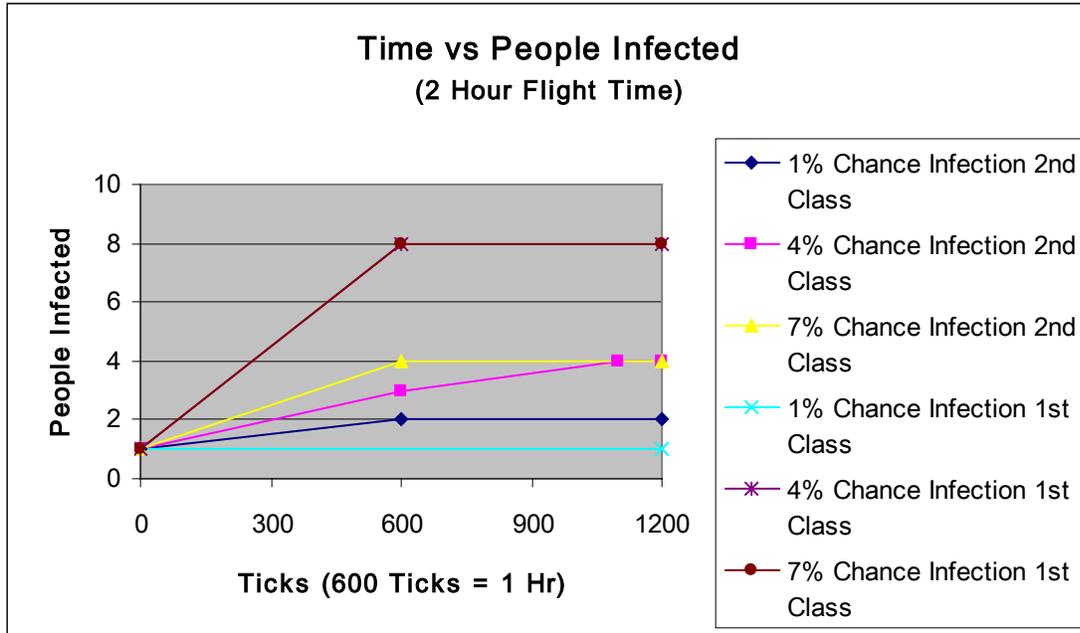
Figure 2: Chart 2

The graphs will be discussed next. All four depict the relationship between the passage of time and the number of people infected. Each graph illustrates a different flight time, but they all show the time of infections during the flight. The results in Graph A are unremarkable, as are the outcomes in Graph B (figure 3).

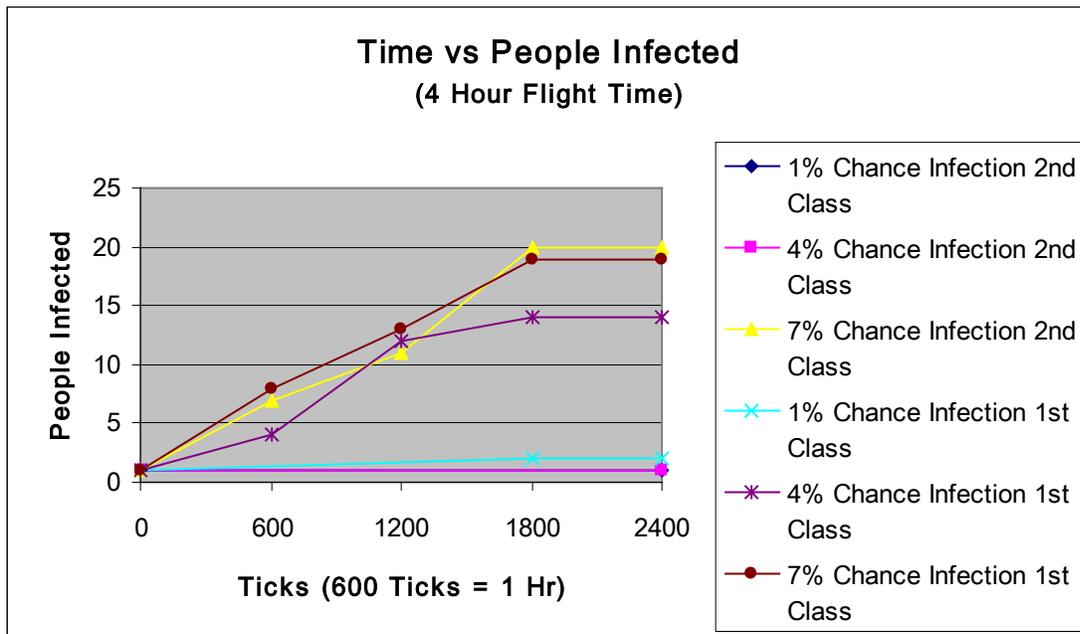
There are no patterns to indicate a bug in the program, and the number of infected people changes in correlation to the length of the flight and the probability of infection. Graph C, however, depicts something different (figure 4). The trend created with the results from a 7% chance of infection and the sick person in first class indicates a pattern, perhaps produced from the program. There is an obvious dip in the trend at exactly half of the test time period. After this anomaly, the trend seems to double the values it received from the first half of the test period.

Graph D has a similar phenomenon, with the same variables. At half the test time period, there is a lull, and then a doubling of the values. All other trend lines on graphs C and D appear have no recognizable patterns such as this.

