A Fire Evacuation Simulation Tool in Java

Project by Guillaume Ardaud *(09010297@brookes.ac.uk)* - ***Supervised by Dr. Hong Zhu*** *(hzhu@brookes.ac.uk)*

Abstract

The project's goal was to design and implement an application that would allow the simulation of a fire evacuation in a building, and generate relevant data. One of the purposes of this is to analyze the pertinence of the fire exits' placement throughout a given building, and to estimate how successful a fire evacuation would be. The project consisted of two parts: first of all, the implementation of the simulation environment itself; and then, the multi-agent algorithms that ruled how the agents tried to evacuate the building and other algorithms related to the modeling of the fire.

Topics studied include multi-agent simulation, cellular automaton, and graph theory.

Multi-Agent simulation

Multi-agent simulation, as opposed to single-agent simulation, refers to applications where intelligent agents interact with one another to solve a problem, instead of the problem being solved by one single entity. Multi-agent systems are often used for complex problems which could not be realistically defined and solved using a single agent system.

Multi-agent simulation has naturally evolved to simulate processes which study can then be used to learn about them. In our case, the process is pedestrian flow, or how people move and organize themselves in specific contexts. In our case, the agents will be the persons evacuating the building, following various behaviors modeled in order to achieve a realistic representation of how actual people would react.

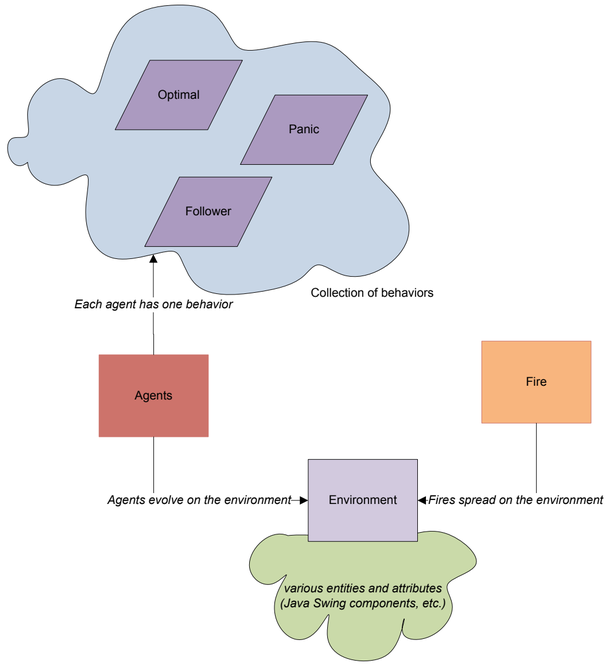
Objectives

The objectives of the project were as follows:

- Provide a thorough and reasonably accurate modeling of the pedestrian flow
- Credible simulation of the fire
- Obtain a usable and relevant software

Photo by Victoria Peckham

Structure of the system



Results

The resulting software, named Kitsune, is programmed in Java using the Swing framework. The user can easily design a building with its doors, fire exits, etc. using visual tools.

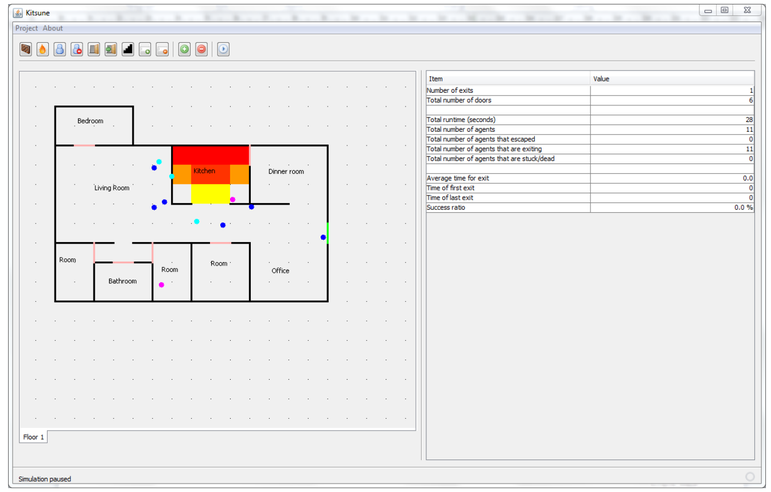
Agents, represented by dots on the visualization, use various AI algorithms to react to their environment.

Blue agents find the optimal path to the fire exit, using Floyd-Warshall's algorithm.

Cyan agents follow the agent nearest to them, to wherever he goes.

Fuchsia agents go to a random adjacent square that they have not visited.

The agent system has an easily extendable behavior mechanism, allowing anyone in the future to create additional behaviors, of any complexity.



The fire (represented in the visualization by the yellow, orange and red tiles) is modeled by a cellular automaton using— each cell can have a certain burn intensity, which changes depending on the burn intensities of the cells neighboring it, and for how long it has been at the current intensity.

Finally, the right part of the interface is dedicated to the presentation of the results gathered during the simulation. These results include, for example, the average time it took for agents to exit the building or the success ratio of the evacuation.

Related works

Similar software include buildingEXODUS and SimWalk. The related work typically answers to the following points:

- Use of physics based simulations (ie. particle modeling)
- Separation between the software that simulates the fire and the one that simulates the pedestrian flow
- Very restricted choice of agent behaviors

Future work

While a solid functional starting point, a lot can be done to improve Kitsune.

- **Improvements relative to the interface and design of the environment, and metrics collecting**
- **Improvements relative to the simulation of the fire and pedestrian flow**
- **Improvements relative to the simulation pedestrian flow**