

# 최종 결과 보고서

HARMS / SARL Project

---

세종대학교 조우석

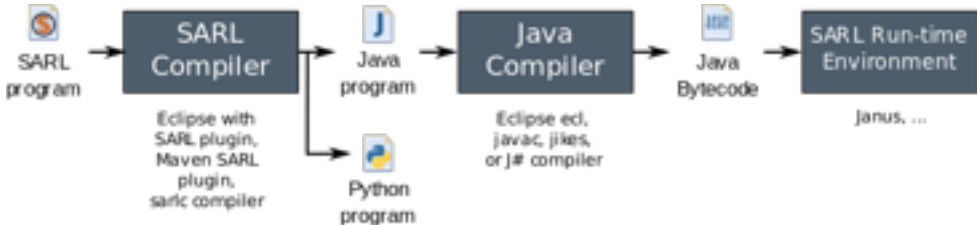
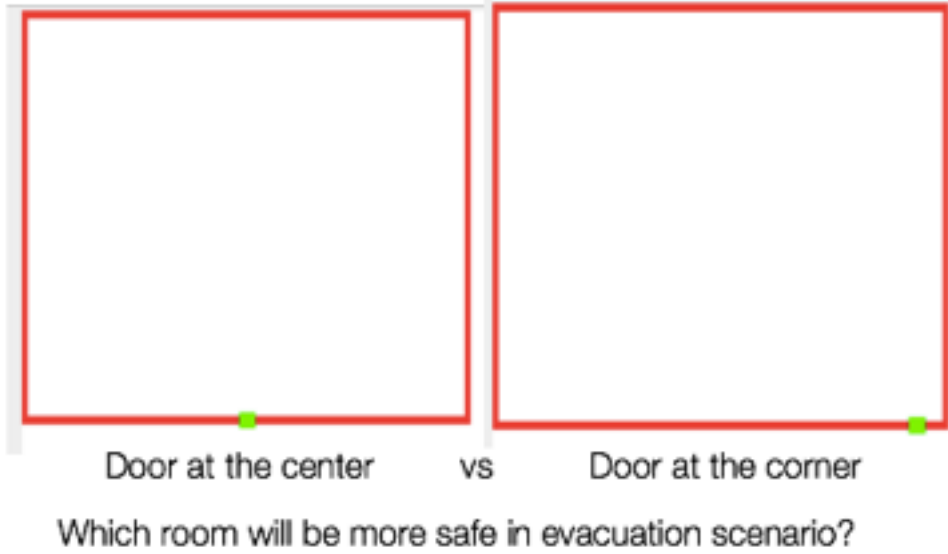
세종대학교 이충래