Figure 3: Graphs A and B

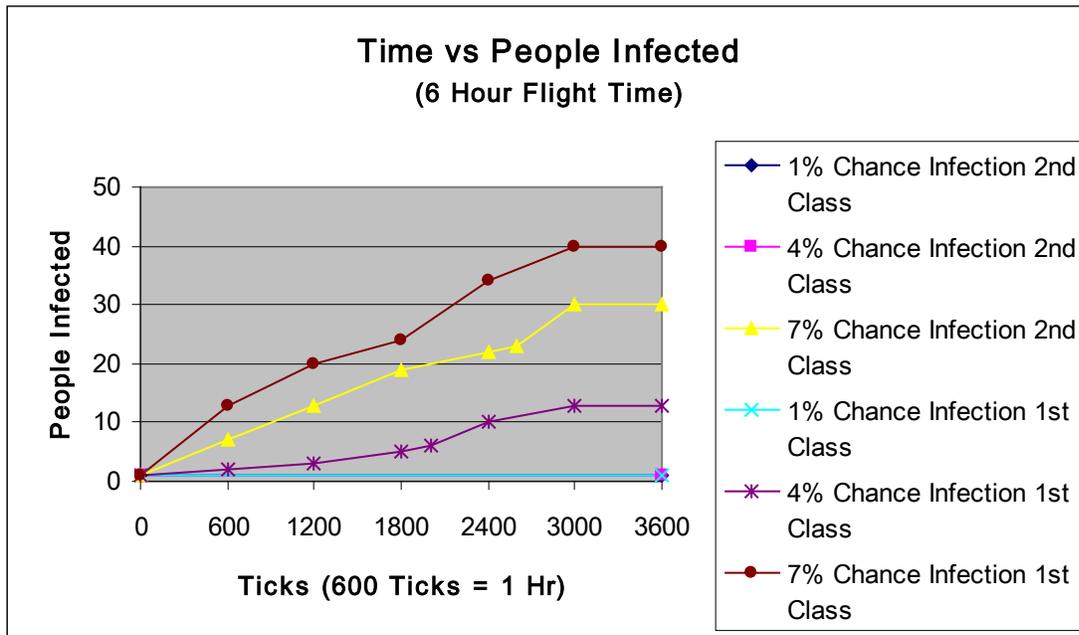


Graph A

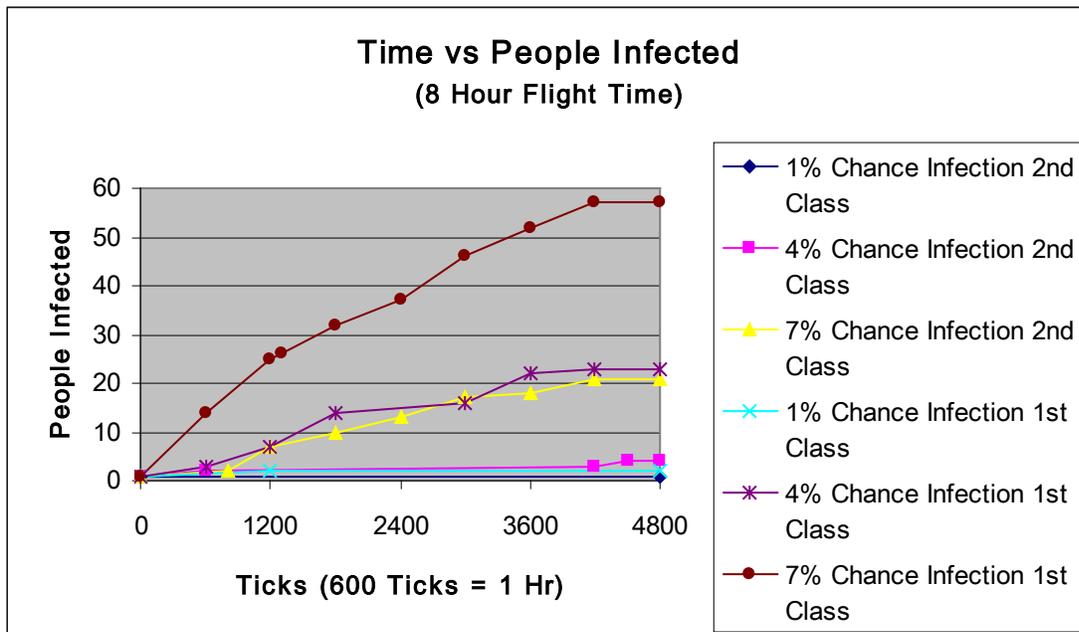


Graph B

Figure 4: Graphs C and D



Graph C



Graph D

5.0 Conclusions

5.1 Results

Computer Program

The program satisfies the project proposal. It successfully models simple interaction on an airplane and the spread of a Norwalk virus. The program can also adapt to several different strains of Norwalk Virus, which is done by giving the user control over the probability of infection.

The results indicate that the most likely manner of infection on an airplane is through contact with the flight attendant. This can be explained by the fact that the flight attendant physically interacts every passenger on the plane. On an individual basis, the likeliness of contracting the virus by this method is the same as by the lavatory, but this can only be taken at face value for two reasons. The first is that not all passengers use the lavatory, and those that do use it on a random basis. This is unlike the contact all passengers have with the flight attendant. The second reason is the passengers do not use the lavatory on a set time scale. The flight attendant contacts everyone on an hourly basis.

The performance of the program is very good. The program produces results that are reasonable based on the settings on which it runs. The program also displays a 2-dimensional model of a Boeing 757. See Appendix D for a screen shot of the program.

