Forge4Flame (F4F): a user-friendly dashboard designed to simplify the definition of ABM environment for FLAME GPU 2

Daniele Baccega, Marco Beccuti, Marco Frattarola, Simone Pernice, Irene Terrone

Department of Computer Science, University of Turin, Italy

Who are we?

Research Objectives:

Patient stratification

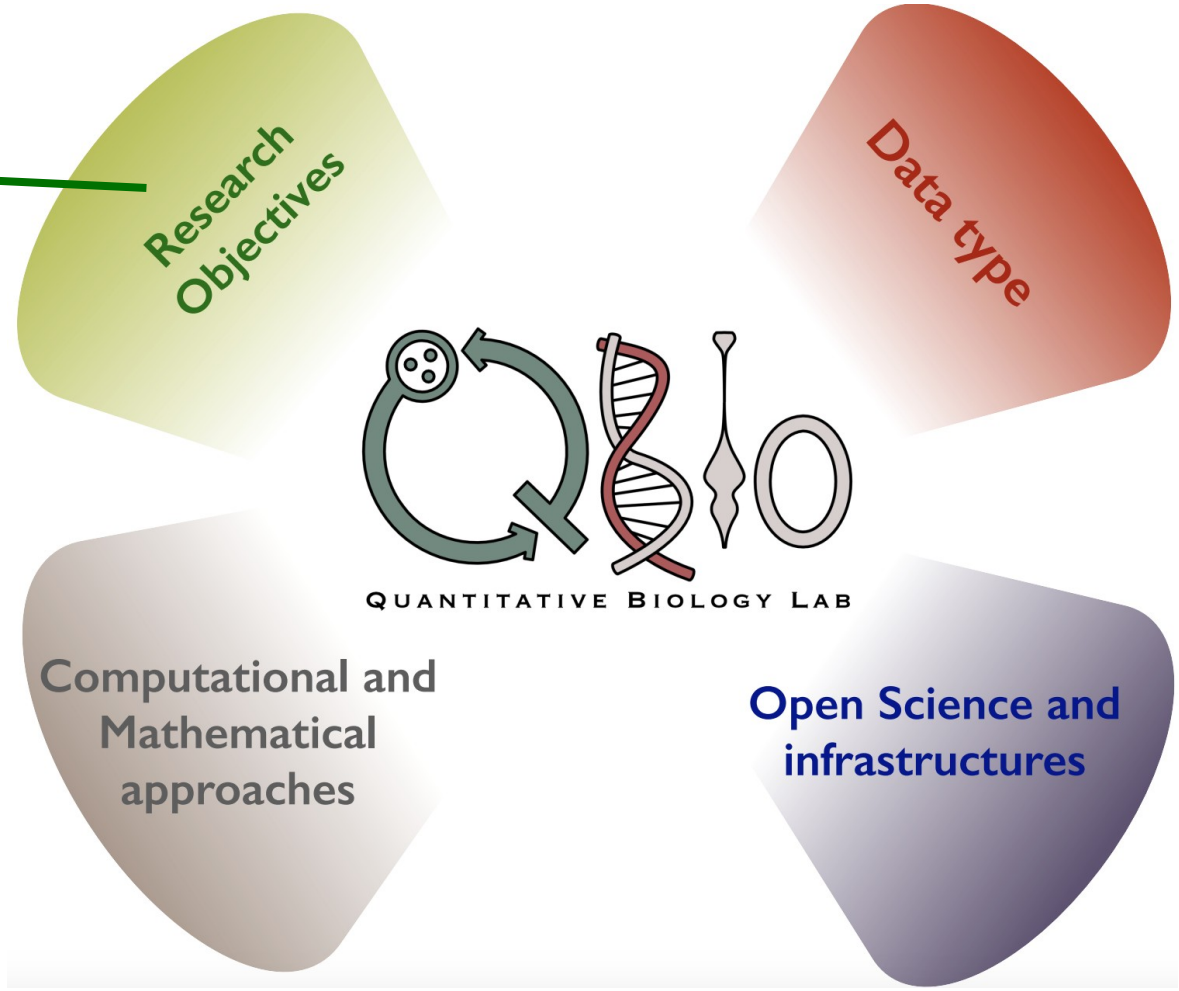
Identification of diagnostic and prognostic biomarkers

Epidemiological models

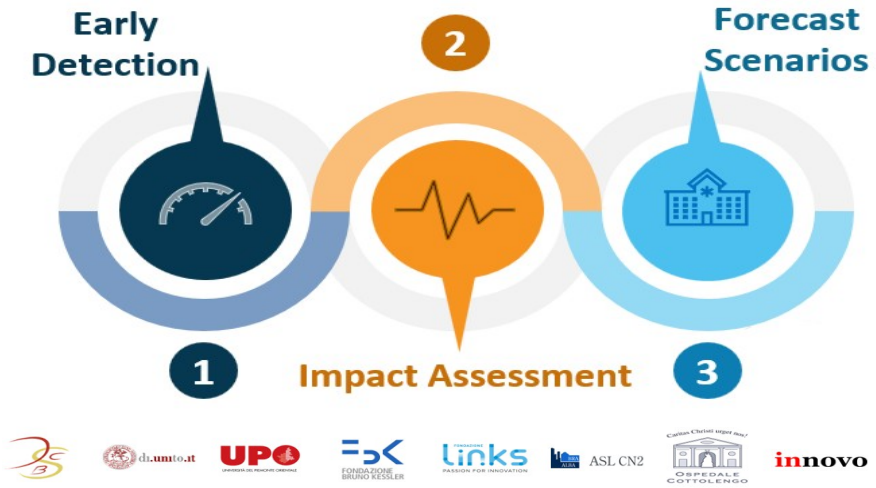
Mathematical models for cancer progression and treatment response

Evolutionary Genomics and Modelling

Basic Research: Cancer Genomics and Epigenomics



How it started...



TrustAlert Platform for the analysis of near real-time stream of healthcare data

Founded by Compagnia San Paolo

1M Euro Project
7 Collaborators

Our objectives in this project:

evaluate the potential impacts of a health emergency in a real case scenario (Cottolengo Hospital in Turin) at different levels of detail, spanning from **different simulation paradigms** (e.g., ABM)

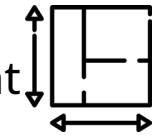
Expected results:

Being able to simulate the behavior of the "Hospital" system, for what happens outside, will allow Cottolengo Hospital to be able, in the medium term, not only to be effective but to improve the efficiency (cost / result ratio) and response capacity to ensure the delivery of health services

Introduction

User-friendly dashboard designed to simplify the definition of an ***Agent-Based Model environment*** without the need to write code.

It enables to define:

1. model's environment 
2. the agents interacting within it 
3. the disease model 
4. other components relevant to an agent-based simulation of disease spread 
5. Simple visualisations

User-friendly

Approachable by scientists without specific computational knowledge

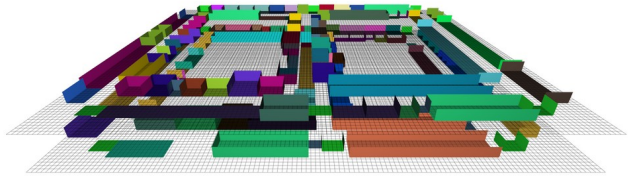
Automatic

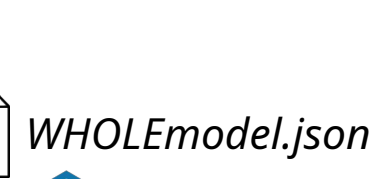
Automatic generation of 3D model environment for FLAME GPU 2 [5, 6]

Reproducibility

Scientific results can be consistently replicated by independent researchers using the same methodology and data

Pipeline



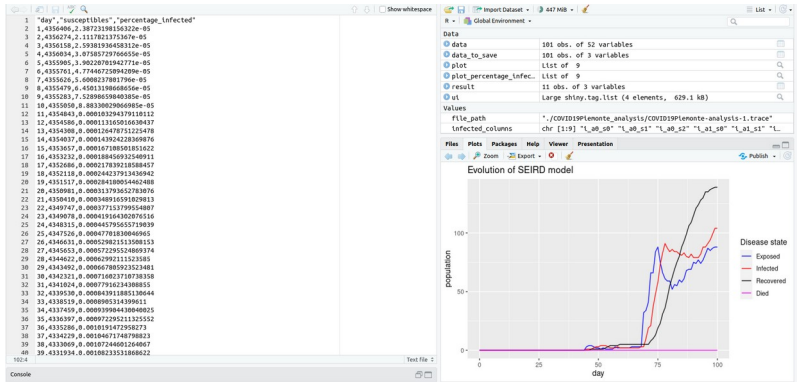


3D ABM Visualisation

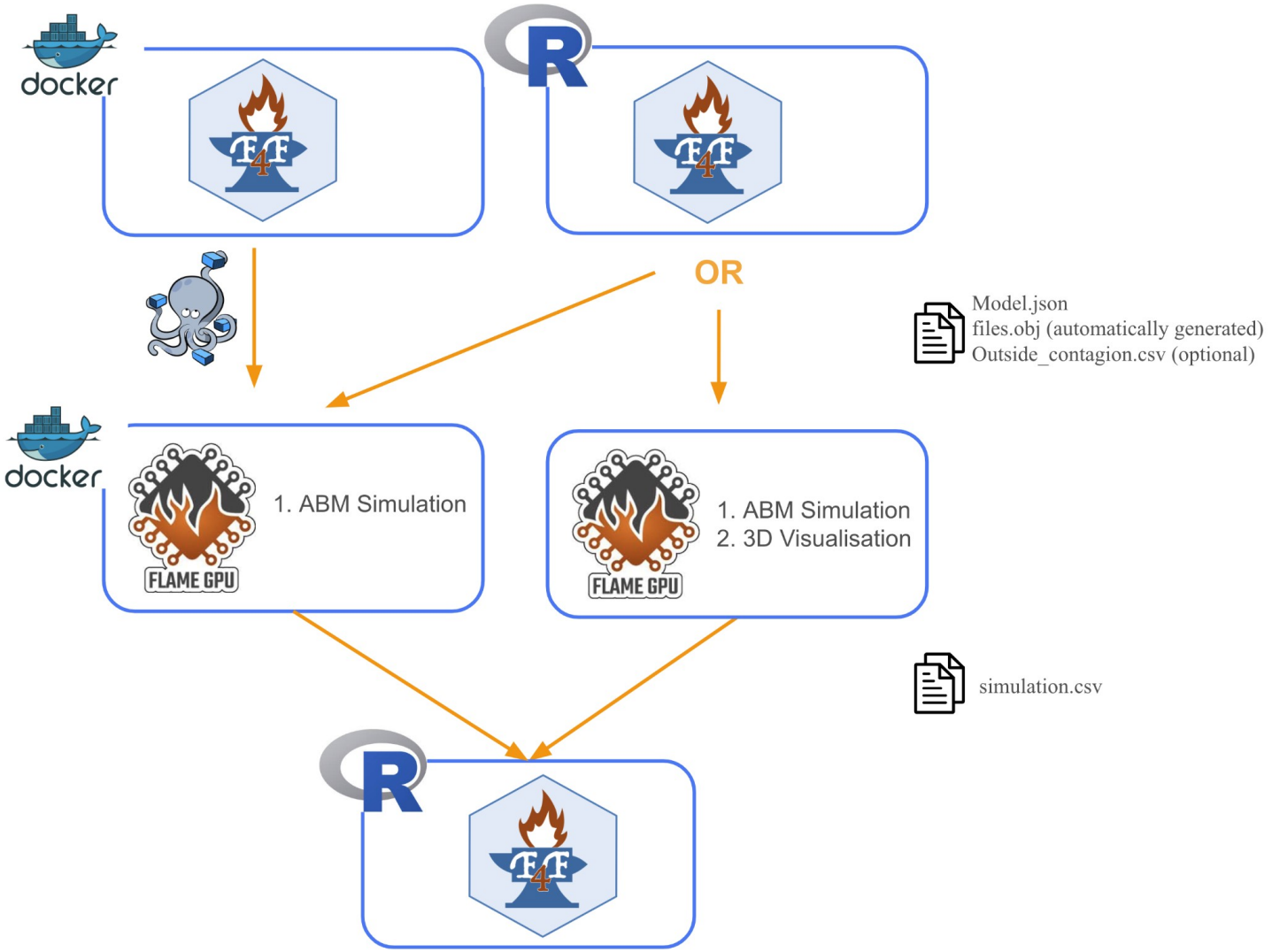
WHOLEmodel.json

ABM Environment

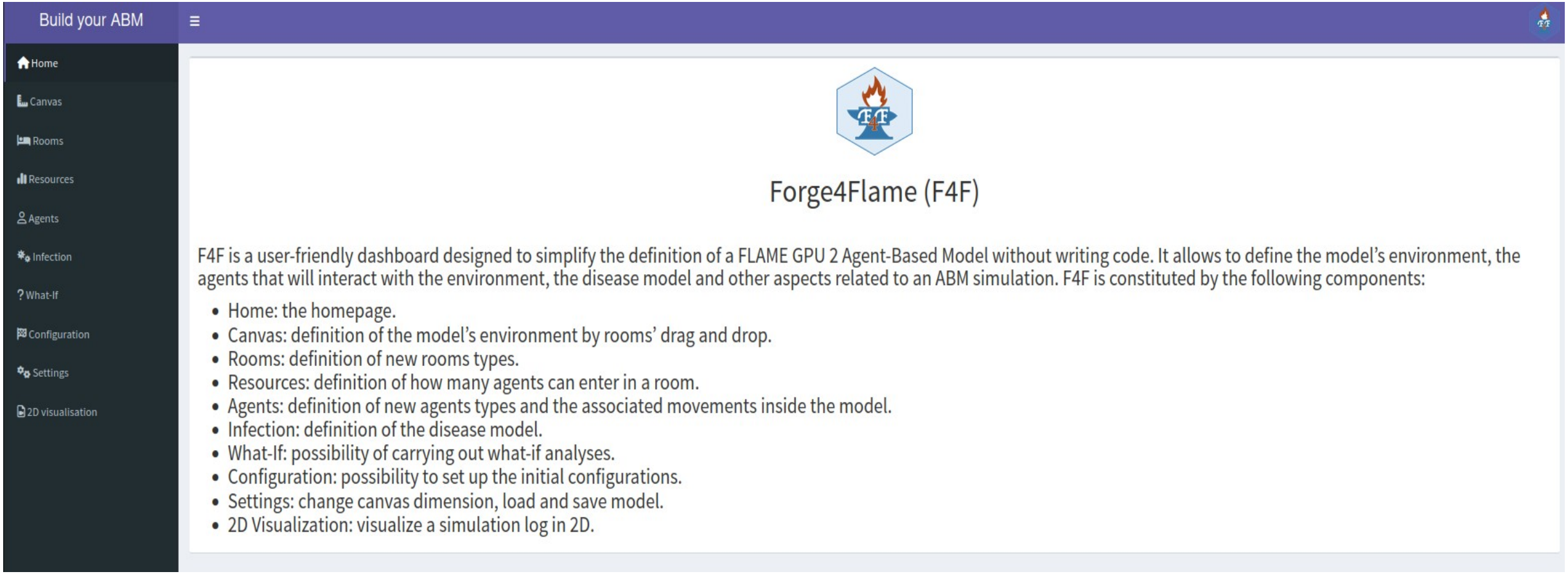
Epidemic Evolution



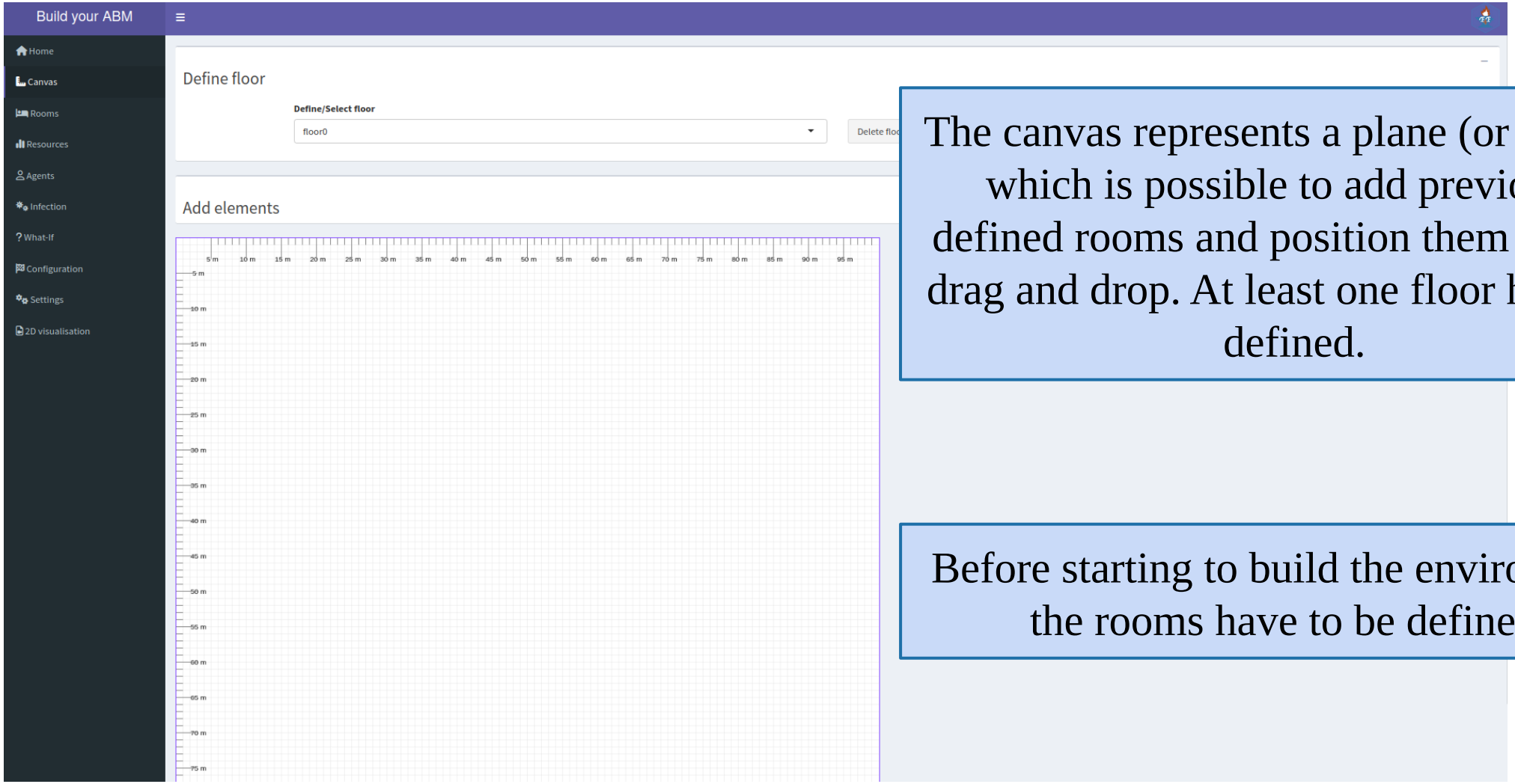
Docker Pipeline



Homepage


Build your ABM
Home
Canvas
Rooms
Resources
Agents
Infection
What-If
Configuration
Settings
2D visualisation
Forge4Flame (F4F)
F4F is a user-friendly dashboard designed to simplify the definition of a FLAME GPU 2 Agent-Based Model without writing code. It allows to define the model's environment, the agents that will interact with the environment, the disease model and other aspects related to an ABM simulation. F4F is constituted by the following components:
• Home: the homepage.
• Canvas: definition of the model's environment by rooms' drag and drop.
• Rooms: definition of new rooms types.
• Resources: definition of how many agents can enter in a room.
• Agents: definition of new agents types and the associated movements inside the model.
• Infection: definition of the disease model.
• What-If: possibility of carrying out what-if analyses.
• Configuration: possibility to set up the initial configurations.
• Settings: change canvas dimension, load and save model.
• 2D Visualization: visualize a simulation log in 2D.

