Technical University of Munich

Unity - Can We Ever Celebrate Again?

Fabian Höltke, Maximilian Geitner, and Sven Andabaka

December 23, 2021

1 Introduction

In this experiment, we want to find out if we can ever celebrate the Unity in the MI Magistrale again. We want to find out what would have happened if these celebrations took place during different periods of the Corona pandemic and what measurements need to be taken to prevent dangerous spreading events.
Our party layout and guest movements for this scenario are based entirely on memory and information publicly available at www.unity.bayern. Looking at the pictures and especially the "beer counter graphics" from the earlier parties, we estimate that around 2000 students attended the parties simultaneously.

2 Running the Simulation

Please clone the following repository containing our modified version of The One:
https://github.com/fabianh001/the-one.

To run the different scenarios, one has to select a predefined configuration file in the folder **./unity_party_scenarios** and load it in The One simulator.

After the simulation was successfully completed, three reports about the location of guests, movement, and spreading are created. These reports can be found in the root directory's folder **./reportsUnity<ConfigurationName>**.

3 Map Creation

To design our scenario map, we used the images and experiences from the last Unity 2019. The images are publicly available under www.unity.bayern.
As you can see there, Unity took place mainly in the front part of MI Magistrale and the beer garden behind MI Bistro.

Unity was divided into several dance, bar and service areas. In the dance areas, which were located in a single room or in the center of the MI magistrate, people were free to dance and move around. The bar areas were spread throughout the place. There, guests could relax from partying and enjoy food and drinks. Lastly, there were service areas, such as the outdoor chill area, bathrooms and the wardrobe.
Table 1 gives a brief overview of all areas that could be found during a Unity party. Figure 1 shows where each area was located on the map.

3.0.1 Implementation

To design a meaningful map for the Unity scenario, we first searched for a map section with the MI Magistrale in Garching on OpenStreetMaps and downloaded it. This map was then edited using JOSM, where we removed all paths, nodes and other waypoints not related to the MI building itself.
After that, we used the tool osm2wkt[1] to convert the file to the wkt format, which is also used by The One simulator. This file is then used as background and outline for the MI building.

[1]https://github.com/julianofischer/osm2wkt

Figure 1 Map of the Unity party and its areas

Unity Areas		
Dance Areas	Food Areas	Service Areas
Metal Bunker	Beer Bar	Outdoor Chill Area
Techno Bunker	Cocktail Bar	Side WC
Main Stage	Shot Bar	Main WC
	Pizza Bar	Entry
	Shisha Bar	Exit
		Wardrobe entering
		Wardrobe leaving
		Waiting Queue

Table 1 Different areas within the Unity map

To model the different Unity areas as mentioned above, we used several polygons. Movement behavior within area polygons differ between dance and food/service areas. Within dance area polygons, nodes are allowed to move freely around. Within food and service areas, nodes are stationary and are not allowed to move.
In order to retrieve the exact locations of those polygons, we modified the implemented mouseEventListener functions of the One simulation. This allowed us to retrieve the exact coordinates of our areas in the playfield. Those coordinates were then saved to the area's state.
To prevent nodes from moving through walls or along incorrect routes within the MI building, several "door coordinates" were implemented. These door coordinates were then added to a single node path when a movement was to be performed between two separated and non-adjacent areas. For example, this included movements between indoor areas, such as the "Main Stage" and outdoor areas, such as the "Outdoor Chill Area".

4 Movement Model

All Unity guests are following the same movement model. We created the 'UnityGuestMovementModel', which is based on stateful movement, time-variant movement and polygon movement.
All nodes have an internal state, which resembles one of the Unity areas defined in tab: areas and therefore the node's current location. Each of the areas/states is modeled as an own entity extending the abstract class 'NodeState'. Therefore, each state needs to implement different attributes, namely:

- the name of the respective area
- the polygon of the respective area
- a flag if the node should move according to the random waypoint model inside of the polygon
- the minimum time the node has to spend in this state
- an array with probabilities moving to other areas

4.1 State Selection

Each node is initialized in the Queue state. Each time a new path is picked, the current state of the node is updated. The selection of the next state consists out of 3 major steps:

1. check if minimum time passed by
2. check event-based state changes
3. get next state based on a transition table

4.1.1 Minimum Time Passed

Each node saves the timestamp when it enters a new state. If the time spent in the current state is less than the minimum time defined by the current state, the current state is not changed. Otherwise, the algorithm continues with the selection based on the current time.

4.1.2 Event-Based State Changes

There are multiple interesting events during the Unity which occur in specific time frames. Each of this events has a probability assigned, which represents the likelihood that this node will take part in this event. Table 2 gives an overview all events.
Based on the current time of the simulation (where 0 denotes 20:30) the next state is chosen according to the mentioned probabilities.

4.1.3 State Transition Table

If none of the first two mechanisms (Minimum Time Passed, Time-Based State Changes) are used for a state change, a state transition table is applied (Table 3).

4.2 Path Selection

After a state update, the node calculates its path to the new state. If the updated state is the same as the previous state, nothing has to be done. However, there are exceptions for states representing the dance areas. The nodes move according to the random waypoint model within these dance floors to simulate dancing. If the updated state is new, the nodes choose a random waypoint within the polygon of the new state.
Additionally, an optional waypoint representing doors can be set. This makes the movement between non-adjacent areas more realistic, such as movement between indoor and outdoor areas.