Chart and Graph Indications

The graphs and charts indicate that the number of infected people at the end of a flight is directly related to a longer flight time and/or a higher chance of infection. The difference, however, between the significance of the numbers produced by a higher infection chance rather than a longer flight time show that, no matter how long the exposure to the virus, the infection chance determines the number of people infected.

The anomalies within graphs C and D could be a result of two things. One, the program has bugs in the program's routines insuring randomness of the model; or two, that, the program accurately models the virus and these patterns emerge naturally. The latter of the two is more probable based on the models criteria.

5.2 Recommendations

The only change to the program could be the extent of the program itself. After encountering NetLogo's limitations, several variables had to be removed, such as the fact that a flight attendant does not contact everyone on an airplane in real life.

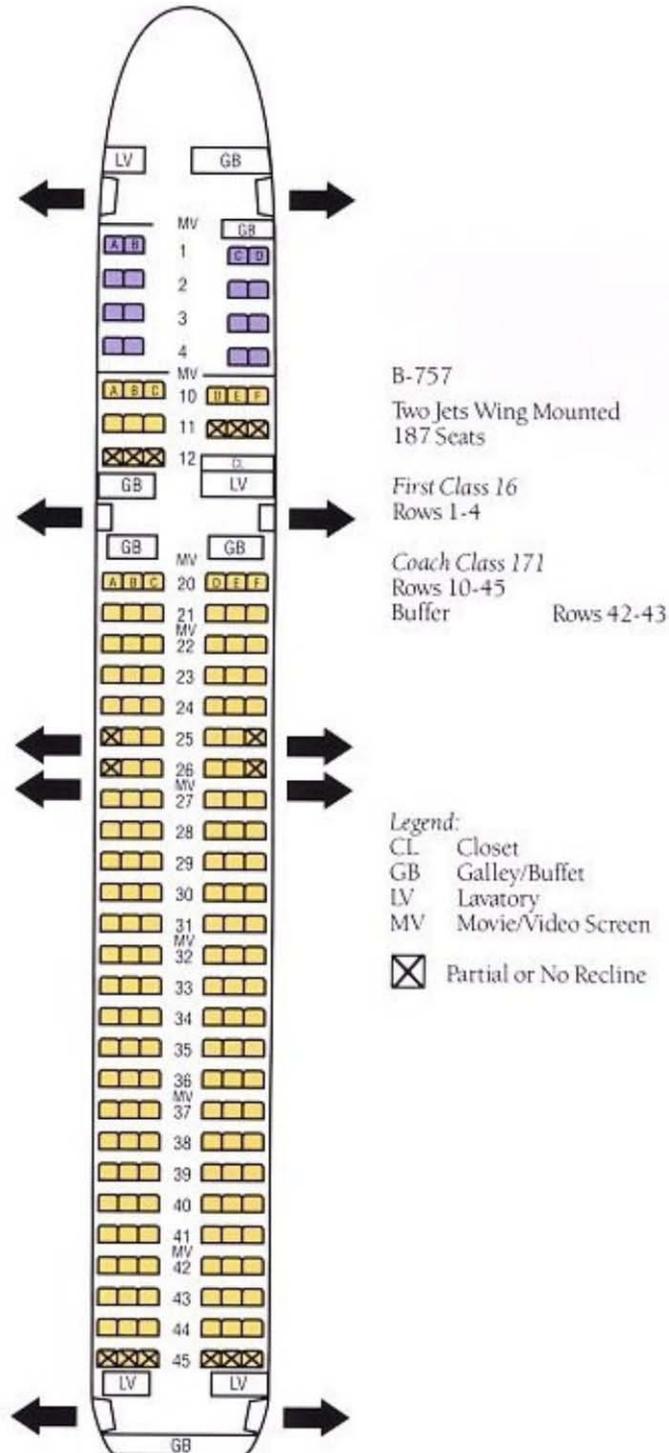
In the future, the program could improve the scope of the applicability. Currently, the results can only be used for a Boeing 757. It is reasonable to assume that other airplanes would have different results, because of total passenger capacity, number of lavatories, flight range, and etcetera.

References

1. "Mode of Transport." Wikipedia. March 19 2010. Accessed March 29 2010. <http://en.wikipedia.org/wiki/Mode_of_transport>
2. "Norwalk Virus: how infectious is it?" Pubmed. 2008. Accessed March 30 2010. <<http://www.ncbi.nlm.nih.gov/pubmed/18551613>>
3. "Norovirus: Technical Fact Sheet." Center for Disease Control. February 23 2010. Accessed February 25 2010. <<http://www.cdc.gov/ncidod/dvrd/revb/gastro/norovirus-factsheet.htm>>
4. "Airline Seating Charts." Aviation Explorer. December 3 2003. Accessed January 25 2010. <www.aviationexplorer.com/aircraft_airline_seating_charts.html>

Appendix A Boeing 757

This is the basic layout of a Boeing 757 used to create the model. Since different styles of 757s have been created, there are actually several different passenger capacities. To allow for this, the model has forty more seats than this layout.



Appendix B Charts

Chart 1:

Chart 1 displays the results from twelve scenarios tested in the program. It shows the relationship between the length of the flight and the people infected. For each flight time tested, three different chances of infection were tested as well. This chart only shows the results of tests while the infected passenger was in first class.

