UNIVERSITY OF OXFORD BROOKES

FINAL YEAR PROJECT

Simulation of the pedestrian flow in the context of fire evacuation in buildings

Author:
Guillaume ARDAUD
(09010297)

Supervisor:
Dr. Hong ZHU

Department of Computer Science - Module U08096

Contents

1 Introduction

This report is the interim report for my final year project. It will cover the progress made so far

1.1 About the name

I have decided to name the application developed for the purpose of the project 'Kitsune', a word which refers to fox spirits in the Japanese folklore. These magical and intelligent spirits are said to be able to generate fire from their mouthes and tails. [1]

Figure 1: Detail from a print by Kuniyoshi Utagawa, 19th century

2 Contents

Following is a table of contents for the final report, as if all the work had been completed. It aims to give an overview of what the final report will look like.

Introduction
This chapter gives the background of the research in this project.
1. Motivation *In this section we discuss how the research topic is relevant from both research and practical point of views.*
2. Related Work *In this section we review the existing work and research in the concerned field, as well as their advantages and flaws, the open problems and the state of the art in this topic.*
3. Overview of the project *In this section we give an overview of the project, its main results, and the structure of the report.*

I. The Proposed Approach
This chapter outlines our solution to the problem by presenting the general ideas, the functionnalities of our application, and what limitations of existing approaches were overcome. We also discuss the research methodology.
1.1. The basic ideas
1.2. The main functions of the system
1.3. The architectural design
1.4. Discussions *Advantages and disadvantages of the proposal approach and comparison with existing approaches, research methodology.*

II. Design and Implementation
This section gives the details of the design and implementation of the software system, where we describe its components.
2.1. Environment
2.2. Simulation of fire
2.3. Simulation of human behaviour in the environment
2.4. Discussions *This section will discuss the issues related to the design and implementation of the software, such as the choice of technologies, languages and implementation techniques, testing, etc.*

III. Evaluation and case study

This chapter will present the experiments with the software system by a number of case studies to demonstrate the usability of the system. Each case study will be presented in one section, which will describe the environment setting, the data obtained in the experiment, etc.

CONCLUSION

This chapter concludes the report with a summary of the main results of the project, a discussion of the advantages and disadvantages of the approach, the limitations of the current version of the software and future work that can be envisioned.

1. Summary of the main results of the project

2. Comparison with existing work

3. Future work

REFERENCES

APPENDICES

3 Schedule

According to the Gantt diagram included in the project proposal, by this date the environment should be fully implemented and a phase of tests and debug should start after handing in the interim report.

Things have gone differently, and the project is slightly late compared to what the initial schedule planned (see the 'Progress' chapter for complete details). This is due to the fact that the learning curve for the Swing framework was steeper than what I had initially though, and the development is taking more time than initially planned.

However, I have realized that due to my iterative approach to the development, implementation and testing are done in paralell. Thus, there is little to no need for a phase dedicated to testing and debug. As a consequence, the 'Implementation' and 'Tests and debug' tasks on my Gantt diagram are merged into one, leaving me 7 more days. With hard work, it should be enough to compensate for the lost time. The only thing related to testing that will have to be done are case studies. However, these will be done only when the project is completed- thus during the 'Testing and debug' part of the agents and environment interfacing phase.

4 Abstract

The following abstract is written as if the project had been completed.

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. The purpose of this is to analyze the pertinence of the fire exits' placement throughout a given building. 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.

After a brief overview of the project and motivations behind it, we review the proposed approach and the design process of the application, before covering the architecture of the environment simulator, written in Java, exposing the algorithms used for the spreading of the fire, the agents' behavior, and how they were interfaced with the environment. Discussed topics include modelisation of fire using cellular automaton, pedestrian flow simulation, multi-agent modeling and swarm intelligence.

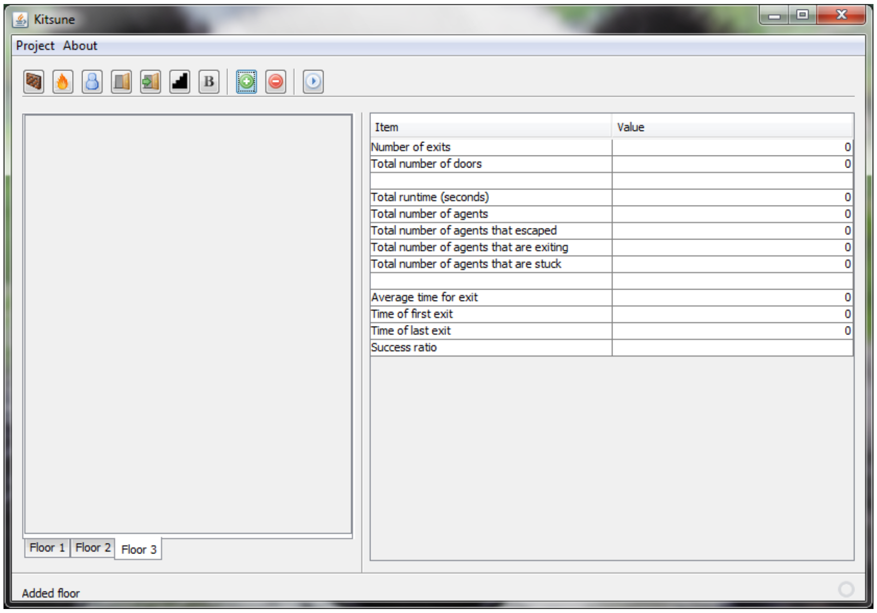
We conclude with case studies and an evaluation of the work achieved, known problems with the application and reflect on what would have been changed if the project had to be done again. We also compare it with existing solutions, consider its relevance in real life situations, and how it could be enhanced in that respect.