Time	Location	Probability	Description
20:30-21:00	Queue	100%	The Unity opens at 21:00. So all nodes have to wait till at least this time in the queue.
21:00-22:00	BeerBar	15%	There is a beer happy hour. So there is a higher chance of people visiting the beer bar.
21:00-22:00 and 03:00-03:30	PizzaBar	10%	The likelihood of people getting snacks is higher in the beginning and end of the party.
01:00-01:30	Exit	25%	The last regular subway leaves at 1:30. So the chance of people leaving is higher during this time.
3:30-5:00	Exit	90%	The party closes at 4:30 so people tend to leave during the last hour.

Table 2 Events during the Unity Pary

Places	Queue	Entry	Wardrobe	Main Stage	Outdoor Area	Metal Bunker	Techno Bunker	Cocktail Bar	Beer Bar	Shot Bar	Shisha Bar	Pizza Bar	Side WC	Main WC	Exit	Wardrobe before exiting
Queue	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Entry	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wardrobe	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Main Stage	0	2,5	65	73	20,6	20	20	78	76,8	48	19,9	30,8	55	61,4	0	0
Outdoor Area	0	1	10	8	42	10	10	7	7	30,8	25	32	5	7	0	0
Metal Bunker	0	0,5	2,2	3	3	50	2	3	4	4	5	7	1,8	4	0	0
Techno Bunker	0	0,5	3,5	4	5	3	50	5,3	4	4	1	3	4	2	0	0
Cocktails Bar	0	0,5	6	3	3	2	5,8	2	1	1	2	3	15	5	0	0
Beer Bar	0	2	8	4	5	8	5	1	2	1	2	8	10	15	0	0
Shot Bar	0	0,5	3	1,5	8	1,7	2	1	1	3	2	5	4	2	0	0
Shisha Bar	0	0,5	0,1	0,1	0,3	0,2	0,1	0,1	0,1	0,1	35	0,3	0,1	0,1	0	0
Pizza Bar	0	1	0,2	0,3	8	1	1	1	1	3	5	7	5	3	0	0
Side Wc	0	0	0	1	2	1	2	1	1	2	1	2,8	0	0	0	0
Main Wc	0	1	2	2	3	3	2	0,5	2	3	2	1	0	0	0	0
Exit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100
Wardrobe before exiting	0	0	0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,5	0	0

Table 3 State Transition Table (all probabilities in %)

Thus, the node moves with respect to the previously mentioned state table, time-varying dependencies, and polygons as long as it has not left the party. Once a node enters the initial state, it moves to a fixed coordinate representing the subway station. Once at the subway station, the node's movement is disabled.

5 Spreading Model

The network model simulates the spread of the coronavirus between guests at the Unity party. In this scenario, there are a fixed number of infected hosts at the beginning of the event, which spread the virus to neighboring uninfected hosts at regular intervals throughout the event. The simulation must deal with hosts in the magnitude of $\mathcal{O}(1000)$, so we decided to simulate only direct transmissions of the virus. Any indirect indoor airborne transmissions or infections from sharing food, drinks, or equipment in the Shisha bar are not explicitly modeled in our simulation due to increased computational and implementation costs. Instead, the risk is modeled indirectly by increasing the viral load of hosts in these areas.

5.1 Scenario Configuration

For our infection model, we have defined a set of parameters as weights and simulated the viral load of hosts as particles. An infected host will broadcast messages to neighbors that indicate a direct transmission. In the default setting without weights, non-infected hosts receive messages with a payload equivalent to 1000 particles per second from neighbours. After every 1000 particles, the receiver invokes an infection event and gets infected with a probability p.

This type of modeling ensures that people can be infected with even low amounts of particles received, but are more likely to be infected with more contact with infectious neighbors. An article from Lelieveld et al. describes the risk of an infection from one single COVID-19 virus to be approximately 0.22%. In experiments, values between 0.0001 and 0.001 seemed suitable for the probability p and led to believable simulation results [3].

In addition to the risk of infection p for each event, we have weights for the distance between hosts, the vaccination status of both hosts, and their current locations that affect the final viral load. A common global measure to prevent virus spread is to maintain a distance of 1.5 meters from other people in public. Therefore, we reduced the risk of infection at greater distances between hosts [5]. In the simulation, only neighboring connections with a maximum distance of 3 meters are considered for virus transmission, and the highest viral loads are transmitted only within a distance of one meter; at larger distances, the received viral load for the recipient decreases sharply.

Another important factor is the guest's vaccination status. There are reports of a lower viral load spread by vaccinated individuals compared to non-vaccinated individuals [2] [6] [4].
To model the viral load on the transmitter side, we reduce the amount of sent particles in the simulation when the sender is a vaccinated person. On the receiver side, we apply a similar scheme that should slightly reduce the risk of vaccinated individuals contracting the virus.
Additionally, we added a third group for individuals with booster vaccination with even better parameters to avoid infecting individuals.

This simulation of the Unity party does not include the modeling of guests wearing masks, since the party is very unlikely to exist with mask mandates. A precedence case was set with the Christmas University party in 2021, which would have required a mask mandate according to regulations in Bavaria. As a consequence, the organizers decided to cancel the event in advance[2].

In the party setting, the risk for infections is significantly higher compared to other settings like in schools, offices or at choir practices [3]. People on the dance floor are quite active and are very likely to sing with the music, which leads to a higher infection risk in areas like the main stage, the techno and the metal bunker. We also increased the risk for infection at the Shisha Bar due to the close contact between guests and sharing of utensils between each other. In other outdoor areas, we halved the risk due to the better airflow compared to the indoor areas.