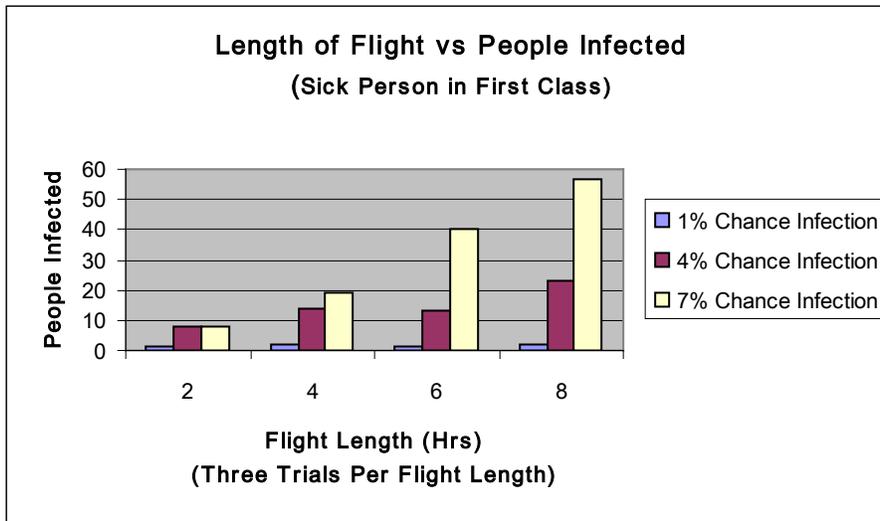
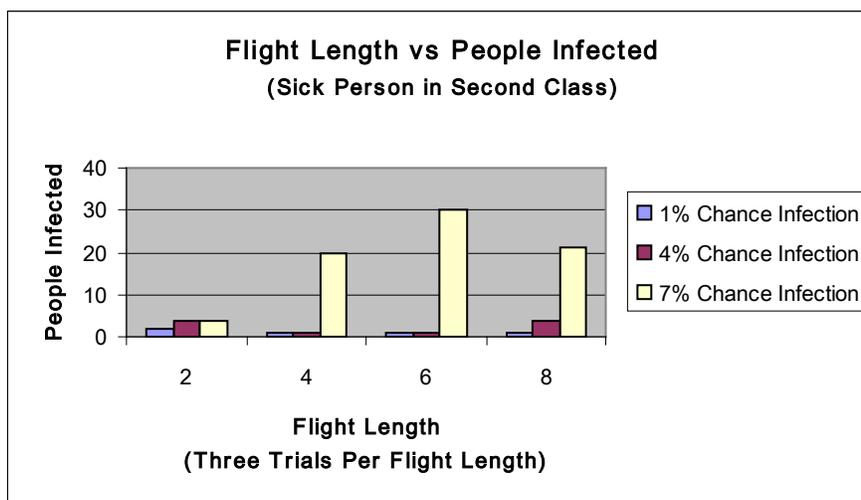


Chart 2:

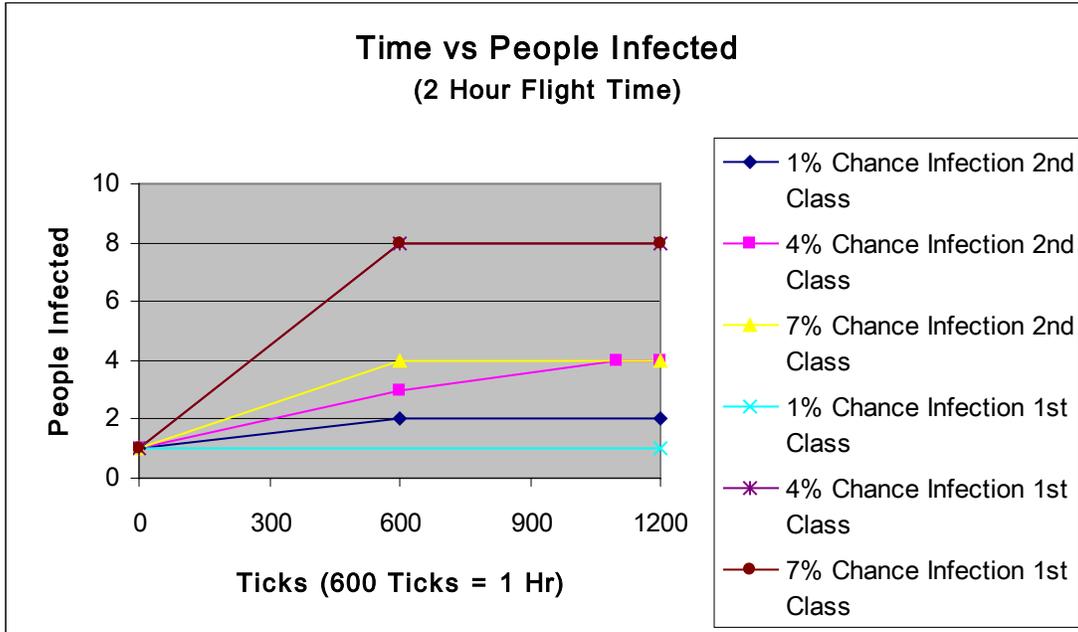
Chart 2 shows the same scenarios, but the infected passenger is in second class. Notice the difference in the number of infected people between Chart 1 and 2. Chart 2 verifies that when the infection chance is 1%, the amount of infected people is insignificant.