세종대학교 임지훈

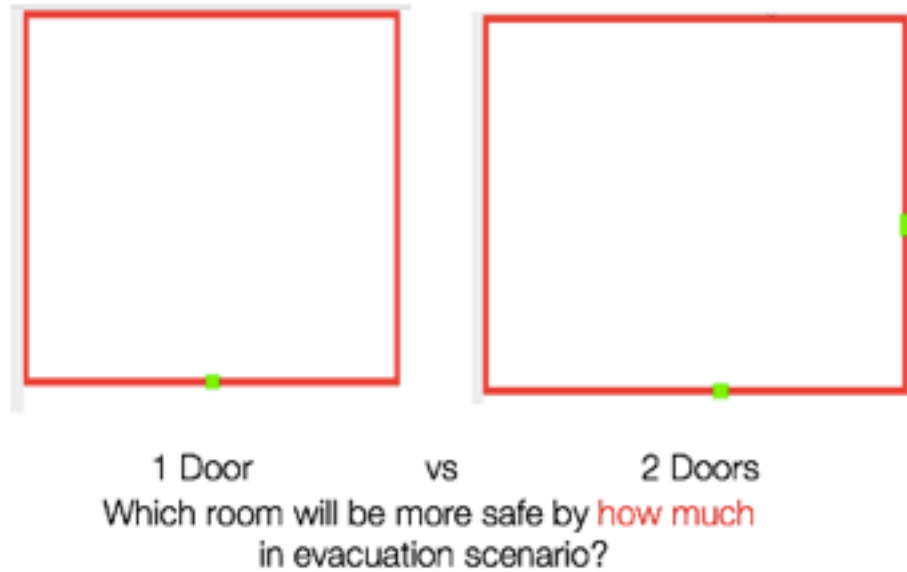
**PURDUE**  

---

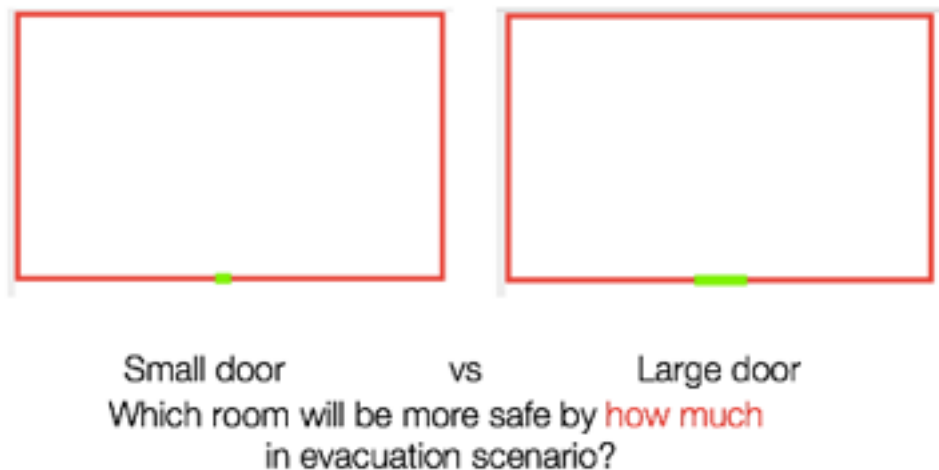
**U N I V E R S I T Y**®

주제	Evacuation Simulation using Sarl language			
구성원	학교	학과	학번	이름
	세종대학교	컴퓨터공학과	122116	이총래
	세종대학교	컴퓨터공학과	13010979	임지훈
	세종대학교	컴퓨터공학과	15011023	조우석
멘토	Purdue University	-	-	Ahmad Esmaeili
들어가기 앞서	<p>퍼듀대학교에서 진행 중인 여러 프로젝트에서 우리 팀의 프로젝트 주제는 Harms &amp; Sarl이다. 따라서 Harms와 Sarl의 개념을 우선 소개하고자 한다.</p> <p>HARMS란 Human, Agent, Robot, Machine, Sensor 의 앞 자를 딴 단어로 이 5개의 요소들이 서로 유기적으로 연결되어 각각의 요소들이 유연하게 작업을 하여 궁극적인 목표를 이루는 개념으로 볼 수 있다.</p>  <pre> graph LR     SARL[SARL program] --&gt; SARL[SARL Compiler]     SARL --&gt; Python[Python program]     SARL --&gt; Java[Java program]     SARL --&gt; Java[Java Compiler]     SARL --&gt; Java[Java Bytecode]     SARL --&gt; SARL[SARL Run-time Environment]     SARL --&gt; Janus[Janus, ...] </pre> <p>SARL은 Java를 기반으로 한 프로그래밍 언어로써, Agent-oriented 프로그래밍 언어로 볼 수가 있다. 따라서 개발자 입장에서는 multi-agent 기반의 프로그램을 구현하는데 있어서 좀 더 직관적이고 간결하게 코드를 개발할 수 있다는 장점이 존재한다. 따라서 이번 프로젝트는 Multi-agent based model simulation을 토대로 Sarl 언어를 이용하여 구현을 하였다.</p>			
프로젝트 필요성 및 개요	<p>본 탈출 시뮬레이션의 목표는 크게 3가지로 볼 수 있다. 첫 번째로, 건물에 있는 문의 수, 크기, 위치에 따른 방의 안전도를 확인해 보는 것이다. 따라서, 우리는 3가지 실험을 실시하였다.</p>  <p>Door at the center vs Door at the corner</p> <p>Which room will be more safe in evacuation scenario?</p>			

