

Intermediate Report

|  |
| --- |
| School of Computing  Faculty of Engineering |

Automatically Designed Rooms Using Machine Learning and Crowd Simulation

Edward A H Webb

Submitted in accordance with the requirements for the degree of  
BSc Computer Science

**2019/2020**

Type of Project: Exploratory Software - ESw

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)

© 2019/2020 The University of Leeds and Edward A H Webb

***Reminder about basic requirements of layout and format:***

***The report must be in typescript, sequentially page numbered, on A4, with 2.5cm margins. Point size 11 and one-and-a-half line spacing should be used.***

*Typically the intermediate report will consist of two chapters, ‘Introduction’ (including the aims, objectives, deliverables, and initial project plan) and ‘Background review,’ although this is not strictly enforced. There will also be a reference list. You should also address ethical issues somewhere in this intermediate report, even if just to state that there are none (a full discussion will be required in the final report).* ***No other material should be included*** *– deliverables should* ***only*** *be included in the Final report.*

*Note that it is not acceptable to solicit assistance on ‘proof reading’ which is defined as “the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the text”; see http://students.leeds.ac.uk/info/103552/taught\_student\_policies\_and\_procedures/945/proof-reading\_taught\_components*

***Page Numbering: The pages preceding the body of the text, i.e. from "Summary" to "Contents" inclusive, should be sequentially numbered in Roman numerals. All the remaining pages should be numbered in a single sequence of Arabic numerals.***

***Length: This report should be no more than 10 pages, excluding the references, the table of contents and the title page.***

# Table of Contents (example of how to format)

Table of Contents ii

Chapter 1 Level 1 Heading with ‘heading 1’ Style Applied by Pressing Ctrl Shift 1 1

1.1 Level 2 Heading with ‘heading 2’ Style Applied by Pressing Ctrl Shift 2 1

1.1.1 Level 3 Heading with ‘heading 3’ Style Applied by Pressing Ctrl Shift 3 1

1.1.1.1 Level 4 Heading with ‘heading 4’ Style Applied by Pressing Ctrl Shift 4 1

Chapter 2 Tables and Figures 2

2.1 Tables using the ‘table caption’ and ‘table description’ Styles 2

2.2 Figures using the ‘figure caption’ and ‘figure description’ Styles 2

List of References 3

# Chapter 1 - Introduction

Safety is a primary concern when architects and others design a building. Serious thought needs to be put into the availability of features like fire escapes and other exits which provide ways out for people if disaster strikes. Crowd simulation software provides insights as to how a group of people may move in certain environments and situations – such as in the event of emergency. Due to the importance of safety in design, crowd simulation is among the most prominent techniques for doing so.(Jin et al.). Machine Learning is a powerful area of computing which can be used to generate results given a scoring criteria.

## Aims

This project aims to follow and deliver the development of a piece of software which will attempt to combine Machine Learning and crowd simulation to automatically design floorplans which are as safe as possible in the event of emergency. The software will randomly place a dangerous event in a floorplan populated with people, at which point their escape will be modelled using crowd simulation. The floorplan will then be assessed based on how many and how quickly people escaped, whilst avoiding the danger. A 2D model will be used, as this will maintain simplicity and because a 2D simulation is sufficient and more appropriate for this project’s evacuation-style simulation.(Jin et al.). Machine learning will then be used to automatically repeat this experiment many times, using previous results to deliver safer floorplans to the user.