Appendix C Graphs

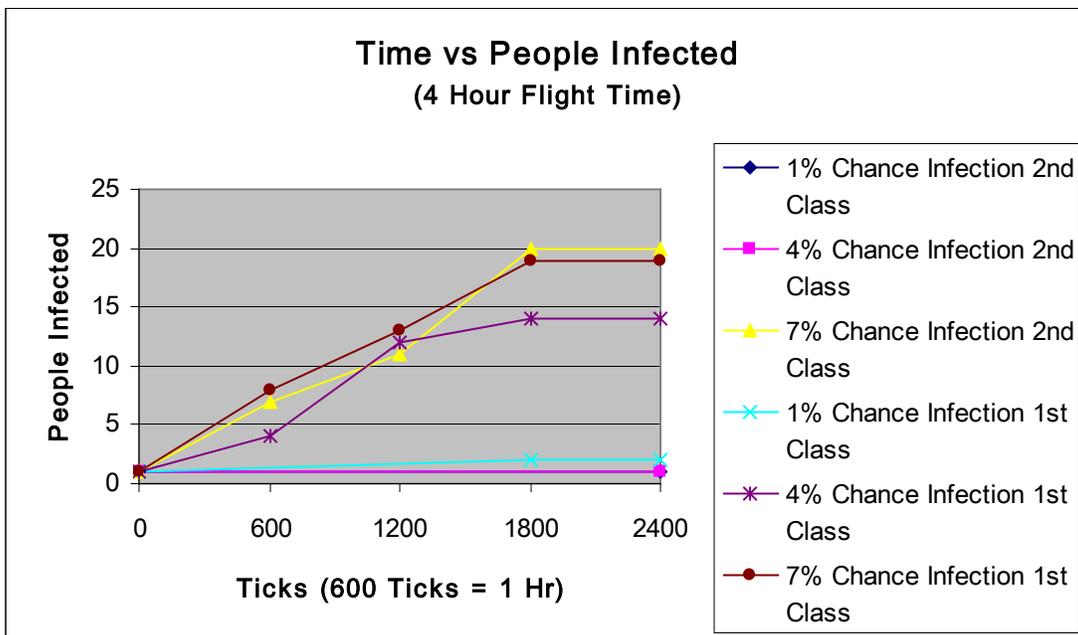
Graph A

Graph A is unremarkable in that the trend lines are to be expected. It does, however, confirm the program is working how it is supposed to. With a higher infection chance, there should be a higher number of people infected.



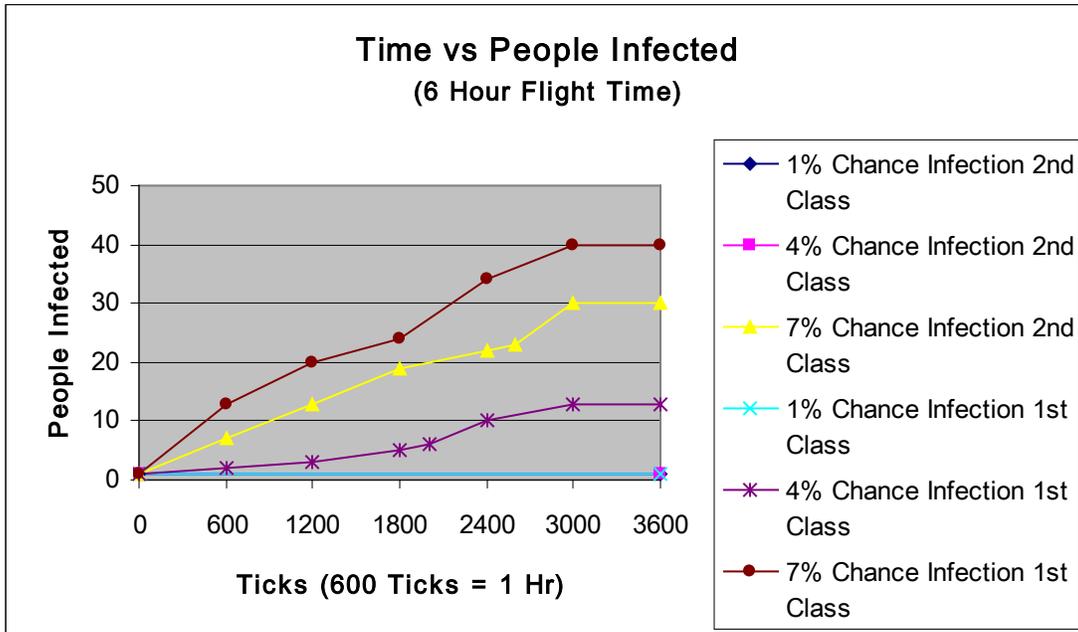
Graph B

Graph B indicates the same thing as Graph A. The trend lines are as expected, and there are no remarkable patterns in them.