Not only the virus transmission can be configured, but also how many and what kind of hosts visit the party. There are six groups in total that can be configured. They are distinguished by their infection status at the beginning of the simulation (infected or not infected) and by their vaccination status, which can be none, fully vaccinated or vaccinated with a booster vaccine. In the configuration file, the first letter of the group prefix indicates the infection status and the second letter indicates the vaccination status. Newly infected individuals at the event do not change their group prefix because the event does not last long enough for the newly infected to become infectious themselves.

5.2 Implementation

The implementation of the spreading model can be divided into three main parts: network model, the report class logging infections and the configuration file. We replaced the EpidemicRouter and the InfectionSimpleBroad-

[2]https://www.sv.tum.de/sv/das-machen-wir/veranstaltungen/weihnachts-uniparty/

castInterface with custom solutions (CustomEpidemicRouter.java and InfectionSimpleBroadcastInterface.java), which only send messages from infected people to non-infected hosts nearby.

5.2.1 Configuration File

The amount of infected people is specified in the configuration file, there are six different host groups with active movement models that we want to observe and later evaluate. Additionally, the weights discussed in subsection 5.1 can also be configured here.

5.2.2 Transmission of Messages

All infected people regularly send out messages to all non-infected people within a three-meter radius. Currently, the update interval is set to 5 seconds. There is a trade-off between the performance of the simulation and tracking close contacts precisely. The message size is set to 5000, which represents 5000 particles or the default amount of 1000 particles per second. The interfaces immediately exchange messages between each other, the actual received amount of particles is only processed in the InfectionReport class. That means, that the effects of distances and communicating with multiple people are not part of the designed network model.

5.2.3 Infection Report Processing

The receivers interpret the message as close contact and evaluate the message in the InfectionReport class by applying all discussed weights in subsection 5.1 and triggering eventual infection events. After processing the message, it is immediately dropped, because we only model direct contact with infectious people and want to avoid flooding the network with too many messages.
Furthermore, connections between people with the same infection status must not be established in this model and led to a speed-up in large crowd simulations with 1000+ hosts by at least the factor 10.

5.2.4 Aggregation of Data

There is an evaluation in the InfectionReport class after the simulation did complete, which provides additional aggregated data. This includes data such as the infections per time interval, the vaccination status of the transmitter and receiver and the area in which the infections took place.

6 Evaluation

In this chapter we evaluate our findings we got from simulating various Unity Party Scenarios.

6.1 Analysis of Movement Model

After each run, two different reports are generated to analyze the movement data of each host, namely the **UnityMovementReport** and the **HostLocationReport**. The movement model is the same for each scenario. Therefore, we will not distinguish between different scenarios.

6.1.1 UnityMovementReport

This report is a MovementListener report and computes how much time each node spends in each area. To do this, it listens for each node's movements and saves the time that has elapsed since the last run and in which area the last time frame occurred. When the simulation is completed, the accumulated times for each node are logged.

Figure 2 shows resulting average times from the UnityMovementReport. It is quickly identifiable that nodes spent most of the time on the Main Stage. This is the largest area and is expected to have the most people. The time spent in the queue is also quite high since the nodes start in the Queue state and wait in this state until they enter the Unity. This is also to simulate the behavior of people who arrive at the party later.

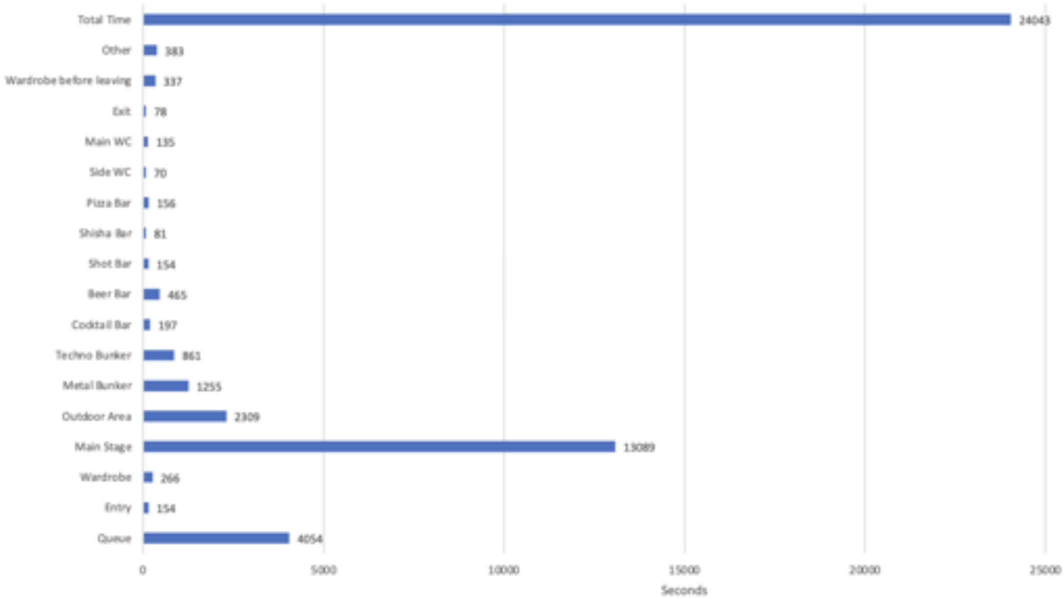

Figure 2 Average Time Nodes Spent in Areas

The average time spent in outside areas at 80 minutes is also quite realistic, as people tend to go outside to take a short break and get some fresh air, but also to eat pizza or have a shisha from the bars outside.
The average time spent on the techno/metal dance areas is quite low, as these places tend to be visited by people with more specific tastes. This results in fewer people visiting these areas but staying there for quite a long time. Bars and restrooms are visited for short periods of time, but more frequently. The beer bar is visited more often than other bars. People stay in the wardrobe area for about 5 minutes because they have to wait in line there.