## Objectives

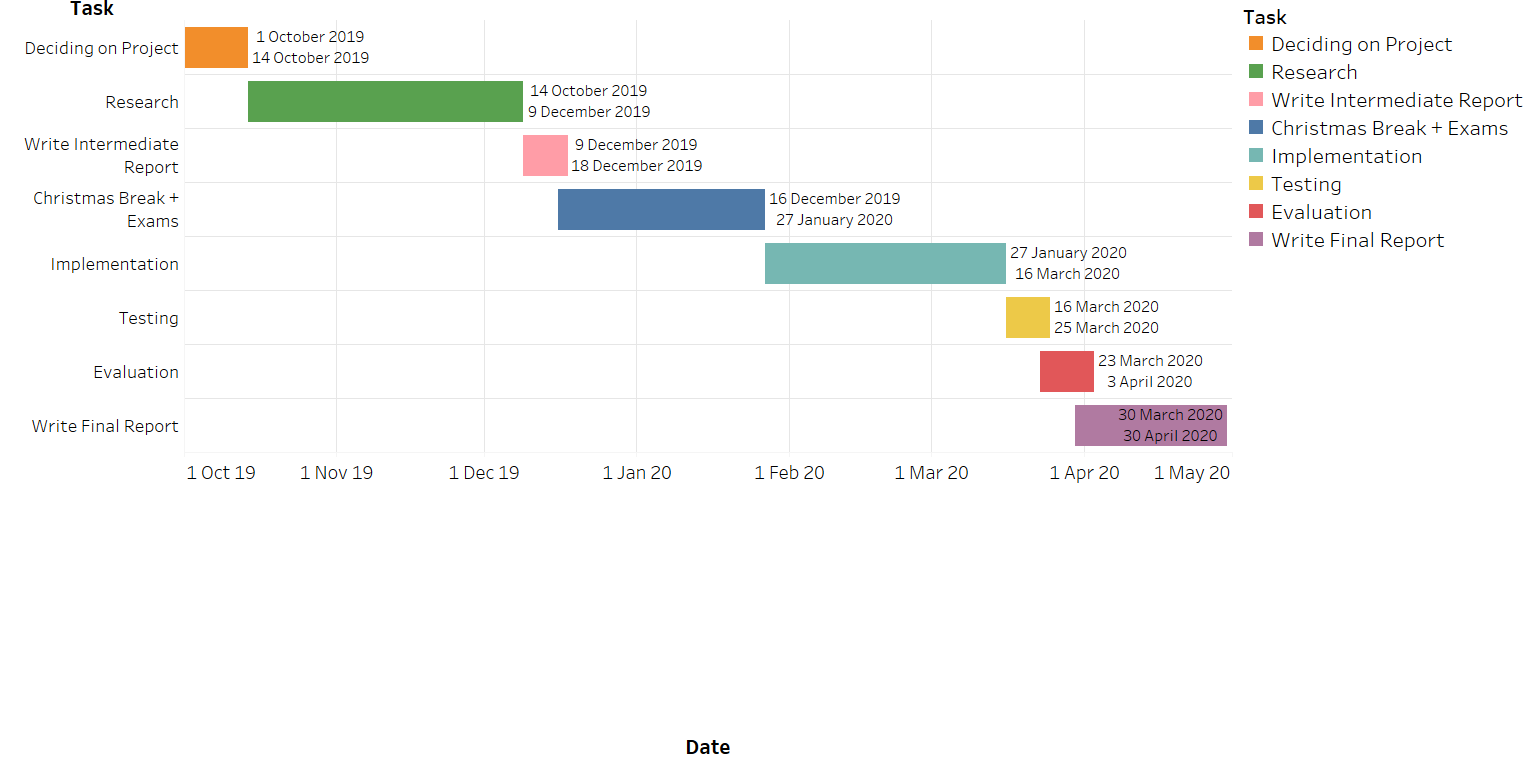
* Implement the crowd simulator using the Menge framework.
* Devise a way of placing danger in the simulation domain and scoring agents’ ability to get away from it.
* Implement the machine learning software to change the layout of a floorplan depending on its safety rating.

## Deliverables

* Demonstration of the software
* Report on the project

## Initial Plan

The project’s panned time allocation is shown on the following page as a Gantt chart. The different areas have deliberately been left relatively vague and open ended, as it is difficult to know what will take large chunks of time in the Implementation stage for example.



## Methodology

## Risk Mitigation

A key aspect of risk mitigation is guarding against the loss of work due to lack of version control. To this end, Git will be used to keep an online repository of work safe. This means that in the event of computer failure or theft, the project will be backed-up online. Furthermore the use of Git will allow the use of commit and merge points to return to in the event of faults in code; and makes developing new features easier, as branches allow for testing and prototyping before being included in the codebase.

Another important risk for this project is the general aspect of venturing into the unknown in many ways, as previous lack of experience in crowd simulation and neural networks throw up steep learning curves. Therefore there is the danger of the project’s scope becoming too large, resulting in planned features having to be cut short or revised. Because of this, constant evaluation will be needed as well as advice from the project coordinator to keep the project in check and make sure it doesn’t get too complicated and result in failure and disappointment.

There are two main directions in crowd simulations, realism of behavioural aspects and highquality visualisation [Thalmann and Musse 2007]. In the former, simulations are achieved in a simple 2D environment so that more quantitative validation can be compared to real world observations, for example, a simulated emergency evacuation and a hazard drill. In the latter idea, one considers visual results that can be presented in a convincing and believable manner, for example, animated movies and 3D games.

# Chapter 2 - Background Research

## 2.1 Crowd Simulation

When simulating a crowd there are two major ways in which to do so. One approach, under the name of macroscopic models, offers higher realism of general large scale crowd behaviour and models the crowd as more of a gas or liquid-like object (Henderson,1971). The others, microscopic models, are concerned with higher detail in individual simulation of movement, but with the downside of not behaving properly in highly dense crowd situations (Jin, et al., 2016).

As macroscopic models are more suited to evacuation and emergency simulations, this project will attempt to implement a simulation of this type.

## 2.1.2 Crowd Simulation Paradigm

A common paradigm in crowd simulation is to break the problem down into four subproblems: goal selection, path computation, path adaptation, and spatial queries (Funge et al., 1999). Each of these subproblems need to be addressed and together they make up a crowd simulator’s basic structure and operation. Many simulators follow this abstraction, and there are others which differ slightly – combining the path computation and path adaptation steps for example. Menge, the crowd simulation framework selected for this project, provides ease of use when changing and using the four different subproblems.

