Intro

Simulating a pandemic:

- not an easy task because so complex
- User shouldnt have to do settings like these, because how would the user know the fatality rate between Omicron and Delta variant
- confusing, User nearly needs a medical degree to use and understand the simulator
- So many options and information to digest
- Game-like approach, overview significantly better

Stochastic Algorithms (Dis-)Advantages

- Haven't put much deep thought in it, simple reasons
- Decided in favor of Gillespie because adding new rules was easy, nevertheless rewritten the rule set about 3 times
- Handling of the numbers easy

Gillespie-Algorithm

- t, number of events match the time frame: So if you want 1sec that there arent a huge number of infections

Gillespie Algorithm Example:

- Iota infect rate, rho recovery rate, delta death rate

Extended SIRD-Model

- My Rule set for the Gillespie-Algorithm, every arrow is a rule
- 149 rates that decide over rules, many rates carry out the same rule
- Heavily relied on a paper for rates
- I2, R2, D2 unaware changes of infection, negative tested and getting infected then
- Every state in Germany has this model, commuter get traded

Basic structure

- Game-Management acts as an adapter between data classes and game interface

Main Thread inside Dash Control class

- Showing the path from the button press to the simulation
- simulation runs in a different thread than the main thread, so that interface stays responsive

Inside Game-Management Class

- another function call, now to the Country-Class
- checks if there are no new infections in a certain time frame, chose 75 days

Inside the Country Class

- Handles vax production and distribution, and commuter distribution between states
- starts the threads for the states

Distributing Commuters

- Commuter array in state really complicated, indexing is kind of hard
- Commuter Rates based on data of the Agentur für Arbeit
- Basically distribute Commuter evenly on neighboring states
- Randomly pick individuals from the various groups

Inside the state Class

- where the magic happens, Gillespie Iteration here
- time frame of one day, time difference from last event is considered and added before starting the new day, overhead event statistically not significant, because measures don't work in an instant in reality as well
- waiting period realised as a conveyor belt, where the individuals move one step each day

Saving the data

- Godot Arrays allow several data types in the same array, 0-Index Name of the group, after that numbers

Simulating – Options

- Godmode even shows unrecorded cases
- Sim factor: for example 0.2, simulation only calculates with 20% of the population and projects the numbers to 100%

Available mesures against the spread of the disease

- Strictness of the measures is just a rough indication on how many measures are taken at the moment
- Progress bar below map that shows the progress of the simulation while it runs

Implementing the measures

- All measures are realized through the infect rate and the commuter rate, base infect rate gets modified
- Data on how much the baseInfect-Rate is modified is taken from papers or from german statistical Bundesamt

Overall Indicator behind map

- Scale for the incidence scaling with the highest recorded incidence, highest incidence state always colored the darkest, be it with incidence 20 or 2000
- Indicator behind map gives feeling for the big picture, if green incidences on a low level, at red on a high level

DEMONSTRATION

- Terminal Window opens too, see some runtime stats
- Interesting phenomenon, show that tests act time-delayed
- Some unrealistic stuff can be tried out, can vaccinate all people at once or have 80 million hospital beds
- Show the Ending