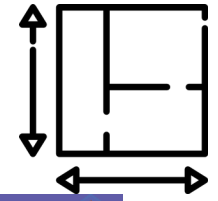
Canvas page with empty canvas

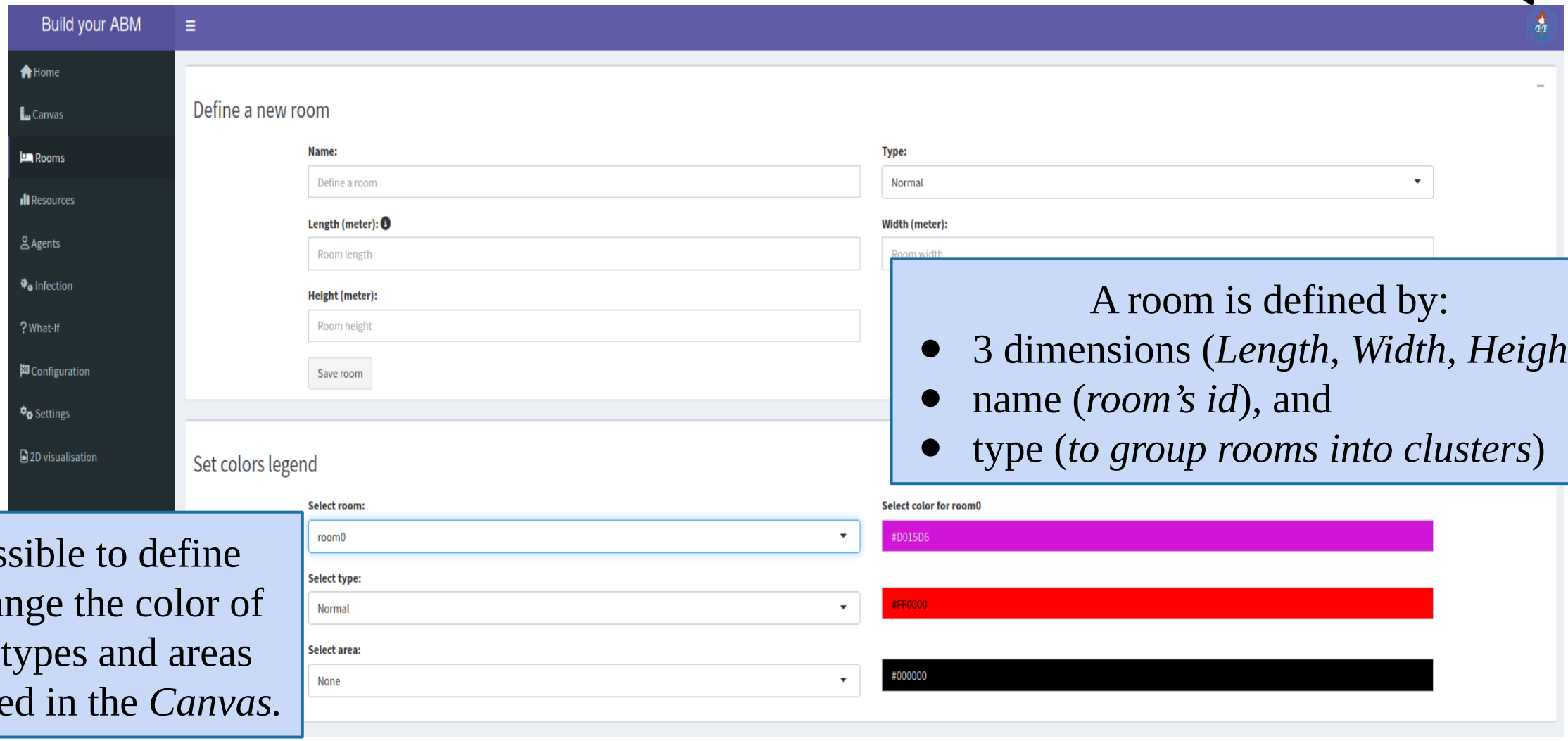


The canvas represents a plane (or floor) in which is possible to add previously defined rooms and position them through drag and drop. At least one floor has to be defined.

Before starting to build the environment, the rooms have to be defined.

Rooms page



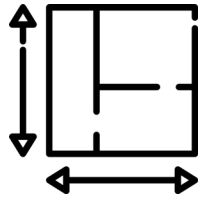


A room is defined by:

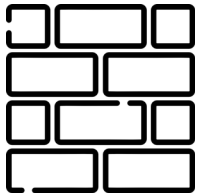
- 3 dimensions (*Length*, *Width*, *Height*),
- name (*room's id*), and
- type (*to group rooms into clusters*)

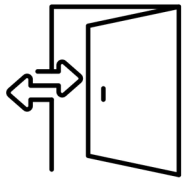
It is possible to define and change the color of rooms, types and areas displayed in the *Canvas*.

Rooms page: predefined rooms' type



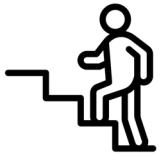
Filling-rooms: room type utilized by agents to simulate walls or useless rooms.





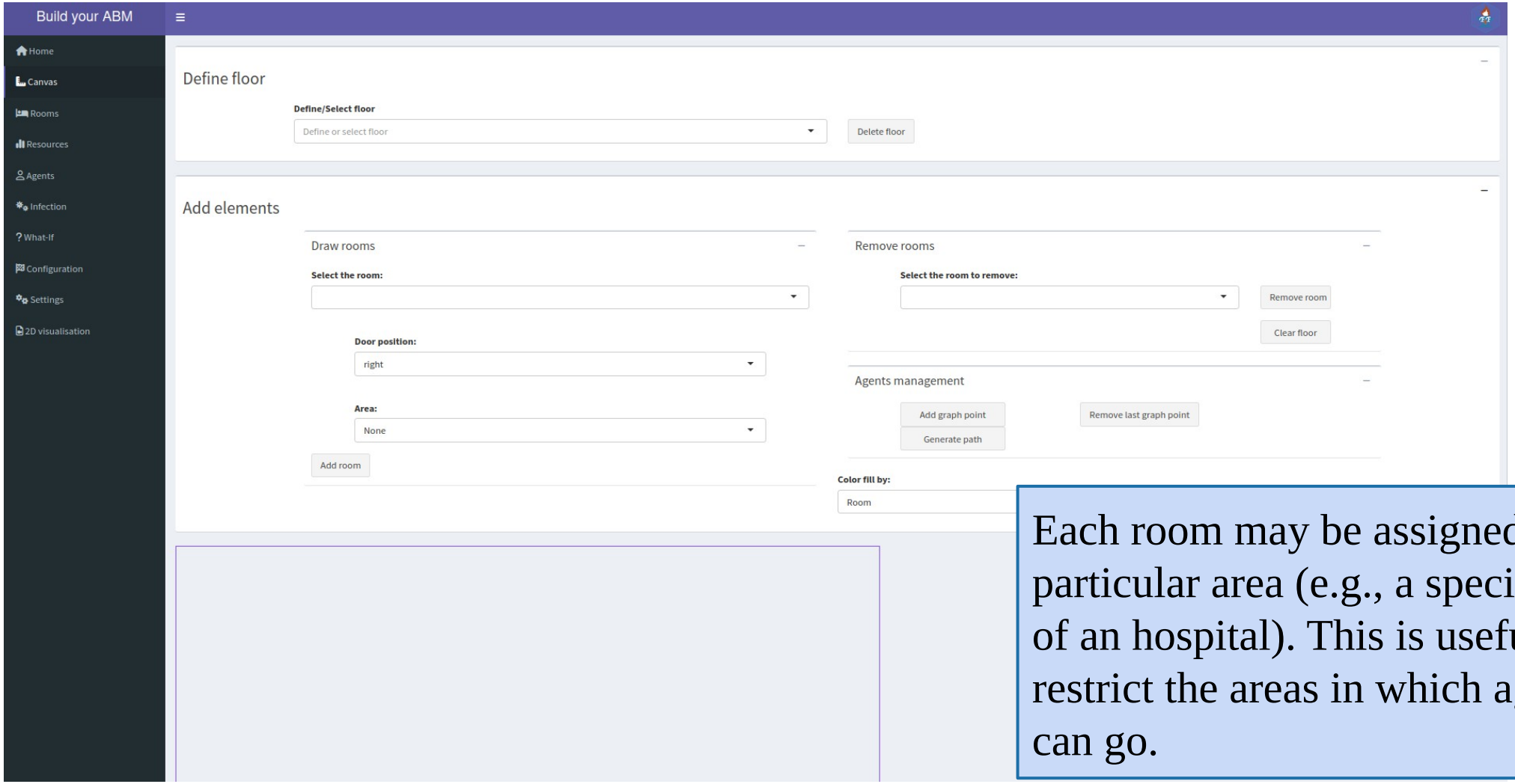
Spawn-rooms: room type utilized by agents to enter (and exit) the environment.

Stairs: room type utilized by agents to move among floors.



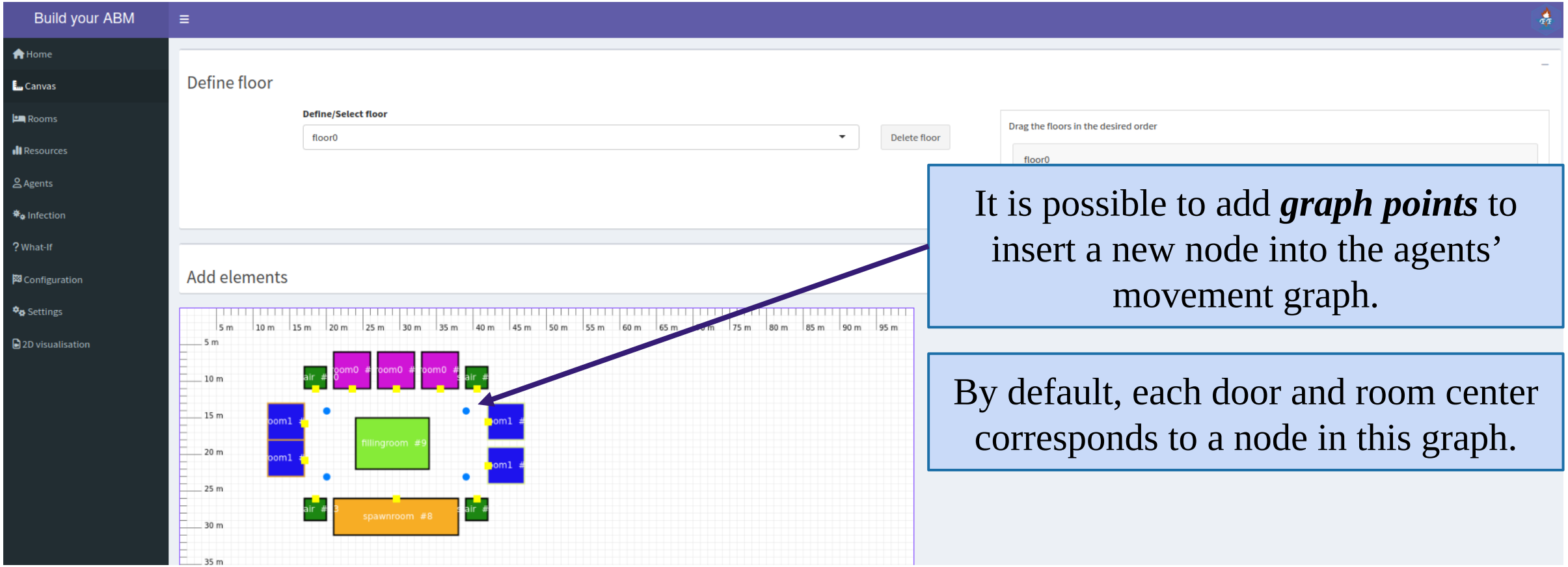
Waiting-rooms: room type utilized by agents to wait to enter in another room.

Canvas page



Each room may be assigned to a particular area (e.g., a specific area of an hospital). This is useful restrict the areas in which agents can go.

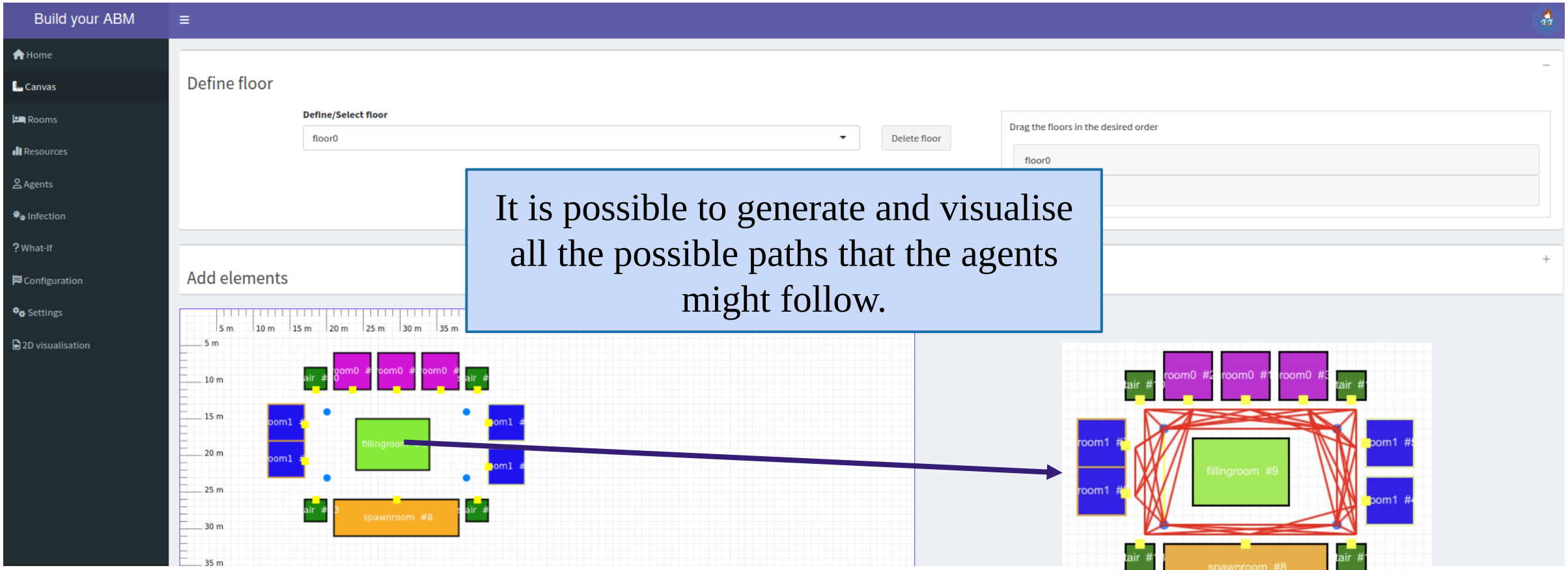
Canvas page: floor0



It is possible to add ***graph points*** to insert a new node into the agents' movement graph.

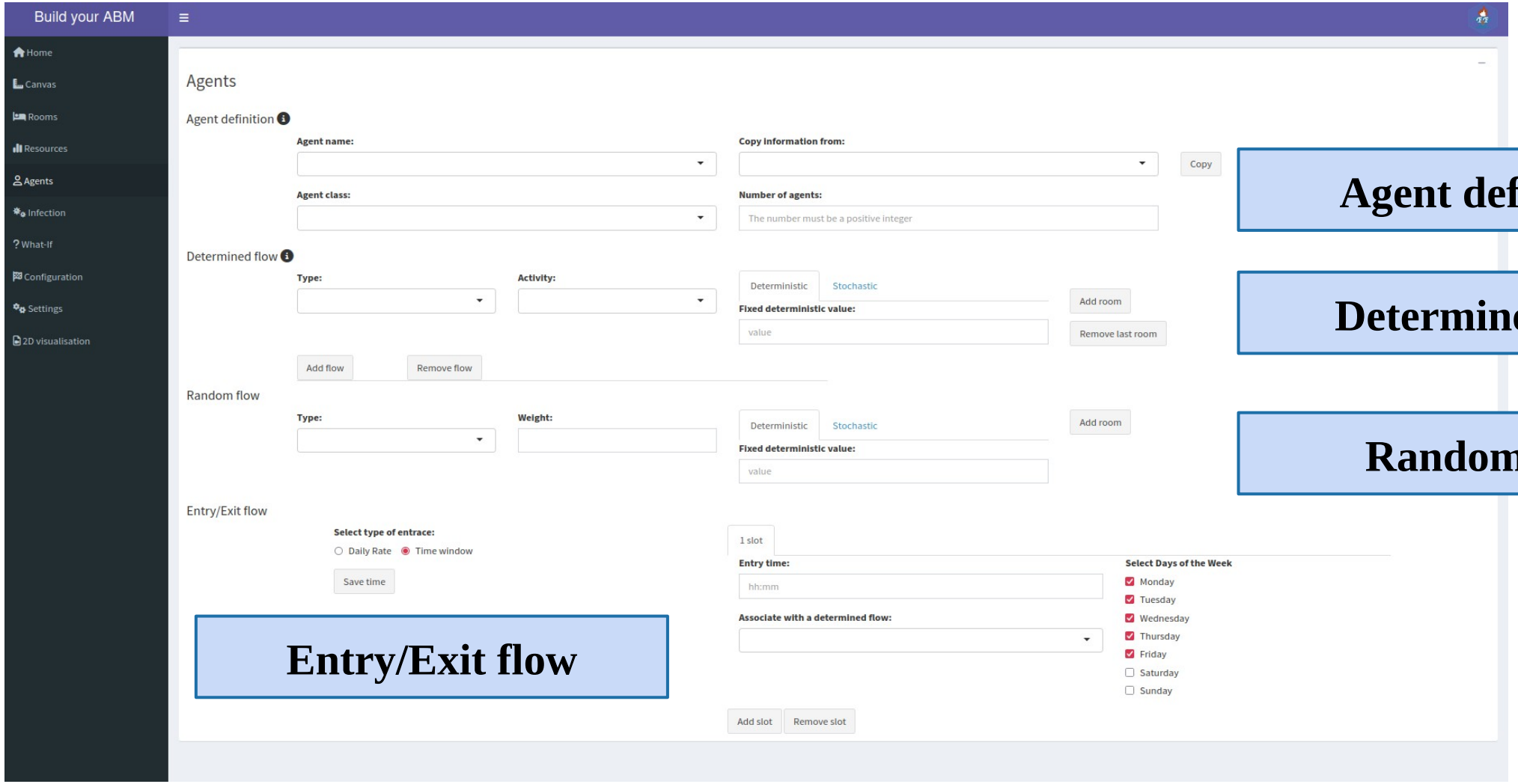
By default, each door and room center corresponds to a node in this graph.