### 실험 1. 문의 위치에 따른 방의 안전도 실험



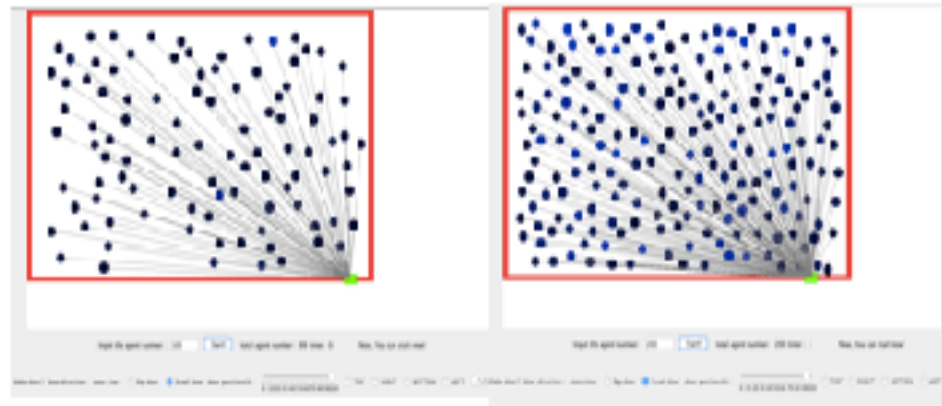
### 실험 2. 문의 수에 따른 방의 안전도 실험



### 실험 3. 문의 크기에 따른 방의 안전도 실험

두 번째 목표는 인원수에 따른 pressure/push 양을 구하는 것이다. 이의 필요성은 Hillsborough disaster를 들 수 있다. 이 대참사는 각각 상대방에 존재하는 축구팬들의 충돌을 막기 위해 구역을 분리하였는데, 각 팀의 축구팬들의 수용인원수를 잘못 배분하여 대규모의 인원이 좁은 공간에 들어가게 되어 96명이 압사를 당한 사고이다.

## 2. What is the maximum capacity allowed in the building?



How would the difference be between  
100 people in a room vs 200 people in a room?

### 실험 4. 인구 수에 따른 방의 안전도 실험

따라서, Hillsborough disaster와 같은 재난이 발생하는 것을 방지하고자 pressure/push 개념을 도입하여 몇 명의 부상자가 발생할 것인지 예측할 수 있는 장치를 구현하고자 하고, 시각화 시키도록 하였다.

마지막으로 부상자의 수가 몇 명이 될지를 알아내는 작업을 하는 것이다. 탈출 시뮬레이션에서 건물의 문의 위치, 크기, 수, 인원수에 따른 방의 안전도를 확인하기 위해서 부상자를 알아내는 것이 필요하다.

