Pranav Agarwal (NUID - 001099801)
Jyoti Vaswani (NUID - 001005711)
INFO 6205 - Program Structures and Algorithms (Fall 2020)

Virus Transmission Simulation

1. Introduction:

With every passing day, we are witnessing the global pandemic crashing health care systems, stock markets, and businesses throughout the world. People all around the globe are intently following news reports and other data about the COVID-19 pandemic. The entire world has the same goal: cases of COVID-19 need to go to zero, but unfortunately, people continue to die at an alarming rate.
In this project, we simulate the virus transmitted pandemic of influenza virus and coronaviruses such as SARS-CoV (2003) and SARS-CoV-2 (2019). We explore what makes SARS-CoV-2 different from pandemic influenza virus and the epidemic severe SARS-CoV virus.

2. Aim of the project:

In this project we are aiming:

- To simulate the spread of the viruses SARS-CoV-2 i.e. the pathogen behind COVID-19, SARS-CoV-1 and Influenza in rural, sub-urban and urban areas.
- Analyse the data generated by simulation to understand the R^o and standard deviation in individual spread for these 3 viruses and the environmental settings or precautions that can contribute in keeping these values under control.
- Use the data and observations to draw conclusions on the efficacy of precautionary measures like lockdown, quarantining, enforcing masks, social distancing, etc. in preventing the spread of the virus.

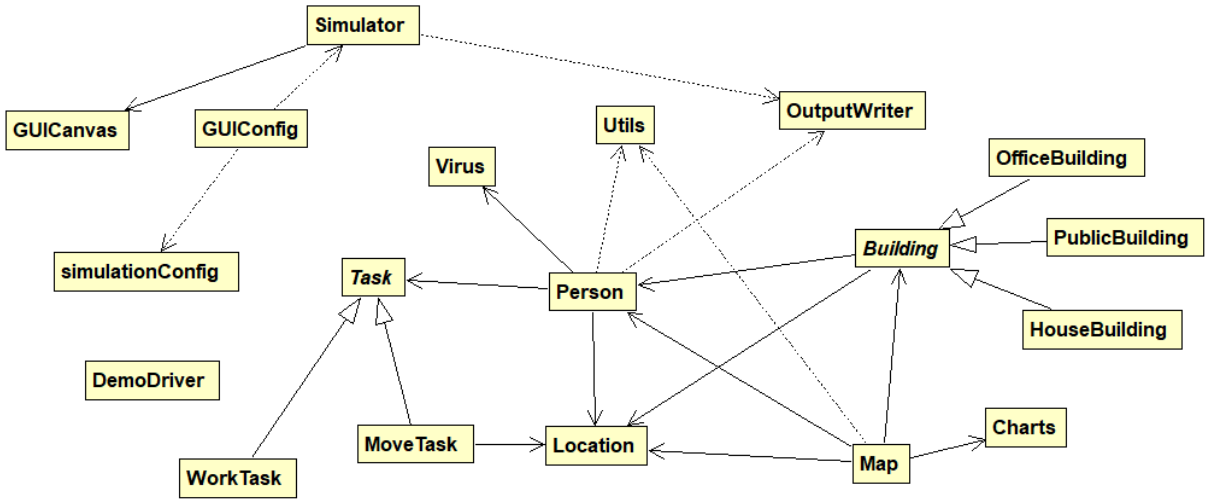
3. Project Details:

We have used Java and its libraries to model the virus and it's spread in a population. Excel was used to find the R^o value and the standard deviation, and Tableau was used to create visualizations of the data output by the simulation. Other than the standard Java JDK, we used Knowm XChart (https://knowm.org/open-source/xchart/) to create the real time graphs, and Junit for unit testing.

The complete code and the executable can be found at https://github.com/Pranav-Agarwal/disease-spread-simulation

4. Model Entities:

The model simulates the spread of a disease (Virus) in a certain population (Persons) as they move around in a particular way around a 2D grid area (Map), which contains some discrete contiguous segments of grid squares (Buildings) of various types. A detailed explanation of the class object structure, and algorithms used to realize this model follows.



1 Approximate Class Diagram

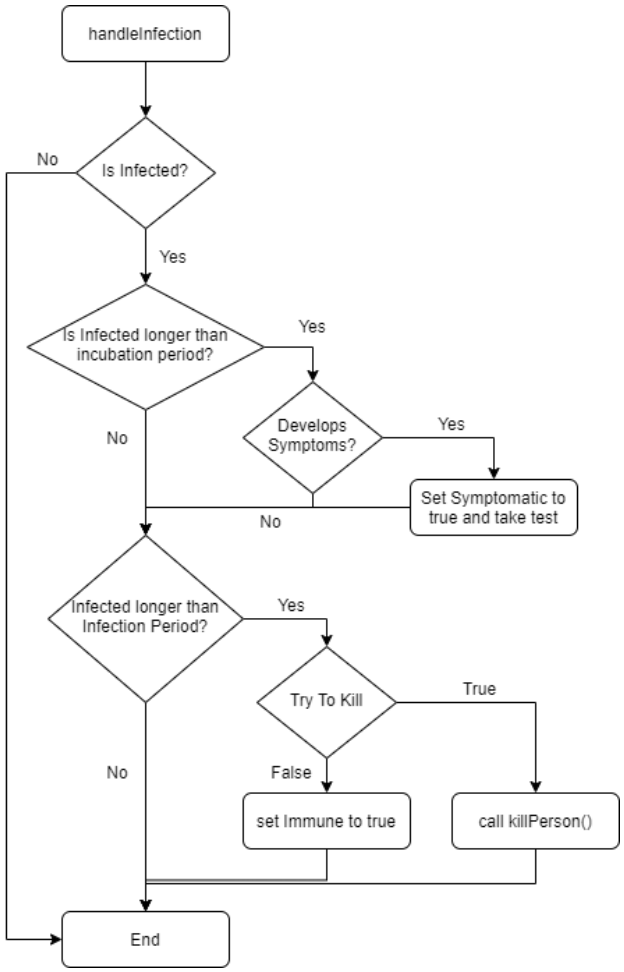
This is an approximate Class Diagram of the model. Some relationships have been excluded for clarity and only the most important ones have been drawn. A brief explanation of the classes created follows –

1. DemoDriver – Main entry point for the program. Initializes other objects and drives the simulation.
2. simulationConfig – Class that reads all properties from the configuration files when it is initialized and stores them in static variables for the rest of the program to use.
3. Location – Represents a particular square on the 2D grid. Has a set of the people currently in this spot, its own x and y in the grid, and of what type it is depending on which building it belongs to.
4. Building – Represents a continuous collection of Locations on the map. It is important to the model as the disease can only be spread between two people in the same building. It contains a list of locations that comprises it, a set of people currently present in it, and a property defining whether it is in lockdown or not.

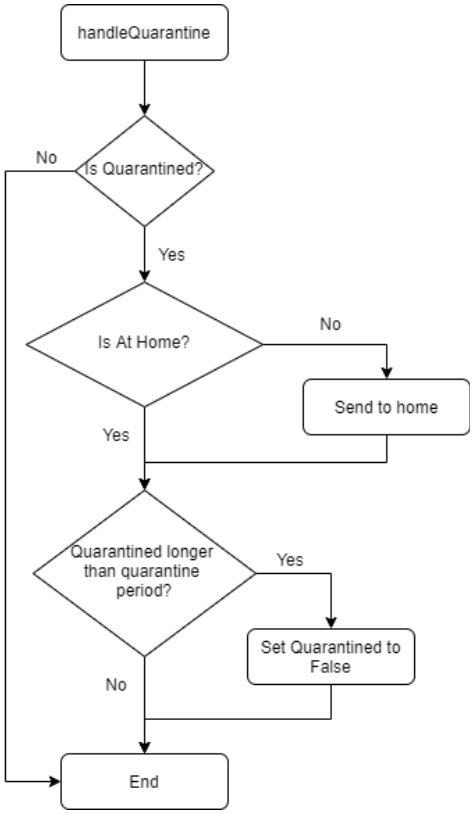
5. Map – Represents a 2D grid of Location objects containing people and buildings on which the virus will be released. Also contains the counters for various simulation statistics. Has methods for initializing buildings, people and enforcing various conditions the user may impose upon the simulation.
6. Person – Represents a person that will move around on the map in a particular fashion. Contains information about it's position on the map, and a task queue that will determine it's behaviour. Also has various properties such as isDead, isInfected etc. that represent the state of the person at that time in the simulation. Has methods that determine it's movement, chance to die, get infected etc.
7. Virus – Represents a pathogen that will be released in the population. It has properties such as infectivity, incubation period, lethality etc. These properties are initialized at start time and are not changed for a simulation instance.
8. Simulator – Extends Java's TimerTask. Its run method is called at regular intervals and it tells the Map to update itself and the people and buildings in it so the simulation can move to the next state.
9. Task – Abstract class that represents a task that a person will work upon until it is completed. It can take multiple simulation ticks for a Person to complete a task, until which time it's run method will be called.
10. MoveTask – Subclass of Task that instructs a Person to move from their current location to a destination location defined in the Task object during its initialization. Will be re-run every tick until the Person reaches the defined destination location.
11. WorkTask – Subclass of Task that instructs a Person to wait at their current location until a timeout number of ticks is reached which is defined in the Task object during its initialization. Will be re-run every tick until the number of ticks equalling timeout is reached
12. OutputWriter – Whenever a new infection happens, a death occurs, or a simulation tick happens, this class will save certain simulation statistics to csv files in an output folder. This is so this data can be analysed later for insights.
13. Utils – Contains static utility methods for general use by other classes
14. GUIConfig – Class that initializes the configuration swing component, and defines it's interaction with the rest of the program.
15. GUICanvas – Class that initializes the visualization of the simulation. Contain methods that read the map data and decide how to render it to accurately display the simulation state.
16. Charts – Uses the XChart external library to display a real time chart and statistics pertaining to the map state.
17. OfficeBuilding – Inherits off the Building Class to represent a Workplace. Buildings of this type are rendered in a particular color.
18. HouseBuilding – Inherits off the Building Class to represent a House. Buildings of this type are rendered in a particular color.
19. PublicBuilding – Inherits off the Building Class to represent a Public Area such as a park. Buildings of this type are rendered in a particular color.

