The Flu in Schools

New Mexico
Supercomputing Challenge
Final Report
March 28, 2009

Team 60
Los Alamos Middle School

Team Members
George Barnum
Eric Begnaud
Mohit Dubey
Ben Liu

Teacher
Mr. Bob Dryja

Project Mentor
Dr. Michael Begnaud

Table of Contents

Table of Figures

Execute Summary

The flu is a seasonal infection caused by the Orthomyxoviridae virus. There are between 25 to 50 million cases of influenza in the United States alone, leading to 150,000 hospitalizations and from 30,000 to 40,000 deaths yearly. We chose to model the spread of the flu virus in schools where classrooms can cause an isolation or concentration of the infection. Our agent-based simulation uses students, germs, and contaminated surfaces to model the infection of the flu in a school. Students become infected and contaminate surfaces, then become sick and sneeze, spreading germs to other students, and becoming immune to the flu strain for that year. Initial modeling results tell us that the sneeze rate during the simulation is what affects the infection the most. After changing the sneeze rate from 1-20%, we found that the peak number of infected students changes the most as well as the number of students never infected. The time until no more students are infected and the day the peak infection occurs does not change much with sneeze rate.

Introduction

We studied the flu for our program. We did this because we have an annual flu season, and we wanted to model the spread of the flu in a school environment. The school environment is for testing how a standard flu season occurs in a close-quarters environment. Our goal is to model how the flu spreads, so we are modeling parameters based on infection. This is modeling one strain of the virus over the course of a year, assuming that the virus does not mutate into a new strain over the course of the season. Also, we wanted to simulate the contamination and sneezes accurately. We observed that the variable that affects the number of infected students the most is the time that the contamination stays on a surface.

Flu Facts

As we were working on our program, we learned several things about the flu (influenza)[1]. We learned that the flu is caused by the Orthomyxoviridae virus. Also, we learned that there are between 25 to 50 million cases of influenza in the United States alone, leading to 150,000 hospitalizations and from 30,000 to 40,000 deaths yearly. We also found out what symptoms the flu has, which are: swelling of the throat, headache, muscle pain, coughing, weakness, general discomfort, and possible death. Another fact is that the flu is spread by touching anything with the flu germs and then touching your eyes mouth, or nose. The last thing that our team learned is that vaccinations work about 65% of the time for the elderly and about 30% of the time for people with diabetes and heart disease.

Project Setup

There were many steps in setting up our project. To start, we set up a school-like area, completely open without any interior boundaries.

---

[1] *http://www.flufacts.com*

In Net logo there are ticks, which represent one loop. In our program 1 tick represents 5 minutes.

The students we set up could have many different behaviors. They could be infected, in which the student has caught the flu but is not sneezing. Then they become sick in which the students can sneeze and infect other students. After they are sick, the students become immune so they can no longer become sick.

In our simulation we set up boundaries that acted like classrooms in a school. There were six rooms in our program. We also set up students that could become sick with the flu and a nurse that could send sick kids home (Figure 1). We programmed our setup so that sneeze germs and students do not pass through the walls of the boundaries.

Next, we added contaminations and sneeze germs. Contaminations are left behind when a student moves and are immobile. Sneezes are projectiles that die immediately after they are shot out (they are shot out and can move forward in ten positions). If students come in contact with either of these germs then the student will become infected. We could also change the percent of the time that students sneeze in each interval and also change the percent chance of the contaminations to die.