Canvas page: floor0



It is possible to generate and visualise all the possible paths that the agents might follow.

Agents page



Agent definition

Determined flow

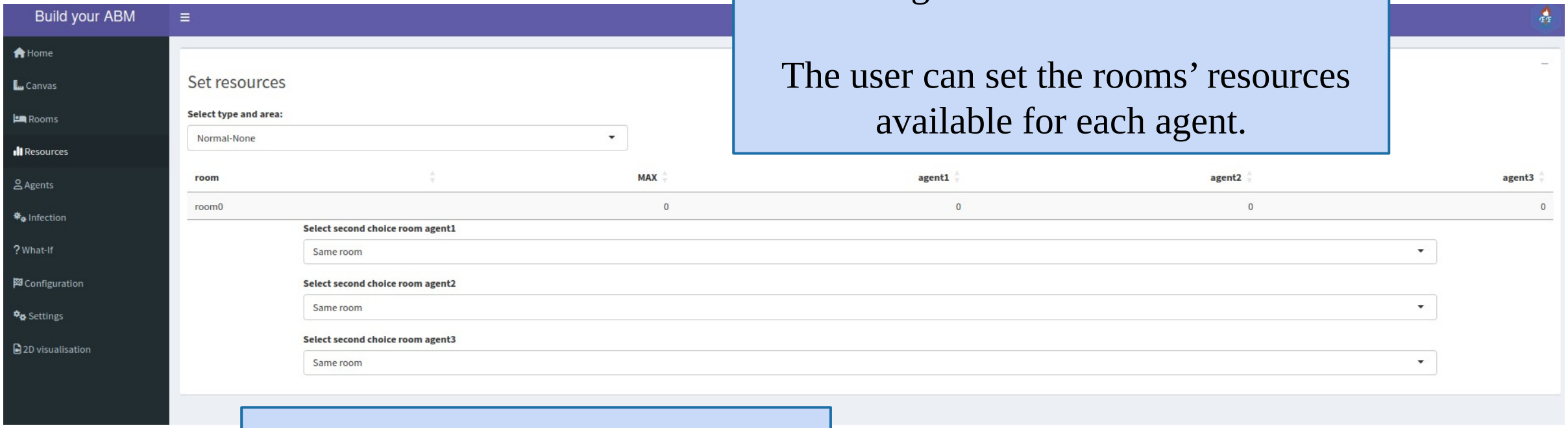
Random flow

Entry/Exit flow

Resources page

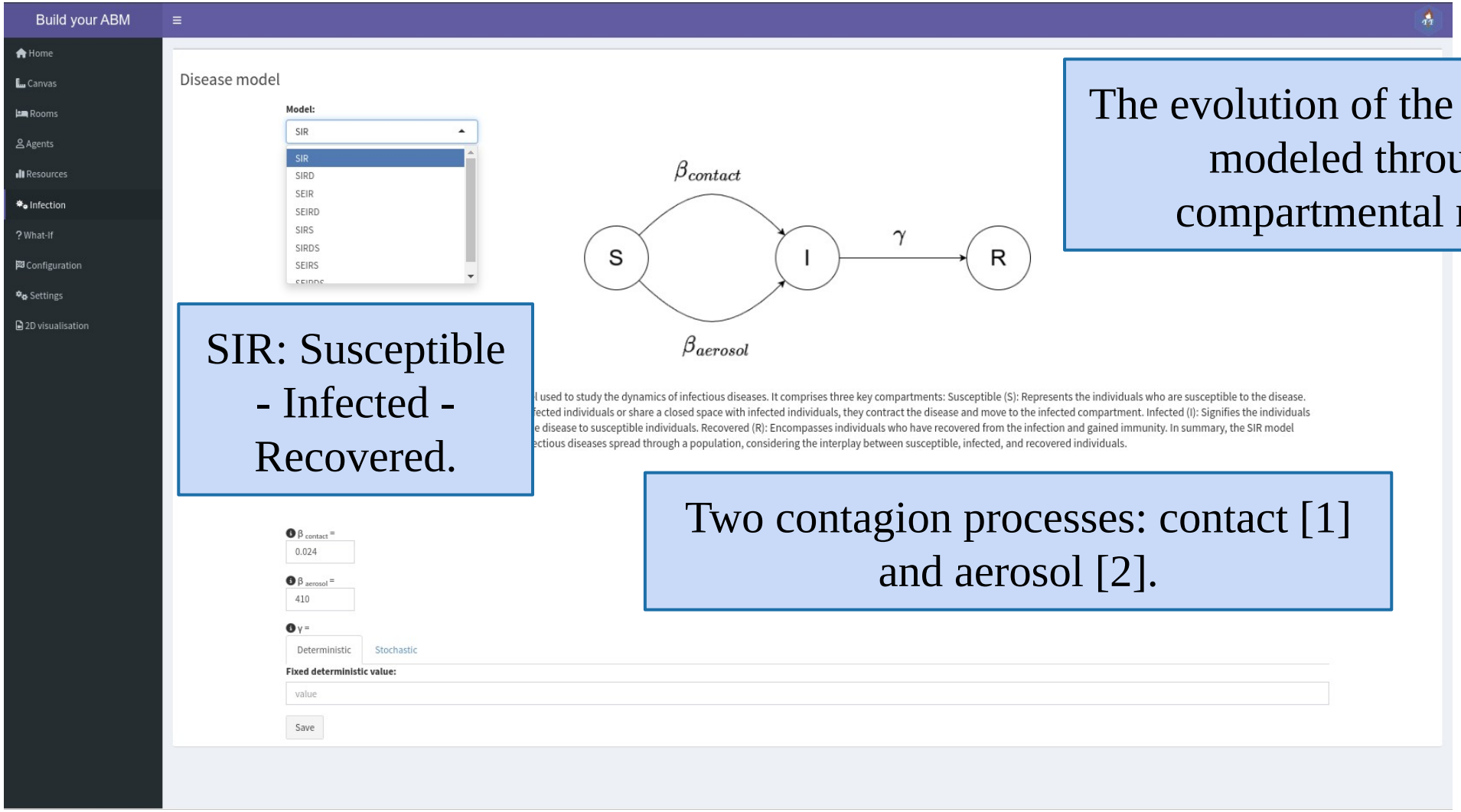
For each room, the list of all agents that might reach it is showed.

The user can set the rooms' resources available for each agent.



Resources:
The number of agents that can occupy a room simultaneously.

Infection page

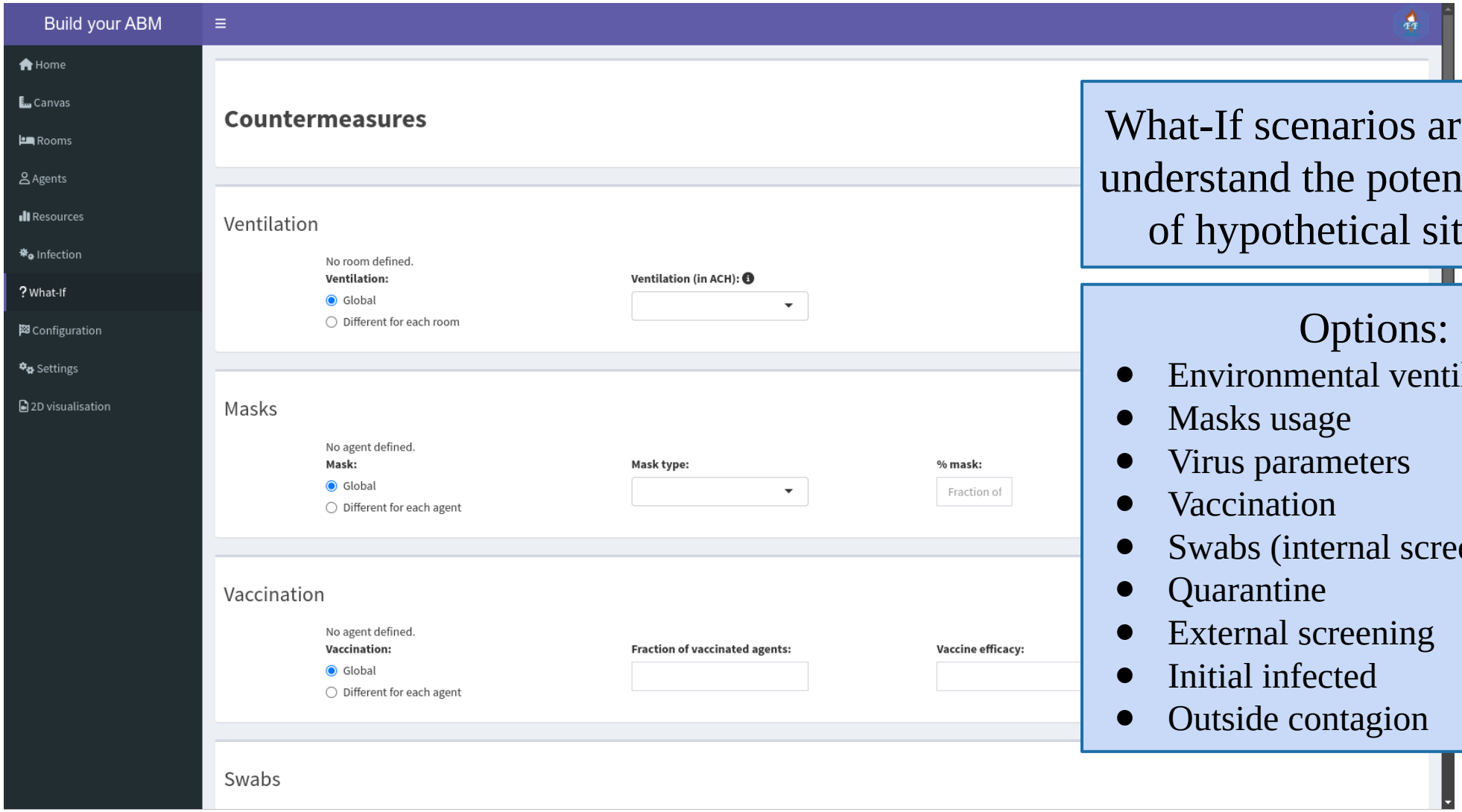


The evolution of the infection is modeled through a compartmental model.

SIR: Susceptible - Infected - Recovered.

Two contagion processes: contact [1] and aerosol [2].

What-If page

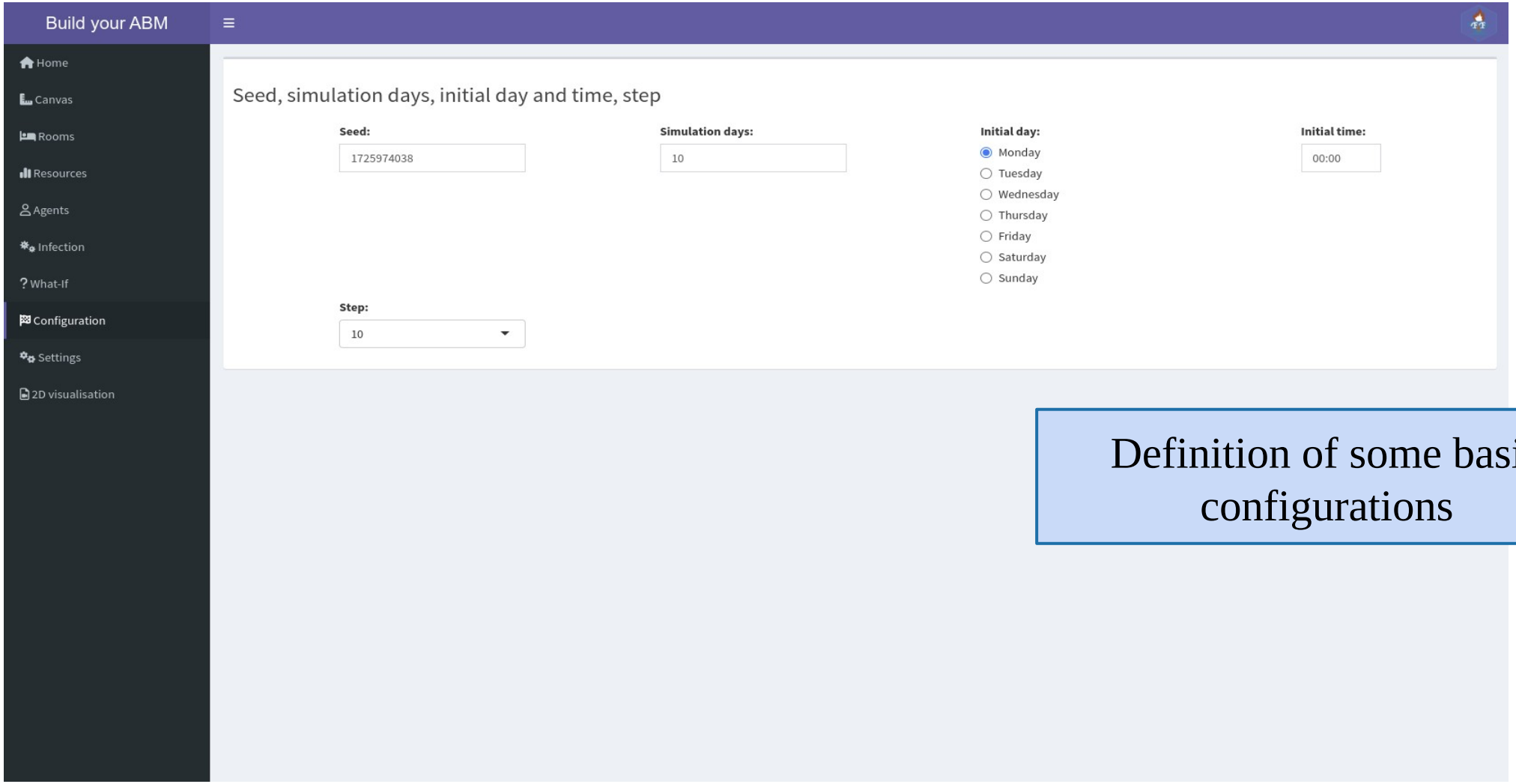


What-If scenarios are useful to understand the potential impact of hypothetical situations.

Options:

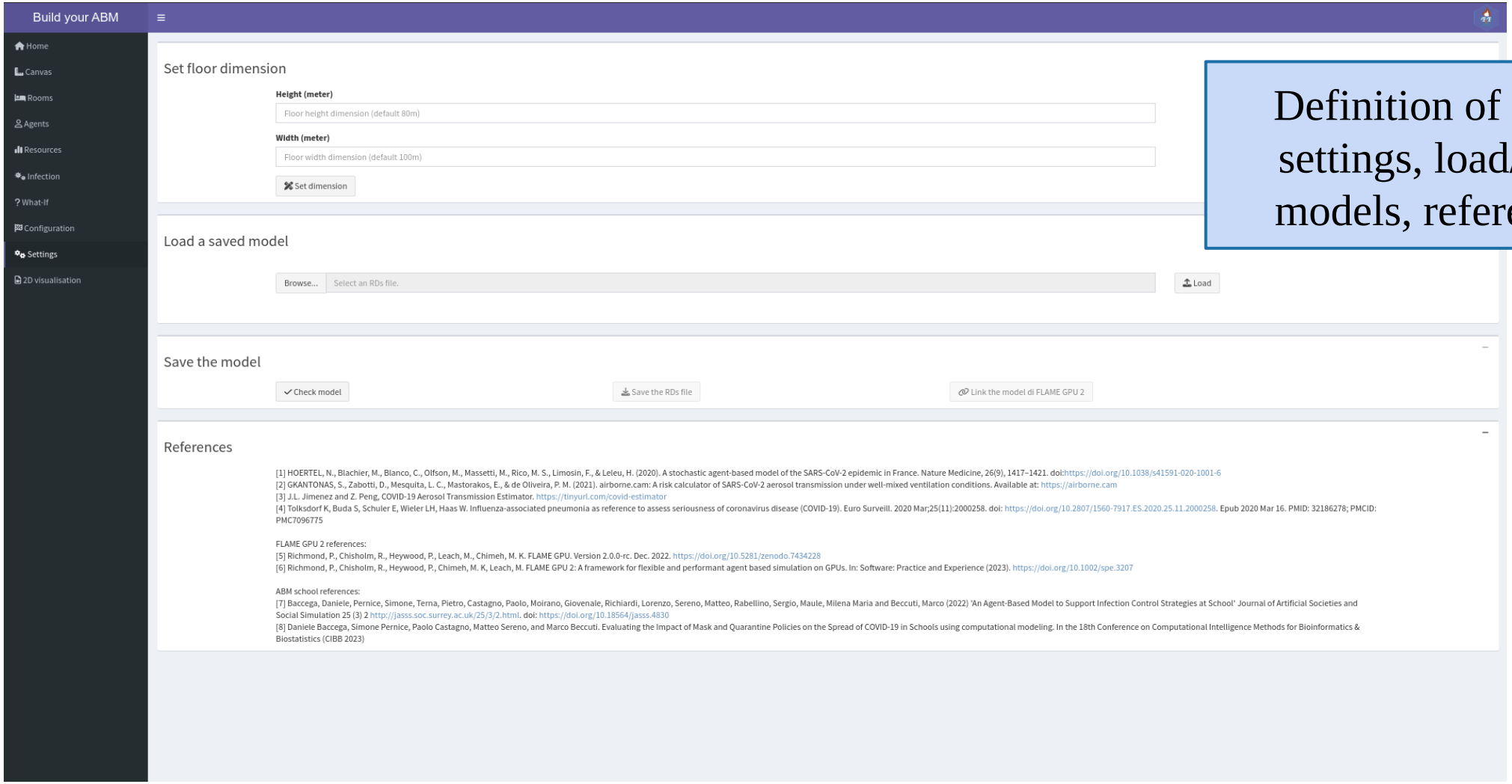
- Environmental ventilation
- Masks usage
- Virus parameters
- Vaccination
- Swabs (internal screening)
- Quarantine
- External screening
- Initial infected
- Outside contagion