5 Progress

The development of the application has started. As the Gantt chart in the project proposal states, the first part of the development focuses on the environment.

So far, the environment is still in development. The main part of it is the building design tool, which allows to design the map of a multi-floor building, by placing walls, doors, fire exits and staircases on a grid. It is also possible to save the designed building to a file, and to import an existing building from a file into the environment. Finally, an inspector pane displays information regarding the simulation, and it is possible to export this information to a text file for further use and analysis.

The base of the UI itself is fully implemented and functionnal. It is possible to view details concerning the simulation and export said details in a text file, and the toolbar for editing is completed.



Item	Value
Number of exits	0
Total number of doors	0
Total runtime (seconds)	0
Total number of agents	0
Total number of agents that escaped	0
Total number of agents that are exiting	0
Total number of agents that are stuck	0
Average time for exit	0
Time of first exit	0
Time of last exit	0
Success ratio	

Added floor

Figure 2: Kitsune's UI

The UI is fully interfaced with the Environment class, allowing it to save it into a file or import it from a file.

The Environment class is the class that contains all the information concerning the building the position of the walls, exits, stairs, etc.

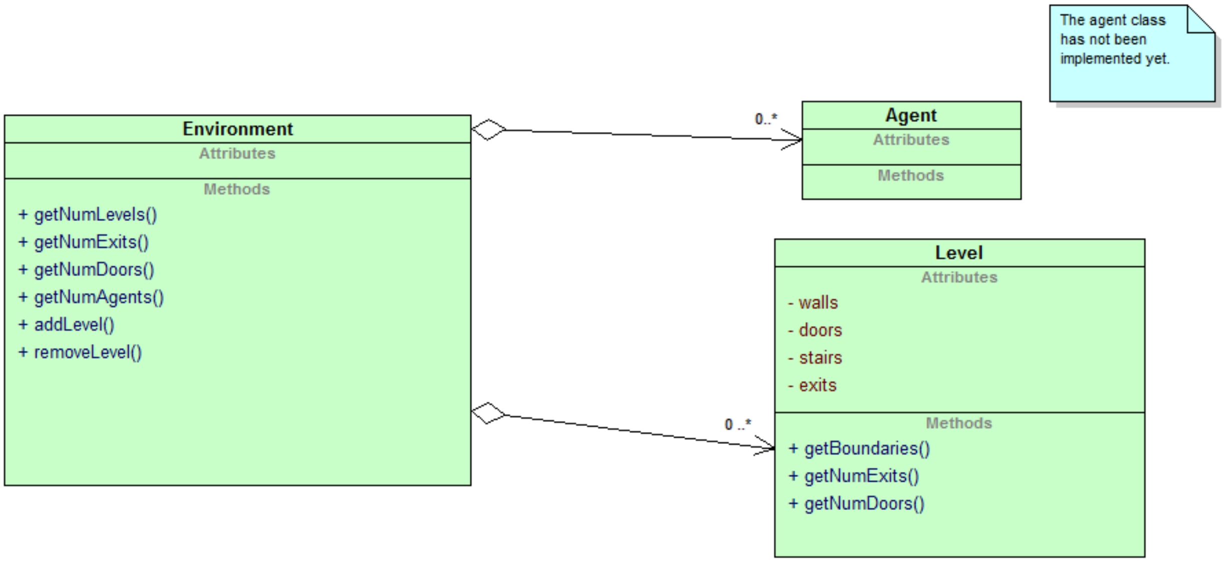


Figure 3: UML Diagram for the Environment class (this diagram is not final and might be completed as the development progresses)

Finally, the canvas of the UI, where it is possible to design the building, is still in development. Again, this is taking more time than expected due to the fact that I have very little experience with the Swing framework, which can be quite complex for tasks such as this one.