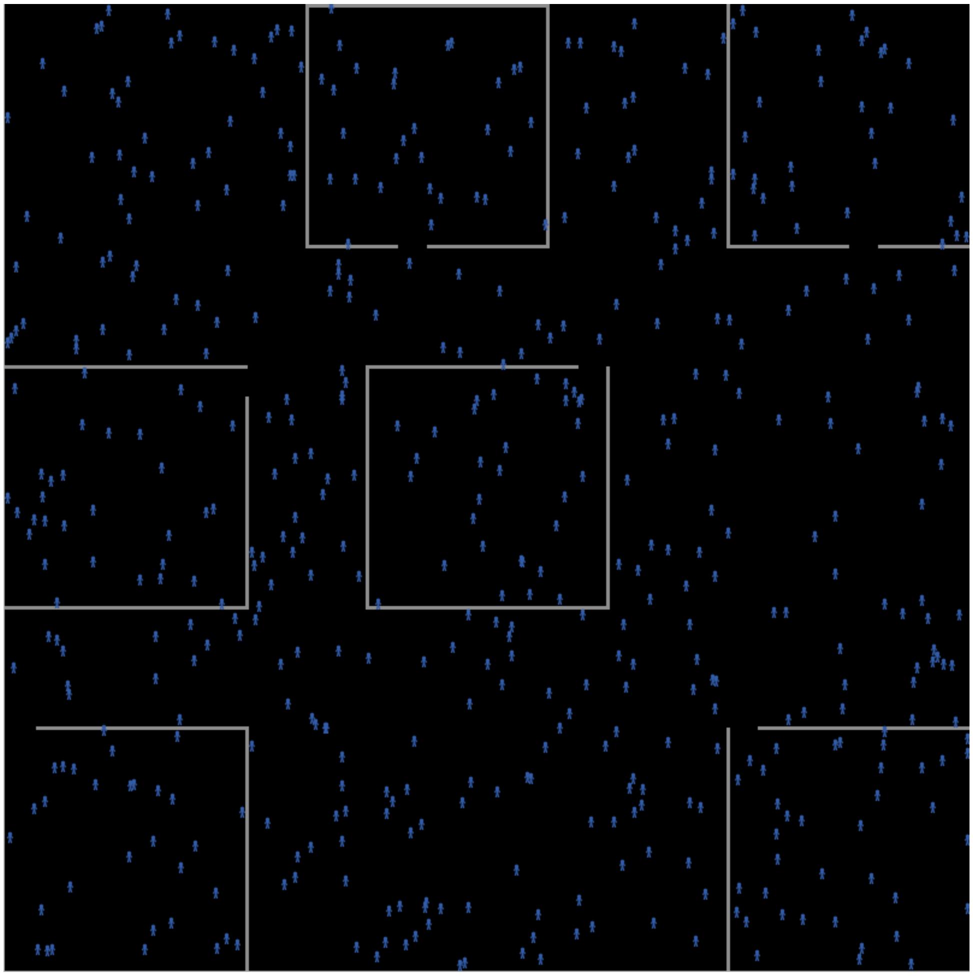

Figure 1. Plot of the Netlogo view area with classrooms (gray patch areas) and students (blue).

There were many parameters in our program that we could change. One is the effectiveness of the nurse. This number was the percent chance that would change the chance that a student would be sent home by the nurse. Another is the sneeze percentage, which is the chance that the student will sneeze in a certain time.

Lastly, we added a nurse to our program. The nurse was not an agent but a behavior of the system. If someone is sick, there was a chance that the nurse would send him home (which immediately makes the student immune).

Our program was an almost perfect simulation of the flu in a school. There were separate classrooms in which students could enter. Also, there were students that could become sick and get other students sick. Plus, the sick students could sneeze and create contaminations just as in real life. Finally, there was a nurse who could send sick students home. Using this simulation, we could simulate how long the flu lasts in a school.