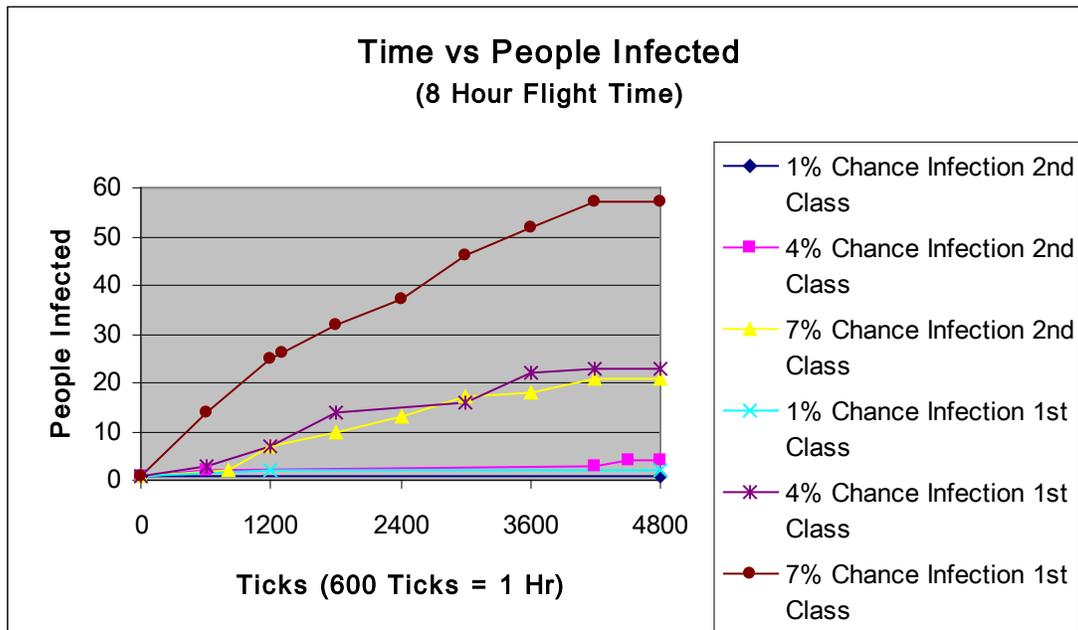
Graph C

Graph C has an anomaly. The trend created from a 7% chance of infection has a dip at half the time period. This may be created by a bug in the program, or it occurs naturally.



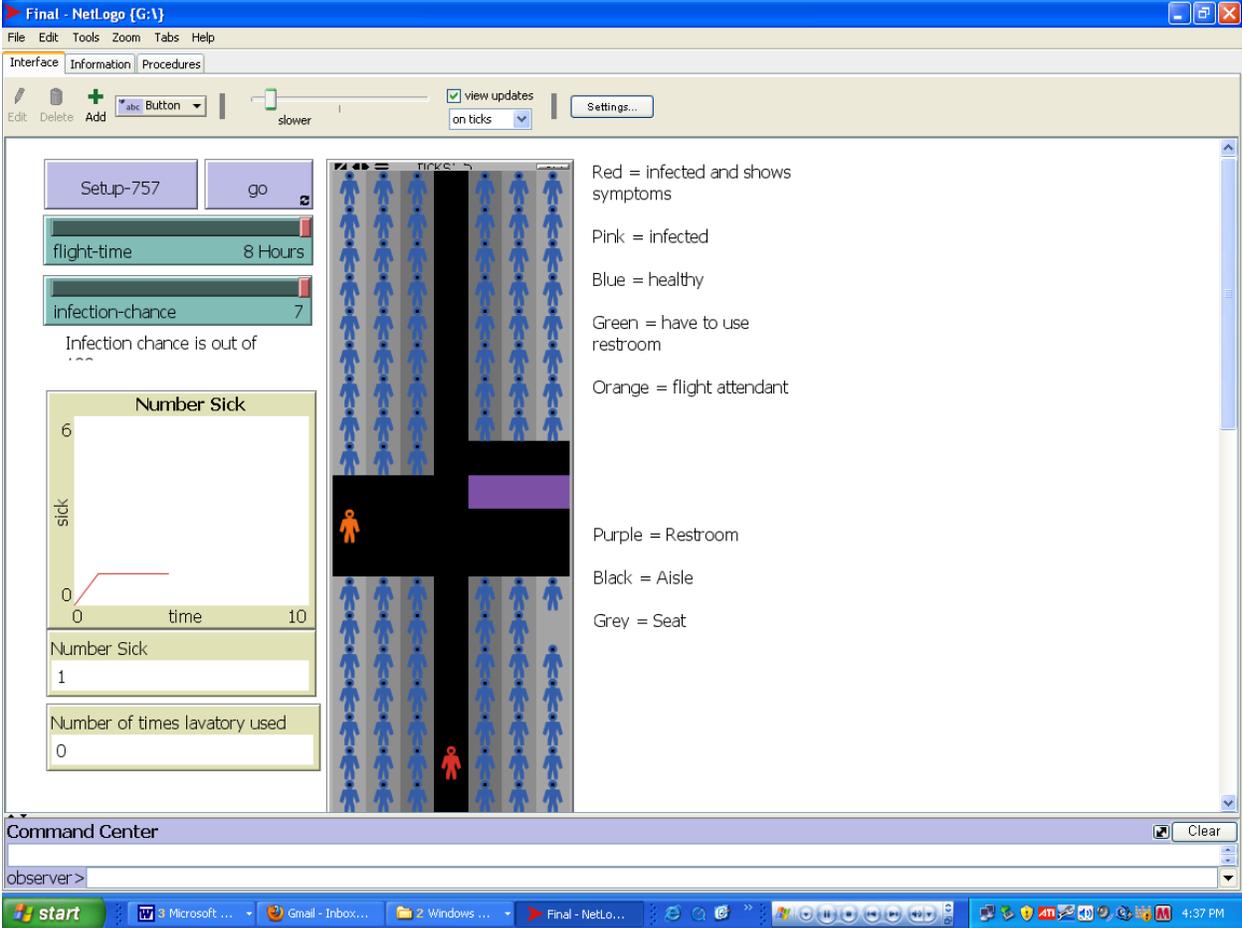
Graph D

Once again, there is a dip in the trend with a 7% chance of infection at half the time period. Note that there are no other anomalies such as this in any other trend line.