6.1.2 HostLocationReport

This report is an UpdateListener report and calculates for each time step how many nodes are currently in which area. To calculate this, it iterates over each node, identifies its location and increases the counter for the respective area. The report then writes the count for each area for each time step.

Figure 3 shows the amount of people in each area per time-frame. It is quickly visible that once the Unity opens its doors at 21:00, people start to enter the party in waves. It can be also seen that not all guests arrive at once, some arrive at the party later. Only at 22:30 around 90% of the guests are at the party. Afterwards, it can be seen that about 2/3 of all guests are on the Main Stage party area, while about 12% of the people remain at the outdoor areas. Both, the Metal and Techno Bunker host about 5% of the guests. The Beer Bar holds an average of 35 guests during the main time of the party (22:00 - 01:00). The other beverage areas hold between 15 and 20 guests during this time as these are smaller. About 20-25 people are occupying the two toilets during the main time. Between 1:00-2:00 about 40% of the guests leave the party as the last subways leave Garching at this time. The other guests leave the party between 3:30 and 4:30 as the first subways run again and the party ends.

6.1.3 Evaluation of the Mobility Reports

Both, the results of the HostLocationReport and the results of the UnityMovementReport match in regards to the time spent in areas and the total number of guests in the respective areas. The Main Stage is the area where guests spend their most time and therefore also where most of the nodes are. Side stages as the Techno and Metal Bunker are more specific and therefore fewer people visit these places and they do not stay there as long. Other areas have a more specific purpose, e.g. bars offer drinks, so people tend to spend only as much time as needed in these areas. This movement seems very realistic and confirms our own experience of the Unity.

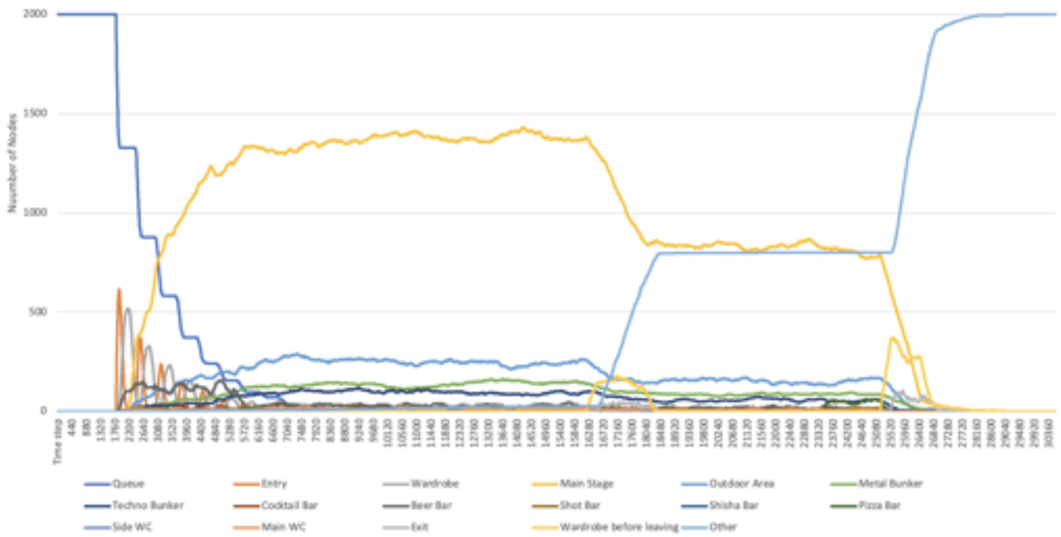

Figure 3 Number of nodes per area per time-frame

However, some problems in the mobility model aroused. One problem is that all nodes have to spawn right from the beginning of the simulation and therefore all nodes have to spend a long time in the queue. Also, very few guests visit the Shisha Bar, as this area is a very specific place. This leads to the average number of people being in this area lower than in real conditions. One last problem of the mobility model is that people leave the party in waves instead of a more linear behavior.
All of these problems could be tackled by adapting, testing and fine-tuning the state table (see Table 3) and the time-variant behavior (see Table 2). This would lead to even more realistic movements.
In conclusion, we can say that the created mobility model could be a bit more fine-tuned to ensure very realistic movement. For our modeling purpose, the movements are accurate enough.

6.2 Analysis of different spreading scenarios

We have selected six scenarios that could have happened at a Unity party. For all scenarios, we assume that 0.5% of the guests are infected. For 2000 people, this would be around 10 infected people, if no counter measurements are taken.
Within these six different scenarios, we can change the parameter of the probability p to get infected after 1000 particles. We will use the proposed value in [3] of 0.22%. For the last Omicron scenario, we increase the infection probability to 0.44%.

We apply the same weights to all tested scenarios. The distance weights are 1.0, 1.0, 0.2 and 0.01 for the distances 0 meters, 1 meter, 1.5 meter and 3 meters respectively with linear interpolation in between. The sender side has the weights 2.0, 1.6, 1.4 for non-vaccinated, vaccinated and booster groups. The receiver side uses the values 1.2, 0.8, 0.2. We assume that vaccinated hosts spread less viral load and that vaccinated guests have better protection against infections.