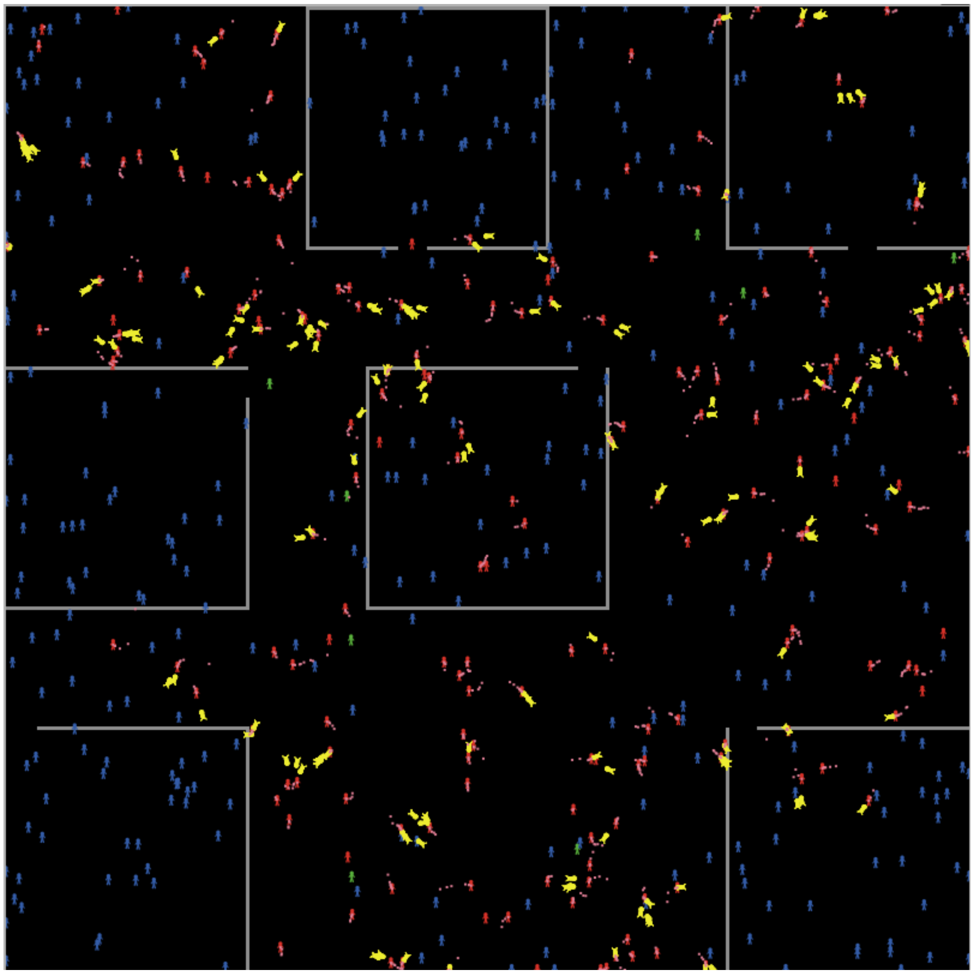

Figure 2. Plot of Netlogo viewing area with students (blue), infected students (red), sick students who sneeze (red with yellow germs), immune students (green) that have “healed” from the flu or had been sent home by the nurse, and contamination (pink).

Code Description

Our program procedure was constructed in Netlogo. The full Netlogo code can be found in the Appendix at the end of the report.

First we set up the boundaries of the rooms in the field. Each line of code sets up one wall, and we added another to make the doors. This is the code to set up the upper right room:

```
ask patches with [pxcor <= max-pxcor and pxcor >= 200 and pycor = 200] [ ;
```
This makes the lower boundaries of the first room–
```
   set pcolor gray ; -and colors it gray
 ]
 ask patches with [pycor <= max-pycor and pycor >= 200 and pxcor = 200] [ ;
```
Makes left boundaries of first room
```
   set pcolor gray
 ]
 ask patches with [pxcor <= max-pxcor and pxcor >= 300 and pxcor <= 325 and
pycor = 200] [ ;
```
Makes door for first room –
```
   set pcolor black ;
```
–and makes it black, the default color for the field
```
 ]
```

We then make all of the walls thicker with a single statement:
```
ask patches with [pcolor = gray] [ask neighbors [set pcolor gray]]
```

Then we create the students:
```
create-students number-of-students [
   set color blue
   set shape "person"
   set size 10
   if random 100 <= sick-percent-at-start [
     set infected 1
     set immune 1
   ]
```

–and then scatter them across the field:

```
   set xcor (random 800) - 400
   set ycor (random 800) - 400
   ; Randomize location if in a wall
   if pcolor = gray [
     set xcor (random 800) - 400
     set ycor (random 800) - 400
   ]
 ]
```

Finally, we set up some global parameters:

```
set #days 0
set-default-shape germs "bug"
set-default-shape contaminations "circle"
ask students [set day 1]
```

–and clear the plots:

```
set-current-plot "Number Infected" ; Set up plots
clear-plot
set-current-plot "Number Immune"
clear-plot
```

For the “go” procedure, which makes the program go, we started with making the students move:

```
ask students [
      ifelse pcolor = gray [
        right 180
        forward 1
      ][
        forward 1
      ]
      right random 45
      left random 45
```

We also added the procedure for making infected students sneeze:

```
      if sick >= 1 [
        if random 100 <= sneeze-percentage [
          hatch 5 [
            set breed germs
            set color yellow
            set size 10
          ]
        ]
      ]
```

and the one for contamination:

```
      if contagious > 0 [
        hatch 1 [
          set breed contaminations
          set color pink
          set size 2
        ]
      ]
```