5. Simulation behaviour flowcharts:

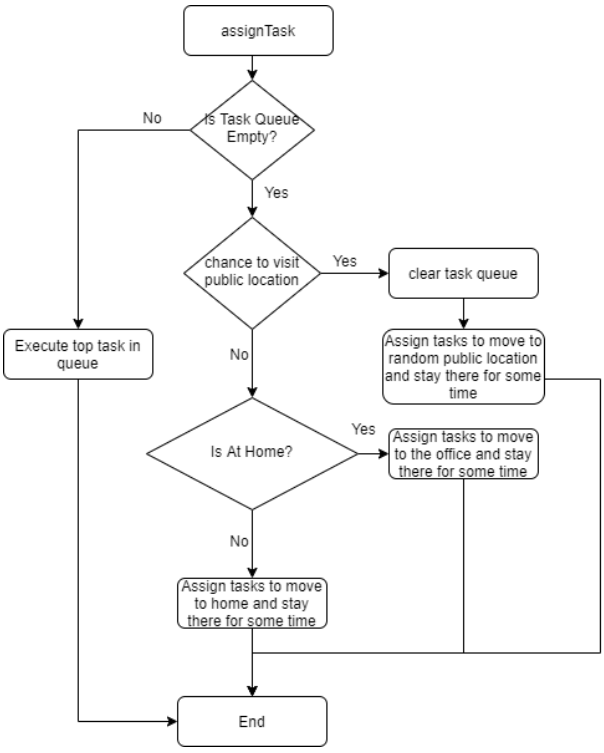
- **Person behaviour –**
  Every simulation tick, 3 methods are called on each person in the map in order, handleInfection, handleQuarantine, and assignTask. The flowcharts for each of these 3 methods are as follows -



2 Handle Infection Flowchart

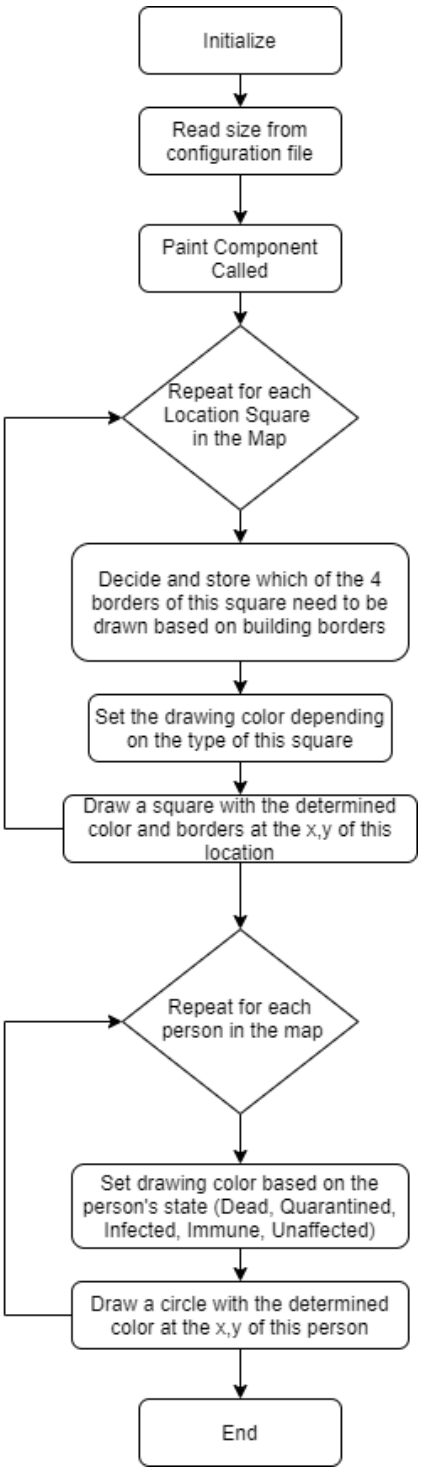


3 Handle Quarantine Flowchart



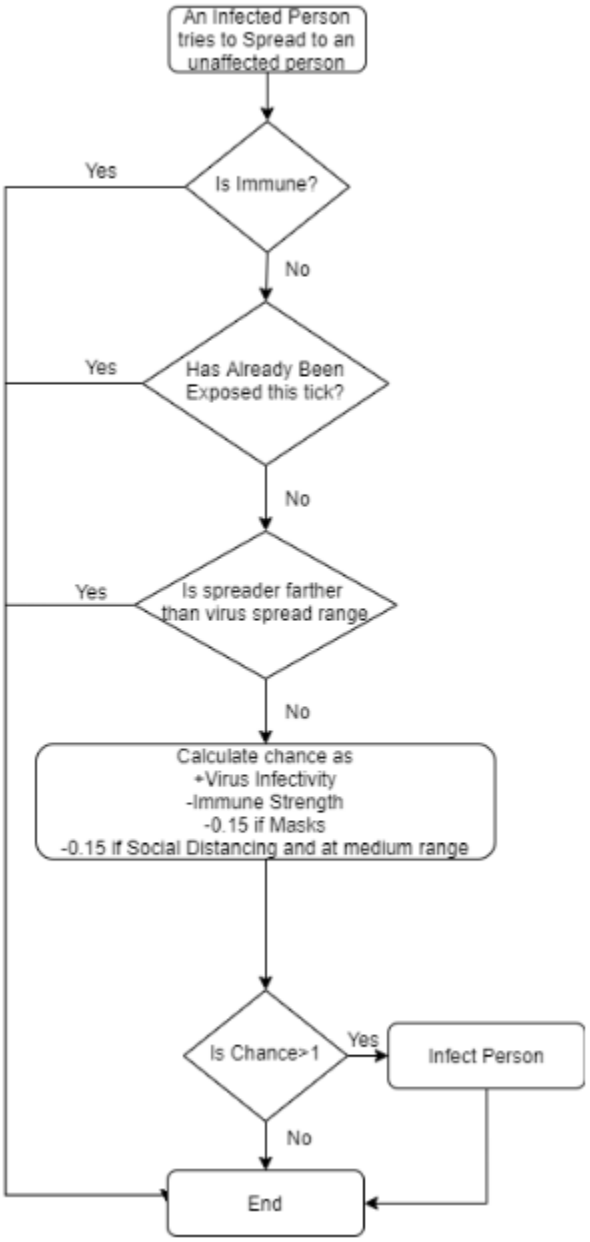
4 Assign Task Flowchart

- **Rendering logic –**
  Every ‘day’ (100 ticks), the visualization of the simulation is repainted according to the following logic -



5 Rendering logic flowchart

- **Infection logic –**



6. Complexity Analysis and Optimizations:

All the algorithms used in this simulation can be divided broadly into 2 categories, either run once at initialization, or run at regular intervals. The ones run at intervals need to complete before the next update instance is called so they are performance critical. A brief explanation of the complexity of major algorithms is below –

- Initialization Algorithms –
  i) Read configuration file – O(1) : Reads about 50 properties into static fields
  ii) Initialize Map – $O(S^2)$: Creates S x S Location objects in a 2D array
  iii) Create Buildings – $O(B * S_B)$: Creates B buildings of size roughly S_B Locations.
  iv) Create People – $O(P * B_O)$: Creates P people and allots them semi randomized office location based on the distance from their home. Distance of all offices need to be measured from this person's home, creating the extra term B_O , the number of offices.
  v) Seed Virus – O(1) : Randomly selects some people and infects them.

All in all, The edge length (S) of the map and number of people (P) affect the performance of initialization the most. However as this will be run only once (usually in a couple of milliseconds), it doesn't really have any practical effect. Performance of initialization is approximated to :

$$O(S^2 + B * S_B + P * B_O)$$

- Repeating Algorithms –
  i) Update Persons – O(P) amortized : calls the update method of each person on the map. This leads to the 3 flowcharts under "Person Behaviour" above being executed. Handle Quarantine and Handle Infection are O(1) , but Assign Task may become $O(B_P)$ if the random check for a person to visit a public location succeeds, because then they will choose a semi randomized location close to them, for which you need to consider all B_P public buildings. However, since this chance is very low, coupled with the fact that the value of B_p itself is usually very low (20-50) , this makes it effectively a constant time operation. So the net complexity is O(P), as each person's update is effectively constant.
  ii) Execute Task – O(1) : after each Person's update with the 3 methods above, the top task in the queue is executed. This could be either a WorkTask , which is nothing but a simple integer increment and compare with a timeout making it O(1), or a MoveTask, which moves the person 1 square closer to a defined destination. In addition to changing the x and y or a Person, the MoveTask also removes the Person object from the old location and building, and adds it to the new one. However since a HashSet has been used to store the list of people, adding and removing Person objects is a O(1) operation, resulting in the whole MoveTask execution for a tick being O(1) as well. The tasks are also stored in a Queue implemented as a Linked List, ensuring constant time addition and deletion of tasks, leveraging the fact that only the top task will ever need to be accessed.
  iii) Update Office Buildings – $O(B_O)$: This method is only used in the very specific case when an office building has been auto locked down and the timer needs to be updated and checked with a timeout to make it lift the lockdown automatically after some interval.
  iv) Spread Disease – $O(P^2/(B_P+B_O))$: Takes every combination of 2 people in a public or office building, and tries to spread the virus between them. If we assume the people are spread roughly uniformly throughout the offices and public locations, the number of people in each location would be P/B . So this algorithm would run for every building, depending quadratically on the number of people in each, making it $B * (P/B)^2 = P^2/B$. It is actually even less in practice, as a lot of people may be in House Buildings or moving around on the map.

This function is also called only every 100 ticks as that interval represents a “day”, improving the averaged per tick performance even more.

v) Take Test without contact tracing – O(1) : This method is called whenever a Person develops symptoms. It is a simple percentage chance then for the test to show positive.

vi) Take Test with contact tracing – θ(1) but may in extremely rare circumstances go to O(P) . When contact tracing is enabled, whenever a person tests positive, all their housemates and co-workers are tested as well recursively (if they have not been tested recently). This could potentially lead to a recursive chain of testing spreading to every person in the Map, however it is almost impossible, since the person has to be infected and tested positive to call a further recursion. Also, since office buildings are assigned close to People’s homes, the chances are high that there will only be a small block of nearby homes and offices this testing will extend to. In practice, it can be assumed that this testing will extend to a few buildings i.e., a few dozen Persons. To prevent infinite recursion and improve performance, a person tested recently cannot be tested again.

vii) Refresh visualization – $O(P + S^2)$: For every person and grid square on the map, draw a shape depending on certain conditions. Detailed flowchart for the same is above.

Overall, the complexity for each update tick is approximated to be :

$$(P) + (B_O) + (P^2/(B_P+B_O))/100 + (P+S^2)$$

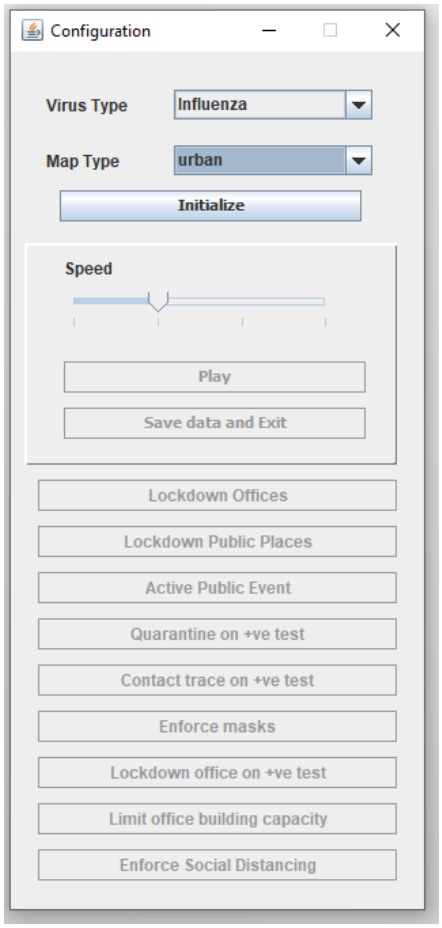
In its most simplified form, the performance per tick is quadratic on edge length and number of people in the simulation.

$$O(P^2+S^2)$$

7. Operation Manual:

- **Setup**
  i) Go to https://github.com/Pranav-Agarwal/disease-spread-simulation and download the executable folder.
  ii) Copy the configuration file and the Jar to the same location.
  iii) Edit the configuration file as desired
  iv) Double click the Jar to begin the program
  v) Click “Initialize” on the panel that shows up.