To evaluate the different scenarios, each scenario was run 3 times and the average values between these runs were taken.

6.2.1 Unity 2020

First, we want to establish a baseline of only non-vaccinated guests at a fictitious Unity event in 2020. During this time, no vaccine or any other measurements against Covid-19 were yet available. If a Unity in this time would have taken place with 2000 guests, a total of 540 guests would have been infected after the event, or around 27% of all attendees.

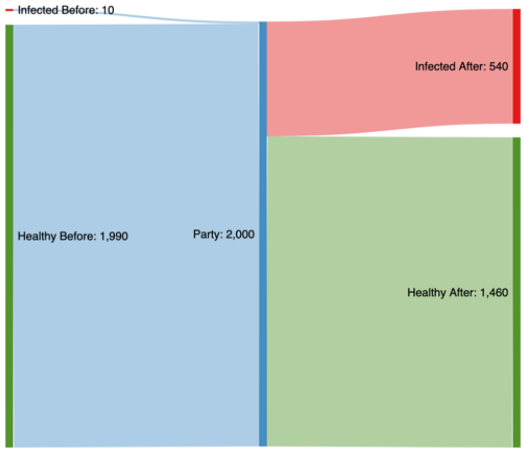

Figure 4 Infections before and after the party - No active measurements

With a total of 455 new infections, most virus spreading took place on the party's main stage. Other dance areas, such as Techno and Metal Bunker, are the other major spreading events.
Even though the risk of getting infected in the Shisha bar was implemented very high, only 2 infections were reported in one out of 3 runs for this area. The chances of an infected guest entering the Shisha area are significantly lower than in other areas.

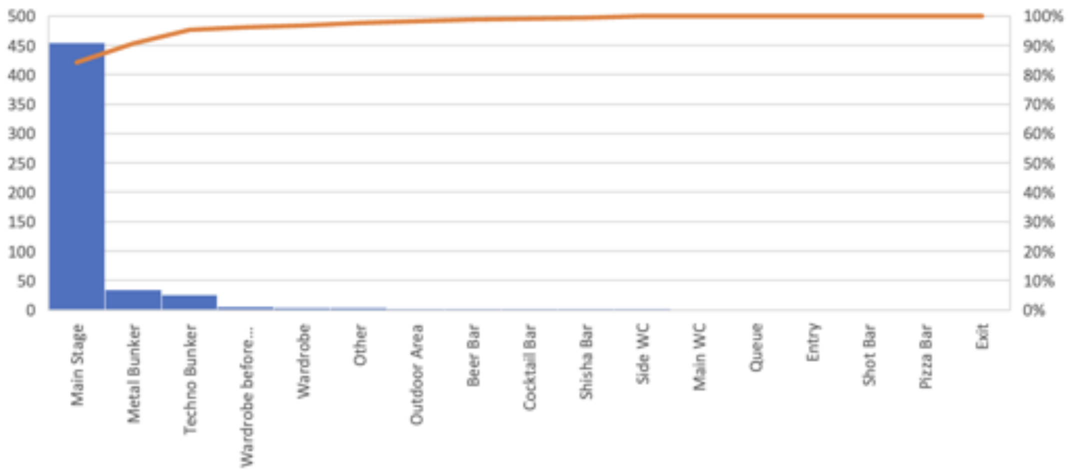

Figure 5 Places where new infections were reported

Most infections took place between the second and third hours of the party. The time before midnight is usually the busiest during the whole event. Most guests have arrived at the party and not many guests left. After midnight, the infection rate seems to follow a decreasing trend, as people start to leave the party.

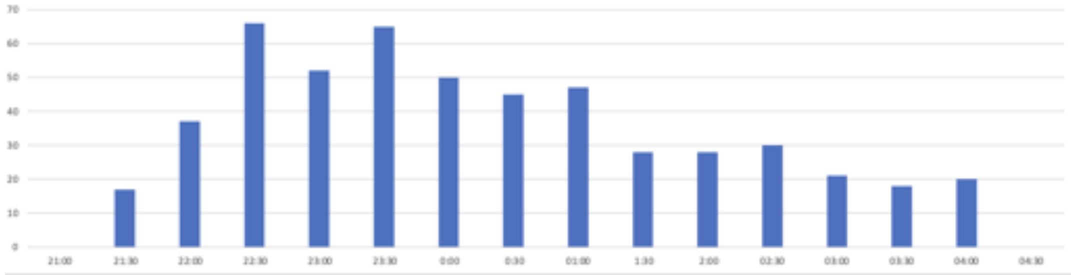

Figure 6 New infections per 30min party intervals

6.2.2 Unity 2021 - 3G, no booster

For the following year, This scenario would describe an Unity event as if it was in July 2021. The 3G rule (rapid-test, vaccinated, recovered) is applied for all bigger events during this time. We assume 60% fully vaccinated guests for the event. This is a very optimistic assumption, as in reality only about 40% were fully vaccinated and 60% of eligible individuals were vaccinated once.