Then come the end-of-day procedures:

```
      set #d #days
      ; each loop = 5 minutes, so 288 loops = 24 hours = one day
      if day = 288 [
        set #days #days + (1 / count students)
        set day 0

        ; if a student is infected, make them contagious
        if infected >= 1 [
          set infected infected + 1
          set contagious contagious + 1

          ; if contagious for more than a day, make them sick
          if contagious > 1 [
            set sick sick + 1

            ; send some students home (make them immune immediately)
```

```
            ; later add slider to say how effective nurse is at sending
students home
            if random 100 <= nurse [
              set immune 1
            ]

            ; if they are sick at least a certain number of days, chance they
will get better
            if sick >= days-til-better [
               if random sick >= days-til-better [
                 set infected 0
                 set contagious 0
                 set sick 0
               ]
            ]
          ]
        ]
      ]
```

Then we added a piece to make students change colors depending on if they are infected, healthy, or immune:

```
      ifelse infected >= 1 [
        set color red
      ] [
        ifelse immune = 1 [
          set color green
        ][
          set color blue
        ]
      ]
```

Then we make the germs move:

```
    ask germs [
      repeat 3
        [if pcolor != gray
          [
          forward 1
          ]
        ]
      set counter counter + 1
      ; if the sneeze is greater than 10 ticks, germs die
      if counter >= 10 [
        die
      ]
    ]
```

The contamination does not move, and just checks to see if it should die now:

```
    ask contaminations [
        if random 100 <= contamination-chance-die [
```

```
        die
      ]
    ]
```

Finally, we update the plots and increase the tick counter:

```
  if #days > #d [
      set-current-plot "Number Infected"
      plot count students with [infected >= 1]
      set-current-plot "Number Immune"
      plot count students with [immune >= 1]
    ]
  tick
```

The last procedure infects the students:

```
to infect-students
  ask students [
    ; can only catch cold if not immune
    if immune = 0 [
      if count (turtles-here with [breed = germs]) >= 1 or count (turtles-here
with [breed = contaminations]) >= 1 [
        set infected 1
        set color red
        set immune 1
      ]
    ]
  ]
end
```

Results

Initial Results

We ran many runs varying different parameters to see how it would affect the length and peaks of the flu. For the first run, we set the number of students to 600, the effectiveness of the nurse to 5%, and the sneeze-percentage to 15%. It took 3 days for students to get better. Two percent of the students were sick at the start, and there was a 50% chance that contaminations would die. From this run we got this data: Peak Infected Students- 294 on day 12, Peak Immune- 570 at the end of the run (that is always the case), and the flu lasted 37 days. For the next run, we left the inputs the same as the first run except we changed the efficiency of the nurse to 50% to see if the flu would not be as strong. As it turns out, the results were not that different from the run before besides the fact that the flu only lasted 34 days. After further testing, we found out that the key factor in flu was not the nurse, but was actually the sneeze percentage. Apparently, if kids just use their tissues, the flu wouldn't be so bad.

Final Results

Because the initial results suggest that the sneeze percentage rate is the parameter that most strongly affects the model results, we chose to vary that parameter and observe the results. We ran each sneeze percentage (1, 5, 10, 15, and 20%) ten times and collected the averages and standard deviations for the peak number of infected students, the day of the run the peak occurred, the number of students that were never infected, and the number of days until the program finished (when no students were infected).

Figure 3 shows the results of varying the sneeze percentage rate of each student during the 5 minute tick intervals. The peak number of infected students does appear to increase with sneeze rate. The number of students never infected gets smaller as sneeze rate increases. The day of the infected student peaks and the number of days until the simulation completes does not change much with sneeze rate. Also, when the sneeze rate is less than 10%, the variation of the number of student never infected is large.