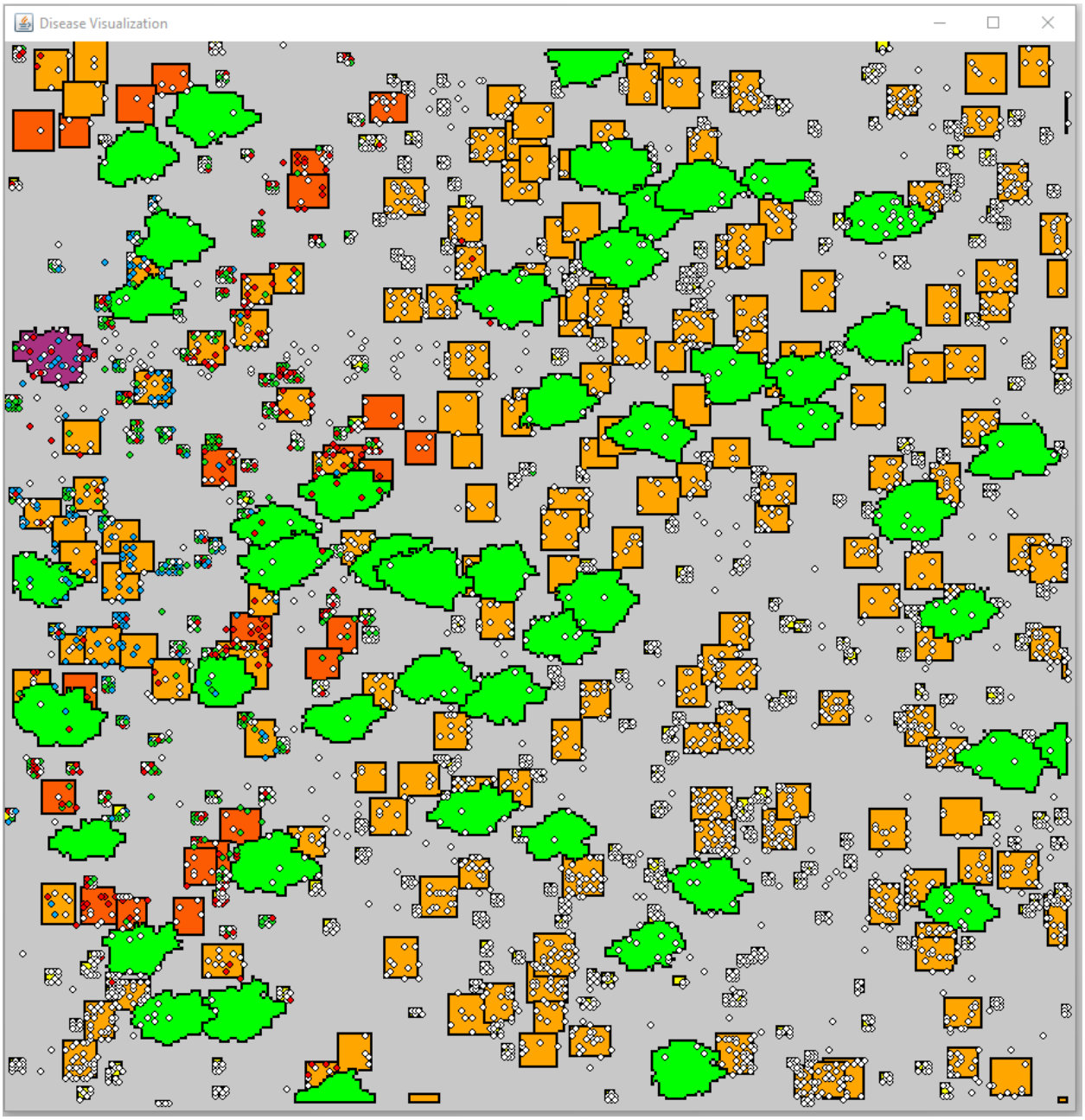
- **Runtime Configuration**
  Used to dynamically alter the simulation as it is running (except for map and virus type)



1. Virus type – User can select 3 different virus types, each with different properties
2. Map type – User can select from 3 different Map types, which vary in population and density
3. Lockdown Offices – Prevents move tasks to office locations being assigned
4. Lockdown Offices – Prevents move tasks to public locations being assigned

5. Active Public Event – Greatly increases the chances people will visit a particular public location and form a large crowd
6. Quarantine on +ve test – If a person tests positive, they will go into quarantine
7. Contact trace on +ve test – If a person tests positive, all their co-workers and housemates are also tested
8. Enforce masks – Reduces infection chance
9. Lockdown office on +ve test – If a person tests positive, automatically lockdown their office for a short duration
10. Enforce social distancing – reduces infection chance at medium ranges

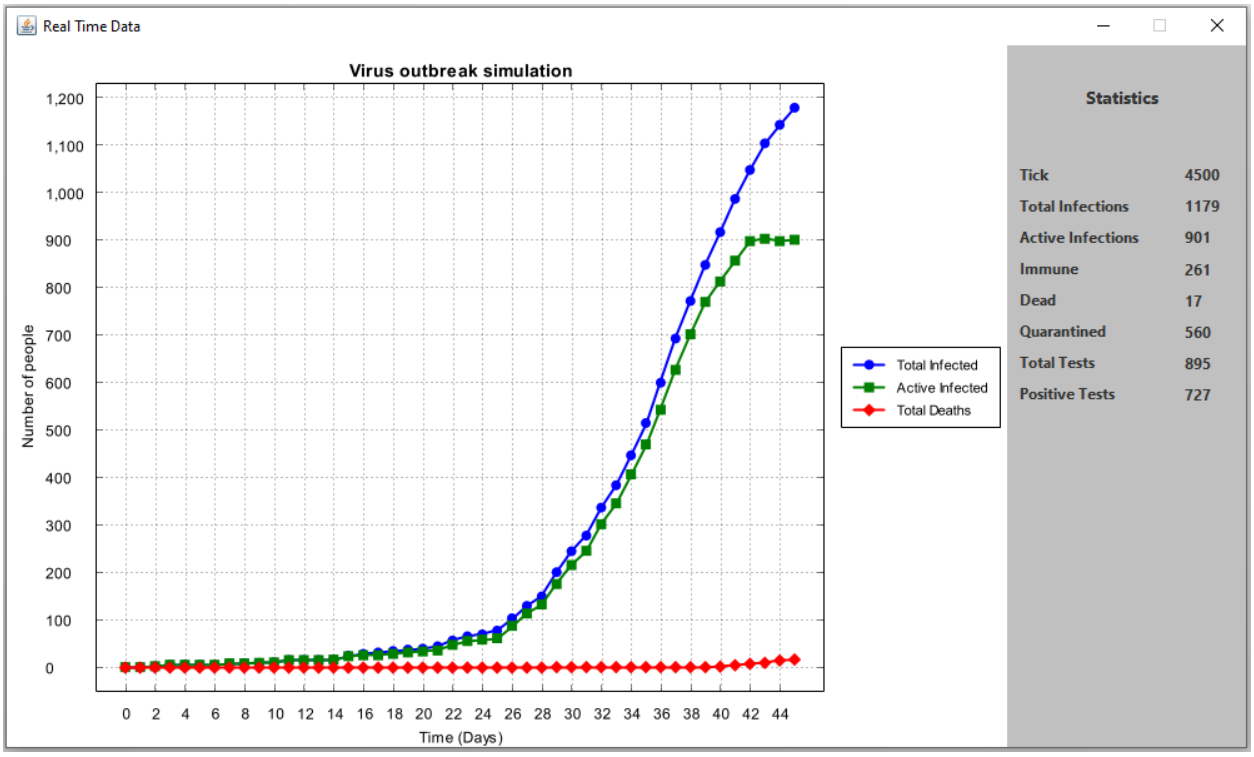
- **Visualization Panel**
  Represents the state of the simulation graphically.



Legend –

i) Orange Square – Office
ii) Green Square – Public Location
iii) Yellow Square – House
iv) Dark Orange – Locked down building
v) Purple building – Public event active there
vi) White circle – unaffected people
vii) Red circle – Infected people
viii) Green circle – quarantined people
ix) Blue circle – Immune people

- **Real time graphs**
  A real time graph and statistics window will also be shown



- **Output**
  There will be an output folder created in the same location as the Jar after the “Save and Exit” button is pressed. This folder will contain timestamped folders for each simulation run. Each of these folders has 3 CSV files for raw simulation data, infection data, and death data, and also a snapshot of the how the real time graph looked at the time of exit.

8. Results and Analysis:

In this section, we are comparing the effect of each of the precautionary measure taken for controlling the spread of virus in an urban (densely populated) area.
Note: Keeping the below mentioned configuration as constant in all the scenarios
VirusSeedCount=5
People on Map= 7000
SpreadDisease()- only after 100 simulation ticks(1 Day).

Scenario 1. Quarantine the people who tested positive with active contact tracing.

Result:

- Figure 6. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 7. shows the spread of SARS-CoV-2 when we actively quarantine the infected people (who have been tested positive) and trace people proactively who were in contact.

The number of days required for virus to infect 5000 people on the map is:

- 105 days when no precautionary measures are taken and
- 180+ days when we quarantine the people who have been tested positive and trace people proactively who were in contact. The curve starts to flatten on 4500.

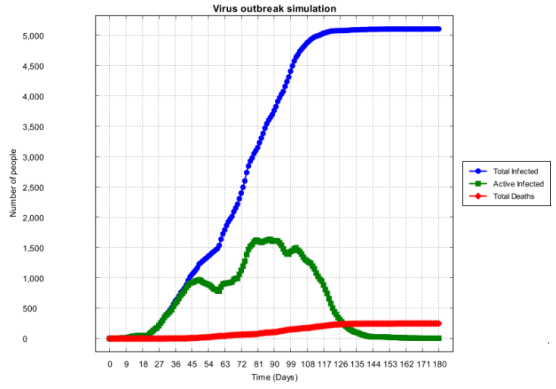


Figure 6:no precautionary measures

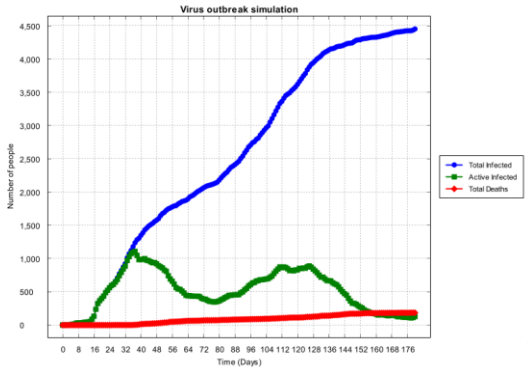


Figure 7: quarantine people who tested positive + tracing

Analysis:

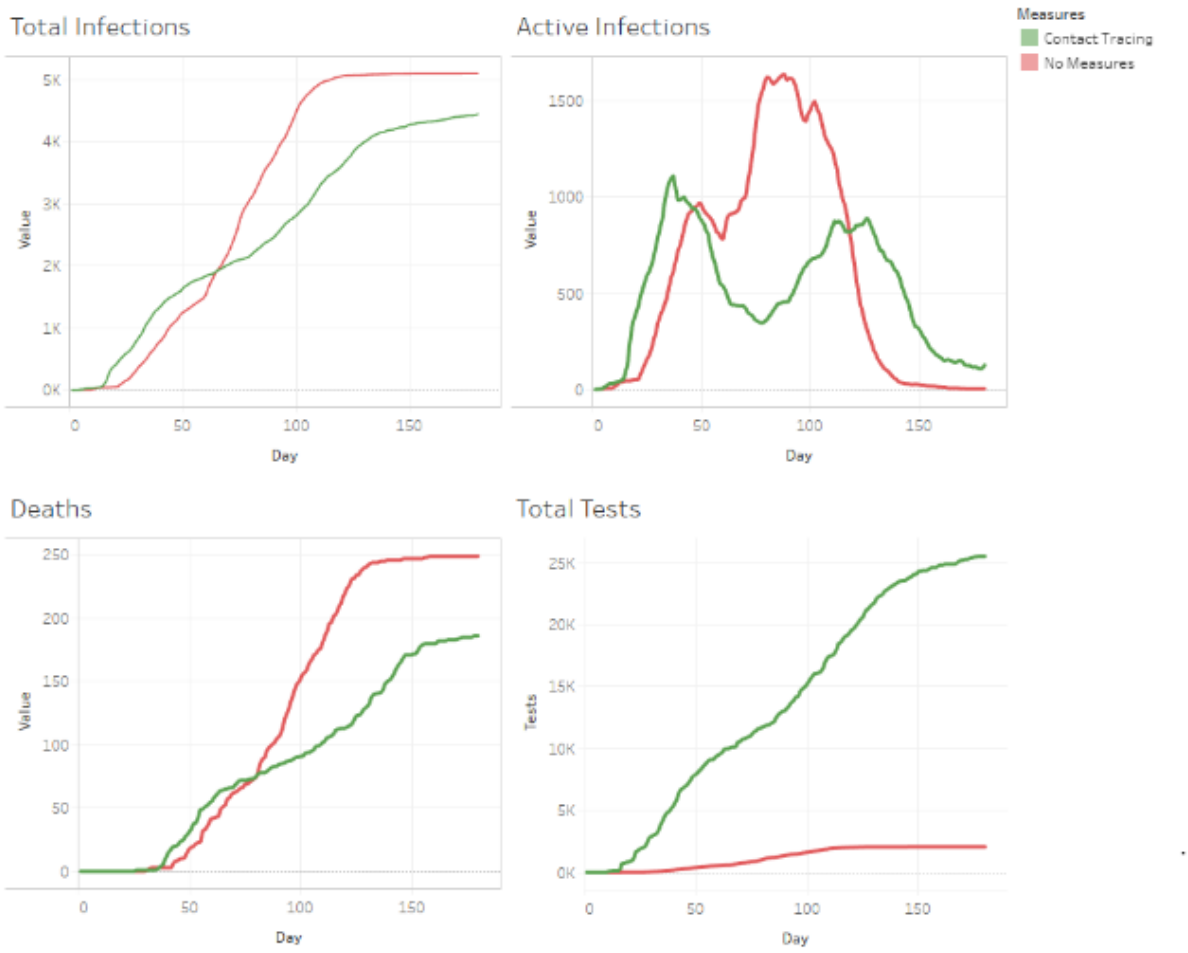


Figure 8: Comparative study

- For the SARS-CoV-2 virus, the incubation period is observed to be 5 days. Hence, the measure like quarantining and the active contact tracing is highly effective in containing the spread of virus.
- The proactiveness of contact tracing and immediate quarantine provides the required time to stop the unknown spreader from spreading the virus further.
- Thereby reducing the R0 to 2.21809.

Scenario 2. Enforcing people to wear masks.

Note: For this scenario, we have assumed that on enforcing usage of masks in simulation, all the people on the map are wearing a high quality 5- layered masks. Our study focuses on over-all effectiveness of masks in controlling transmission of virus and not on comparative study of types of masks (all such variations are ignored).

Result:

- Figure 9. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 10. shows the spread of SARS-CoV-2 when people are enforced to wear masks.

SARS-CoV-2 virus infects:

- 4000 people on the map in 90 days when no precautionary measures are taken and
- 4000 people on the map in 145 days when people are enforced to wear masks.

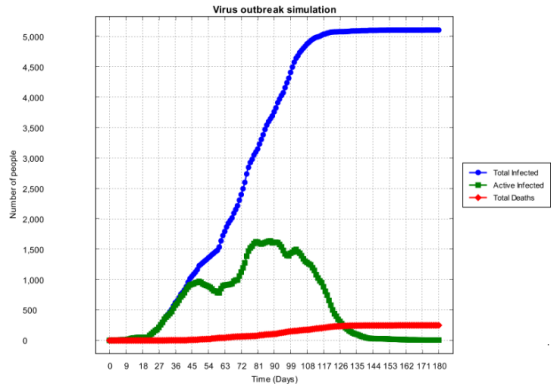


Figure 9:no precautionary measures

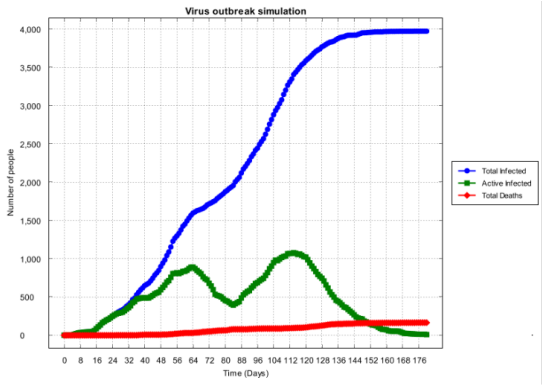


Figure 10: people are enforced to wear masks

Analysis:

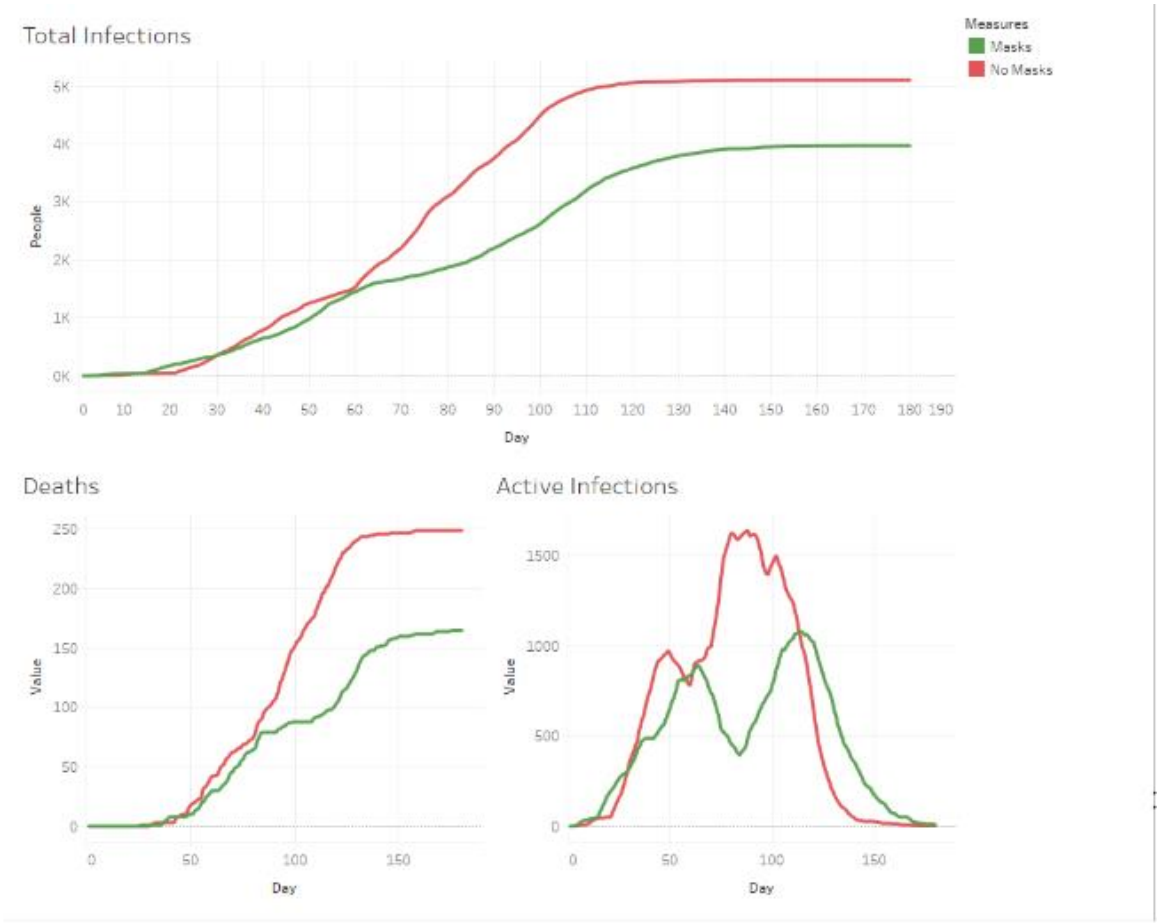


Figure 11: Comparative study

- For SARS-CoV-2, the nature of major transmission route is known to be the respiratory droplets during regular speech of the people. When we enforce people to wear masks it adds a physical barrier and is able to limit the range till which the infecting droplets can travel thereby preventing spreader from spreading and victims from contracting.
- The above simulation proves the same. The mark of infecting 5000 people on map is not reached and the total infection curves starts to flatten on the mark of 4000 people.
- It took 90 days to infect 4000 people when no precautions are taken and it increases further till 5000 before flattening whereas when masks are enforced it took 145 days to infect 4000 people and the curve starts to flatten soon.
- The enforcement of masks thus prevents the spread by more than 60%.
- The R0 is noted to be 2.158868.

Scenario 3. Enforcing social distancing on all places.

Note: For this scenario, we have assumed that on enabling social distancing there would still some people within the range of droplet travel distance and hence will be more prone to infection. Also, the transmission through touching various surfaces (doors, elevators) would still have an impact.

Result:

- Figure 12. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 13. shows the spread of SARS-CoV-2 when people are maintaining social distancing.

SARS-CoV-2 virus infects:

- 5000 people on the map in 108 days when no precautionary measures are taken and
- 4800(~5000) people on the map in 145 days when people are maintaining social distancing.