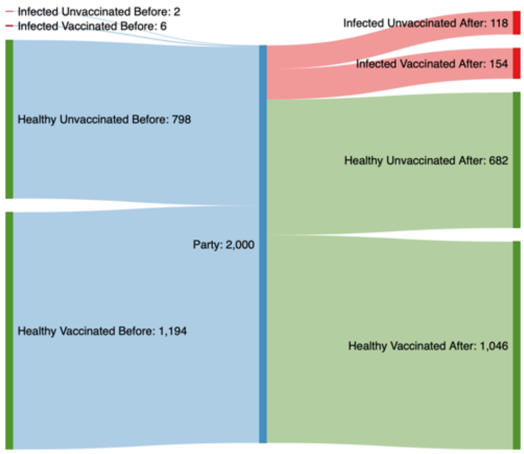

Figure 7 Infections before and after the party - 3G

If a Unity in this time would have taken place with 2000 guests, a total of 264 guests would have been infected during the event, while about 56% of these infected guests would be vaccinated (see Figure 7). The infection rate for unvaccinated people is about 15%, while the one for vaccinated people is about 12%. This shows that vaccination does only help a bit in preventing the infection itself.

6.2.3 Unity 2021 - 3G plus

This scenario would describe an Unity event as if it was in July 2021. The difference in this scenario is that we apply the 3G Plus rule instead of just the 3G rule. This means unvaccinated guests need to do a PCR test instead of just a rapid test, leading to no infected unvaccinated guests as the PCR test should exclude all positive cases. We again expect to have a vaccination rate of 60%.

If a Unity in this time would have taken place with 2000 guests, a total of 45 guests would have been infected during the event, which is significantly less than the scenario without the PCR tests (see Figure 8). The infection rate for unvaccinated people is lower than 3%, while the infection rate for vaccinated people is even less than 2%. This means that PCR tests are very effective against super-spreading events.

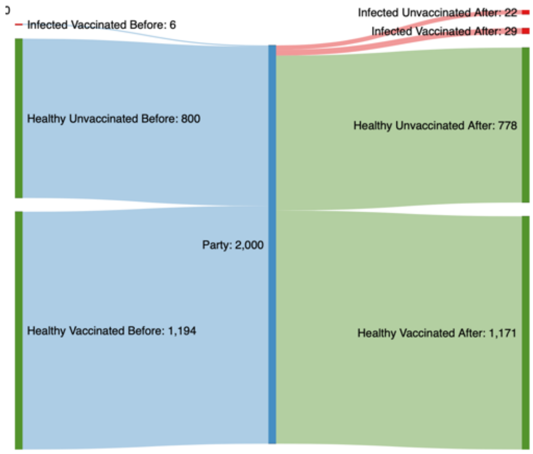

Figure 8 Infections before and after the party - 3G Plus

6.2.4 Unity 2021 - 2G, booster

A winter party with 2G rules would exclude unvaccinated guests, therefore we only consider vaccinated people with or without the booster shot for our simulation. In this scenario, we simulate an event at the start of December 2021 with 85% of vaccinated people, the remaining 15% already have the booster shot. There are no testing requirements for entering the event, therefore we have ten potential spreaders on the event.
This scenario has 358 infected people after the event which is a little bit more than in both 3G scenarios and surprising at first glance. One reason for this outcome is probably the missing testing requirement, which was always mandatory for non-vaccinated guests in the 3G scenario.

Higher vaccination requirements can help, people with the booster shot (6.1 % infection rate) are better protected against infection than vaccinated people (19.5 % infection rate). In our simulation model, the vaccination and booster protection are not more effective than rapid tests and the exclusion of some spreaders from the event.

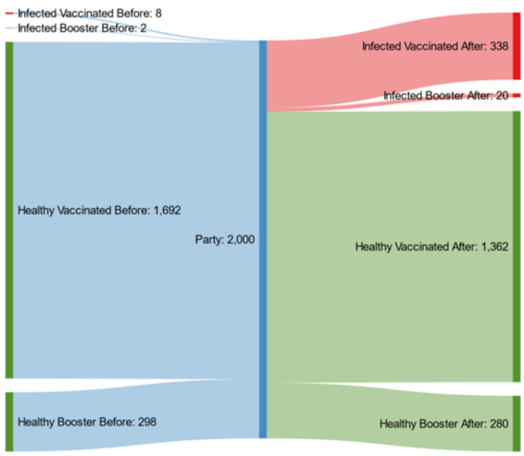

Figure 9 Infections before and after the party - 2G

6.2.5 Unity 2021 - 2G Plus

With 2G plus, each individual uses rapid tests that can detect 50% of the infected people. With rapid test as an entry requirement for all guests, we reduce the number of infected people by 50 percent and re-evaluate the scenario.
The number of infected people are significantly lower with an average of 41 newly infected guests over three runs. This is a lot better compared to 2G. During the event, the infections occur slightly later than in the 2G

scenario and it seems that with fewer infected people at the event, the threshold for an infection is often not reached until the party ends. In our simulation, the infection rates for vaccinated people (2.2%) and with booster shots (1.0%) are much lower than in the regular 2G scenario.