The first subproblem of goal selection deals with what each agent wants to do, as they each need a goal to strive for when thinking about modelling how they’d move to fulfil that goal. In the case of this project, each agent will want to escape the danger presented to them by reaching an exit without harm.

Path computation is how the simulator works out the path that the agent should take in order to try and fulfil their goal. A complete path for the agent isn’t always needed, and path computation can simply serve to compute the agent’s preferred direction of travel at that moment of time. Path computation can be achieved by modelling the space as a graph and then searching it using algorithms such as A\*. Potential fields can also be used, wherein the space is subdivided into a grid and negative goals push the agent away from them, and positive goals pull the agent towards them using gradients of cost functions. This method doesn’t compute a complete path for the agent, instead providing a preferred direction of travel, which is then used as input for the next computation.

Path adaptation describes how agents deal with changes to their paths on the fly. For example, an agent walking straight towards their goal would be able to steer around another agent coming the opposite way using path adaptation, instead of crashing into the and stopping completely.

Spatial queries refer to an agent’s ability to recognise such things as if stimuli are in their field of view, or if stimuli are in their proximity. Effective use of spatial queries would prevent an agent from starting to run from a fire it couldn’t logically see or otherwise detect.

## 2.2 Generative Adversarial Networks

A Generative Adversarial Network (GAN) is a type of machine learning system in which two neural networks compete in a game. One of the networks, the generative network, generates new objects (such as images), which are then judged by the other network, the discriminative network. The discriminative network’s goal in the game is to identify which objects are from the real data set and which have been made by the generative network, while the generative network’s goal is to produce objects which fool the discriminative network, making it think the generated objects are in fact from the data set (Goodfellow et al., 2014)

The discriminative network is trained on an existing data set, and the generative network is trained according to how well it manages to fool the discriminative one. Through the use of Backpropogation, the generative network is able to improve at generating new objects whilst the discriminator improves at catching the fake ones. In this way the generator is eventually able to manufacture objects which are extremely like those in the original data set. The power of GANs has been shown by the use of StyleGAN(CITE HERE), which can produce incredibly realistic portraits of human faces which are indistinguishable from real ones.

It is this project’s hopes to use a GAN to produce floorplans which are as safe as possible when disaster strikes in a crowd simulation.

## <http://gamma.cs.unc.edu/Menge/files/MengeTechReport.pdf>

## <https://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1095&context=cs_etds>

## <https://pdfs.semanticscholar.org/5b84/a2bfaeacb3a44ad8a56a384490ec50940be8.pdf>

## <https://link.springer.com/content/pdf/10.1007%2Fs11390-014-1469-y.pdf>

# Chapter 3 - Ethics

As this project will involve software to evaluate the safety of different building layouts, ethical issues are non-trivial. It’s in this project’s hopes that something like its end result could be used by an architect to more easily design a building’s layout which is as safe as possible for use. However, if a building was constructed based on a design supplied by the program, and something like a fire struck it, serious ethical concerns could be raised. Perhaps most obviously is the question of who may be responsible should the design be found weak to fires. Some parties may claim that it was the builders, the architect, or potentially this project’s author – as they may claim it could have provided a false sense of security to the architect using it.

As such, if this project were ever to be used by designers, legal measures may have to be put in place to make sure that no writers of the code would be liable if something like the previously described event occurred. The author of this project makes no claim to be accredited by the Royal Institute of British Architects (RIBA) and so shouldn’t be held accountable – the onus is on the architect to ensure that a building’s design is sufficiently safe.

# Chapter 2 Tables and Figures

## 2.1 Tables using the ‘table caption’ and ‘table description’ Styles

Text before table. Text before table. Text before table. Text before table. Text before table. Text before table. Text before table. Text before table. Text before table. Text before table.

**Table 2.1** Caption of Table — automatically appears in the List of Tables when that is updated The ‘table caption’ style has been applied to this paragraph by pressing Ctrl Shift T.

This is the table description in the ‘table description’ style. It is optional text to give more information about the table and does not appear in the List of Tables.

|  |  |  |
| --- | --- | --- |
| **Heading One** | **Heading Two** | **Heading Three** |
| 1.1 | 1.2 | 1.3 |
| 1.21 | 1.22 | 12.3 |
| 12.31 | 12.32 | 12.33 |

## 2.2 Figures using the ‘figure caption’ and ‘figure description’ Styles

Figures can be added using the Illustrations section of the Insert tab.



**Figure 2.1** Caption of Figure — automatically appears in the List of Figures when that is updated. The ‘figure caption’ style has been applied to this paragraph by pressing Ctrl Shift F.

This is the figure description in the ‘figure description’ style. It is optional text to give more information about the figure and does not appear in the List of Figures.

# List of References

*<It is expected that the list would reflect the breadth and depth of scholarly research undertaken by the student during the course of the project.>*

Henderson L. The statistics of crowd fluids. Nature, 1971, 229(5284): 381-383.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In Proc. of SIGGRAPH, 29–38.

Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014) Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.