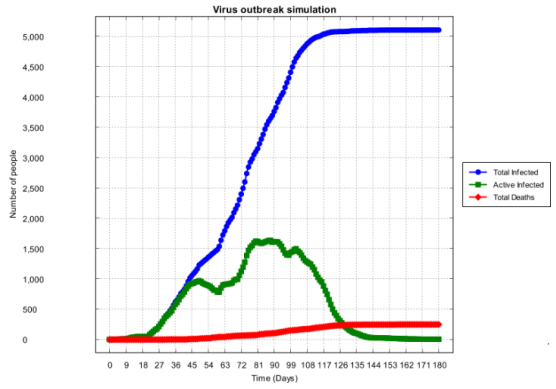


Figure 12:no precautionary measures

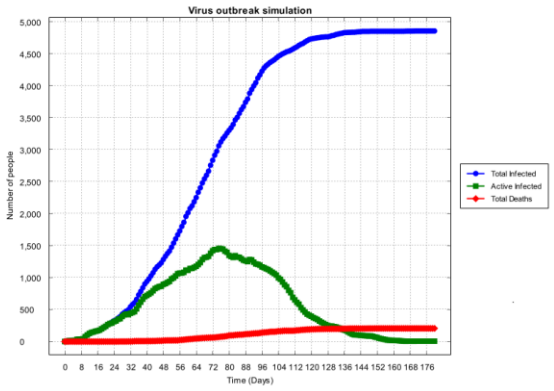


Figure 13: people maintaining social distancing

Analysis:

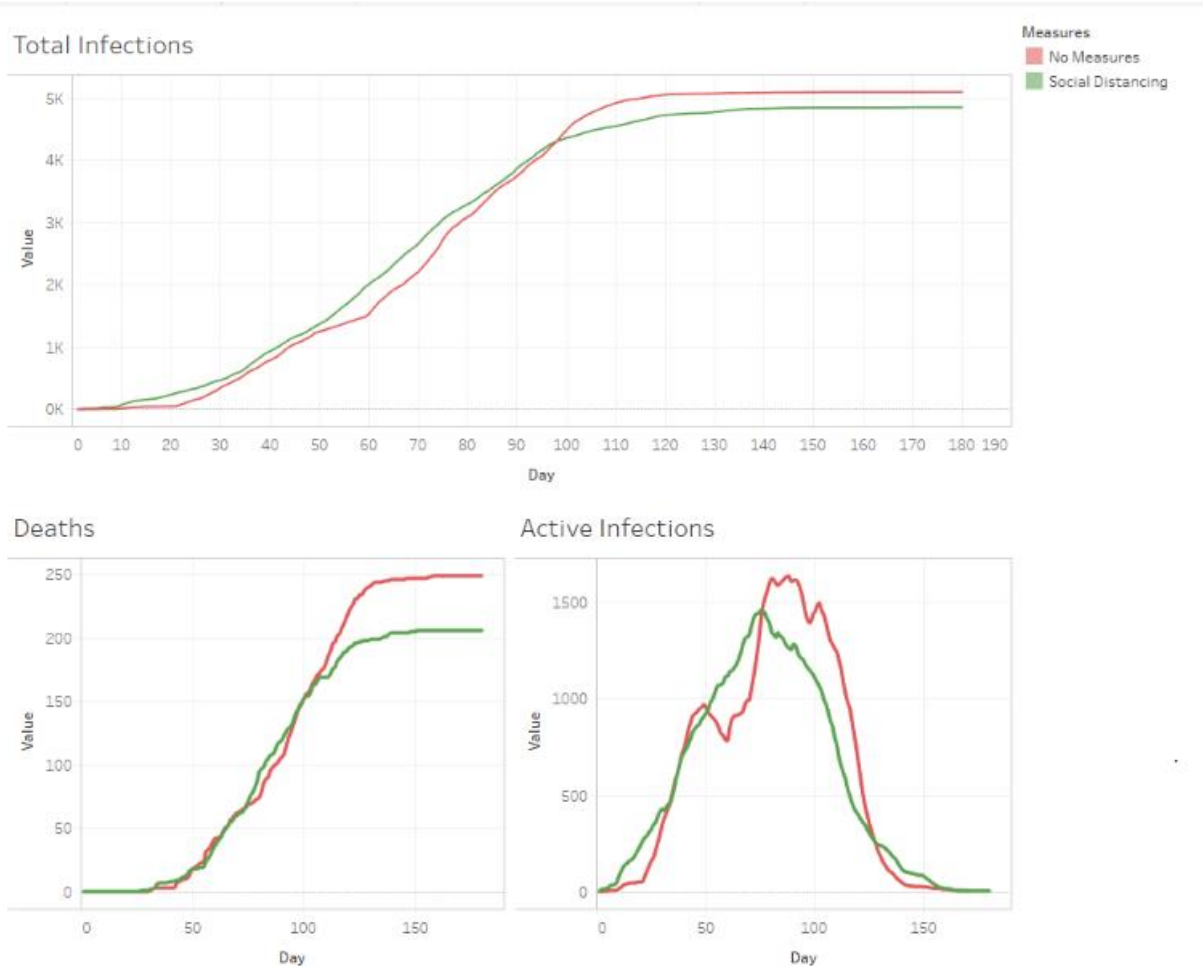


Figure 14: Comparative study

- For SARS-CoV-2, the nature of major transmission route is known to be the respiratory droplets during regular speech of the people. When we enforce people to maintain distance, it prevents the droplets to travel and infect the victim as it doesn't fall into its max travel range.
- The above simulation proves the same. The mark of infecting 5000 people on map is not reached and the total infection curves starts to flatten on the mark of 4800 people.
- The total infection curve reaches its peak of 4800 on 96th day when people maintained distancing whereas when no measures are taken its peak of 5000 was reached on 108th day.
- The social distancing thus prevents the spreading of virus by reducing R0 to 2.169348

Scenario 4. Combined effect of enforcing masks and social distancing

Result:

- Figure 15. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 16. shows the spread of SARS-CoV-2 when people are enforced to wear masks and maintain social distancing among themselves.

SARS-CoV-2 virus infects:

- 4000 people on the map in 90 days when no precautionary measures are taken and
- 3800 people on the map in 180 days when people are enforced to wear masks and maintain social distancing among themselves.

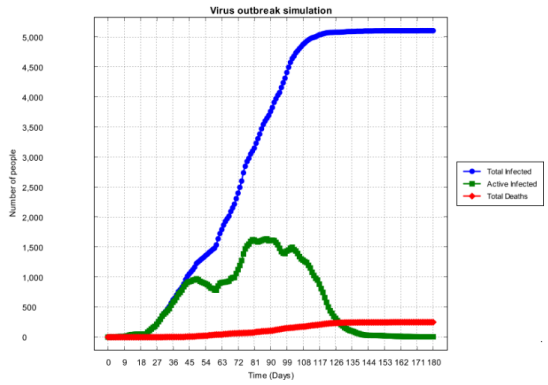


Figure 15:no precautionary measures

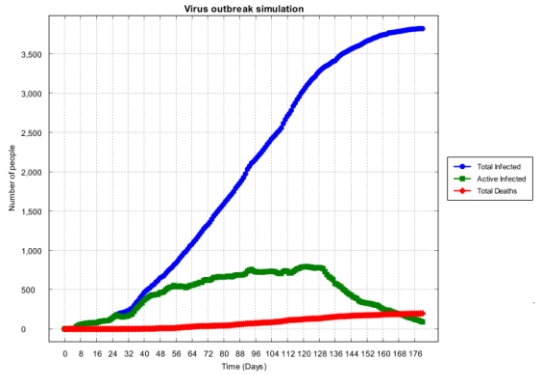


Figure 16: enforced masks + social distancing

Analysis:

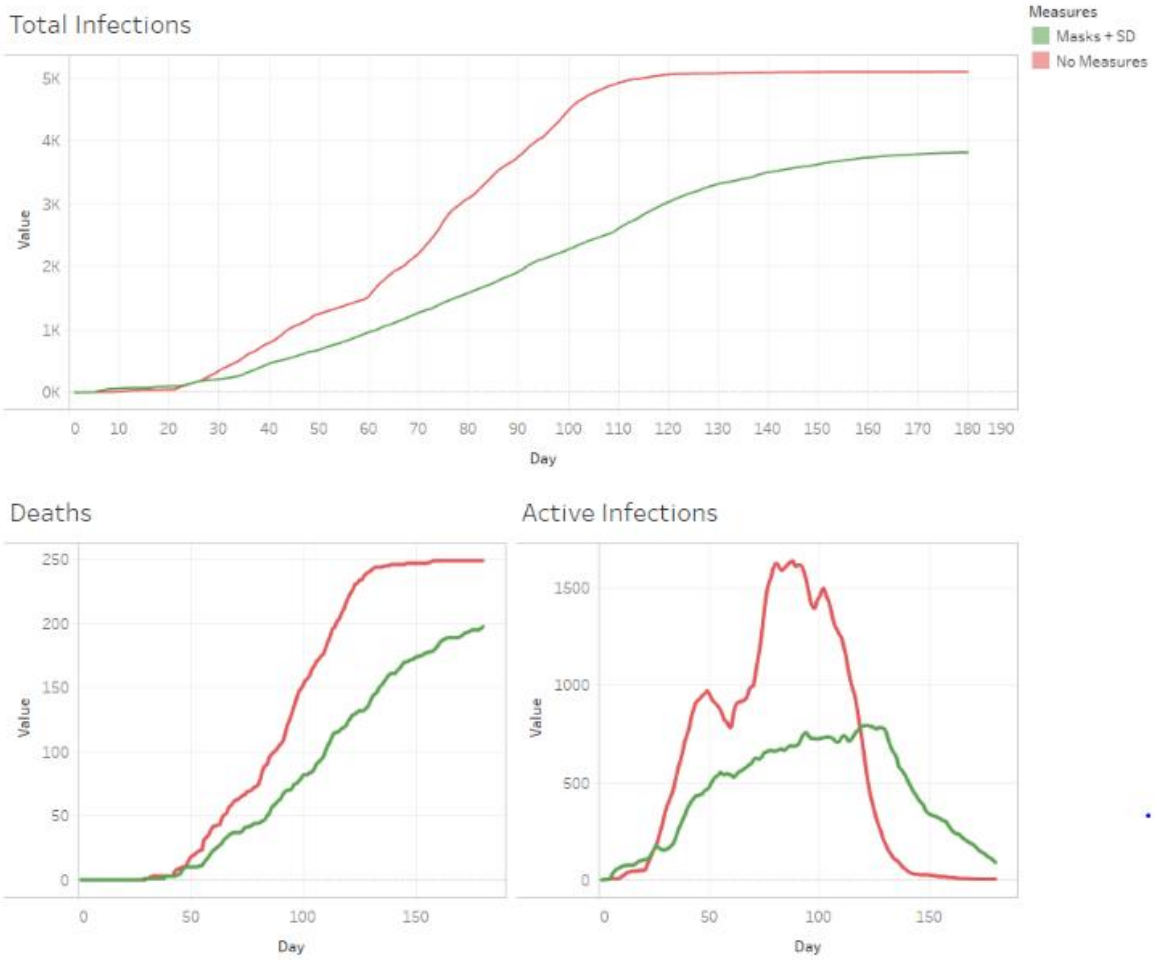


Figure 17: Comparative study

- Indeed, hygiene goes a long way.
- when people are enforced to wear masks and maintain social distancing the over-all probability/chance to catch infection reduces greatly.
- As per the simulation data, the total infection curve starts to flatten in **just half the time** compared to the time needed when there are no measures taken for controlling the spread.
- The R0 is noted to be 2.088316

Scenario 5. Occurrences of public social events (concerts, festivals, protests)

Result:

- Figure 18. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 19. shows the spread of SARS-CoV-2 when people are attending social events actively.

SARS-CoV-2 virus infects:

- 5000 people on the map in 108 days when no precautionary measures are taken and
- 5000 people on the map in just 56 days when people are attending social events actively.