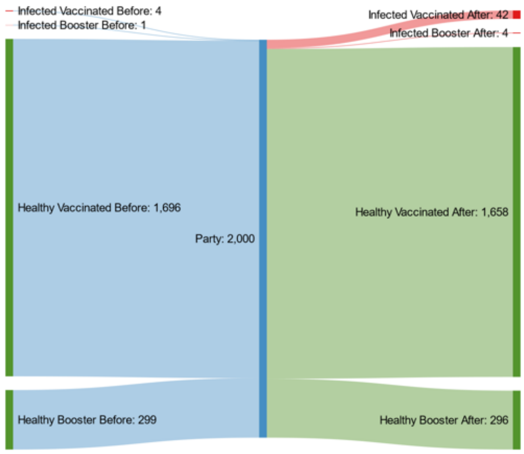

Figure 10 Infections before and after the party - 2G Plus

6.2.6 Unity 2021 - Omicron, Booster, 2G plus

Lastly, with the rise of the upcoming Omicron variant, we decided to model a Unity party in the future, as of February 2022. We expect that by this time, 80% of all vaccinated guests are already boostered. In December 2021, we reached a booster rate of about 40% within one month.
Currently, there is very limited data available about the infection rate of the Omicron variant. Based on a very recent paper [1], it is estimated that Omicron is by factor 2 more infectious than the Delta variant. We therefore increased the infection probability to 0.44%.

In our three tested runs, this would lead to around 35 infections on average. Out of these 35 infections, around 22 happened among boostered guests. 13 took place around guests which received only two vaccinations. 22 out of 1600 guests gives us an infection rate of 1.4%. The infection rate is significantly higher in the group with only 2 vaccination shots, with 13 out of 400 people producing a rate of 3.25%.

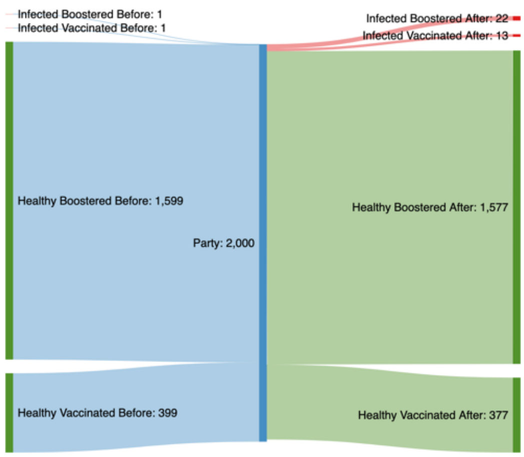

Figure 11 Infections before and after the party - Omicron, Booster, 2G plus

While the places of infection do not differ from any other scenario, a difference could be detected when an infection took place. As can be seen in Figure 12, a lot of infections happened after midnight, when many guests left the party already. It seems that through increased defence against infections, an infection takes longer to be transmitted successfully.

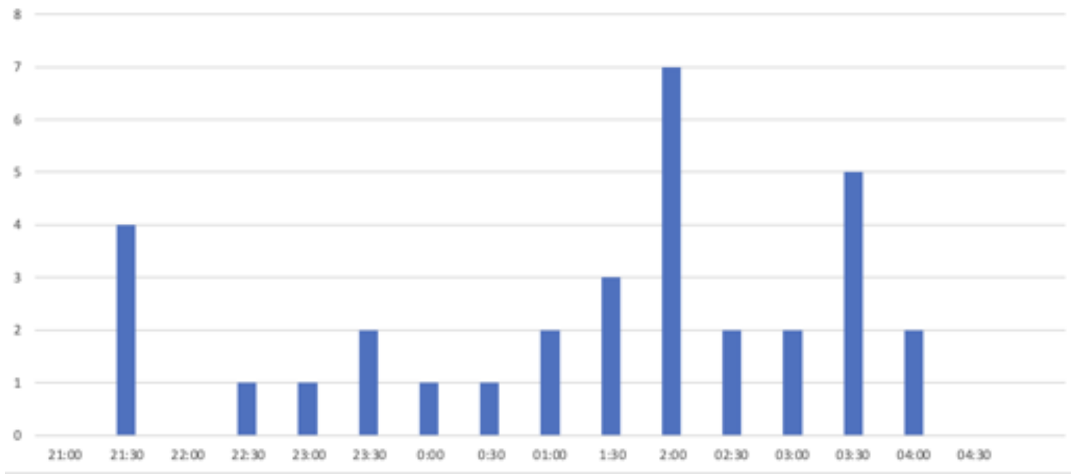

Figure 12 New infections per 30min party intervals

6.3 Shared aspects and Main Findings

In all tested configurations, the major spreading took place on the main stage. Here, many guests are partying very close to each other for a decent amount of time. Singing and dancing increases the probability of getting infected significantly. On the other hand, almost no infections were reported in outdoor areas throughout all scenarios.

Testing is very effective against having high infection rates during parties. It results in having fewer infected guests attending the party, which then reduces the infection rate significantly. Furthermore, PCR tests are a lot more effective than rapid tests and would lead to safer parties.

Time intervals when infections happen vary between scenarios. In the 2020 scenario, most infections happen during the prime time of the party before midnight. If only vaccinated people attend a party, an infection takes longer to be transmitted successfully. Here, infections happen later in the night.

Our brightest finding was that even if we have to deal with a new Omicron variant in the upcoming months, a high booster rate is a great way to tackle this variant. If the infection probability of Omicron and the booster defends guests as stated in the cited papers, a party with 80% of boostered attendees should be considered safer than a regular 2G+ party in Delta setting.

7 Challenges

During our implementation step, the major challenge was to deal with the One's performance capabilities. These challenges occurred primarily when we attempted to model more than 1000 nodes simultaneously. All of those nodes had their own movement model and spreading model implemented. By improving and updating our own and predefined algorithms by the One, we were able to run a single simulation within 10 minutes.

Another problem was the visualization of several thousand nodes. Each node usually has its message spreading range visualized as a green circle around its center. In our case, this leads to a diameter of 6 meters.
Especially in crowded areas, such as the Main Stage, many nodes are standing close to each other. Since the diameter of a single node is that wide, it is very hard to visualize a guest's behavior within a larger group gathering.