Configuration page



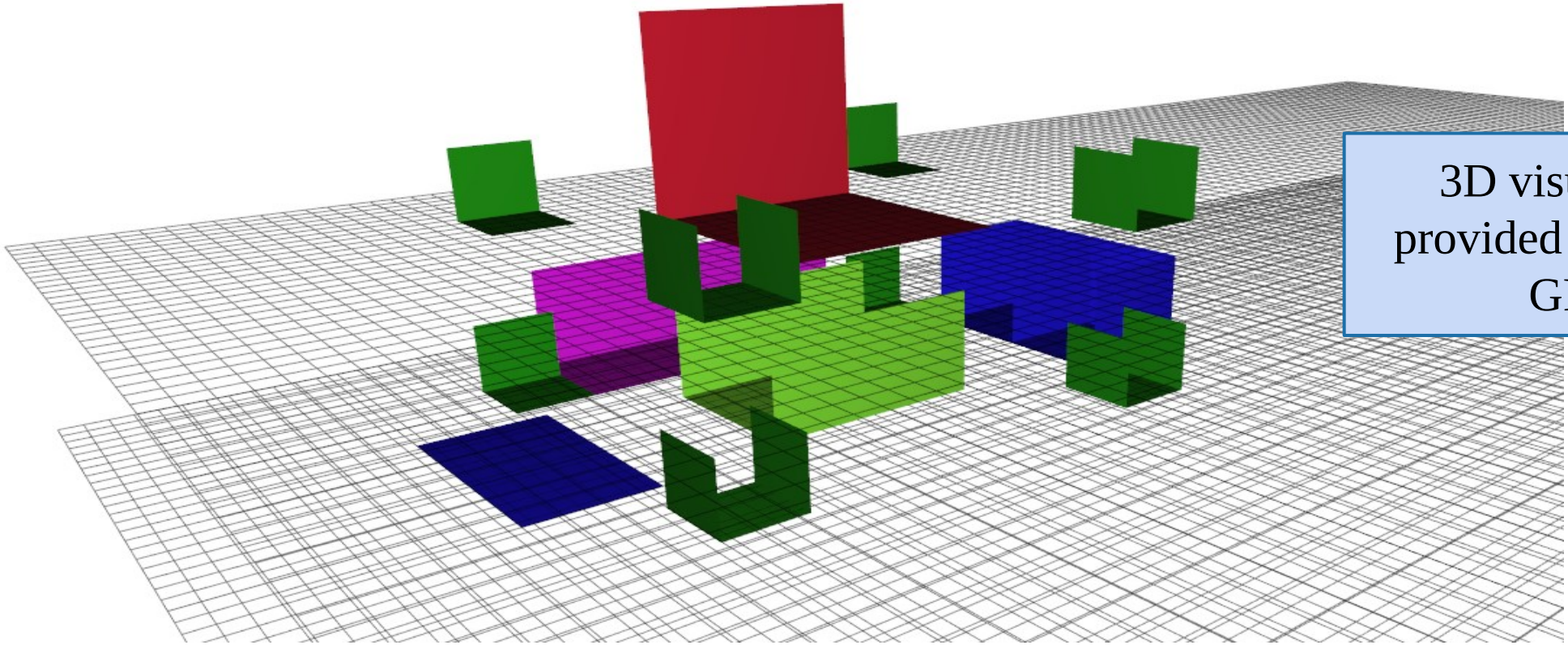
Definition of some basic configurations

Settings page



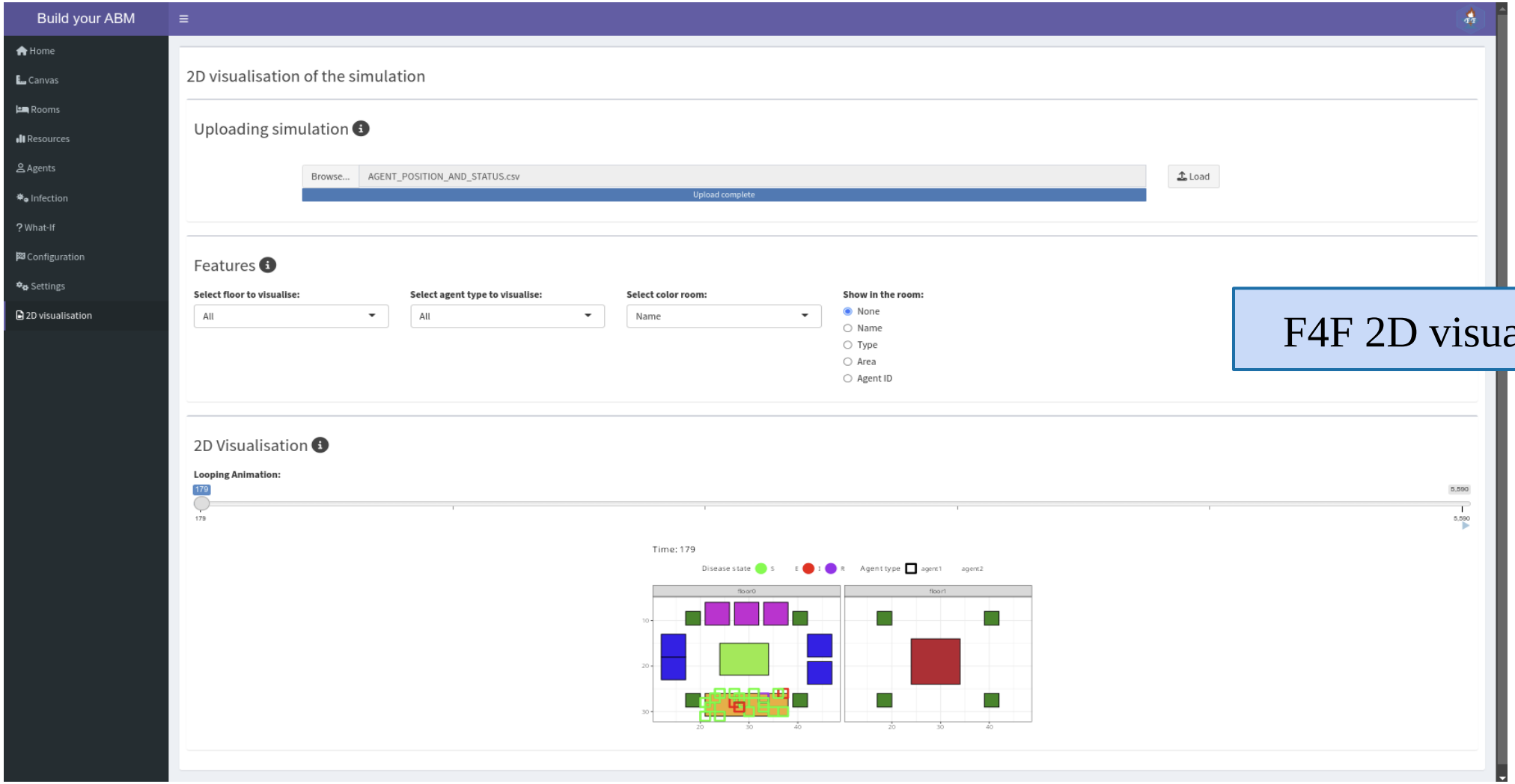
Definition of some settings, load/save models, references

FLAMEGPU2 3D mapping



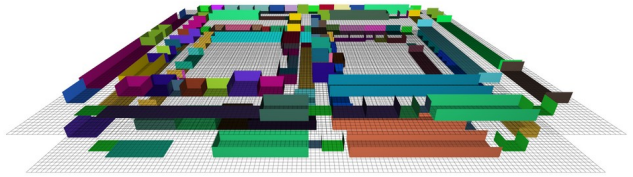
3D visualisation provided by FLAME GPU 2

2D visualisation page



F4F 2D visualisation

From F4F to FLAME GPU 2



glm

3D ABM Visualisation

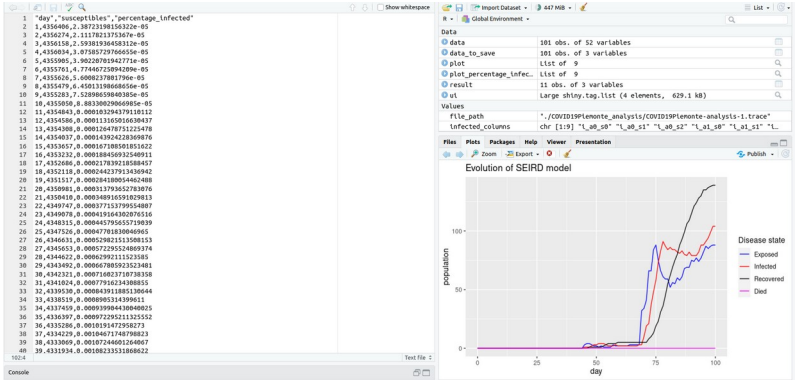
WHOLEmodel.json

Shiny

ABM Environment

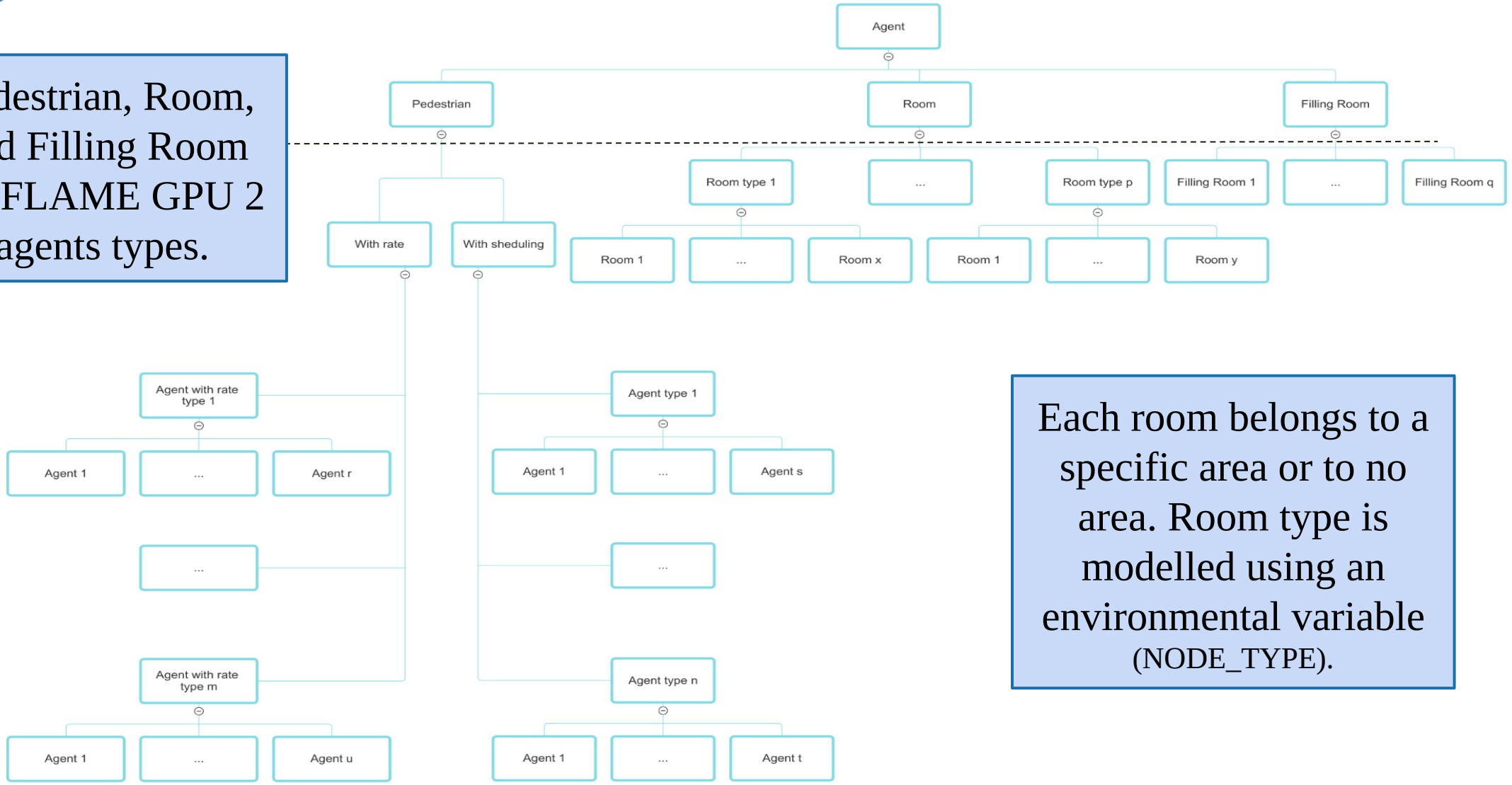
FLAME GPU

Epidemic Evolution



From F4F to FLAME GPU 2: agents

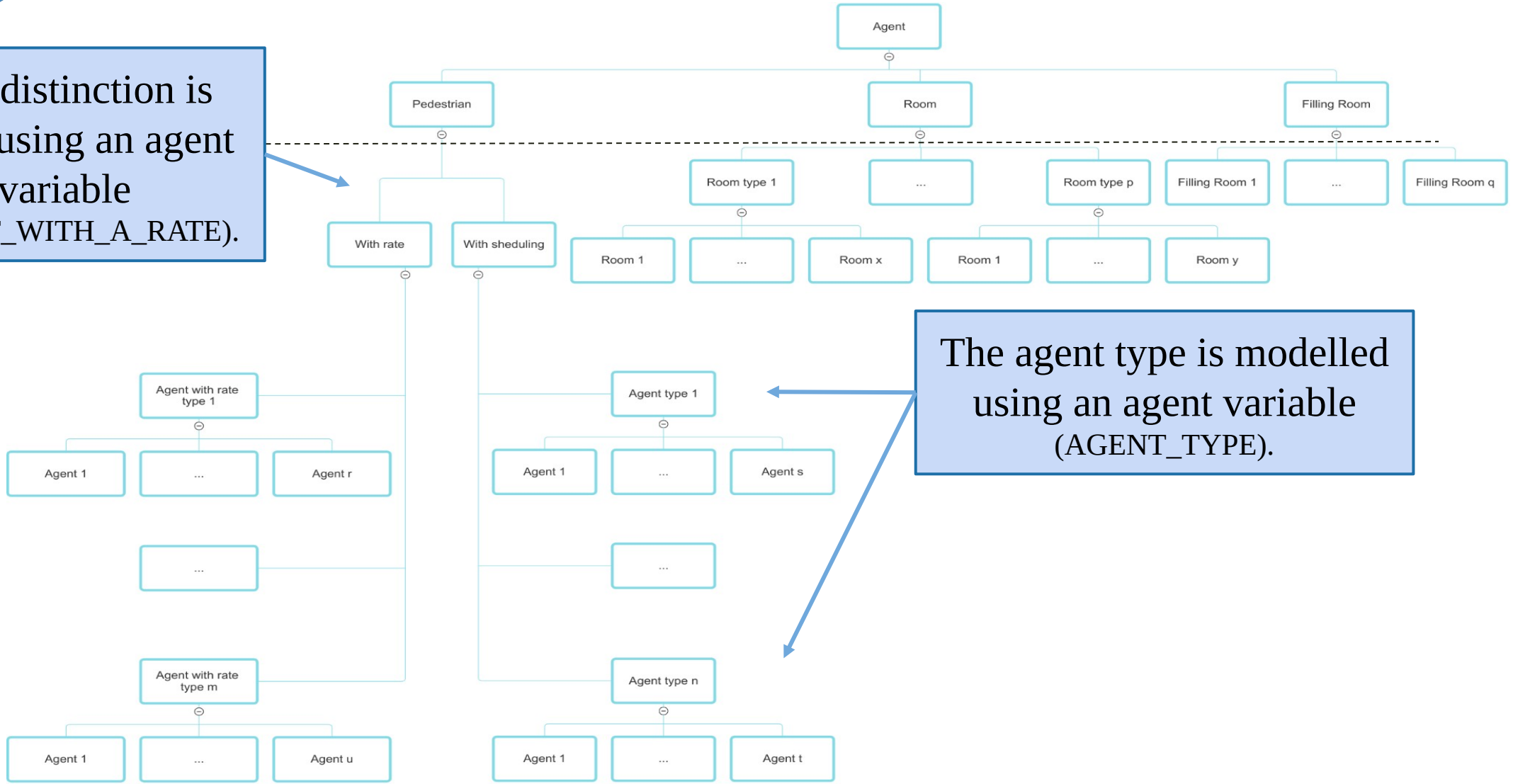
Pedestrian, Room, and Filling Room are FLAME GPU 2 agents types.



Each room belongs to a specific area or to no area. Room type is modelled using an environmental variable (NODE_TYPE).

From F4F to FLAME GPU 2: agents

This distinction is made using an agent variable (AGENT_WITH_A_RATE).



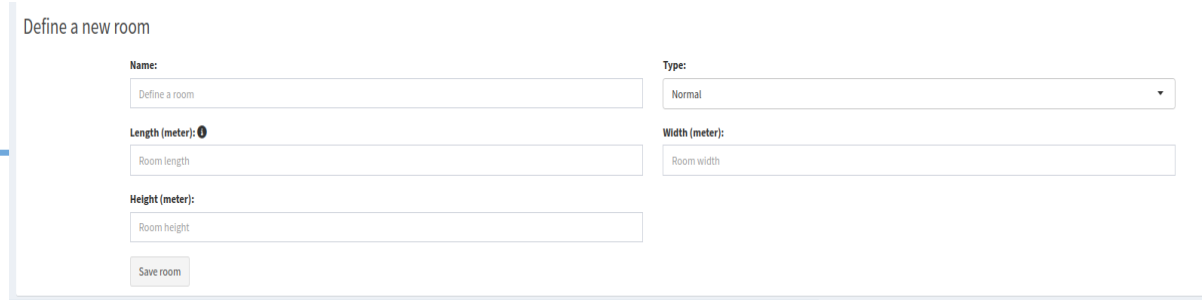
The agent type is modelled using an agent variable (AGENT_TYPE).

From F4F to FLAME GPU 2: rooms

```
// Define model's agents room
void define_room(ModelDescription& model){
    // Room agents
    string rooms_types[NUM_ROOMS_TYPES] = ROOMS;

    for(string room_type: rooms_types){
        {
            AgentDescription room = model.newAgent(room_type);
            room.newVariable<float>(X, 0.0f);
            room.newVariable<float>(Y, 0.0f);
            room.newVariable<float>(Z, 0.0f);
            room.newVariable<float>(LENGTH_OBJ, 0.0f);
            room.newVariable<float>(WIDTH_OBJ, 0.0f);
            room.newVariable<float>(HEIGHT_OBJ, 0.0f);
            room.newVariable<float>(YAW, 0.0f);
            room.newVariable<unsigned char>(INIT_ROOM, 0);
            room.newVariable<int>(AREA, -1);
            room.newVariable<int>(COLOR_ID, 0);
            if(room_type != FILLINGROOM_AGENT_STRING){
                room.newVariable<float>(VOLUME, 0.0f);
                room.newVariable<float>(VENTILATION, 0.0f);
                room.newVariable<float>(ROOM_QUANTA_CONCENTRATION, 0.0f);
                room.newVariable<unsigned short>(X_CENTER, 0);
                room.newVariable<unsigned short>(Y_CENTER, 0);
                room.newVariable<unsigned short>(Z_CENTER, 0);
            }

            define_room_functions(room, room_type);
        }
    }
}
```

From F4F to FLAME GPU 2: rooms

```
// Define model's agents room
void define_room(ModelDescription& model){
    // Room agents
    string rooms_types[NUM_ROOMS_TYPES] = ROOMS;

    for(string room_type: rooms_types){
        {
            AgentDescription room = model.newAgent(room_type);
            room.newVariable<float>(X, 0.0f);
            room.newVariable<float>(Y, 0.0f);
            room.newVariable<float>(Z, 0.0f);
            room.newVariable<float>(LENGTH_OBJ, 0.0f);
            room.newVariable<float>(WIDTH_OBJ, 0.0f);
            room.newVariable<float>(HEIGHT_OBJ, 0.0f);
            room.newVariable<float>(YAW, 0.0f);
            room.newVariable<unsigned char>(INIT_ROOM, 0);
            room.newVariable<int>(AREA, -1);
            room.newVariable<int>(COLOR_ID, 0);
            if(room_type != FILLINGROOM_AGENT_STRING){
                room.newVariable<float>(VOLUME, 0.0f);
                room.newVariable<float>(VENTILATION, 0.0f);
                room.newVariable<float>(ROOM_QUANTA_CONCENTRATION, 0.0f);
                room.newVariable<unsigned short>(X_CENTER, 0);
                room.newVariable<unsigned short>(Y_CENTER, 0);
                room.newVariable<unsigned short>(Z_CENTER, 0);
            }

            define_room_functions(room, room_type);
        }
    }
}
```



The room's volume is important for the aerosol transmission.