Appendix D Screenshot

This is a screenshot of the model. It is too big to fit into one view, so this is only the front half of the plane in the model.



Appendix E Code

```
.....  
.....  
;; Declarations ;;  
.....  
.....  
  
globals  
[  
  num-sick  
  run-number  
  delay  
  
  flight-attendant  
  carrying-virus  
  current-ycor  
  serving?  
  to-station  
  
  lav-use      ;;output the number of times lavatory used  
  inlav  
  
  Lav1        ;;first class lavatory  
  Lav1occupied? ;;is it occupied?  
  Lav1infected? ;;is it infected?  
  
  lav2        ;;second class lavatory  
  Lav2occupied? ;;is it occupied?  
  Lav2infected? ;;is it infected?  
  
  lav3  
  lav3occupied?  
  lav3infected?  
  
  first-class ;;turtles in first class  
  
  a  
  b  
  c  
  d  
  o  
  
  x          ;;variable used to count down time of flight left  
  y          ;;variable used to count number of times the bathroom has been used  
  z          ;;variable used to set turtles that wash hands  
  
  original   ;;turtle(s) at the beginning of the model that show symptoms  
]  
  
turtles-own
```

```
[
  seat-ycor
  seat-xcor
  infected?
  already-gone
  going?
  coming?
  washer?
]
```

```
.....
;; Setup Functions ;;
.....
```

```
to setup-757
  clear-all
  set run-number 1
```

```
.....
;; set up seats for Boeing 757;;
.....
```

```
ask patches with [pxcor = 1 or pxcor = -1]
[set pcolor 4]
```

```
ask patches with [pxcor = 2 or pxcor = -2]
[set pcolor 5]
```

```
ask patches with [pxcor = 3 or pxcor = -3]
[set pcolor 6]
```

```
.....
;; set up restrooms and spaces;;
.....
```

```
ask patches with [pycor = min-pycor]
[set pcolor black]
```

```
ask patches with [pycor = min-pycor]
[set pcolor violet]
```

```
ask patches with [pycor = 13 and pxcor > 0]
[set pcolor violet]
```

```
ask patches with [pycor = 12]
[set pcolor black]
```

```
ask patches with [pycor = 11]
[set pcolor black]
```

```
ask patches with [pycor = 14 and pxcor > 0]
```

```

[set pcolor black]

ask patches with [pycor = 13 and pxcor < 0]
[set pcolor black]

;; set up end of plane
ask patches with [pycor = min-pycor and pxcor = -2]
[set pcolor 75]

ask patches with [pycor = min-pycor and pxcor = -3]
[set pcolor 75]

ask patches with [pycor = min-pycor and pxcor = 2]
[set pcolor 75]

ask patches with [pycor = min-pycor and pxcor = 3]
[set pcolor 75]

ask patches with [pycor = min-pycor and pxcor = 0]
[set pcolor black]

.....
;; create passengers;;
.....

ask patches with [pcolor = 4 or pcolor = 5 or pcolor = 6]
[sprout 1]

set-default-shape turtles "android"

;; -3 12 : patch where flight attendant will start

setup-world
end

to setup-world
set a 1
set b 1
set c 1
set d 1
set o 1
set x 0
set y 0
set z 0
set num-sick 0
set delay 0

set Lav1 patch 2 13
set Lav1occupied? false
set Lav1infected? false

```

```

set Lav2 patch 1 -22
set Lav2occupied? false
set Lav2infected? false

set lav3 patch -1 -22
set lav3occupied? false
set lav3infected? false

setup-plot
do-plot
ask turtles [
  set color blue
  set heading 0
  set infected? false
  set already-gone false
  set going? false
  set coming? false
  set flight-attendant turtles with [color = orange]
  set serving? false
]

ask one-of turtles [
  set infected? true
  set color red
]

ask patch -3 12 [
  sprout 1
  ask turtles-here [
    set color orange
    set serving? false
  ]
]

set original turtles with [color = red]
set flight-attendant turtles with [color = orange]

ask turtles [
  set washer? false
  set seat-xcor xcor
  set seat-ycor ycor
  set inlav false ]

while [z < 30] [
  ask one-of turtles with [washer? = false] [
    set washer? true
  ]
  set z z + 1
]

ask original [set washer? false]

