The Emgergency Egress Simulation: The Guide

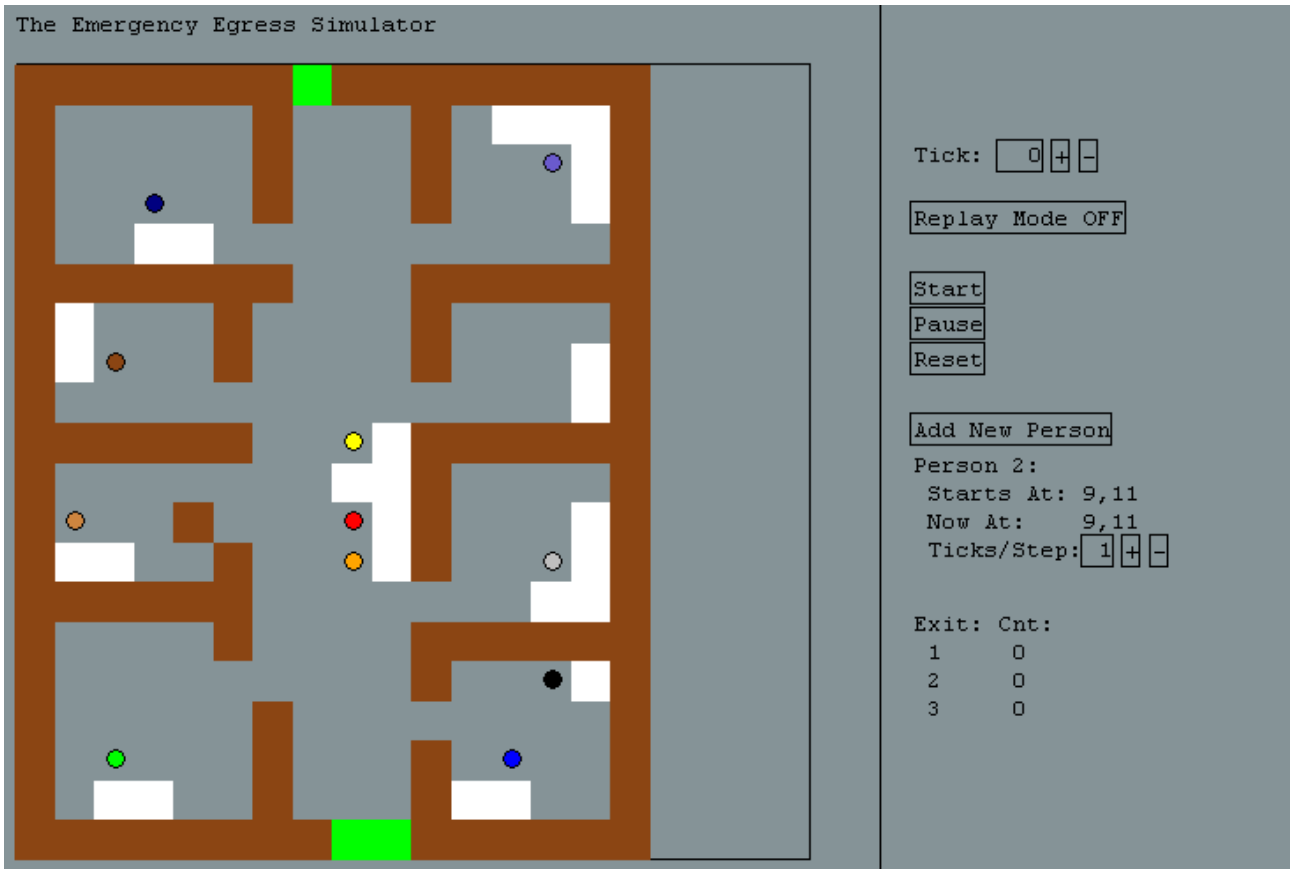
Weighting

I am submitting my work to be marked at 70% Model, 30% Paper.

Running the Model

1. Execute the included `Run.e`, or both of `emergencyexit.e` and `interface.e`
2. Load a `room_design_XXX.e` file of your choice to pre-load different simulations or start creating your own simulation from scratch as described below.

Using the model



The idea of the model is to provide a simulation of people exiting a room or area in an emergency. The user can adjust the properties of the environment to experiment with the effects of changing the placement of walls, people, etc. In the graphical interface:

- Coloured circles are people. A person has properties such as speed and initial position, which can be inspected if you click on them. Click 'Add New Person' to randomly place another person on the model.
- Brown squares are walls – a person can neither see or move through them.
- White squares are obstructions – a person cannot move through them, but they can see over them
- Green squares are the exits.

Click 'Start' to begin the simulation. 'Pause' temporarily halts the simulation, pressing 'Start' again resumes it. When the simulation has finished or is paused you can enter a replay mode by clicking on the 'Replay Mode OFF' button. The tick count can then be incremented and decremented, stepping through the steps that have been made by the people so far. The replay mode is driven entirely by dependency. 'Reset' returns people to their original locations, resets the tick

counter to 0, and clears the memory of people's movement.

A list at the bottom right shows the number of people that go through each exit. More detailed information is available by looking at the console output of tkeden.

Included Room Designs

`room_design_3_cs104.e` – A simulation of the lecture room CS1.04

`room_design_4_cs_corridor.e` – A simulation quite like a corridor in CS.

Creating and Editing Simulations

1. `reset_room(X);` will clear and re-size the simulation area to a grid *X* by *X* cells
2. Left clicking on a cell within the simulation area will cycle that cell between being empty (grey), white (obstruction), and brown (wall).
3. Right clicking on a cell will cause it to become an exit, and clicking again removes the exit.
4. Clicking on a person selects them – clicking and dragging them moves them. (it also re-assigns their start position to that point. To move them temporarily, hold CTRL when dragging (NumLock may need to be off).
5. Once a person is selected you can see their current and start position and adjust their speed.
6. The observable `room_file_output` should contain the necessary script to re-load your room design at your next session.

Extending and Altering the model – observables of interest

`occupancelist`: a two-dimensional array, in which each element corresponds to the current status of each cell on the grid. A positive value represents a the index of a person in `person_positions`. A negative number is: -1:wall, -2:exit, -3:obstruction. Walls and obstructions are defined by changing the requisite value here – exits are defined in `exittargets`, which is a list of exit positions.

`person_positions`: defines each person. Each person is represented by a list, where the first two elements are their current position, the second two are their starting position, the fifth element is the index of exit they used (if they have exited), the sixth is the tick on which they exited, and the seventh is their speed (in terms of how many ticks must pass between each of their possible steps). If the speed is omitted then a value is randomly assigned.

`moveperson`: procedure which defines the decision making process for each step a person takes.

`ticks_per_sec`: controls how many ticks there are per second, and so the overall speed of the animation.

The scripts also contain suggestions in the comments for places where they could be extended.

A note about the AI

Modelling a persons behaviour has been an extremely challenging exercise. Whilst the behaviour in this model is complex, there is always more that could be done to improve it. The model has the ability to determine the whether a person can see an exit or not, which took a good deal of effort to get to work, and is core to a persons behaviour. Because of the unforeseen difficulty of implementing the AI, not all of the desired features have been able to be developed, so the focus of the model has shifted slightly, to trying to develop some of the surrounding features to the simulation, such as the replay mode.

Known Issues and work-arounds

- If a person tries to walk off the edge of a simulation then an error will be generated and the simulation will stop.
- `reset_room` causes *X*X* cells to be generated – there is therefore a limit on the room size due to limitations in the number of possible shapes in Donald.