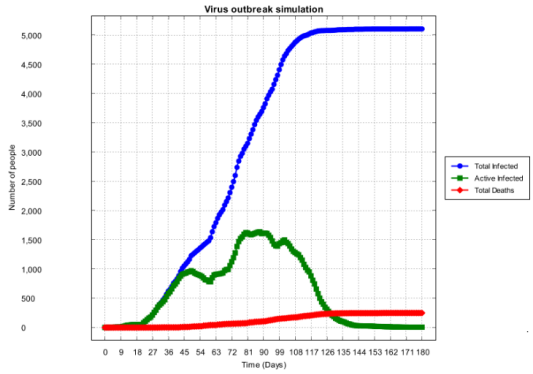


Figure 18:no precautionary measures

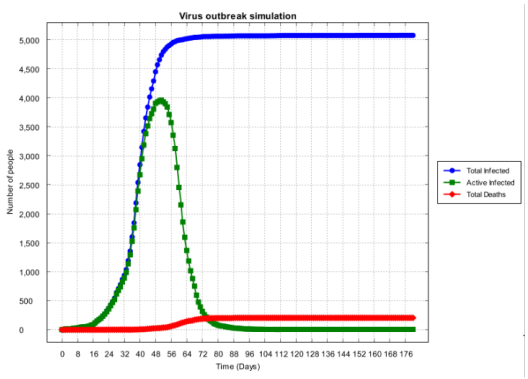


Figure 19: people attending social events actively

Analysis:

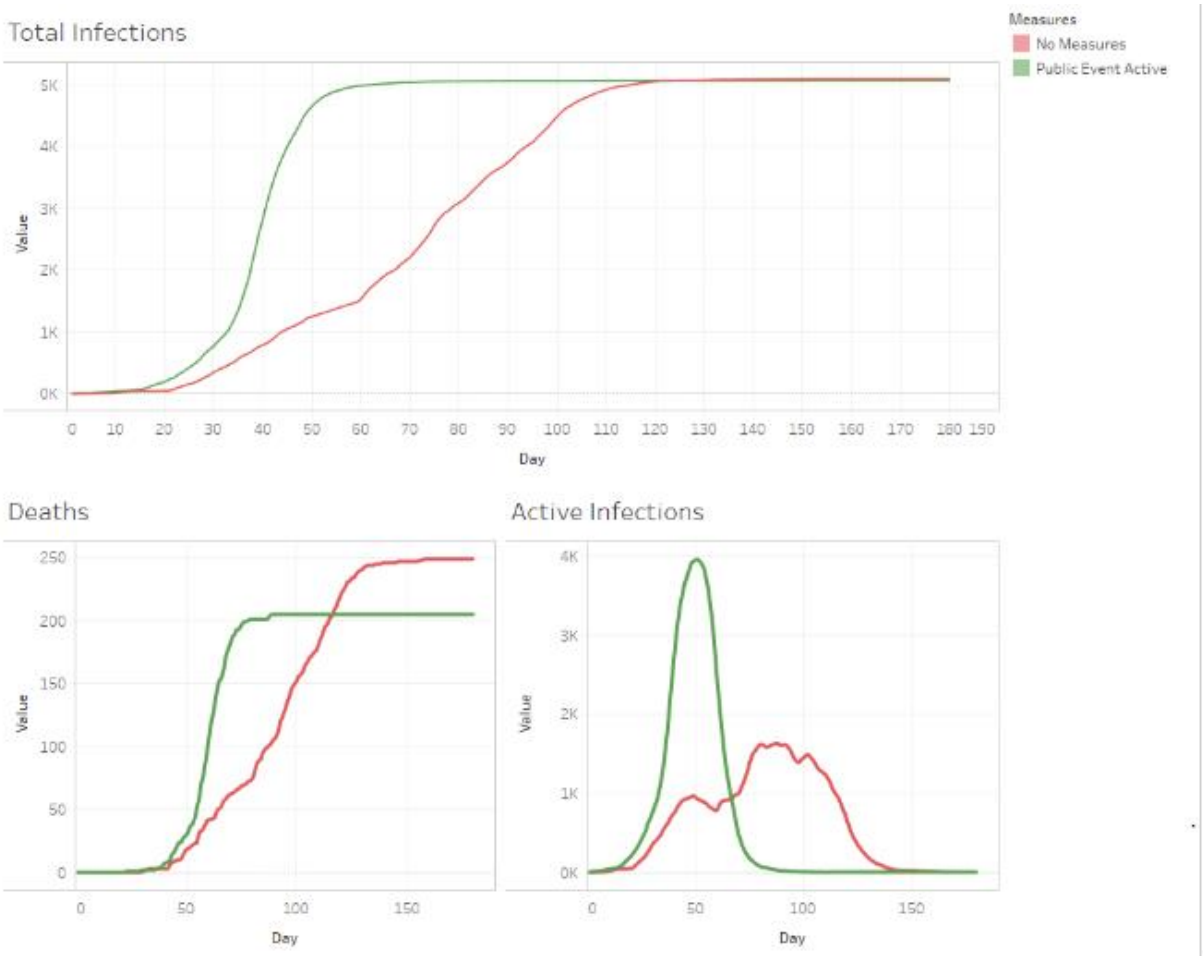


Figure 20: Comparative study

- The above simulation shows that with active social contact the mark of infecting 5000 people on map is reached in just 56 days and the total infection curves starts to flatten after the peak.
- Whereas when no measures are taken the total infection peak of 5000 was reached in 108 days.
- The active social gathering of people allows virus to reproduce twice its rate.
- The R0 in this scenario increases to 2.337172.

Scenario 6. Lockdown the city and then reopening (Lifting lockdown)

Result:

- Figure 21. shows the spread of SARS-CoV-2 when no precautionary measures are taken.
- Figure 22. shows the spread of SARS-CoV-2 when city was locked down and then partially reopened followed by complete lift of lockdown.

SARS-CoV-2 virus infects:

- 5000 people on the map in 108 days when no precautionary measures are taken.
- the total infection flattens at 3 levels-
  - 1st at the peak of 200 when an immediate lockdown was enforced on city
  - which increased to 300 people in just 2 days on partial reopening.
  - Further, the curve grew exponentially when lockdown was completely lifted and a peak 1300 total infections was reached in 27 days before it starts to flatten again for 3rd time.

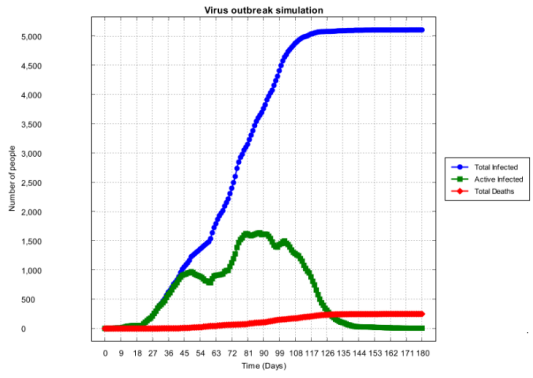


Figure 21:no precautionary measures

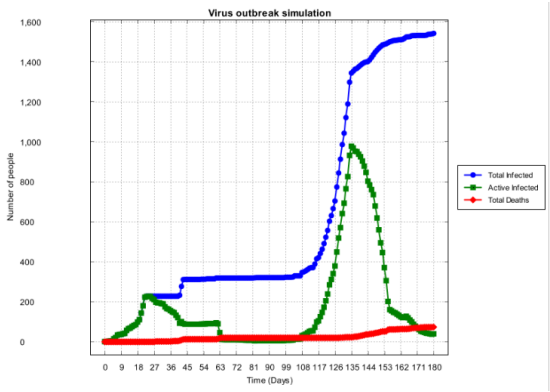


Figure 22: people attending social events actively

Analysis:

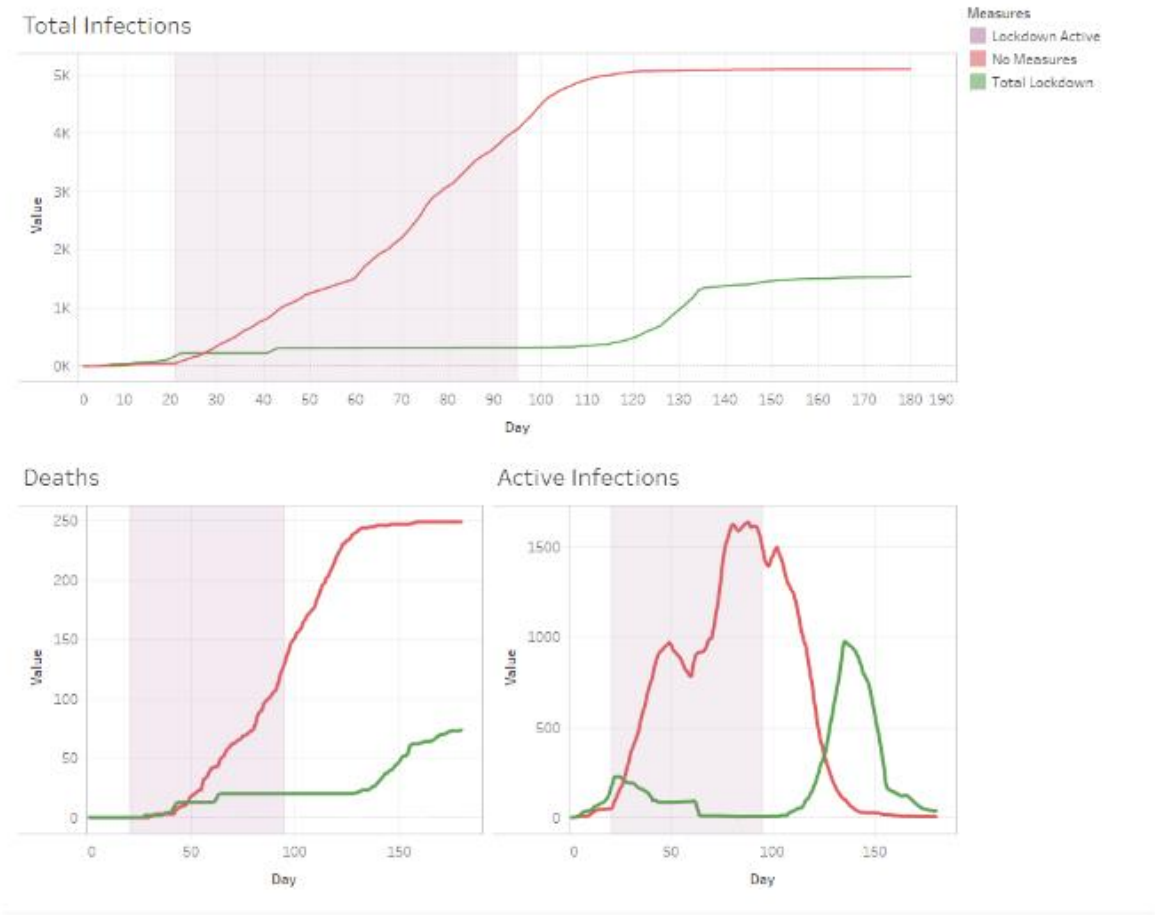


Figure 23: Comparative Study

- The above simulation shows that on imposing lockdown of total infection curves flattens immediately as less and less people come in contact. Further, on lifting lockdown the total infection increases exponentially.
- Imposing lockdown has the greatest control on spread of virus as it reduces overall exposure of people.
- It reduces the R0 to 2.292793

Scenario 7. Comparing the spread in densely(urban), moderately(sub-urban) and sparsely(rural) populated places.