```

end

```
.....  
;; Runtime Functions ;;  
.....
```

to go

```
;;limit flight time
```

```
while [x < (flight-time * 600)] [
```

```
  every 0.1
```

```
  [  
    set num-sick count turtles with [ infected? = true ]  
    set lav-use y  
    tick  
    do-plot  
    set x x + 1  
  ]
```

```
if ticks = a * 175 or ticks = 1[
```

```
  ask original [  
    set xcor 0  
    set heading 180  
    set going? true  
    set a a + 1 ] ]
```

```
if b * 600 = ticks or ticks = 1 [
```

```
  ask turtles with [seat-xcor = patch-here] [  
    set already-gone false  
  ]  
  set b b + 1  
]
```

```
if o * 600 = ticks [
```

```
  move-flight-attendant  
  set o o + 1 ]
```

```
serve-1-class
```

```
ask-passengers
```

```
station
```

```
.....  
;;call bathroom subroutines;;  
.....
```

```
have-to-go
```

```
going-to
```

```
checking-restroom-1
```

```
checking-restroom-2
```

```

go-to-seat
if c * 10 = ticks [
  if Lav1occupied? = true [set y y + 1]
  if Lav2occupied? = true [set y y + 1]
  if lav3occupied? = true [set y y + 1]
  ask turtles-on patch 2 13 [leave-rest-1]
  ask turtles-on patch 1 -22 [leave-rest-2]
  ask turtles-on patch -1 -22 [leave-rest-3]
  set c c + 1
]
]
if x = (flight-time * 600) [stop]
end

;; set the appropriate variables to make this turtle sick
to get-sick ;; turtle procedure
  if (infected? = false) and (random 100 < infection-chance) [
    if washer? = false and ycor != current-ycor [
      set infected? true
      if color = green [set color pink]
      if color = blue [set color pink ] ]
    if washer? = true [get-sick-2]
    if ycor = current-ycor [get-sick-2]
  ]
end

to get-sick-2
  if (infected? = false) and (random 100 < infection-chance) [
    set infected? true
    if color = green [set color pink]
    if color = blue [set color pink ] ]
end

.....
;;bathroom subroutines;;
.....

to have-to-go
  if d * 100 = ticks or ticks = 1 [
    ask one-of turtles with [color = blue and already-gone = false ] [
      set color green
      set xcor 0
      set heading 180
      set already-gone true
      set coming? false
      set going? true
    ]
    set d d + 1
  ]
end

```

```

to going-to
  ask turtles with [going? = true] [
    if ycor != 13 [
      if ycor != -22 [
        fd 1
      ]
    ]
  ]
end

```

```

to checking-restroom-1
  ask turtles with [going? = true] [
    if ycor = 13 [
      if Lav1occupied? = false [
        restroom-1
      ]
    ]
  ]
end

```

```

to checking-restroom-2
  ask turtles with [going? = true] [
    if ycor = -22 and xcor = 0 [
      if Lav2occupied? = true [
        restroom-3
      ]
      if Lav2occupied? = false [
        restroom-2
      ]
    ]
  ]
end

```

```

to check-restroom-3
end

```

```

to restroom-1
  set xcor 2
  set Lav1occupied? true
  Lav1infect
end

```

```

to restroom-2
  set xcor 1
  set Lav2occupied? true
  Lav2infect
end

```

```

to restroom-3
  if Lav3occupied? = false [
    set xcor -1
  ]
end

```

```
    set lav3occupied? true
    lav3infect ]
end
```

```
to leave-rest-1
  if infected? = false [
    set color blue ]
  set going? false
  set xcor 0
  set heading 0
  set Lav1occupied? false
  set coming? true
end
```

```
to leave-rest-2
  if infected? = false [
    set color blue ]
  set going? false
  set ycor -22
  set xcor 0
  set heading 0
  set Lav2occupied? false
  set coming? true
end
```

```
to leave-rest-3
  if infected? = false [
    set color blue ]
  set going? false
  set ycor -22
  set xcor 0
  set heading 0
  set lav3occupied? false
  set coming? true
end
```

```
to go-to-seat
  ask turtles with [coming? = true] [
    if ycor != seat-ycor [
      fd 1]
    if ycor = seat-ycor [
      set xcor seat-xcor
      set coming? false ]
  ]
end
```

;;spread infection to either the bathroom or another person

```
to Lav1infect
  if Lav1infected? = false [
    if washer? = false [
```

```

        if infected? = true [
            set Lav1infected? true ] ;;infect bathroom
    ]
]

if Lav1infected? = true [
    if washer? = false [
        get-sick ] ;;infect person
    ]
end

to Lav2infect
    if Lav2infected? = false [
        if washer? = false [
            if infected? = true [
                set Lav2infected? true ] ;;infect bathroom
            ]
        ]
    ]

    if Lav2infected? = true [
        if washer? = false [
            get-sick ] ;;infect person
        ]
    end

to lav3infect
    if lav3infected? = false [
        if washer? = false [
            if infected? = true [
                set lav3infected? true ] ;;infect bathroom
            ]
        ]
    ]

    if lav3infected? = true [
        if washer? = false [
            get-sick ] ;;infect person
        ]
    end

.....
;;flight attendant(s);
.....

to move-flight-attendant
    ask flight-attendant [
        set xcor 0
        set heading 0
        set serving? true
    ]
end

```

```

to serve-1-class
  ask flight-attendant [
    if serving? = true [
      fd 1 ]
    if ycor = max-pycor [
      set heading 180 ]
    if ycor = min-pycor [
      ask flight-attendant [
        set serving? false
        set to-station true ] ] ]
end

```

```

to ask-passengers
  ask flight-attendant [set current-ycor ycor]
  ask turtles with [ycor = current-ycor and xcor != 0] [
    be-served ]
end

```

```

to be-served
  if infected? = true [
    ask flight-attendant [
      set carrying-virus true ] ]
  ask flight-attendant [
    if carrying-virus = true [
      if serving? = true [
        ask turtles with [ycor = current-ycor] [
          get-sick
        ]
      ]
    ]
  ]
end

```

```

to station
  ask flight-attendant [
    if serving? = false [
      if to-station = true [
        if seat-ycor != ycor [
          set heading 0
          fd 1 ] ]
      if seat-ycor = ycor [
        set xcor seat-xcor
        set to-station false
        set carrying-virus false ]
    ]
  ]
end

```

```

.....
;;; Plotting Procedures ;;
.....

```

```
to setup-plot
  create-temporary-plot-pen word "run " run-number
  set-plot-pen-color item (run-number mod 5)
    [blue red green orange violet]
end

to do-plot
  plot num-sick
end
```