From F4F to FLAME GPU 2: rooms

```
// Define model's agents room
void define_room(ModelDescription& model){
    // Room agents
    string rooms_types[NUM_ROOMS_TYPES] = ROOMS;

    for(string room_type: rooms_types){
        {
            AgentDescription room = model.newAgent(room_type);
            room.newVariable<float>(X, 0.0f);
            room.newVariable<float>(Y, 0.0f);
            room.newVariable<float>(Z, 0.0f);
            room.newVariable<float>(LENGTH_OBJ, 0.0f);
            room.newVariable<float>(WIDTH_OBJ, 0.0f);
            room.newVariable<float>(HEIGHT_OBJ, 0.0f);
            room.newVariable<float>(YAW, 0.0f);
            room.newVariable<unsigned char>(INIT_ROOM, 0);
            room.newVariable<int>(AREA, -1);
            room.newVariable<int>(COLOR_ID, 0);
            if(room_type != FILLINGROOM_AGENT_STRING){
                room.newVariable<float>(VOLUME, 0.0f);
                room.newVariable<float>(VENTILATION, 0.0f);
                room.newVariable<float>(ROOM_QUANTA_CONCENTRATION, 0.0f);
                room.newVariable<unsigned short>(X_CENTER, 0);
                room.newVariable<unsigned short>(Y_CENTER, 0);
                room.newVariable<unsigned short>(Z_CENTER, 0);
            }

            define_room_functions(room, room_type);
        }
    }
}
```

Virus concentration inside the room

From F4F to FLAME GPU 2: rooms

```
// Define model's agents room
void define_room(ModelDescription& model){
    // Room agents
    string rooms_types[NUM_ROOMS_TYPES] = ROOMS;

    for(string room_type: rooms_types){
        {
            AgentDescription room = model.newAgent(room_type);
            room.newVariable<float>(X, 0.0f);
            room.newVariable<float>(Y, 0.0f);
            room.newVariable<float>(Z, 0.0f);
            room.newVariable<float>(LENGTH_OBJ, 0.0f);
            room.newVariable<float>(WIDTH_OBJ, 0.0f);
            room.newVariable<float>(HEIGHT_OBJ, 0.0f);
            room.newVariable<float>(YAW, 0.0f);
            room.newVariable<unsigned char>(INIT_ROOM, 0);
            room.newVariable<int>(AREA, -1);
            room.newVariable<int>(COLOR_ID, 0);
            if(room_type != FILLINGROOM_AGENT_STRING){
                room.newVariable<float>(VOLUME, 0.0f);
                room.newVariable<float>(VENTILATION, 0.0f);
                room.newVariable<float>(ROOM_QUANTA_CONCENTRATION, 0.0f);
                room.newVariable<unsigned short>(X_CENTER, 0);
                room.newVariable<unsigned short>(Y_CENTER, 0);
                room.newVariable<unsigned short>(Z_CENTER, 0);
            }

            define_room_functions(room, room_type);
        }
    }
}
```

From F4F to FLAME GPU 2: pedestrian

```
// Define model's agents pedestrian
void define_pedestrian(ModelDescription& model){
    // Pedestrian agents
    AgentDescription pedestrian = model.newAgent("pedestrian");

    pedestrian.newVariable<unsigned short>(CUDA_INITIALIZED);
    pedestrian.newVariable<float>(X);
    pedestrian.newVariable<float>(Y);
    pedestrian.newVariable<float>(Z);
    pedestrian.newVariable<float>(ANIMATE);
    pedestrian.newVariable<float>(VELX);
    pedestrian.newVariable<float>(VELY);
    pedestrian.newVariable<float>(VELZ);
    pedestrian.newVariable<float>(QUANTA_INHALED);
    pedestrian.newVariable<float, 3>(FINAL_TARGET);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_X);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Y);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Z);
    pedestrian.newVariable<unsigned short, SOLUTION_LENGTH>(STAY);
    pedestrian.newVariable<unsigned short>(NEXT_INDEX);
    pedestrian.newVariable<unsigned short>(TARGET_INDEX);
    pedestrian.newVariable<unsigned short>(FLOW_INDEX);
    pedestrian.newVariable<unsigned short>(INCUBATION_DAYS);
    pedestrian.newVariable<unsigned short>(INFECTION_DAYS);
    pedestrian.newVariable<unsigned short>(FATALITY_DAYS);
    pedestrian.newVariable<unsigned short>(END_OF_IMMUNIZATION_DAYS);
    pedestrian.newVariable<unsigned char>(INIT);
    pedestrian.newVariable<unsigned int>(INFECTED_CONTACTS_STEPS);
    pedestrian.newVariable<short>(ANIMATE_DIR, 1);
    pedestrian.newVariable<short>(CONTACTS_ID, -1);
    pedestrian.newVariable<int>(DISEASE_STATE);
    pedestrian.newVariable<int>(MASK_TYPE, NO_MASK);
    pedestrian.newVariable<int>(ROOM_FOR_QUARANTINE_INDEX, -1);
    pedestrian.newVariable<unsigned int, SOLUTION_LENGTH>(FLOW_STAY_RANDOM);
    pedestrian.newVariable<int>(AGENT_TYPE, -1);
    pedestrian.newVariable<unsigned short>(AGENT_WITH_A_RATE, AGENT_WITHOUT_RATE);
    pedestrian.newVariable<unsigned short>(SEVERITY, MINOR);
    pedestrian.newVariable<unsigned short>(QUARANTINE);
    pedestrian.newVariable<unsigned short>(IDENTIFIED_INFECTED, NOT_IDENTIFIED);
    pedestrian.newVariable<int>(SWAB_STEPS, -1);
    pedestrian.newVariable<unsigned short>(ENTRY_TIME_INDEX);
    pedestrian.newVariable<unsigned short>(JUST_EXITED_FROM_QUARANTINE);

    define_pedestrian_functions(pedestrian);
}
```

From F4F to FLAME GPU 2: pedestrian

```
// Define model's agents pedestrian
void define_pedestrian(ModelDescription& model){
    // Pedestrian agents
    AgentDescription pedestrian = model.newAgent("pedestrian");

    pedestrian.newVariable<unsigned short>(CUDA_INITIALIZED);
    pedestrian.newVariable<float>(X);
    pedestrian.newVariable<float>(Y);
    pedestrian.newVariable<float>(Z);
    pedestrian.newVariable<float>(ANIMATE);
    pedestrian.newVariable<float>(VELX);
    pedestrian.newVariable<float>(VELY);
    pedestrian.newVariable<float>(VELZ);
    pedestrian.newVariable<float>(QUANTA_INHALED);
    pedestrian.newVariable<float, 3>(FINAL_TARGET);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_X);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Y);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Z);
    pedestrian.newVariable<unsigned short, SOLUTION_LENGTH>(STAY);
    pedestrian.newVariable<unsigned short>(NEXT_INDEX);
    pedestrian.newVariable<unsigned short>(TARGET_INDEX);
    pedestrian.newVariable<unsigned short>(FLOW_INDEX);
    pedestrian.newVariable<unsigned short>(INCUBATION_DAYS);
    pedestrian.newVariable<unsigned short>(INFECTION_DAYS);
    pedestrian.newVariable<unsigned short>(FATALITY_DAYS);
    pedestrian.newVariable<unsigned short>(END_OF_IMMUNIZATION_DAYS);
    pedestrian.newVariable<unsigned char>(INIT);
    pedestrian.newVariable<unsigned int>(INFECTED_CONTACTS_STEPS);
    pedestrian.newVariable<short>(ANIMATE_DIR, 1);
    pedestrian.newVariable<short>(CONTACTS_ID, -1);
    pedestrian.newVariable<int>(DISEASE_STATE);
    pedestrian.newVariable<int>(MASK_TYPE, NO_MASK);
    pedestrian.newVariable<int>(ROOM_FOR_QUARANTINE_INDEX, -1);
    pedestrian.newVariable<unsigned int, SOLUTION_LENGTH>(FLOW_STAY_RANDOM);
    pedestrian.newVariable<int>(AGENT_TYPE, -1);
    pedestrian.newVariable<unsigned short>(AGENT_WITH_A_RATE, AGENT_WITHOUT_RATE);
    pedestrian.newVariable<unsigned short>(SEVERITY, MINOR);
    pedestrian.newVariable<unsigned short>(QUARANTINE);
    pedestrian.newVariable<unsigned short>(IDENTIFIED_INFECTED, NOT_IDENTIFIED);
    pedestrian.newVariable<int>(SWAB_STEPS, -1);
    pedestrian.newVariable<unsigned short>(ENTRY_TIME_INDEX);
    pedestrian.newVariable<unsigned short>(JUST_EXITED_FROM_QUARANTINE);

    define_pedestrian_functions(pedestrian);
}
```

Handle movements

Count contacts with infected pedestrian

From F4F to FLAME GPU 2: pedestrian

```
// Define model's agents pedestrian
void define_pedestrian(ModelDescription& model){
    // Pedestrian agents
    AgentDescription pedestrian = model.newAgent("pedestrian");

    pedestrian.newVariable<unsigned short>(CUDA_INITIALIZED);
    pedestrian.newVariable<float>(X);
    pedestrian.newVariable<float>(Y);
    pedestrian.newVariable<float>(Z);
    pedestrian.newVariable<float>(ANIMATE);
    pedestrian.newVariable<float>(VELX);
    pedestrian.newVariable<float>(VELY);
    pedestrian.newVariable<float>(VELZ);
    pedestrian.newVariable<float>(QUANTA_INHALED);
    pedestrian.newVariable<float, 3>(FINAL_TARGET);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_X);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Y);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Z);
    pedestrian.newVariable<unsigned short, SOLUTION_LENGTH>(STAY);
    pedestrian.newVariable<unsigned short>(NEXT_INDEX);
    pedestrian.newVariable<unsigned short>(TARGET_INDEX);
    pedestrian.newVariable<unsigned short>(FLOW_INDEX);
    pedestrian.newVariable<unsigned short>(INCUBATION_DAYS);
    pedestrian.newVariable<unsigned short>(INFECTION_DAYS);
    pedestrian.newVariable<unsigned short>(FATALITY_DAYS);
    pedestrian.newVariable<unsigned short>(END_OF_IMMUNIZATION_DAYS);
    pedestrian.newVariable<unsigned char>(INIT);
    pedestrian.newVariable<unsigned int>(INFECTED_CONTACTS_STEPS);
    pedestrian.newVariable<short>(ANIMATE_DIR, 1);
    pedestrian.newVariable<short>(CONTACTS_ID, -1);
    pedestrian.newVariable<int>(DISEASE_STATE);
    pedestrian.newVariable<int>(MASK_TYPE, NO_MASK);
    pedestrian.newVariable<int>(ROOM_FOR_QUARANTINE_INDEX, -1);
    pedestrian.newVariable<unsigned int, SOLUTION_LENGTH>(FLOW_STAY_RANDOM);
    pedestrian.newVariable<int>(AGENT_TYPE, -1);
    pedestrian.newVariable<unsigned short>(AGENT_WITH_A_RATE, AGENT_WITHOUT_RATE);
    pedestrian.newVariable<unsigned short>(SEVERITY, MINOR);
    pedestrian.newVariable<unsigned short>(QUARANTINE);
    pedestrian.newVariable<unsigned short>(IDENTIFIED_INFECTED, NOT_IDENTIFIED);
    pedestrian.newVariable<int>(SWAB_STEPS, -1);
    pedestrian.newVariable<unsigned short>(ENTRY_TIME_INDEX);
    pedestrian.newVariable<unsigned short>(JUST_EXITED_FROM_QUARANTINE);

    define_pedestrian_functions(pedestrian);
}
```

Agent's disease state

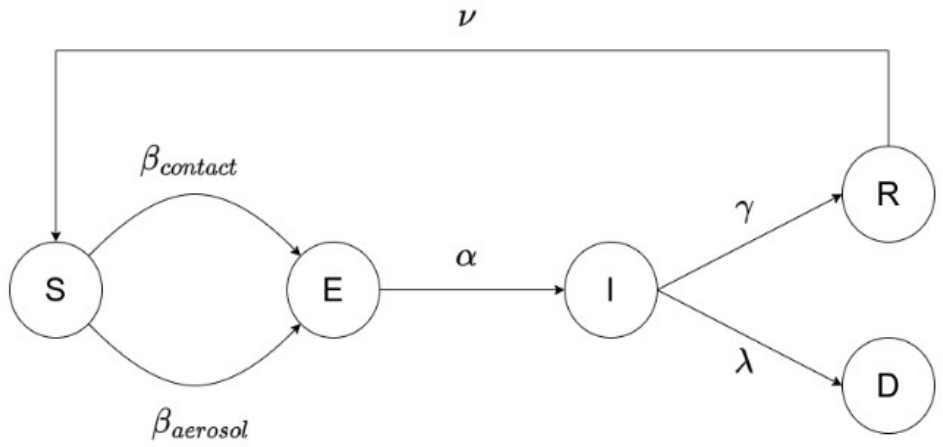
Handle quarantine

Handle swabs

From F4F to FLAME GPU 2: pedestrian

```
// Define model's agents pedestrian
void define_pedestrian(ModelDescription& model){
    // Pedestrian agents
    AgentDescription pedestrian = model.newAgent("pedestrian");

    pedestrian.newVariable<unsigned short>(CUDA_INITIALIZED);
    pedestrian.newVariable<float>(X);
    pedestrian.newVariable<float>(Y);
    pedestrian.newVariable<float>(Z);
    pedestrian.newVariable<float>(ANIMATE);
    pedestrian.newVariable<float>(VELX);
    pedestrian.newVariable<float>(VELY);
    pedestrian.newVariable<float>(VELZ);
    pedestrian.newVariable<float>(QUANTA_INHALED);
    pedestrian.newVariable<float, 3>(FINAL_TARGET);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_X);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Y);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Z);
    pedestrian.newVariable<unsigned short, SOLUTION_LENGTH>(STAY);
    pedestrian.newVariable<unsigned short>(NEXT_INDEX);
    pedestrian.newVariable<unsigned short>(TARGET_INDEX);
    pedestrian.newVariable<unsigned short>(FLOW_INDEX);
    pedestrian.newVariable<unsigned short>(INCUBATION_DAYS);
    pedestrian.newVariable<unsigned short>(INFECTION_DAYS);
    pedestrian.newVariable<unsigned short>(FATALITY_DAYS);
    pedestrian.newVariable<unsigned short>(END_OF_IMMUNIZATION_DAYS);
    pedestrian.newVariable<unsigned char>(INIT);
    pedestrian.newVariable<unsigned int>(INFECTED_CONTACTS_STEPS);
    pedestrian.newVariable<short>(ANIMATE_DIR, 1);
    pedestrian.newVariable<short>(CONTACTS_ID, -1);
    pedestrian.newVariable<int>(DISEASE_STATE);
    pedestrian.newVariable<int>(MASK_TYPE, NO_MASK);
    pedestrian.newVariable<int>(ROOM_FOR_QUARANTINE_INDEX, -1);
    pedestrian.newVariable<unsigned int, SOLUTION_LENGTH>(FLOW_STAY_RANDOM);
    pedestrian.newVariable<int>(AGENT_TYPE, -1);
    pedestrian.newVariable<unsigned short>(AGENT_WITH_A_RATE, AGENT_WITHOUT_RATE);
    pedestrian.newVariable<unsigned short>(SEVERITY, MINOR);
    pedestrian.newVariable<unsigned short>(QUARANTINE);
    pedestrian.newVariable<unsigned short>(IDENTIFIED_INFECTED, NOT_IDENTIFIED);
    pedestrian.newVariable<int>(SWAB_STEPS, -1);
    pedestrian.newVariable<unsigned short>(ENTRY_TIME_INDEX);
    pedestrian.newVariable<unsigned short>(JUST_EXITED_FROM_QUARANTINE);

    define_pedestrian_functions(pedestrian);
}
```

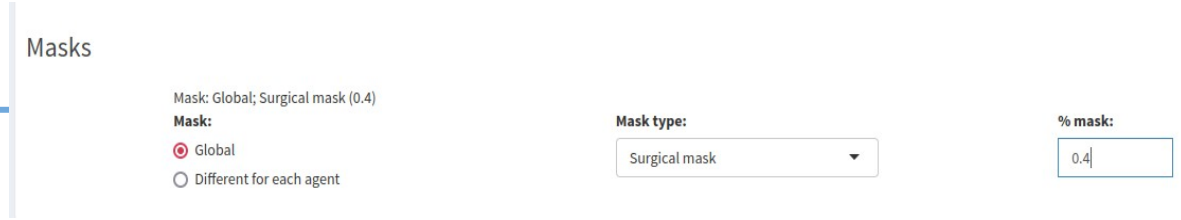
The model is used to calculate the incubation days, the days of infection, the fatality and/or the recovery of the agent.