Result:

- Figure 24. shows the spread of SARS-CoV-2 in an urban place.
- Figure 25. shows the spread of SARS-CoV-2 in a sub-urban place.
- Figure 26. shows the spread of SARS-CoV-2 in a rural place.

- The nature of total infection curve is same in all the 3 places but the peak of infections varies widely as the exposure of people in all 3 is directly proportional to their total population.
- The total infection peaks at:

    1. 5000 for urban,
    2. 2100 for sub urban,
    3. 1100 for rural.

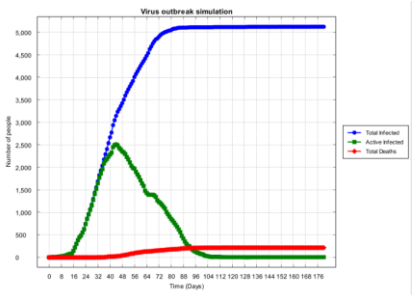


Figure 24: spread of SARS-COV-2 in urban

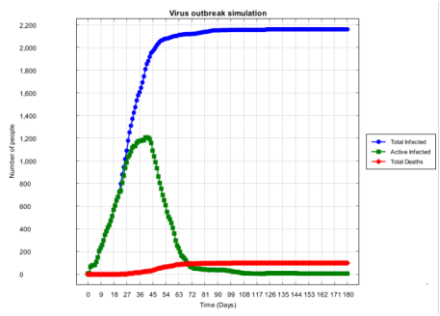


Figure 25: spread of SARS-CoV-2 in sub-urban

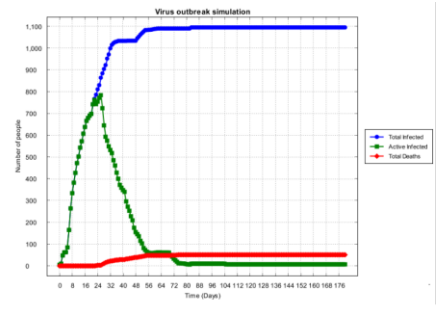


Figure 26: spread of SARS-COV-2 in rural

Analysis:

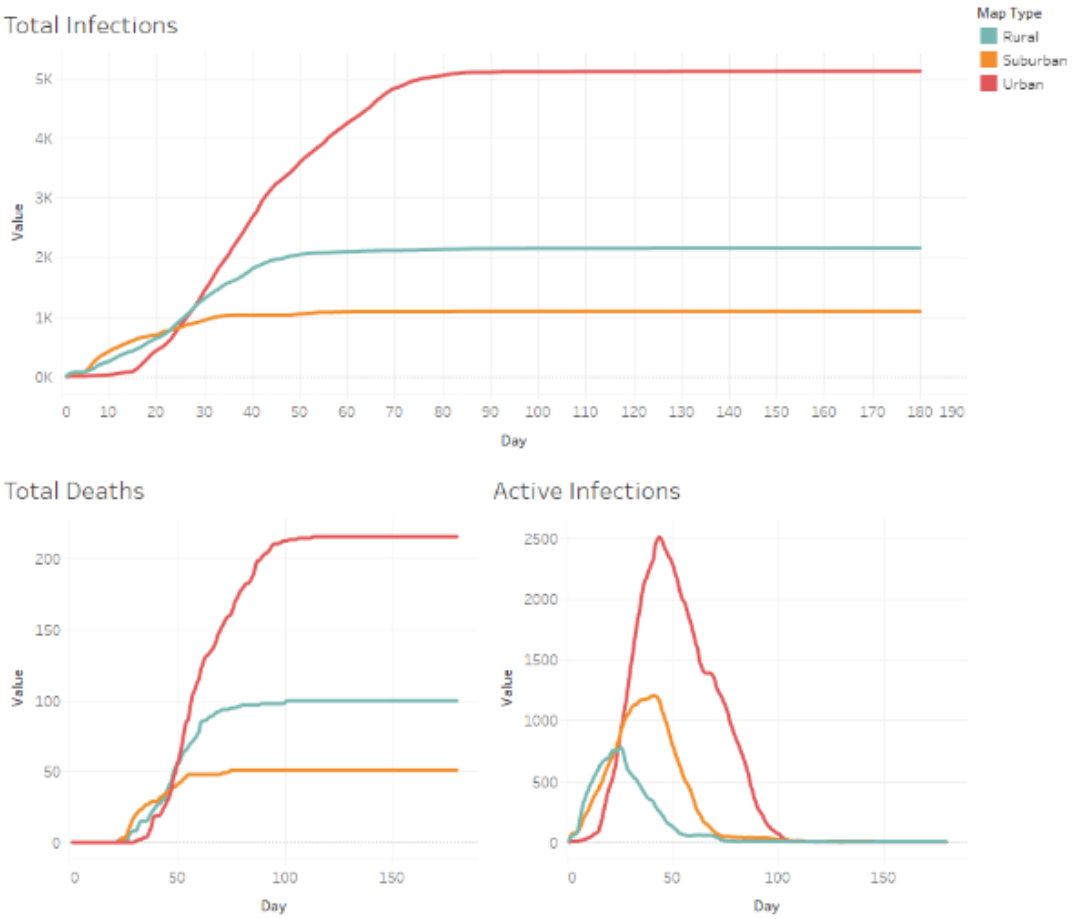


Figure 27: Comparative study

Scenario 7. Comparative analysis of spread of 3 viruses- Influenza, SARS-CoV-1 and SARS-CoV-2.

Result:

- Figure 28. shows the spread of Influenza in an urban place.
- Figure 29. shows the spread of SARS-CoV-1 in an urban place.
- Figure 30. shows the spread of SARS-CoV-2 in an urban place.

The 3 figures below provide the simulation data for the 3 viruses that have caused pandemic in the history of the world. Over the similar Map (urban) the nature of 3 viruses is shown:

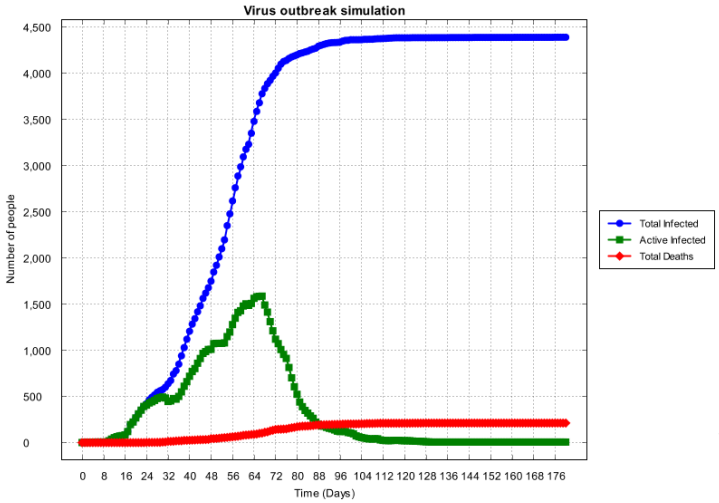


Figure 28: spread of Influenza

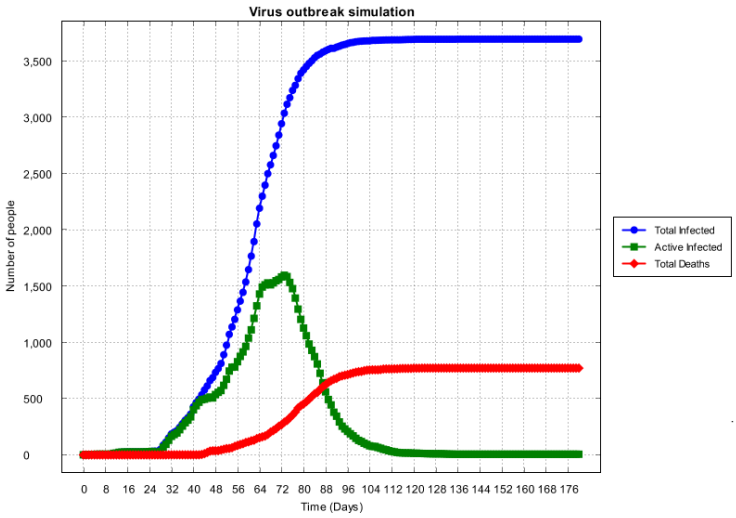


Figure 29: spread of SARS-CoV-1

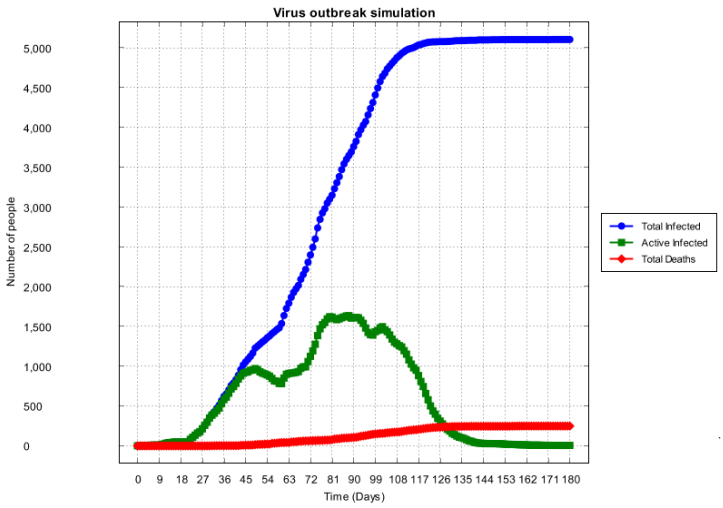


Figure 30: spread of SARS-CoV-2

Analysis:

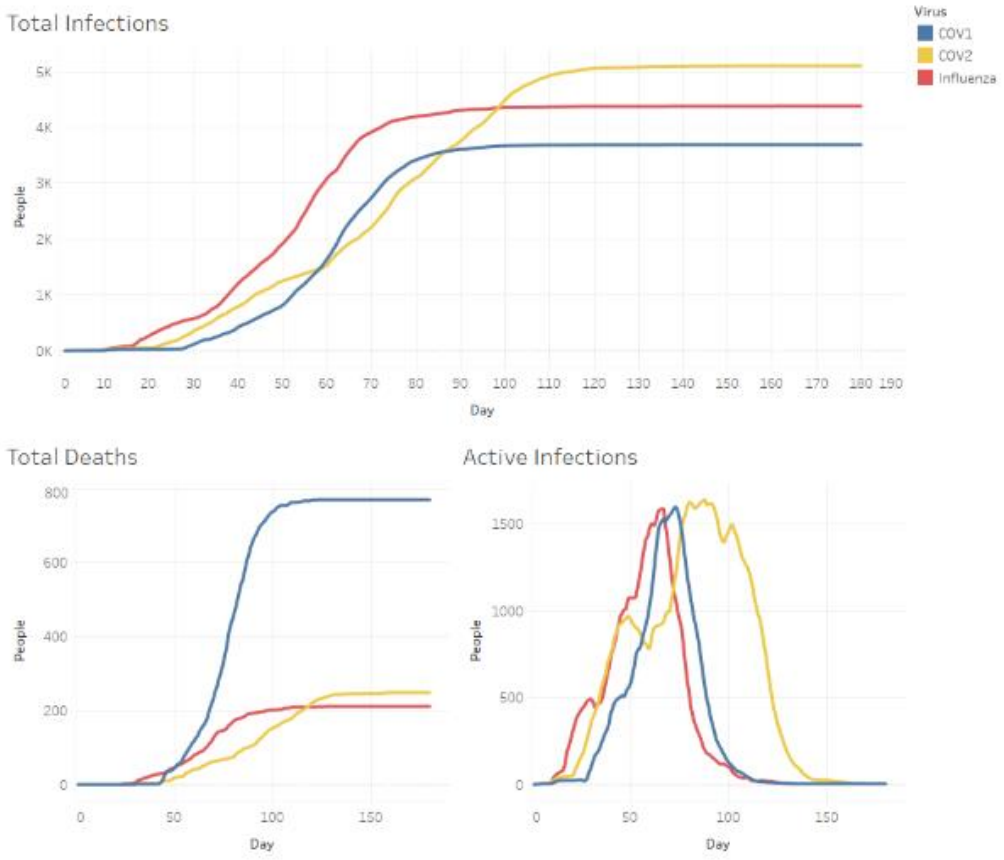


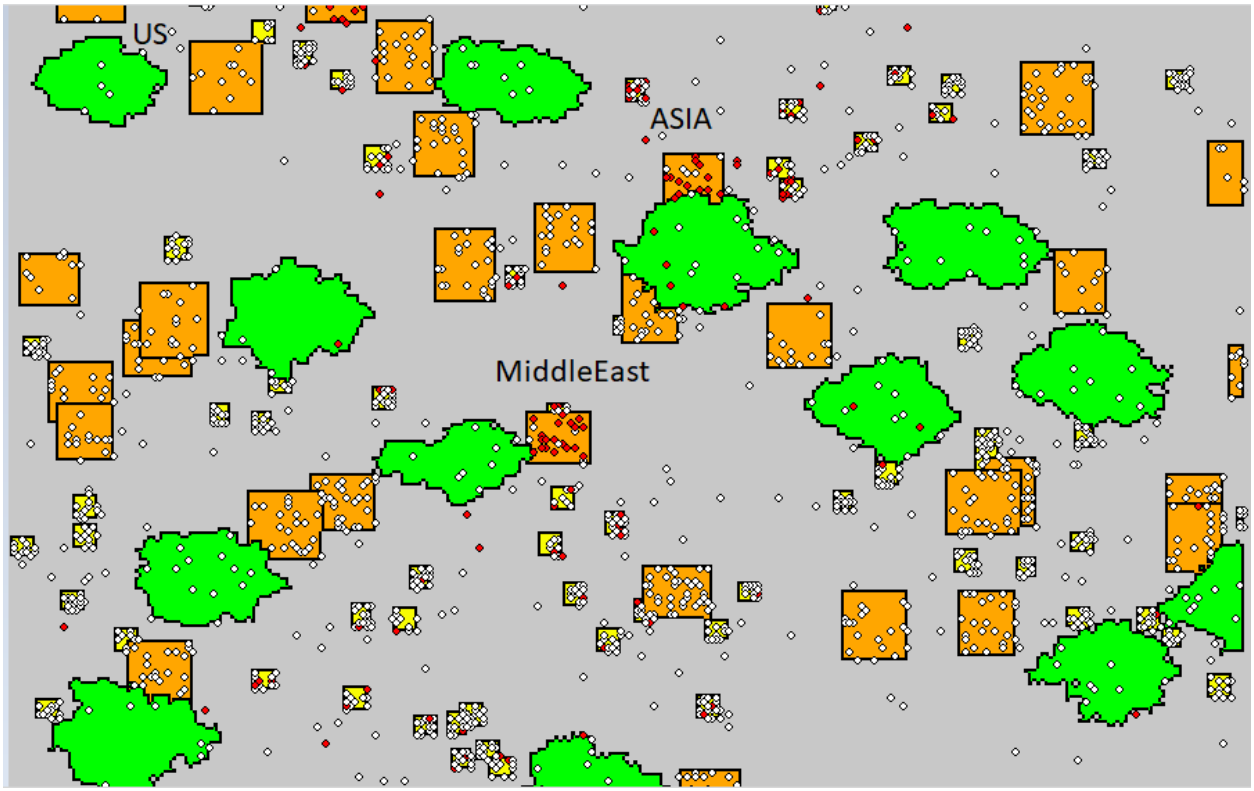
Figure 31: Comparative study

- The death rate (total deaths/total infected) has been recorded maximum for SARS-CoV-1.
- For SARS-CoV-2, the total infection spread curve has taken longest to flatten.
- The R0 noted for each of the case is as listed below:

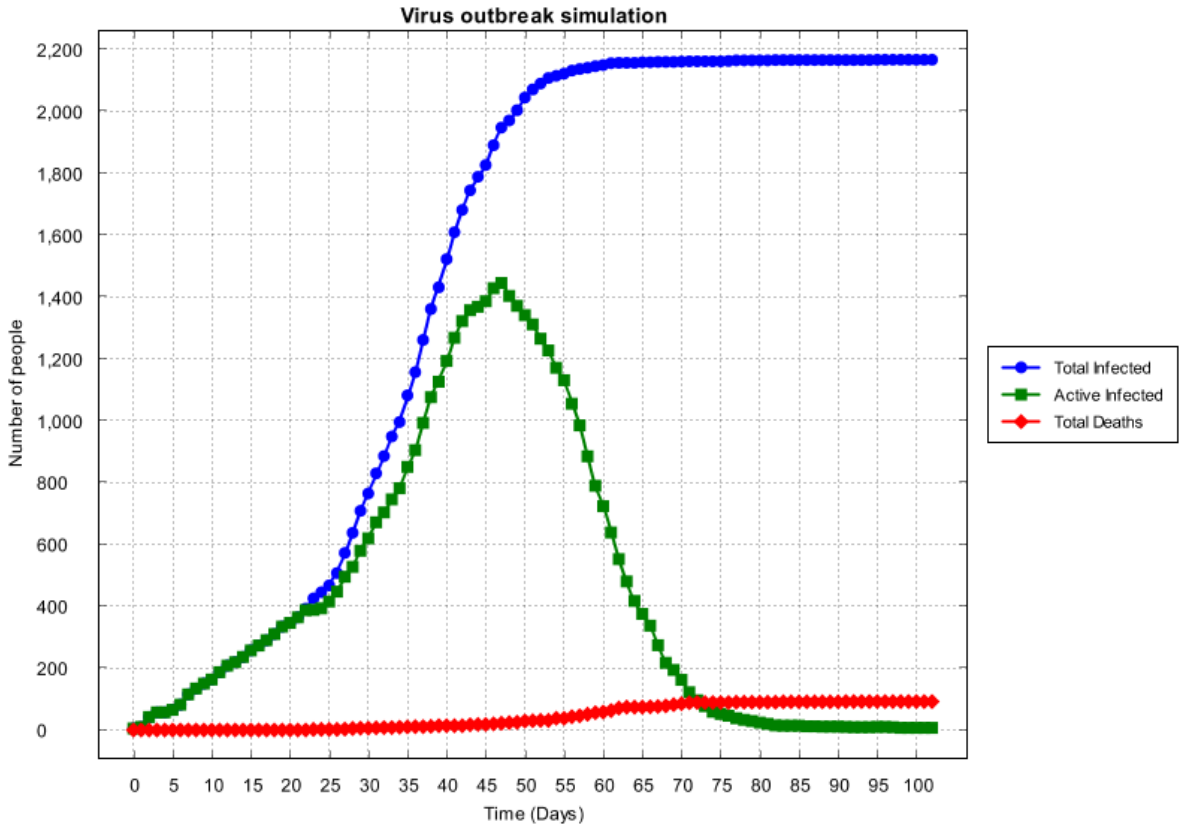
Influenza	Average (R Nought)	2.679511
	Standard Deviation	2.477552
SARS-CoV-1	Average (R Nought)	2.297196
	Standard Deviation	1.779292
SARS-CoV-2	Average (R Nought)	2.298333
	Standard Deviation	1.779978

Scenario 8: Scaling Map simulation to match the world map:

- Using our current sub-urban Map settings, we are scaling this simulation to match the world map and it depicts how SARS_CoV-2 spread across the globe initially.
- In these settings, we have assumed each of office building, house and public buildings to be the countries of that continent depicting the communities that interact with other communities and among themselves to a great extent. The (infected –in red) people moving depicts the air travel that spread the virus.
- It started in Asia and through travel it was transmitted to the other continents in ~3 months.
- Below is the screenshot when 5000 sim ticks (50 days) completed of knowing SARS-CoV-2 infection in Asia.



Analysis:



- **In initial 20 days a total of 346** infections were recorded which **increased to 1521 in next 20 days**.
- **The total number of infections increased 5 folds when there is active inter-community interaction.**
- Hence, various travel restrictions were imposed to control the virus transmissions across global communities. **Note**: we have not analysed the impact of this measure in our simulation for simplicity.
- The value of R0 i.e. virus transmissibility for this simulation is noted to be **2.13228.**

9. Conclusion:

Pinning down key factors:

To control the infection and to keep R0 in check we need to reduce the 3 factors that drive the infection rate:

- **Exposure of the people (Pe):** The SARS-CoV-2 spreads majorly through close contact among people.
  Hence, the precautionary measures that target and reduce the exposure of people reduces the infection spread greatly.
  Measures such as: **Lockdown, cancelling of public gatherings** (socially active events), travel restrictions reduce this key factor and contain the reproduction number (R0) from increasing.

- **People's chance to catch the infection (Pc):** Every human is susceptible to infection but the chance to catch the infection highly depends on their hygiene. Hygiene paves the way, here. **Measures such as: wearing masks and maintaining social distancing** can reduce the spread of virus.

- **Uncontrollable slips/fall outs (U):** There would be some asymptomatic carriers and these are uncatchable under normal circumstances and they add a significant number of infections in any populated place.
  The only measure to keep these in control is to follow **contact tracing and immediate quarantine by** being proactive.

Analysis of Growth rate:
Below formulae describes our simulation conclusions in mathematical way.

$$\textbf{Number of total infections} = (\textbf{Pc} * \textbf{Pe}) + \textbf{U}$$

As mentioned above, reduction in any of these factors will lead to the reduction of R0 value and thereby control the infection growth.

Below we have summarized the R0 value captured in each of the specific simulation.

Summarizing R for all scenarios:

	Influenza	SARS-CoV-1	SARS-CoV-2
Values of Ro transmission from different sources	2·0–3·0	2·0–3·0	2·5 (range 1·8–3·6)
Values of Ro transmission from Simulation data	2.679511	2.297196	2.298333

Scenario	Enforcing Masks	Enforcing Social Distancing	Quarantine and contact tracing	Enforcing Social Distancing and Masks	Lockdown Measures	Active Public Events
Ro transmissibility	2.158868	2.169348	2.21809	2.088316	2.292793	2.337172

10. References:

- https://www.thelancet.com/pdfs/journals/laninf/PIIS1473-3099(20)30484-9.pdf
- https://advances.sciencemag.org/content/6/36/eabd3083
- https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19-similarities-and-differences-with-influenza
- https://www.who.int/csr/sars/en/WHOconsensus.pdf
- https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30484-9/fulltext#seccestitle30
- https://www.youtube.com/watch?v=Kas0tIxDvrg
- https://www.youtube.com/watch?v=gxAaO2rsdIs