



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

# Burning Man: Simulating the behaviour of people in the event of fire

Project Report

Nico Hauser, Andri Horat, Elias Schmid, Jonas Spieler

Zurich  
December 2019

## **Agreement for free-download**

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Nico Hauser

Andri Horat

Elias Schmid

Jonas Spieler

# Contents

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Introduction and Motivations</b>	<b>4</b>
<b>3</b>	<b>Description of the Model</b>	<b>5</b>
3.1	Building model . . . . .	5
3.2	Agent model . . . . .	7
<b>4</b>	<b>Implementation</b>	<b>7</b>
4.1	Javascript and the web . . . . .	7
4.2	Physics (Phaser 3) . . . . .	7
4.3	Map generation . . . . .	7
4.4	Agent behaviour . . . . .	7
4.4.1	Raytracing and pathfinding . . . . .	7
<b>5</b>	<b>Simulation Results and Discussion</b>	<b>7</b>
5.1	Choosing start parameters . . . . .	7
5.2	Interpretation of the simulation result . . . . .	7
5.3	Adapting the parameters for better results . . . . .	7
5.4	Finding the model's limits . . . . .	7
5.4.1	Evaluation of the building model . . . . .	7
5.4.2	Evaluation of the agent model . . . . .	7
5.5	Evaluation of the simulation . . . . .	7
<b>6</b>	<b>Summary and Outlook</b>	<b>7</b>
<b>7</b>	<b>References</b>	<b>7</b>

## 1 Abstract

This paper describes a model for the behaviour of people in the event of fire and it's simulation. The model tries to incorporate the general repulsion of people standing too close, the formation of groups in panic situations, the placement of exit signs and many smaller factor such as different weights and ages. In order to make observations on the defined model, it is simulated using a physics engine using javascript that makes it accessible from any place at any time and allows for reproducible and verifiable results. The paper also defines a general format for describing a ground plan of a building that can directly be imported into the model simulation. Combined with the right simulation parameters, this features allows the evaluation of the safety in buildings in the case of fire.

## 2 Introduction and Motivations

This paper was created during the course *Agent-Based Modeling and Social System Simulation* held by Dr. Nino Antulov-Fantulin and Thomas Asikis at ETH Zurich. The goal was to create a project matching the title of the course meaning the modelling of a complex system where humans are the agents. A complex system is in contrast to complicated system not per se hard to implement but rather consists of many simple small parts that on themselves act based on simple rules. The fact that there are a lot of these so-called agents gives rise to behaviours of the whole system that are not always easy to predict or even understand even if the simple rules of a single agent are well-known. As computers keep getting faster and new technology is developed every day, simulations are a good way of understanding these complex systems by playing with the available parameters while looking for emerging patterns and applying them to the real world.

In particular this paper describes and simulates the event of fire in a building that tries to incorporate many parameters an evacuation in real life depends on. When developing the model and simulation it was a primary goal to have every constant that was either empirically chosen or arbitrarily set to be an adjustable parameter. This allows the simulation of a much larger set of situations that may be needed in real life. Even the building is dynamic and can be changed by creating an appropriate file describing it. In the future it may even be possible to automate this process and directly parse a pdf ground plan which would allow even more people to take advantage of this model by testing their building for possible issues in the event of fire.

We also would like to add a side note about the project title *burning man*. When thinking about a project idea, it was clear quickly that an emergency situation would

be rather interesting to all of us but we didn't immediately know where the evacuation should take place, in a small house, a large building or an event such as a festival. As burning man is the name of a famous festival in Nevada that takes place in the middle of a desert. After some discussions we settled for the evacuation of a building in the case of fire and when thinking about a new name, we realized the name is still kind of fitting and even funny if you have a little bit of dark humour. As software projects often have similar names we thought we'd just leave it that way though it should be made clear that the aim of this project is of course to reduce the number of 'burning men'.

### 3 Description of the Model

The model is separated into different pieces. First of all we had to create a model for the building in which the evacuation takes place. Next a model for the individual agents, here representing humans, had to be created and evaluated.

#### 3.1 Building model

Before agents can react to a fire inside a building, a building has to be defined and created. As the simulation in the end will take place on a computer screen, we first of all have to give the building or the *map* a **width** and a **height**.

The next obvious property any building has are walls. Walls define rooms, corridors and give the map its shape. We define the walls to be lines drawn from some coordinate  $(x_1, y_1)$  to some other coordinate  $(x_2, y_2)$ . In javascript these tuples can be defined as objects `{x: <xVal>, y: <yVal>}` and then be put into an array with two elements. A wall from  $(0,0)$  to  $(100,0)$  is then given by `[{x: 0, y: 0}, {x: 100, y: 0}]`. Multiple walls are then defined like

```

1 {
2   width   : 1900,
3   height  : 1100,
4   wallThickness: 10,
5   fires: [{position: { x: 20, y: 450 }}, {position: { x: 20, y: 500 }}],
6   doors: [
7     {
8       position   : { x: 85, y: 425 },
9       orientation : { x: 0, y: -1 },
10      direction  : { x: 0, y: 1 }
11    },
12    {
13      position   : { x: 445, y: 425 },
14      orientation : { x: 0, y: -1 },
15      direction  : { x: 0, y: 1 }

```

```

16     },
17     ...
18 ]
19 walls : [
20     [{ x: 0, y: 0 }, { x: 1900, y: 0 }],
21     [{ x: 0, y: 1100 }, { x: 1900, y: 1100 }],
22     [{ x: 0, y: 0 }, { x: 0, y: 1100 }],
23     [{ x: 1900, y: 0 }, { x: 1900, y: 1100 }],
24     ...
25 ],
26 signs: [
27     {
28         position: { x: 550, y: 538 },
29         orientation: { x: -1, y: 0 },
30         direction: { x: 1, y: 0 }
31     },
32     {
33         position: { x: 1125, y: 538 },
34         orientation: { x: -1, y: 0 },
35         direction: { x: 1, y: 0 }
36     },
37     ...
38 ],
39 tables: [
40     [{ x: 70, y: 70 }, { x: 110, y: 170 }],
41     [{ x: 70, y: 225 }, { x: 110, y: 325 }],
42     ...
43 ]
44 }

```

For a very basic version of the simulation, this is already enough information. The agents could randomly start moving around and interact with each other but such a simulation wouldn't even remotely be interesting for our purpose, we need at least a way for the agents to escape i.e. define an escape zone or a **despawnZone**, as the agents are removed from the simulation when they reach it. As most buildings have multiple emergency exits, we do not limit the building model to only one escape zone but allow for an arbitrary amount. The escape zones / **despawnZone** are modelled as rectangles meaning two points (the top left and bottom right corner) are essentially enough to exactly describe it. Thus the format looks like

```

1 {
2
3 }

```

### 3.2 Agent model

## 4 Implementation

### 4.1 Javascript and the web

### 4.2 Physics (Phaser 3)

### 4.3 Map generation

### 4.4 Agent behaviour

#### 4.4.1 Raytracing and pathfinding

## 5 Simulation Results and Discussion

### 5.1 Choosing start parameters

### 5.2 Interpretation of the simulation result

### 5.3 Adapting the parameters for better results

### 5.4 Finding the model's limits

#### 5.4.1 Evaluation of the building model

#### 5.4.2 Evaluation of the agent model

### 5.5 Evaluation of the simulation

## 6 Summary and Outlook

## 7 References