From F4F to FLAME GPU 2: pedestrian

```
// Define model's agents pedestrian
void define_pedestrian(ModelDescription& model){
    // Pedestrian agents
    AgentDescription pedestrian = model.newAgent("pedestrian");

    pedestrian.newVariable<unsigned short>(CUDA_INITIALIZED);
    pedestrian.newVariable<float>(X);
    pedestrian.newVariable<float>(Y);
    pedestrian.newVariable<float>(Z);
    pedestrian.newVariable<float>(ANIMATE);
    pedestrian.newVariable<float>(VELX);
    pedestrian.newVariable<float>(VELY);
    pedestrian.newVariable<float>(VELZ);
    pedestrian.newVariable<float>(QUANTA_INHALED);
    pedestrian.newVariable<float, 3>(FINAL_TARGET);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_X);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Y);
    pedestrian.newVariable<float, SOLUTION_LENGTH>(INTERMEDIATE_TARGET_Z);
    pedestrian.newVariable<unsigned short, SOLUTION_LENGTH>(STAY);
    pedestrian.newVariable<unsigned short>(NEXT_INDEX);
    pedestrian.newVariable<unsigned short>(TARGET_INDEX);
    pedestrian.newVariable<unsigned short>(FLOW_INDEX);
    pedestrian.newVariable<unsigned short>(INCUBATION_DAYS);
    pedestrian.newVariable<unsigned short>(INFECTION_DAYS);
    pedestrian.newVariable<unsigned short>(FATALITY_DAYS);
    pedestrian.newVariable<unsigned short>(END_OF_IMMUNIZATION_DAYS);
    pedestrian.newVariable<unsigned char>(INIT);
    pedestrian.newVariable<unsigned int>(INFECTED_CONTACTS_STEPS);
    pedestrian.newVariable<short>(ANIMATE_DIR, 1);
    pedestrian.newVariable<short>(CONTACTS_ID, -1);
    pedestrian.newVariable<int>(DISEASE_STATE);
    pedestrian.newVariable<int>(MASK_TYPE, NO_MASK);
    pedestrian.newVariable<int>(ROOM_FOR_QUARANTINE_INDEX, -1);
    pedestrian.newVariable<unsigned int, SOLUTION_LENGTH>(FLOW_STAY_RANDOM);
    pedestrian.newVariable<int>(AGENT_TYPE, -1);
    pedestrian.newVariable<unsigned short>(AGENT_WITH_A_RATE, AGENT_WITHOUT_RATE);
    pedestrian.newVariable<unsigned short>(SEVERITY, MINOR);
    pedestrian.newVariable<unsigned short>(QUARANTINE);
    pedestrian.newVariable<unsigned short>(IDENTIFIED_INFECTED, NOT_IDENTIFIED);
    pedestrian.newVariable<int>(SWAB_STEPS, -1);
    pedestrian.newVariable<unsigned short>(ENTRY_TIME_INDEX);
    pedestrian.newVariable<unsigned short>(JUST_EXITED_FROM_QUARANTINE);

    define_pedestrian_functions(pedestrian);
}
```

The quanta inhaled are important for the aerosol transmission.



From F4F to FLAME GPU 2: environment variables

```
// Environment properties
env.newProperty<unsigned int>(SEED, 123456789);

env.newProperty<unsigned short, V>(INDEX2COORDX, {0});
env.newProperty<unsigned short, V>(INDEX2COORDY, {0});
env.newProperty<unsigned short, V>(INDEX2COORDZ, {0});
env.newProperty<short, V>(NODE_TYPE, {0});
env.newProperty<float, V>(NODE_YAW, {0.0f});
env.newProperty<int, V>(NODE_LENGTH, {0});
env.newProperty<int, V>(NODE_WIDTH, {0});

env.newProperty<float, 4>(EXTERN_RANGES, {0.0f});

env.newProperty<int, TOTAL_AGENTS_OVERESTIMATION>(CONTACTS_TYPES, {0});

env.newProperty<unsigned short>(EXTERN_NODE, 0);
env.newProperty<short>(NEXT_CONTACTS_ID, 0);
env.newProperty<unsigned short>(DAY, 1);
env.newProperty<unsigned short>(WEEK_DAY, 0);
env.newProperty<unsigned short, 3>(MEAN_INCUBATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_INFECTION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_END_OF_IMMUNIZATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_FATALITY_DAYS, {0});

env.newProperty<float, 3>(EXHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float, 3>(INHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float>(CONTAMINATION_RISK, 0.0f);
env.newProperty<float>(CONTAMINATION_RISK_DECREASED_WITH_MASK, 0.0f);
env.newProperty<float>(NGEN_BASE, 0.0f);
env.newProperty<float>(VL, 0.0f);
env.newProperty<float>(VIRUS_VARIANT_FACTOR, 0.0f);
env.newProperty<float>(VIRUS_SEVERITY, 0.0f);
env.newProperty<float>(DECAY_RATE, 0.0f);
env.newProperty<float>(GRAVITATIONAL_SETTLING_RATE, 0.0f);
env.newProperty<float>(INHALATION_RATE_PURE, 0.0f);
env.newProperty<float>(RISK_CONST, 0.0f);

env.newProperty<float, DAYS>(PERC_INF, {0.0f});

env.newProperty<float>(SENSITIVITY_SWAB, 0.0f);
env.newProperty<float>(SPECIFICITY_SWAB, 0.0f);
```

From room's graph vertex index to coordinates

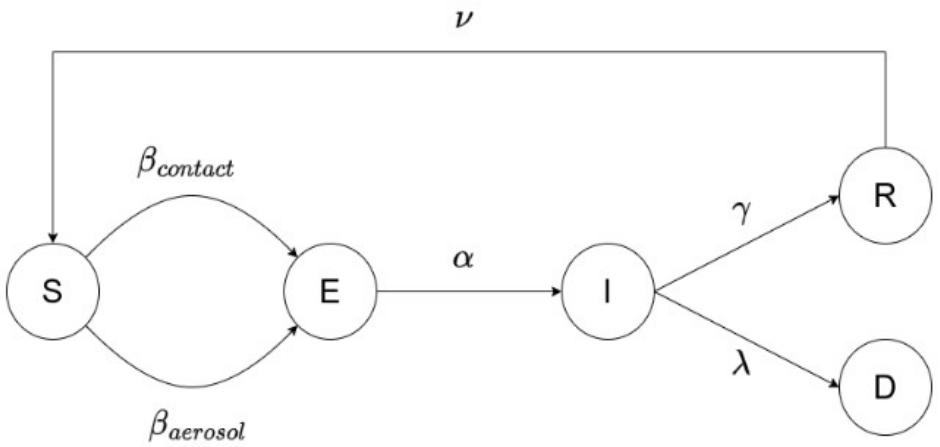
Masks efficacies

Virus' parameters

From F4F to FLAME GPU 2: environment variables

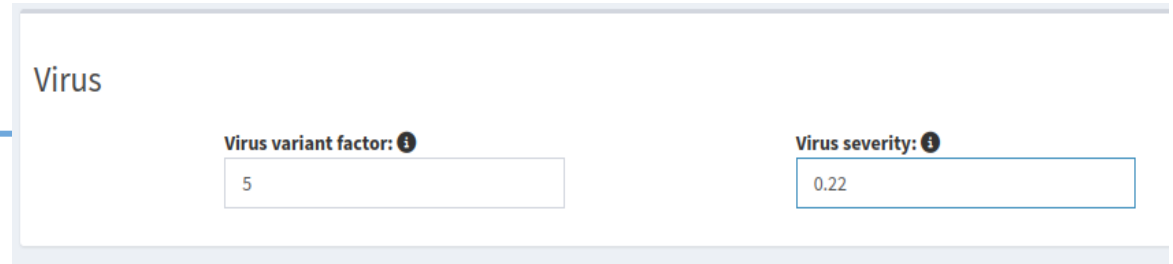
```
// Environment properties
env.newProperty<unsigned int>(SEED, 123456789);

env.newProperty<unsigned short, V>(INDEX2COORDX, {0});
env.newProperty<unsigned short, V>(INDEX2COORDY, {0});
env.newProperty<unsigned short, V>(INDEX2COORDZ, {0});
env.newProperty<short, V>(NODE_TYPE, {0});
env.newProperty<float, V>(NODE_YAW, {0.0f});
env.newProperty<int, V>(NODE_LENGTH, {0});
env.newProperty<int, V>(NODE_WIDTH, {0});

env.newProperty<float, 4>(EXTERN_RANGES, {0.0f});

env.newProperty<int, TOTAL_AGENTS_OVERESTIMATION>(CONTACTS_TYPES, {0});

env.newProperty<unsigned short>(EXTERN_NODE, 0);
env.newProperty<short>(NEXT_CONTACTS_ID, 0);
env.newProperty<unsigned short>(DAY, 1);
env.newProperty<unsigned short>(WEEK_DAY, 0);
env.newProperty<unsigned short, 3>(MEAN_INCUBATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_INFECTION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_END_OF_IMMUNIZATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_FATALITY_DAYS, {0});

env.newProperty<float, 3>(EXHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float, 3>(INHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float>(CONTAMINATION_RISK, 0.0f);
env.newProperty<float>(CONTAMINATION_RISK_DECREASED_WITH_MASK, 0.0f);
env.newProperty<float>(NGEN_BASE, 0.0f);
env.newProperty<float>(VL, 0.0f);
env.newProperty<float>(VIRUS_VARIANT_FACTOR, 0.0f);
env.newProperty<float>(VIRUS_SEVERITY, 0.0f);
env.newProperty<float>(DECAY_RATE, 0.0f);
env.newProperty<float>(GRAVITATIONAL_SETTLING_RATE, 0.0f);
env.newProperty<float>(INHALATION_RATE_PURE, 0.0f);
env.newProperty<float>(RISK_CONST, 0.0f);

env.newProperty<float, DAYS>(PERC_INF, {0.0f});

env.newProperty<float>(SENSITIVITY_SWAB, 0.0f);
env.newProperty<float>(SPECIFICITY_SWAB, 0.0f);
```

Here the model is used to calculate the mean values of incubation, infection etc. of the virus.



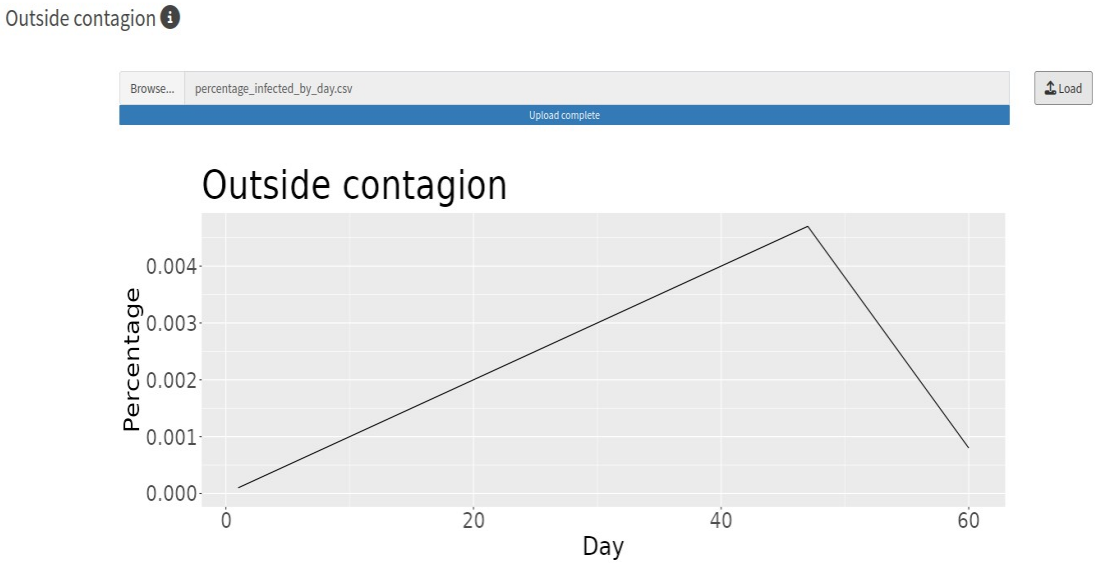
From F4F to FLAME GPU 2: environment variables

```
// Environment properties
env.newProperty<unsigned int>(SEED, 123456789);

env.newProperty<unsigned short, V>(INDEX2COORDX, {0});
env.newProperty<unsigned short, V>(INDEX2COORDY, {0});
env.newProperty<unsigned short, V>(INDEX2COORDZ, {0});
env.newProperty<short, V>(NODE_TYPE, {0});
env.newProperty<float, V>(NODE_YAW, {0.0f});
env.newProperty<int, V>(NODE_LENGTH, {0});
env.newProperty<int, V>(NODE_WIDTH, {0});

env.newProperty<float, 4>(EXTERN_RANGES, {0.0f});

env.newProperty<int, TOTAL_AGENTS_OVERESTIMATION>(CONTACTS_TYPES, {0});

env.newProperty<unsigned short>(EXTERN_NODE, 0);
env.newProperty<short>(NEXT_CONTACTS_ID, 0);
env.newProperty<unsigned short>(DAY, 1);
env.newProperty<unsigned short>(WEEK_DAY, 0);
env.newProperty<unsigned short, 3>(MEAN_INCUBATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_INFECTION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_END_OF_IMMUNIZATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_FATALITY_DAYS, {0});

env.newProperty<float, 3>(EXHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float, 3>(INHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float>(CONTAMINATION_RISK, 0.0f);
env.newProperty<float>(CONTAMINATION_RISK_DECREASED_WITH_MASK, 0.0f);
env.newProperty<float>(NGEN_BASE, 0.0f);
env.newProperty<float>(VL, 0.0f);
env.newProperty<float>(VIRUS_VARIANT_FACTOR, 0.0f);
env.newProperty<float>(VIRUS_SEVERITY, 0.0f);
env.newProperty<float>(DECAY_RATE, 0.0f);
env.newProperty<float>(GRAVITATIONAL_SETTLING_RATE, 0.0f);
env.newProperty<float>(INHALATION_RATE_PURE, 0.0f);
env.newProperty<float>(RISK_CONST, 0.0f);

env.newProperty<float, DAYS>(PERC_INF, {0.0f});

env.newProperty<float>(SENSITIVITY_SWAB, 0.0f);
env.newProperty<float>(SPECIFICITY_SWAB, 0.0f);
```



From F4F to FLAME GPU 2: environment variables

```
// Environment properties
env.newProperty<unsigned int>(SEED, 123456789);

env.newProperty<unsigned short, V>(INDEX2COORDX, {0});
env.newProperty<unsigned short, V>(INDEX2COORDY, {0});
env.newProperty<unsigned short, V>(INDEX2COORDZ, {0});
env.newProperty<short, V>(NODE_TYPE, {0});
env.newProperty<float, V>(NODE_YAW, {0.0f});
env.newProperty<int, V>(NODE_LENGTH, {0});
env.newProperty<int, V>(NODE_WIDTH, {0});

env.newProperty<float, 4>(EXTERN_RANGES, {0.0f});

env.newProperty<int, TOTAL_AGENTS_OVERESTIMATION>(CONTACTS_TYPES, {0});

env.newProperty<unsigned short>(EXTERN_NODE, 0);
env.newProperty<short>(NEXT_CONTACTS_ID, 0);
env.newProperty<unsigned short>(DAY, 1);
env.newProperty<unsigned short>(WEEK_DAY, 0);
env.newProperty<unsigned short, 3>(MEAN_INCUBATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_INFECTION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_END_OF_IMMUNIZATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_FATALITY_DAYS, {0});

env.newProperty<float, 3>(EXHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float, 3>(INHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float>(CONTAMINATION_RISK, 0.0f);
env.newProperty<float>(CONTAMINATION_RISK_DECREASED_WITH_MASK, 0.0f);
env.newProperty<float>(NGEN_BASE, 0.0f);
env.newProperty<float>(VL, 0.0f);
env.newProperty<float>(VIRUS_VARIANT_FACTOR, 0.0f);
env.newProperty<float>(VIRUS_SEVERITY, 0.0f);
env.newProperty<float>(DECAY_RATE, 0.0f);
env.newProperty<float>(GRAVITATIONAL_SETTLING_RATE, 0.0f);
env.newProperty<float>(INHALATION_RATE_PURE, 0.0f);
env.newProperty<float>(RISK_CONST, 0.0f);

env.newProperty<float, DAYS>(PERC_INF, {0.0f});

env.newProperty<float>(SENSITIVITY_SWAB, 0.0f);
env.newProperty<float>(SPECIFICITY_SWAB, 0.0f);
```



From F4F to FLAME GPU 2: environment variables

```
// Environment properties
env.newProperty<unsigned int>(SEED, 123456789);

env.newProperty<unsigned short, V>(INDEX2COORDX, {0});
env.newProperty<unsigned short, V>(INDEX2COORDY, {0});
env.newProperty<unsigned short, V>(INDEX2COORDZ, {0});
env.newProperty<short, V>(NODE_TYPE, {0});
env.newProperty<float, V>(NODE_YAW, {0.0f});
env.newProperty<int, V>(NODE_LENGTH, {0});
env.newProperty<int, V>(NODE_WIDTH, {0});

env.newProperty<float, 4>(EXTERN_RANGES, {0.0f});

env.newProperty<int, TOTAL_AGENTS_OVERESTIMATION>(CONTACTS_TYPES, {0});

env.newProperty<unsigned short>(EXTERN_NODE, 0);
env.newProperty<short>(NEXT_CONTACTS_ID, 0);
env.newProperty<unsigned short>(DAY, 1);
env.newProperty<unsigned short>(WEEK_DAY, 0);
env.newProperty<unsigned short, 3>(MEAN_INCUBATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_INFECTION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_END_OF_IMMUNIZATION_DAYS, {0});
env.newProperty<unsigned short, 3>(MEAN_FATALITY_DAYS, {0});

env.newProperty<float, 3>(EXHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float, 3>(INHALATION_MASK_EFFICACY, {0.0f});
env.newProperty<float>(CONTAMINATION_RISK, 0.0f);
env.newProperty<float>(CONTAMINATION_RISK_DECREASED_WITH_MASK, 0.0f);
env.newProperty<float>(NGEN_BASE, 0.0f);
env.newProperty<float>(VL, 0.0f);
env.newProperty<float>(VIRUS_VARIANT_FACTOR, 0.0f);
env.newProperty<float>(VIRUS_SEVERITY, 0.0f);
env.newProperty<float>(DECAY_RATE, 0.0f);
env.newProperty<float>(GRAVITATIONAL_SETTLING_RATE, 0.0f);
env.newProperty<float>(INHALATION_RATE_PURE, 0.0f);
env.newProperty<float>(RISK_CONST, 0.0f);

env.newProperty<float, DAYS>(PERC_INF, {0.0f});

env.newProperty<float>(SENSITIVITY_SWAB, 0.0f);
env.newProperty<float>(SPECIFICITY_SWAB, 0.0f);
```

From F4F to FLAME GPU 2: macro environment variables

Graph adjacency matrix (used by A*)

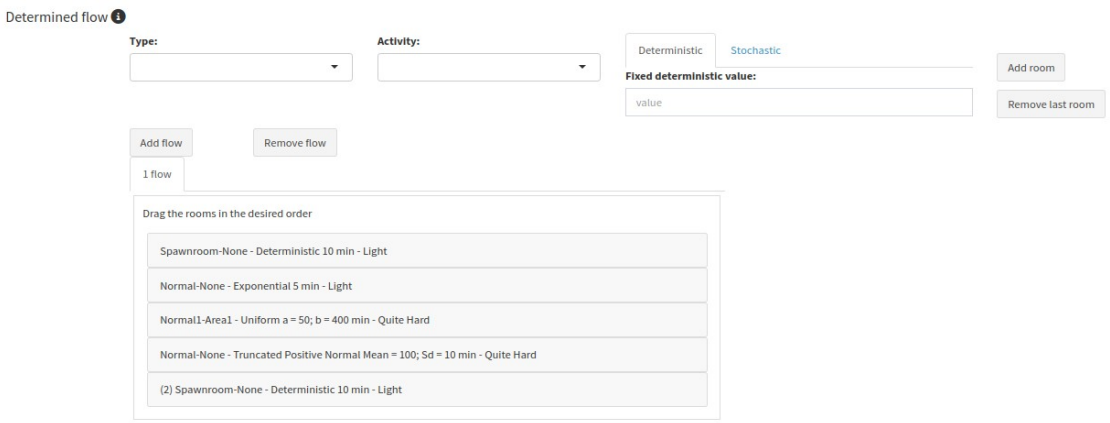
From room's coordinates to graph vertex index