Another major challenge was finding the correct weights for spreading. Sadly there is not much research about parties in corona times available. We therefore tried to orientate on various newspaper headlines and articles to validate parts of our findings.

Implementing group behavior, so that guests do not come alone and move mostly with their group together, was a very challenging approach which we did not implement in the end to avoid over-complexity in our mobility model.

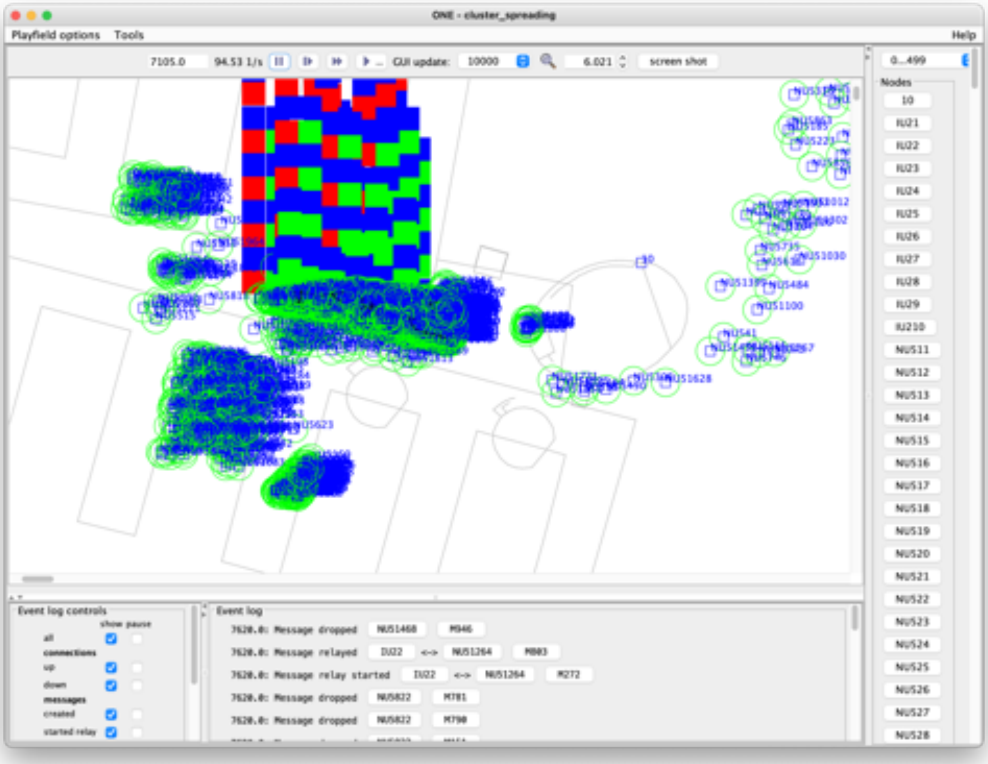

Figure 13 Crowded scenario map during party simulation

8 Conclusion and Outlook

The One proofed as a very powerful tool to forecast and evaluate various party settings. With a little more fine-tuning, it could be easily used to create even more accurate models. After we tested our scenarios, we are quite sure that there will be a point in time when we can finally celebrate Unity again, even with the rise of new Coronavirus variants. On the other side, we can now better understand why the Unity parties could not be celebrated in 2020 or 2021 when fewer counter measurements were available. This would have indeed led to a very great super spreader event.

References

[1] Jiahui Chen, Rui Wang, Nancy Benovich Gilby, and Guo-Wei Wei. Omicron (b. 1.1. 529): Infectivity, vaccine breakthrough, and antibody resistance. *ArXiv*, 2021.

[2] P Elliott, D Haw, H Wang, O Eales, C Walters, K Ainslie, C Atchison, C Fronterre, P Diggle, A Page, et al. React-1 round 13 final report: exponential growth, high prevalence of sars-cov-2 and vaccine effectiveness associated with delta variant in england during may to july 2021.

[3] Jos Lelieveld, Frank Helleis, Stephan Borrmann, Yafang Cheng, Frank Drewnick, Gerald Haug, Thomas Klimach, Jean Sciare, Hang Su, and Ulrich Pöschl. Model calculations of aerosol transmission and infection risk of covid-19 in indoor environments. *International Journal of Environmental Research and Public Health*, 17(21), 2020.

[4] Matan Levine-Tiefenbrun, Idan Yelin, Hillel Alapi, Rachel Katz, Esma Herzel, Jacob Kuint, Gabriel Chodick, Sivan Gazit, Tal Patalon, and Roy Kishony. Viral loads of delta-variant sars-cov-2 breakthrough infections after vaccination and booster with bnt162b2. *Nature medicine*, pages 1–3, 2021.

[5] C Raina MacIntyre and Quanyi Wang. Physical distancing, face masks, and eye protection for prevention of covid-19. *The Lancet*, 395(10242):1950–1951, 2020.

[6] M Catherine McEllistrem, Cornelius J Clancy, Deanna J Buehrle, Aaron Lucas, and Brooke K Decker. Single dose of an mrna severe acute respiratory syndrome coronavirus 2 (sars-cov-2) vaccine is associated with lower nasopharyngeal viral load among nursing home residents with asymptomatic coronavirus disease 2019 (covid-19). *Clinical Infectious Diseases*, 73(6):e1365–e1367, 2021.