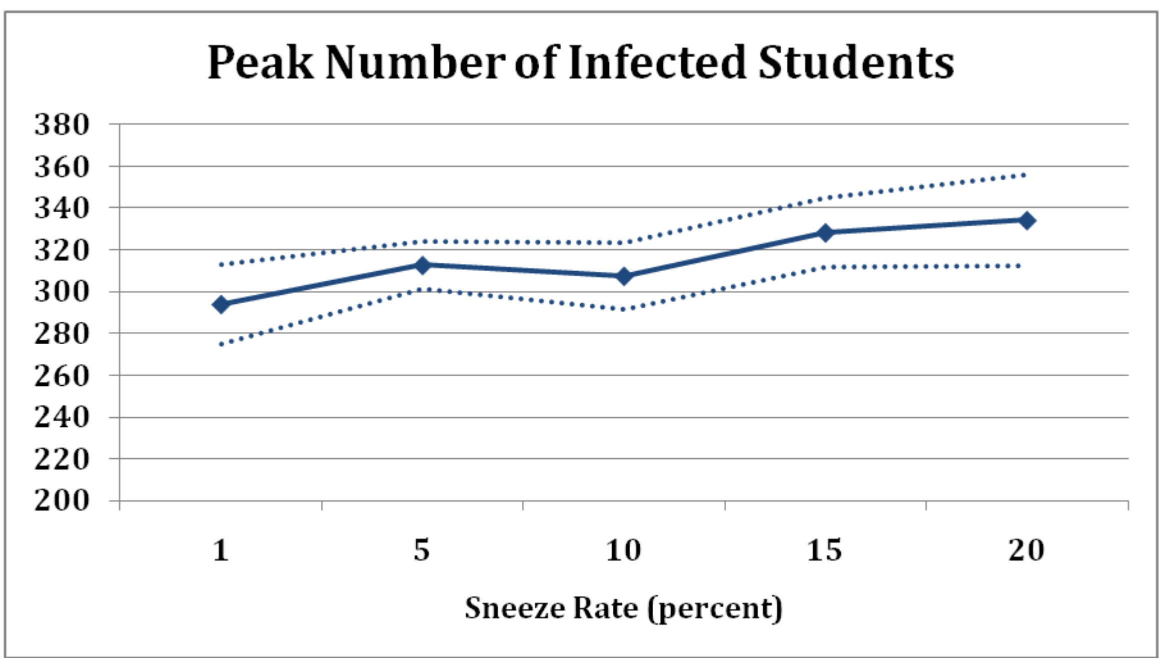

Figure 3. Peak number of infected students with sneeze rate percentage. The peak increases with sneeze rate. There is not much variation.

Throughout all of our program runs, almost all students eventually became infected during the 25-30 days it took until no infected students were left. The flu in our simulation was very easy to catch.

When we added classrooms to the simulation, the infection spread slower and there were more non-infected people at the end then if there were no classrooms. The classrooms also would isolate or concentrate the flu infection.

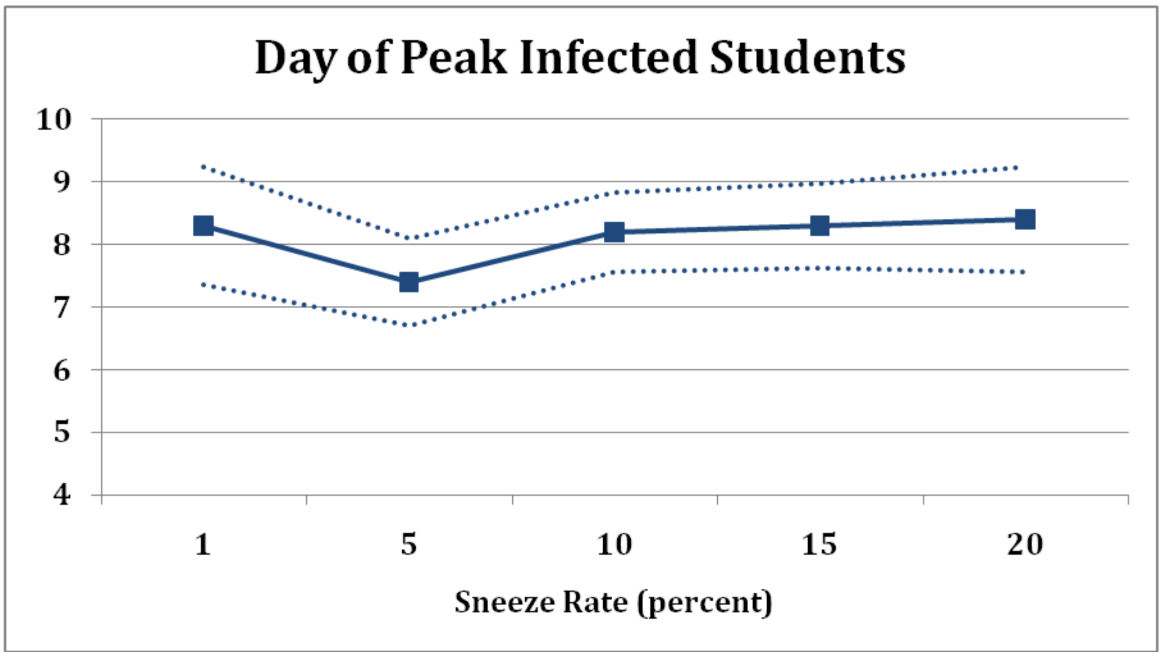

Figure 4. The day number that the peak infection occurred in the simulation. There is not much variation and the peak day usually occurred around 8 days.