Contacts between every pair of agents

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```

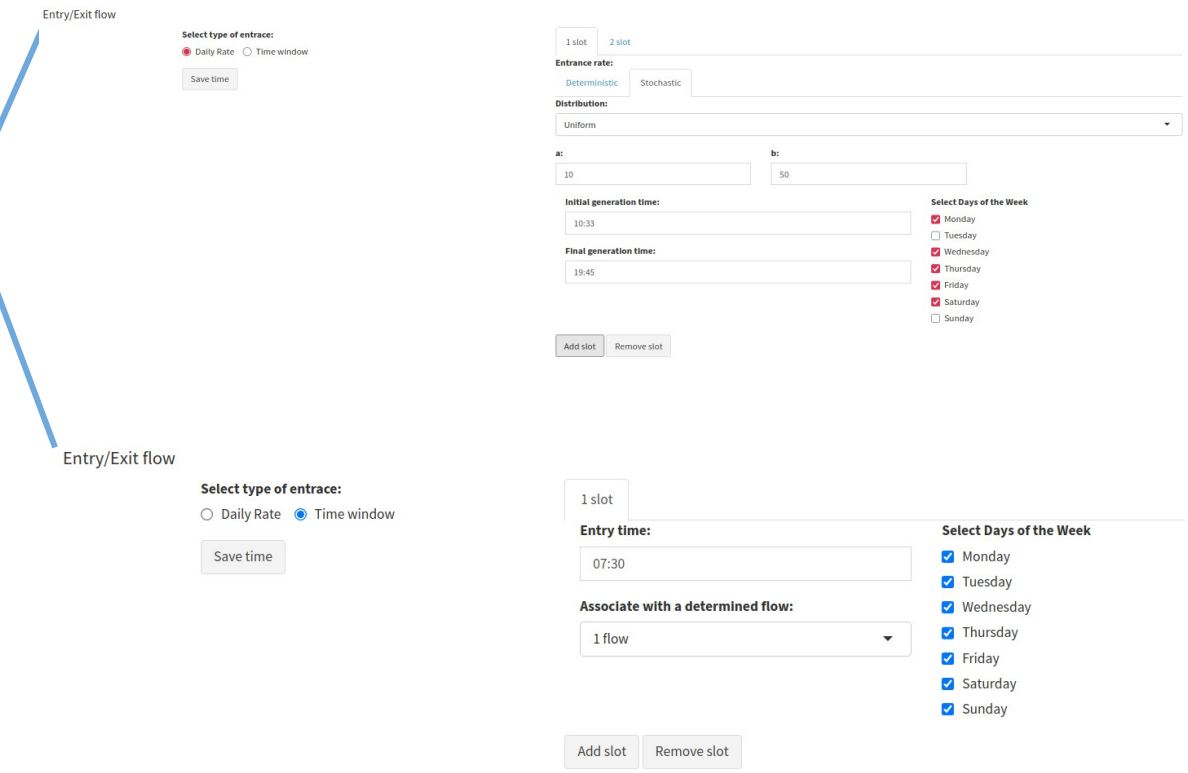
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



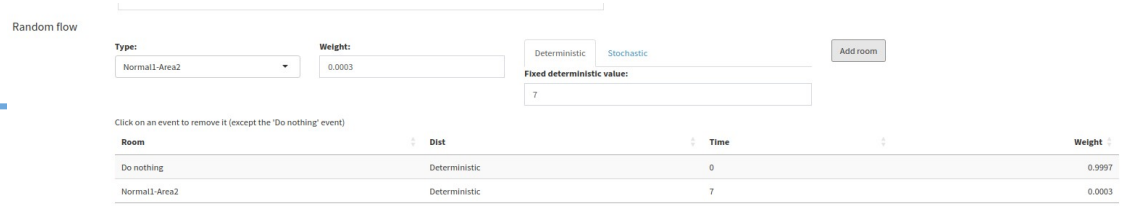
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



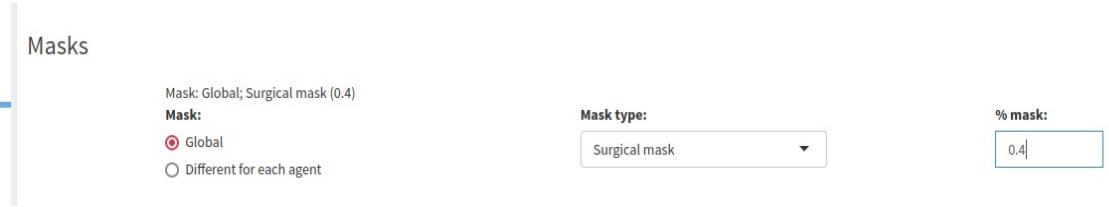
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



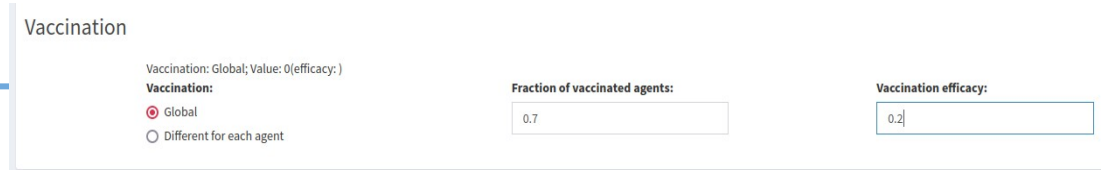
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



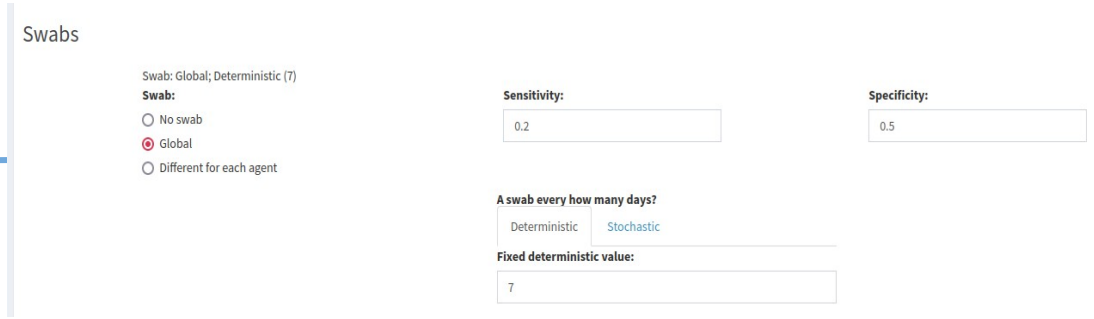
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



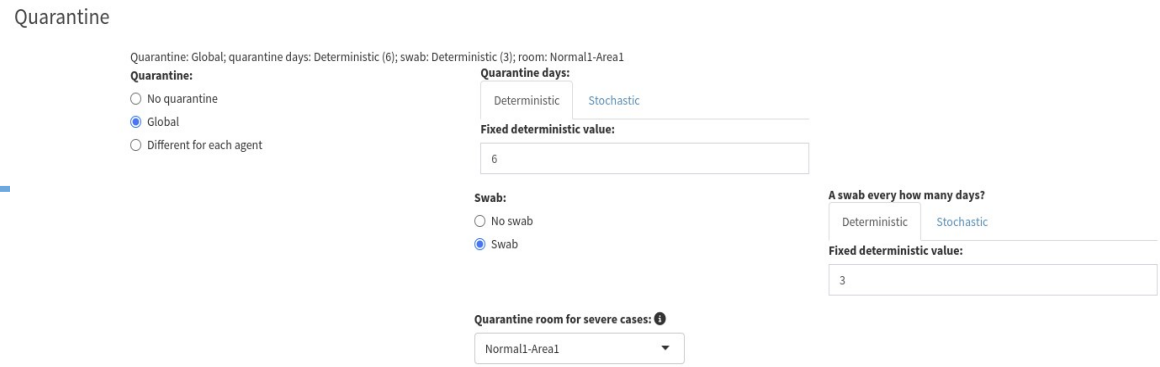
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



From F4F to FLAME GPU 2: macro environment variables

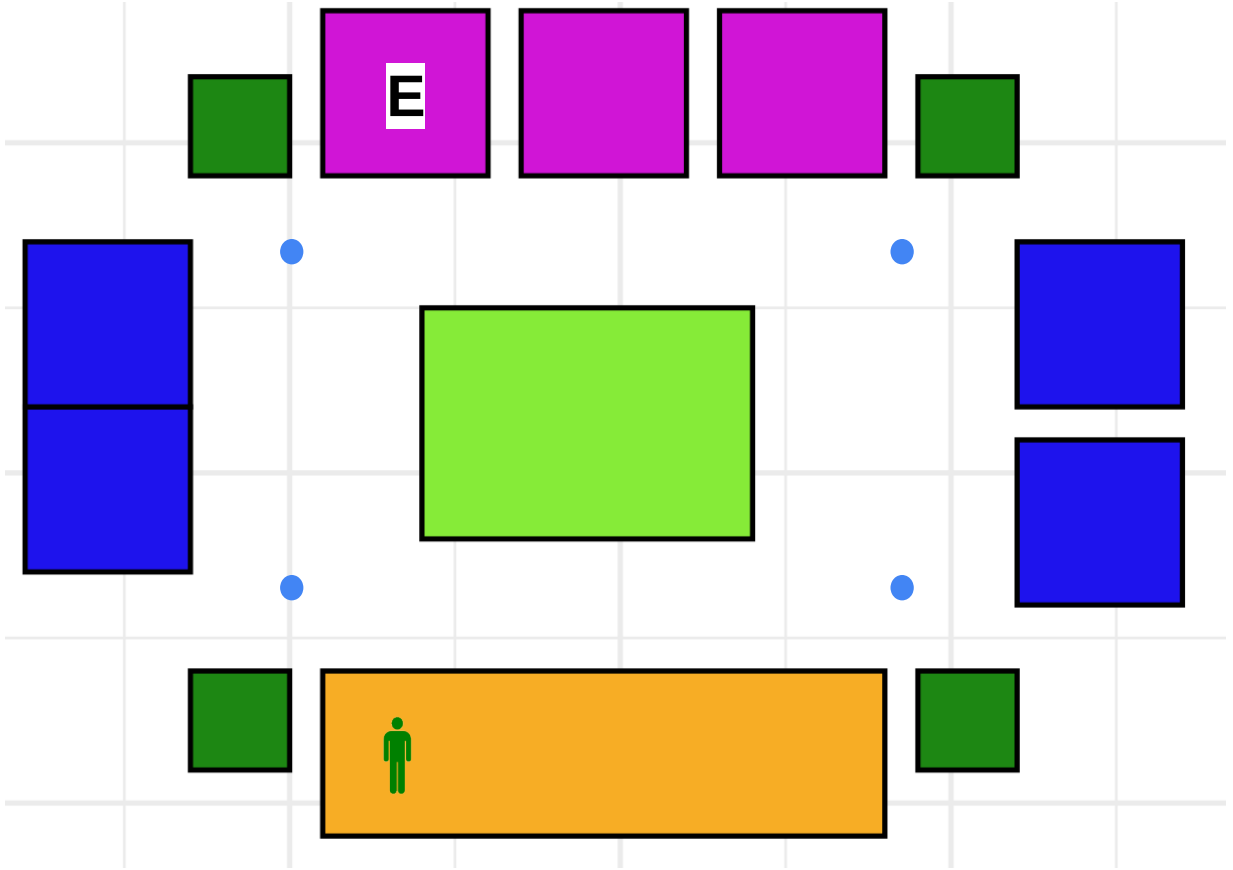
```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```



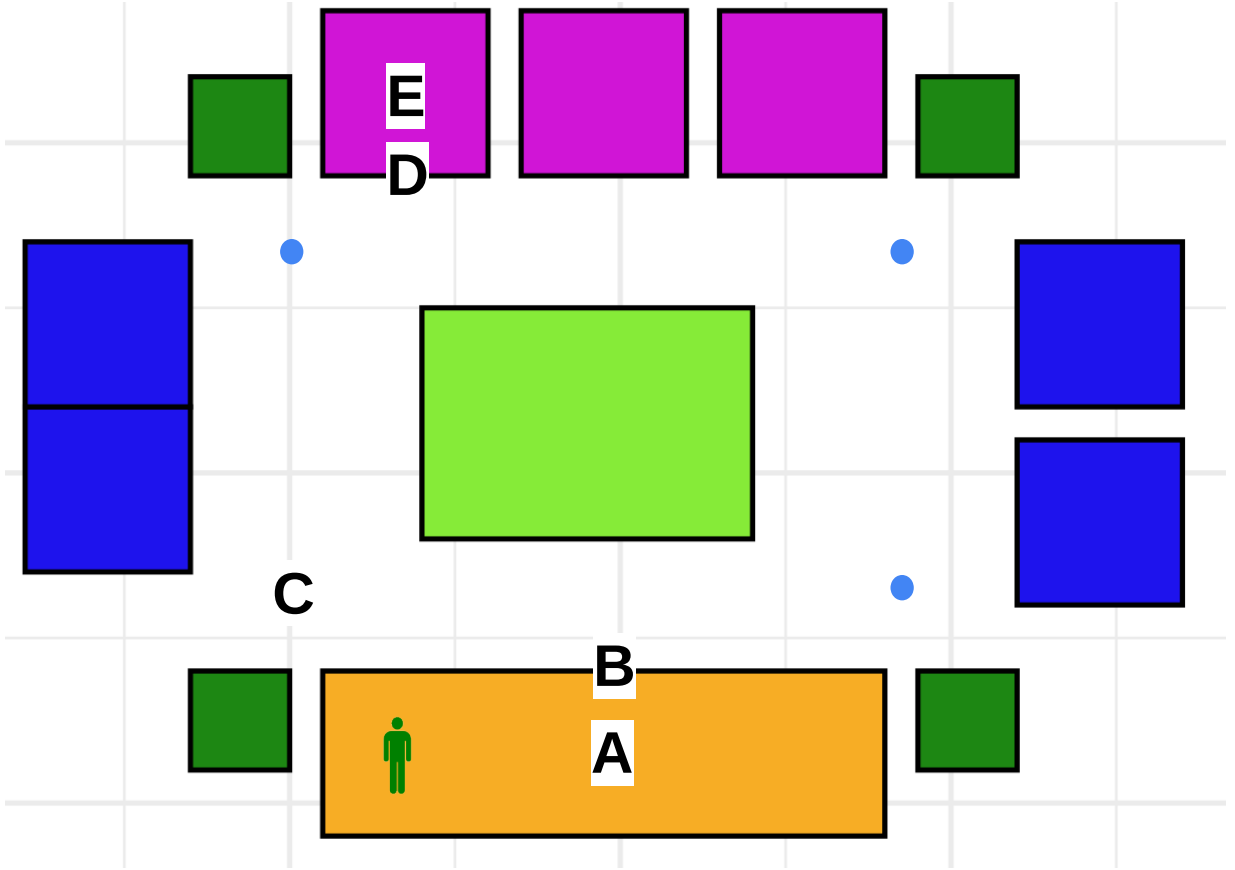
From F4F to FLAME GPU 2: macro environment variables

```
env.newMacroProperty<short, FLOORS, ENV_DIM_Z, ENV_DIM_X>(COORD2INDEX);

env.newMacroProperty<unsigned short, V, V>(ADJMATRIX);

env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_AREA);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_FLOW_DISTR_SECONDPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_ACTIVITY_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_HOURS_SCHEDULE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, DAYS_IN_A_WEEK, SOLUTION_LENGTH>(ENV_BIRTH_RATE_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_CDF);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES, SOLUTION_LENGTH>(ENV_EVENTS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_TYPE);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_MASK_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_FRACTION);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_VACCINATION_EFFICACY);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_SWAB_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_FIRSTPARAM);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_QUARANTINE_SWAB_DAYS_DISTR_SECONDPARAM);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_TYPE);
env.newMacroProperty<int, NUMBER_OF_AGENTS_TYPES>(ENV_ROOM_FOR_QUARANTINE_AREA);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_FIRST);
env.newMacroProperty<float, NUMBER_OF_AGENTS_TYPES>(ENV_EXTERNAL_SCREENING_SECOND);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION, TOTAL_AGENTS_OVERESTIMATION>(NUMBER_OF_STEPS_CONTACTS);

env.newMacroProperty<unsigned int, DISEASE_STATES>(COMPARTMENTAL_MODEL);

env.newMacroProperty<float, V>(ROOMS_QUANTA_CONCENTRATION);

env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES);
env.newMacroProperty<unsigned int, NUMBER_OF_AGENTS_TYPES, V>(GLOBAL_RESOURCES_COUNTER);

env.newMacroProperty<unsigned int, NUM_COUNTERS>(COUNTERS);