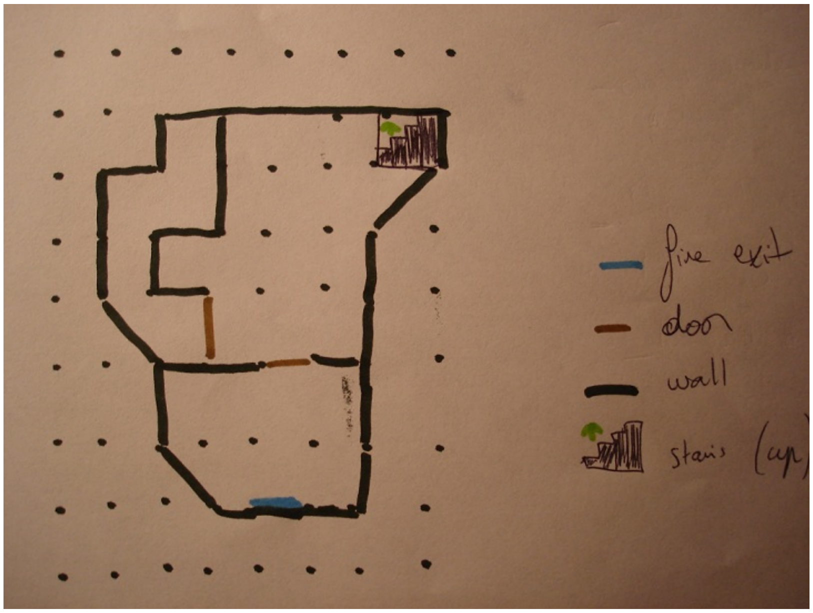


Figure 4: Mockup of the canvas that will be implemented for the design of the building

The interface will display a dotted grid on which the user can place, with the mouse, the different objects that constitute the layout of the building: walls, doors, fire exits, stairs, etc. I believe this approach, inspired by computer tools used by architects, is the most intuitive and flexible one, allowing an easy, instant modification of the whole architecture.

Once this will be complete, the last thing to do concerning the implementation of the environment will be programming the spreading of the fire in the building. I have started to do some background research on techniques and algorithms to simulate the way a fire expands in a building. After reflexion, it appears that the most efficient way, in terms of both computational resources and complexity, is to use an algorithm based on a cellular automaton [2] [3] [4]. The cells would be the same as the squares used in the canvas layout (see above).

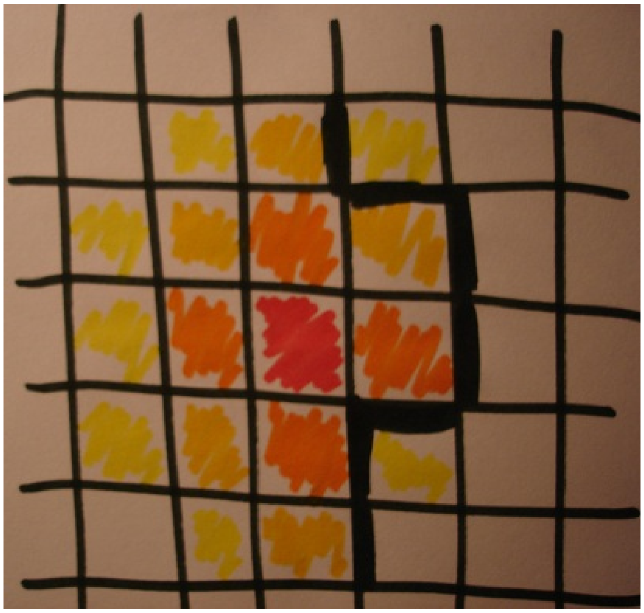


Figure 5: Mockup of how the cellular automaton would look in application

Each cell can take 8 different values: completely burned (black on the interface), unburned (white), and 6 levels of burning (different shades of orange). The algorithm will take into account the status of the adjacent cells, the presence of walls around the cell, and the time since the cell has been burning to dynamically determine the cell's current status. This way, I hope to achieve a simulation of fire that will be realistic enough for our purposes.

The next step will then be the design and implementation of the agents.

6 References

References

[1] Wikipedia the free encyclopedia. Kitsune. Website.
Full article on `http://en.wikipedia.org/wiki/Kitsune`.

[2] Forest fire cellular automaton applet. Website.
Full article on `http://en.wikipedia.org/wiki/CellularAutomaton`.

[3] Wikipedia the free encyclopedia. Cellular automaton. Website.
http://www.eddaardvark.co.uk/fivecell/forest.html.

[4] Marijo Łeri Ljiljana Bodrođi, Darko Stipaniev. Forest fires spread modeling using cellular automata approach.
http://www.fesb.hr/ ljiljana/radovi/Ljiljana_Bodrozic_ceepus2006_2.pdf.