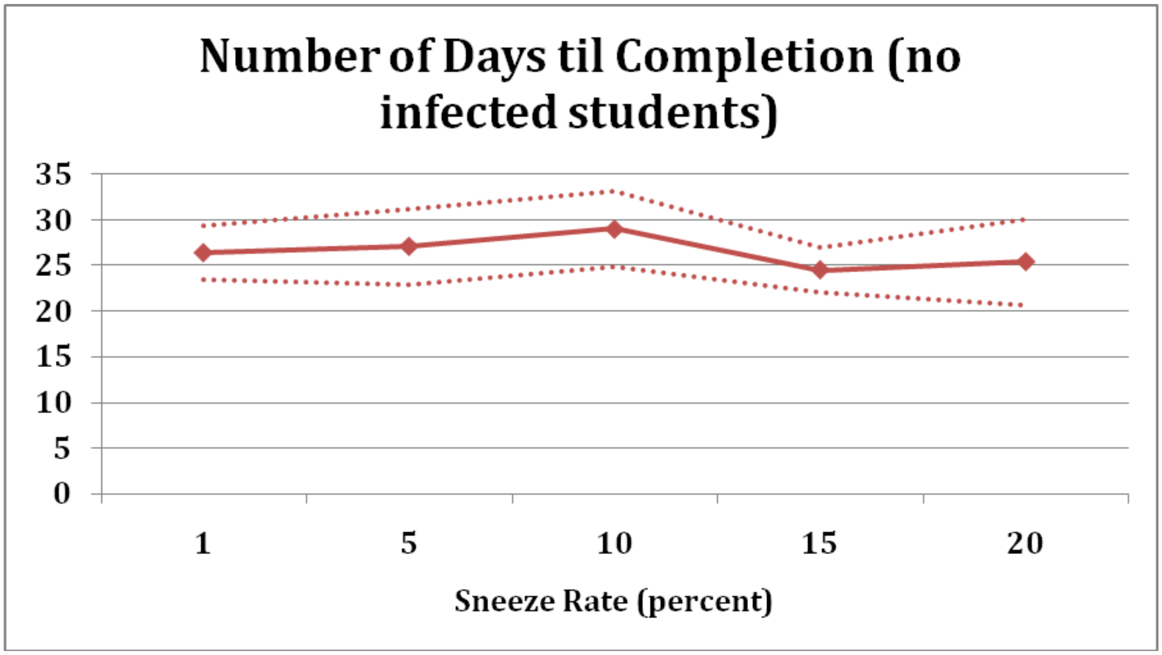

Figure 5. The number of days until program completion plotted against the sneeze rate. The program completed when no students were infected anymore. The value stays around 25-30 days.