env.newMacroProperty<unsigned int, TOTAL_AGENTS_OVERESTIMATION>(CUDA_RNG_OFFSETS_PEDESTRIAN);
env.newMacroProperty<unsigned int, NUM_ROOMS>(CUDA_RNG_OFFSETS_ROOM);
```

From F4F to FLAME GPU 2: pedestrian's flow



From F4F to FLAME GPU 2: pedestrian's flow

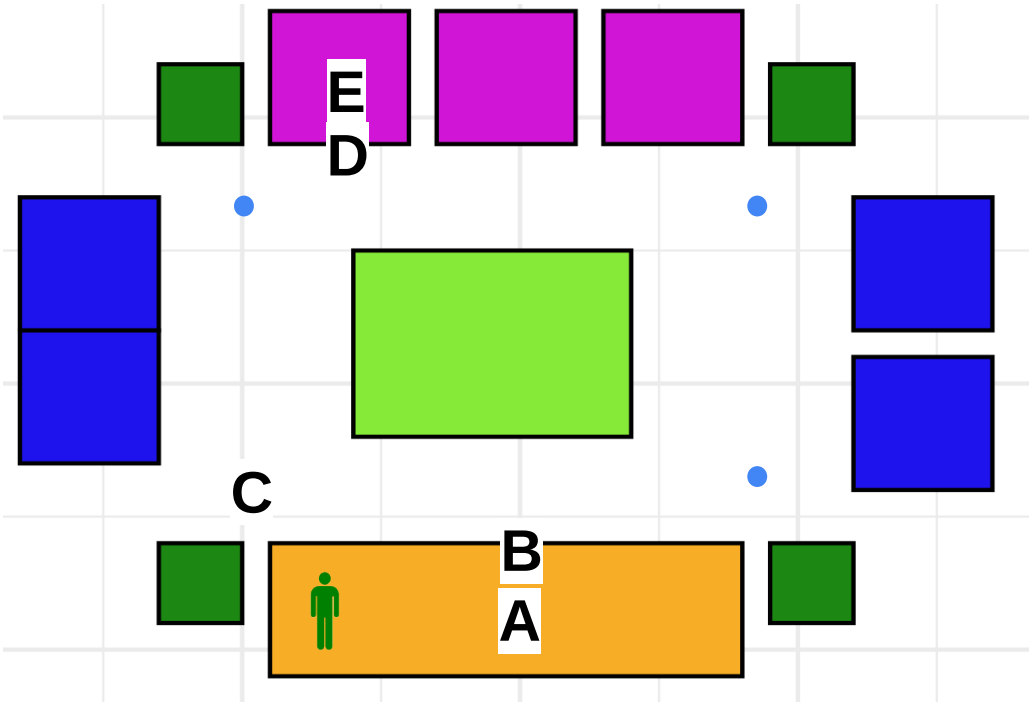


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index

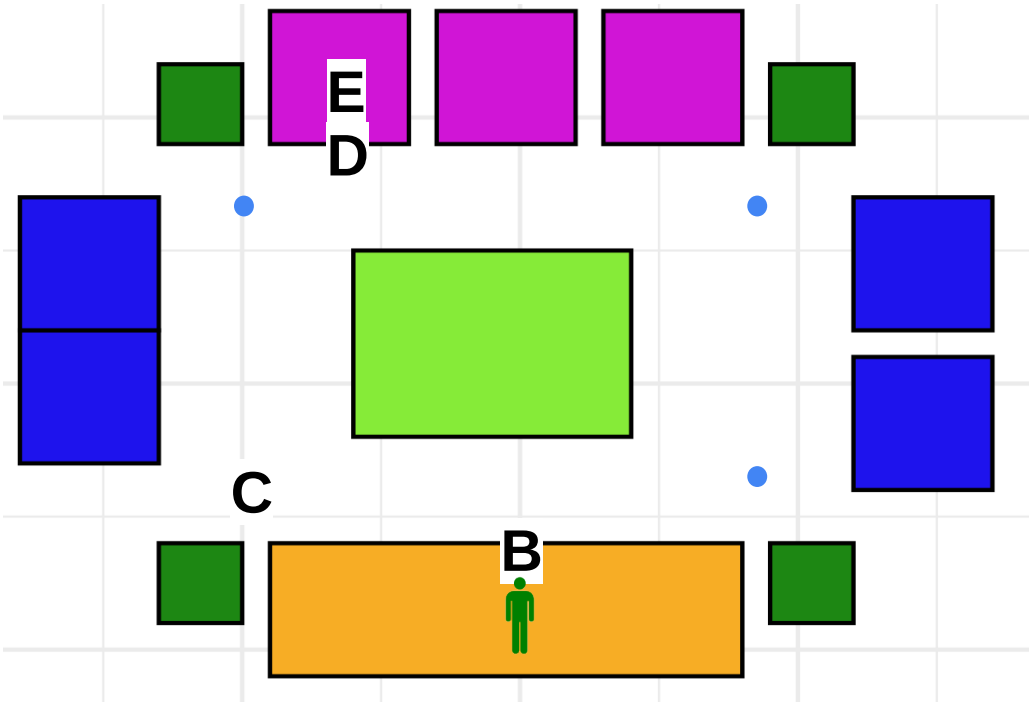


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index

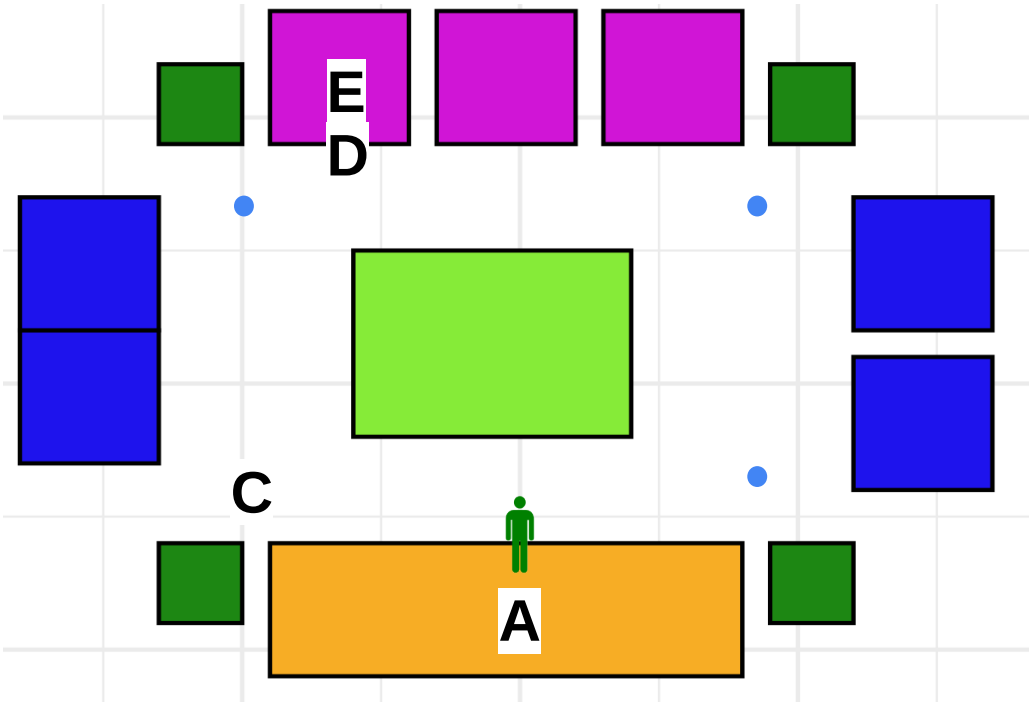


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index

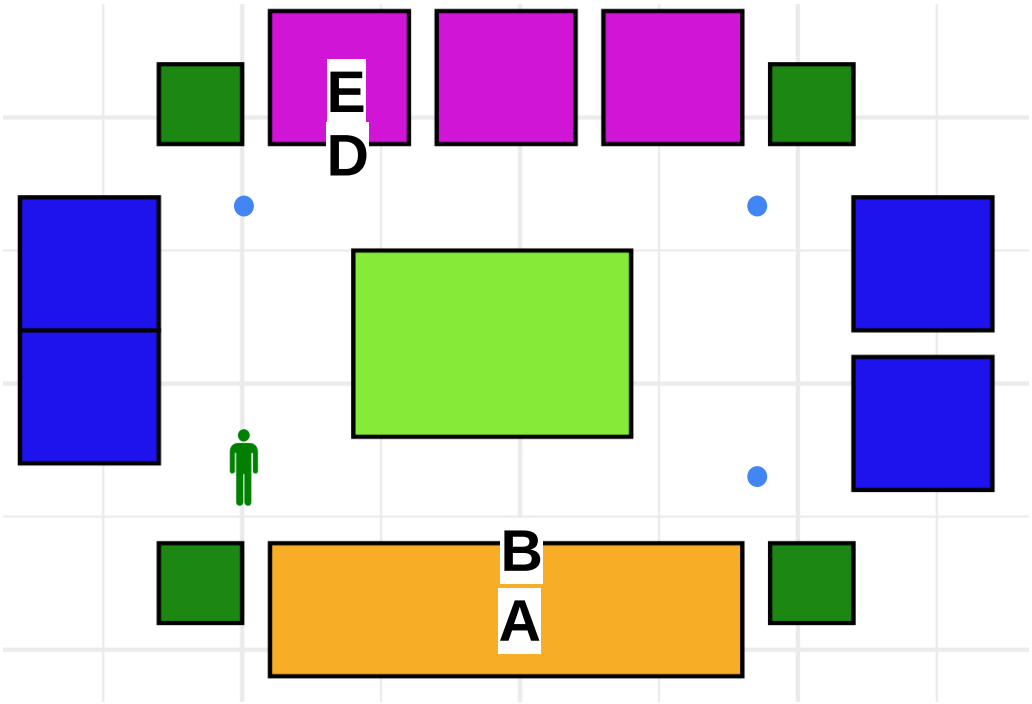


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index

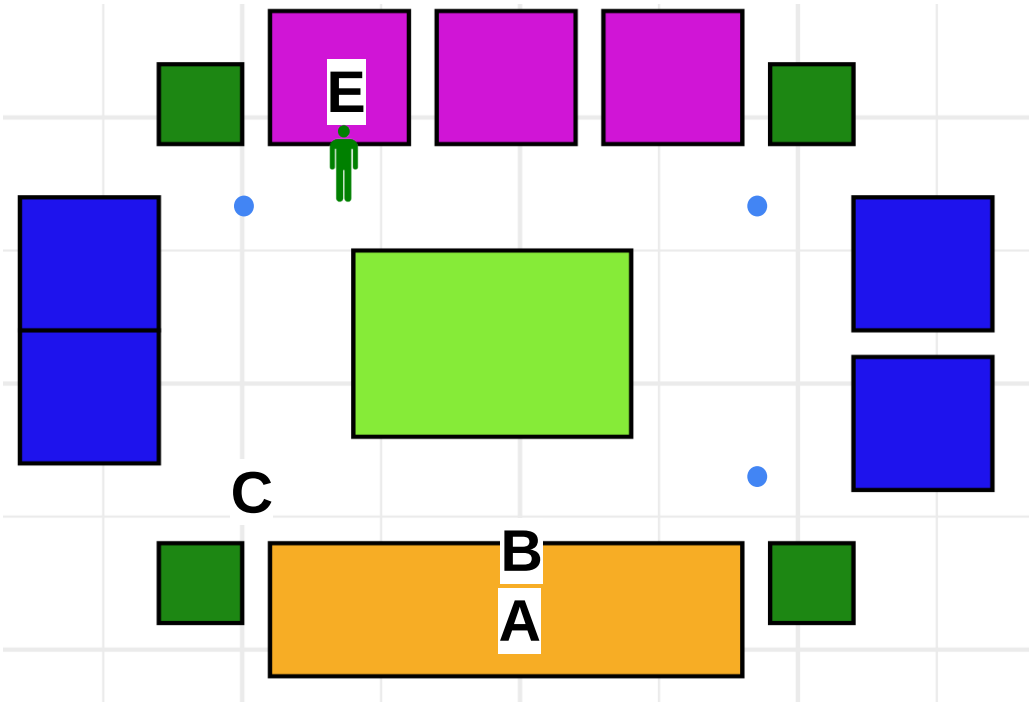


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index

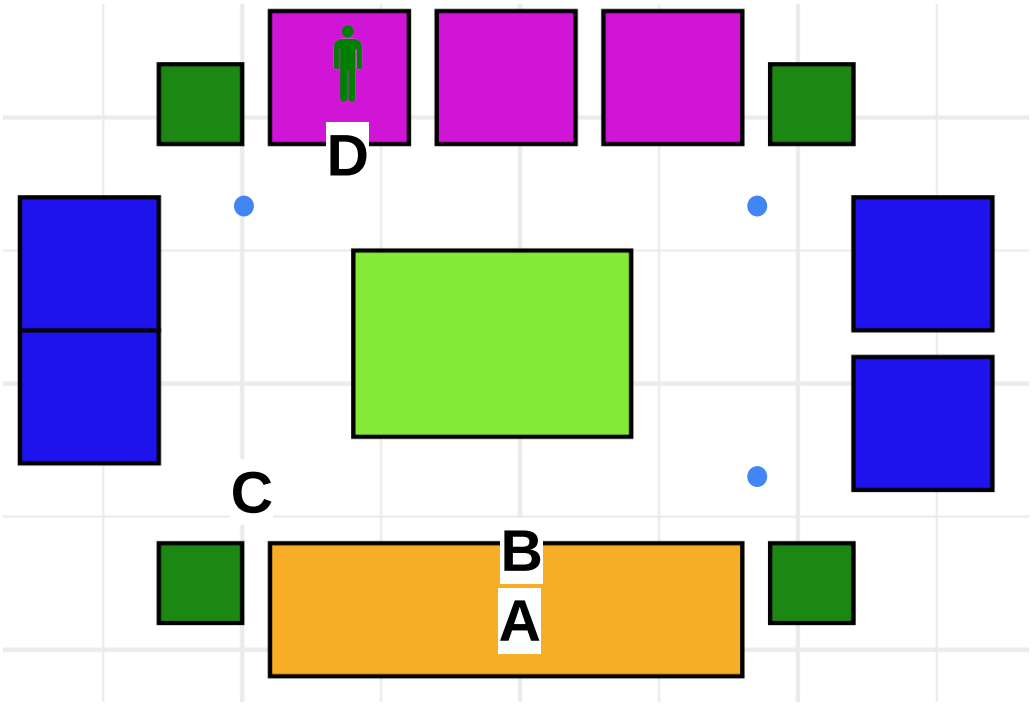


From F4F to FLAME GPU 2: pedestrian's flow

Next index

Intermediate target x	…	A_x	B_x	C_x	D_x	E_x	…	…
Intermediate target y	…	A_y	B_y	C_y	D_y	E_y	…	…
Intermediate target z	…	A_z	B_z	C_z	D_z	E_z	…	…
Stay (steps)	…	0	0	0	0	0	100	…

Target index



From F4F to FLAME GPU 2: pedestrian's messages

```
// Location pedestrian message
MessageSpatial3D::Description pedestrian_message = model.newMessage<MessageSpatial3D>("location");
pedestrian_message.newVariable<id_t>(ID);
pedestrian_message.newVariable<short>(CONTACTS_ID);
pedestrian_message.newVariable<int>(DISEASE_STATE);
pedestrian_message.newVariable<int>(AGENT_TYPE);
pedestrian_message.setRadius(RADIUS);
pedestrian_message.setMin(0, 0, 0);
pedestrian_message.setMax(ENV_DIM_X, ENV_DIM_Y, ENV_DIM_Z);
```

Agent position for counting contacts.

```
// Aerosol counting message
MessageBucket::Description aerosol_message = model.newMessage<MessageBucket>("aerosol_counting");
aerosol_message.newVariable<int>(DISEASE_STATE);
aerosol_message.newVariable<int>(MASK_TYPE);
aerosol_message.newVariable<float>(ACTIVITY_TYPE);
aerosol_message.setBounds(-1, V);
```

Agent position for handling aerosol transmission

From F4F to FLAME GPU 2: room's messages

```
// Room location message
MessageBucket::Description room_message = model.newMessage<MessageBucket>("room_location");
room_message.newVariable<unsigned short>(X);
room_message.newVariable<unsigned short>(Y);
room_message.newVariable<unsigned short>(Z);
room_message.newVariable<short>(GRAPH_NODE);
room_message.newVariable<int>(AREA);
room_message.setBounds(0, V);
room_message.setPersistent(true);
```

Room position for handling events

From F4F to FLAME GPU 2: agent's functions

Pedestrian:

- **FLAMEGPU_AGENT_FUNCTION(CUDAInitContagionMovePedestrianAndGenerateEvent, MessageBucket, MessageNone)**:
  - CUDA RNGs initialization.
  - Contagion processes (aerosol, contacts, and outside contagion) and screening (internal and external).
  - Events.
  - Movements.
- **FLAMEGPU_AGENT_FUNCTION(outputPedestrianLocationAerosol, MessageNone, MessageBucket)**:
  - Output pedestrian position for aerosol.
- **FLAMEGPU_AGENT_FUNCTION(outputPedestrianLocation, MessageNone, MessageSpatial3D)**:
  - Output pedestrian location for contacts.
- **FLAMEGPU_AGENT_FUNCTION(updateQuantaInhaledAndContacts, MessageSpatial3D, MessageNone)** :
  - Update quanta's inhaled by agents and count agents' contacts.

Room:

- **FLAMEGPU_AGENT_FUNCTION(updateQuantaConcentration, MessageBucket, MessageNone)**:
  - Update quanta room's virus concentration.
- **FLAMEGPU_AGENT_FUNCTION(outputRoomLocation, MessageNone, MessageBucket)**:
  - Output room location for events and flows.

From F4F to FLAME GPU 2: host functions

Host:

- **FLAMEGPU_INIT_FUNCTION(initFunction)**:
  - Handles the beginning of the simulation setting the initial day and week day and creating the log files.
- **FLAMEGPU_STEP_FUNCTION(updateDay)**:
  - Update the day of the simulation.
- **FLAMEGPU_STEP_FUNCTION(birth)**:
  - Handles the generation of agents with rate.
- **FLAMEGPU_EXIT_FUNCTION(exitFunction):**
  - Handles the exit from the simulation and generate the log files.

From F4F to FLAME GPU 2: layers

```
void define_layers(ModelDescription& model){
    // Define the execution order
    // Layer 1
    {
        LayerDescription layer = model.newLayer();
        layer.addAgentFunction(CUDAInitContagionMovePedestrianAndGenerateEvent);
    }

    // Layer 2
    {
        LayerDescription layer = model.newLayer();
#ifndef CHECKPOINT
        layer.addAgentFunction(outputPedestrianLocationAerosol);
#endif
        layer.addAgentFunction("room", "outputRoomLocation");
    }

#ifndef CHECKPOINT
    // Layer 3
    {
        LayerDescription layer = model.newLayer();
        layer.addAgentFunction(outputPedestrianLocation);
        layer.addAgentFunction("room", "updateQuantaConcentration");
    }

    // Layer 4
    {
        LayerDescription layer = model.newLayer();
        layer.addAgentFunction(updateQuantaInhaledAndContacts);
    }
#endif

    // Define each host functions
    model.addInitFunction(initFunction);
    model.addStepFunction(updateDay);
    model.addStepFunction(birth);
    model.addExitFunction(exitFunction);
}
```

References

1. Hoertel, N., Blachier, M., Blanco, C., Olfson, M., Massetti, M., Rico, M. S., Limosin, F., & Leleu, H. (2020). A stochastic agent-based model of the SARS-CoV-2 epidemic in France. Nature Medicine, 26(9), 1417–1421. doi:https://doi.org/10.1038/s41591-020-1001-6
2. Gkantonas, S., Zabotti, D., Mesquita, L. C., Mastorakos, E., & de Oliveira, P. M. (2021). airborne.cam: A risk calculator of SARS-CoV-2 aerosol transmission under well-mixed ventilation conditions. Available at: https://airborne.cam
3. J.L. Jimenez and Z. Peng, COVID-19 Aerosol Transmission Estimator. https://tinyurl.com/covid-estimator
4. Tolksdorf K, Buda S, Schuler E, Wieler LH, Haas W. Influenza-associated pneumonia as reference to assess seriousness of coronavirus disease (COVID-19). Euro Surveill. 2020 Mar;25(11):2000258. doi: https://doi.org/10.2807/1560-7917.ES.2020.25.11.2000258. Epub 2020 Mar 16. PMID: 32186278; PMCID: PMC7096775
5. Richmond, P., Chisholm, R., Heywood, P., Leach, M., Chimeh, M. K. FLAME GPU. Version 2.0.0-rc. Dec. 2022. https://doi.org/10.5281/zenodo.7434228
6. Richmond, P., Chisholm, R., Heywood, P., Chimeh, M. K, Leach, M. FLAME GPU 2: A framework for flexible and performant agent based simulation on GPUs. In: Software: Practice and Experience (2023). https://doi.org/10.1002/spe.3207

Acknowledgement

UNIVERSITÀ DI TORINO

di.unito.it DIPARTIMENTO DI INFORMATICA

Early Detection 1 — 2 — Impact Assessment — Forecast Scenarios 3

TrustAlert