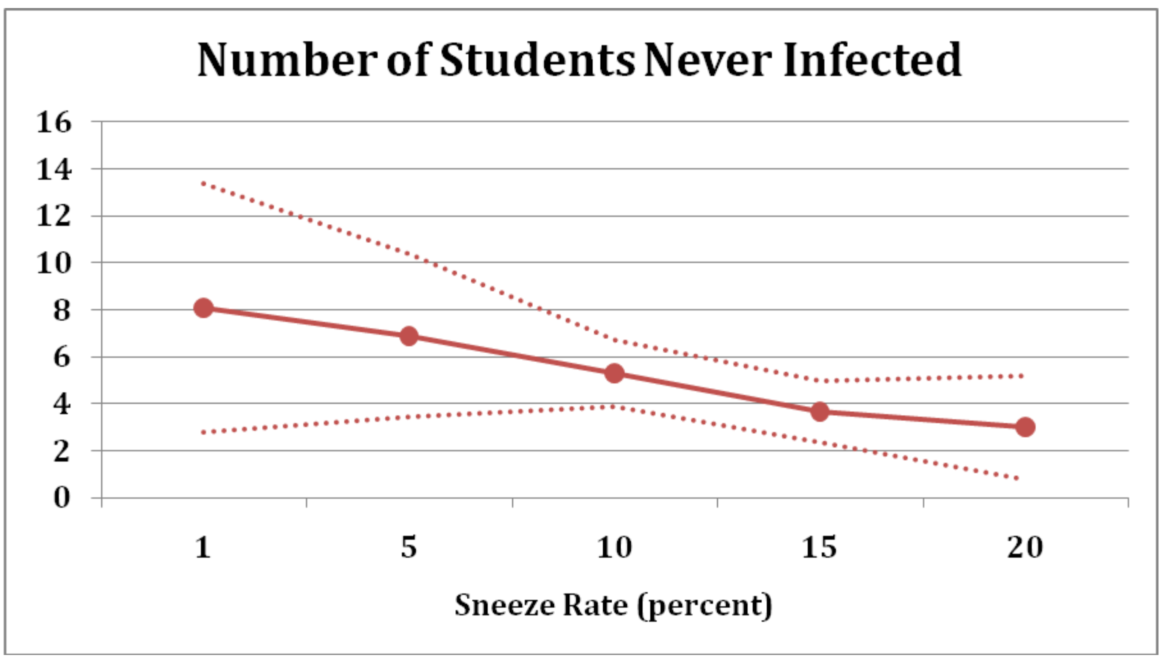

Figure 6. The number of students that were never infected against sneeze rate. As the sneeze rate increases, less students remain uninfected. Below 10% sneeze rate, the variation in the number is larger.

Conclusion

We chose to create an agent-based model for the spread of the flu in a school using the Netlogo software. The agents were students, germs, and contaminated surfaces. The students went from being infected, to becoming sick, then becoming immune. Infected students were able to contaminate surfaces and sick students spread germs by sneezing.

Adding the classrooms to the simulation allowed isolation or concentration of the flu infection and allowed more students to remain uninfected at the end of the simulations.

We found that the rate of sneezing for each 5-minute tick interval was responsible for the amount of infected students. Varying the sneeze rate from 1%-20% showed us that when the sneeze rate was higher, less students were not infected or immune at the end of the simulation. The peak number of infected students did increase with sneeze rate. The day the infection peak occurred and the number of days until the simulation completed did not change much over the sneeze rate. The day the infected peak occurred was almost always about 8 days.

In this model we wanted to determine the behavior of the existing setup. For future work, we want to gather more data on true infection rates at schools, give the students daily class schedules, and add a few more classrooms.

Acknowledgements

We wish to thank the makers of the Netlogo program for a great interactive way to program. We also want to thank our teacher Mr. Dryja and our mentor Dr. Begnaud.

Appendix: Full Netlogo Code

```
breed [students student]
breed [nodes node]
breed [germs germ]
breed [contaminations contamination]

globals [#days #d maxinfected maxinfectedday]

students-own [sick contagious infected immune day]

germs-own [counter]

contaminations-own [counter]

nodes-own [number]

to setup
 ca ; Clear field and create walls
 ; upper right
 ask patches with [pxcor <= max-pxcor and pxcor >= 200 and pycor = 200] [ ;
Make lower boundaries of first room
   set pcolor gray
 ]
 ask patches with [pycor <= max-pycor and pycor >= 200 and pxcor = 200] [ ;
Make left boundaries of first room
   set pcolor gray
 ]
 ask patches with [pxcor <= max-pxcor and pxcor >= 300 and pxcor <= 325 and
pycor = 200] [ ; Make door for first room
   set pcolor black
 ]

 ; center area
 ask patches with [pxcor >= -100 and pxcor <= 100 and (pycor = 100 or pycor =
-100)] [
     set pcolor gray
 ]
 ask patches with [pycor >= -100 and pycor <= 100 and (pxcor = 100 or pxcor =
-100)] [
     set pcolor gray
 ]
 ask patches with [pxcor <= 100 and pxcor >= 75 and pycor = 100] [ ; Make door
   set pcolor black
 ]

 ; offset upper room
 ask patches with [pxcor >= -150 and pxcor <= 50 and (pycor = 400 or pycor =
200)] [
     set pcolor gray
 ]
 ask patches with [pycor >= 200 and pycor <= 400 and (pxcor = -150 or pxcor =
50)] [
     set pcolor gray
 ]
 ask patches with [pxcor <= -50 and pxcor >= -75 and pycor = 200] [ ; Make
door
   set pcolor black
 ]

 ; lower right
```

```
 ask patches with [pxcor <= max-pxcor and pxcor >= 200 and pycor = -200] [ ;
Make lower boundaries of first room
   set pcolor gray
 ]
 ask patches with [pycor >= min-pycor and pycor <= -200 and pxcor = 200] [ ;
Make left boundaries of first room
   set pcolor gray
 ]
 ask patches with [pxcor >= 200 and pxcor <= 225 and pycor = -200] [ ; Make
door for first room
   set pcolor black
 ]

; lower left
 ask patches with [pxcor >= min-pxcor and pxcor <= -200 and pycor = -200] [
   set pcolor gray
 ]
 ask patches with [pycor >= min-pycor and pycor <= -200 and pxcor = -200] [
   set pcolor gray
 ]
 ask patches with [pxcor >= min-pxcor and pxcor <= -375 and pycor = -200] [ ;
Make door for first room
   set pcolor black
 ]

; mid left
 ask patches with [pxcor >= min-pxcor and pxcor <= -200 and (pycor = 100 or
pycor = -100) ] [
   set pcolor gray
 ]
 ask patches with [pycor >= -100 and pycor <= 100 and pxcor = -200] [
   set pcolor gray
 ]
 ask patches with [pycor >= 75 and pycor <= 100 and pxcor = -200] [
   set pcolor black
 ]

; set all wall thickness bigger
 ask patches with [pcolor = gray] [ask neighbors [set pcolor gray]]


create-students number-of-students [ ; Create students
   set color blue
   set shape "person"
   set size 10
   if random 100 <= sick-percent-at-start [
     set infected  1
     set immune 1
   ]
   set xcor (random 800) - 400
   set ycor (random 800) - 400
   ; Randomize if in a wall
   if pcolor = gray [
     set xcor (random 800) - 400
     set ycor (random 800) - 400
   ]
 ]
 set #days 0
 set-default-shape germs "bug"
 set-default-shape contaminations "circle"
 ask students [set day 1] ; Set day

 ; clear plots
 set-current-plot "Number Infected" ; Set up plots
```

```
 clear-plot
 set-current-plot "Number Immune"
 clear-plot
 set maxinfected 0
 set maxinfectedday 0
end

to go
  ask students [
      ; move the students
      ifelse pcolor = gray [
        right 180
        forward 1
      ][
        forward 1
      ]
      right random 45
      left random 45

      ; advance the day
      set day day + 1

      ; if sick, make them sneeze and create germs
      if sick >= 1 [
        if random 100 <= sneeze-percentage [
          hatch 5 [
            set breed germs
            set color yellow
            set size 10
          ]
        ]
      ]

      ; if contagious, leave behind contamination
      if contagious > 0 [
        hatch 1 [
          set breed contaminations
          set color pink
          set size 2
        ]
      ]

      ; check if at end of day
      set #d #days
      ; each tick = 5 minutes
      if day = 288 [
        set #days #days + (1 / count students)
        set day 0

        ; if a student is infected, make them contagious
        if infected >= 1 [
          set infected infected + 1
          set contagious contagious + 1

          ; if contagious for more than a day, make them sick
          if contagious > 1 [
            set sick sick + 1

            ; send some students home (make them immune immediately)
            ; later add slider to say how effective nurse is at sending
students home
            if random 100 <= nurse [
              set immune 1
            ]
```

```
            ; if they are sick at least a certain number of days, change will
get better
            if sick >= days-til-better [
               if random sick >= days-til-better [
                 set infected 0
                 set contagious 0
                 set sick 0
               ]
            ]
          ]
        ]

      ]

      ; change colors for different states
      ifelse infected >= 1 [
        set color red
      ] [
        ifelse immune = 1 [
          set color green
        ][
          set color blue
        ]
      ]
    ]

    ; check for infection
    infect-students

    ; move the germs forward and keep from going across a wall
    ask germs [
      repeat 3
        [if pcolor != gray
          [
          forward 1
          ]
         ]
      set counter counter + 1
      ; if the sneeze is greater than 10 ticks, die (need to find out how long
airborne sticks in a room)
      if counter >= 10 [
        die
      ]
    ]

    ; contamination has a chance of dying each tick
    ask contaminations [
        if random 100 <= contamination-chance-die [
          die
        ]
    ]

    ; update plots at end of each day
    if #days > #d [
      set-current-plot "Number Infected"
      plot count students with [infected >= 1]
      set-current-plot "Number Immune"
      plot count students with [immune >= 1]
      if count students with [infected >= 1] > maxinfected [
        set maxinfected count students with [infected >= 1]
        set maxinfectedday #days
      ]
    ]
```

```
  tick
end

to infect-students
  ask students [
    ; can only catch cold if not immune
    if immune = 0 [
      if count (turtles-here with [breed = germs]) >= 1 or count (turtles-here
with [breed = contaminations]) >= 1 [
        set infected 1
        set color red
        set immune 1
      ]
    ]
  ]
end
```