## 3. How many people will get injured / killed?

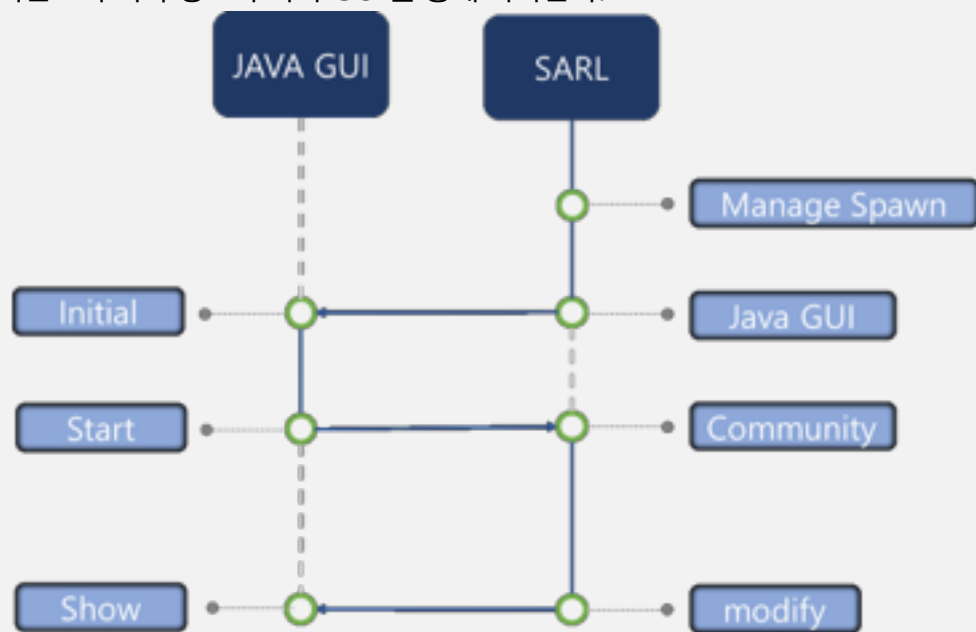


Estimating how many people will get hurt

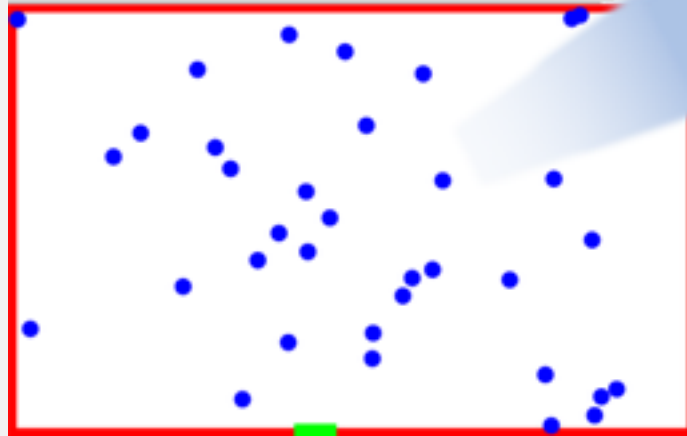
	<p><b>그림. 부상자의 수가 몇 명이 될지를 알아내는데 사용된 그래프</b></p> <p>이는 기존의 시뮬레이션과 비교하여 문의 위치, 크기, 수를 정할 수 있게 하고, 인원수도 손쉽게 지정해 줄 수 있도록 하였으며, pressure/push 기반으로 부상자 수를 산출해낸다는 점에 있어서 차별화를 두었다. 또한 Sarl 언어를 이용하여 구현을 함으로써, Sarl 언어로 구현한 오픈소스 프로젝트 수 증진에 기여를 하고자 한다.</p>
<p><b>프로젝트 수행 결과</b></p>	<p>1. Sarl 언어에서 Java와 연결 및 Manager Agent의 구현과 GUI</p> <p>자바와 SARL을 연결할 수 있는 프레임워크가 존재하여 자바로는 GUI를 구현하여 에이전트의 움직임을 시각화하였다. 그리고 SARL에서는 각각의 에이전트를 정의하여 자바 GUI에서 위치정보를 가져올 수 있도록 연동하였다.</p> <p>각각의 에이전트(Human)는 Skill, Capacity, Behavior 들로 정의된다. 매니저(센서)는 이 에이전트들이 서로 공유하는 네트워크 공간(Default Space)에 화재정보와 최적의 경로 정보들을 제공한다.</p> <div data-bbox="427 772 1273 1348" data-label="Diagram"> <pre> graph TD     Manager[Manager] --&gt; DefaultSpace[Default Space]     DefaultSpace --- Agent1[Agent]     DefaultSpace --- Agent2[Agent]     DefaultSpace --- Agent3[Agent]     DefaultSpace --- Agent4[Agent]     Agent1 --&gt; Behavior1[Behavior]     Agent2 --&gt; Behavior2[Behavior]     Agent3 --&gt; Behavior3[Behavior]     Agent4 --&gt; Behavior4[Behavior]     Behavior1 --&gt; Capacity1[Capacity]     Behavior2 --&gt; Capacity2[Capacity]     Behavior3 --&gt; Capacity3[Capacity]     Behavior4 --&gt; Capacity4[Capacity]     Capacity1 --&gt; Skill1[Skill]     Capacity2 --&gt; Skill2[Skill]     Capacity3 --&gt; Skill3[Skill]     Capacity4 --&gt; Skill4[Skill] </pre> </div> <p>화재정보 이벤트는 모든 에이전트들이 Default Space에서 그 이벤트를 받아들일 수 있으며, 각각의 Agent에게 최적화된 탈출구까지의 최적경로 정보는 Agent의 ID 값을 이용해서 개별로 이동할 수 있도록 한다. 자바와 SARL은 연동이 가능하며 자바에서는 GUI로 시뮬레이션을 보여주었다.</p>



위의 화면이 프로그램의 초기화면이다. 문을 만들어주고 에이전트의 수를 설정한 뒤 Start 버튼을 누르면 SARL의 에이전트가 자바와 서로 연동이 되어서 에이전트의 시각 정보가 자바 GUI를 통해 나타난다.

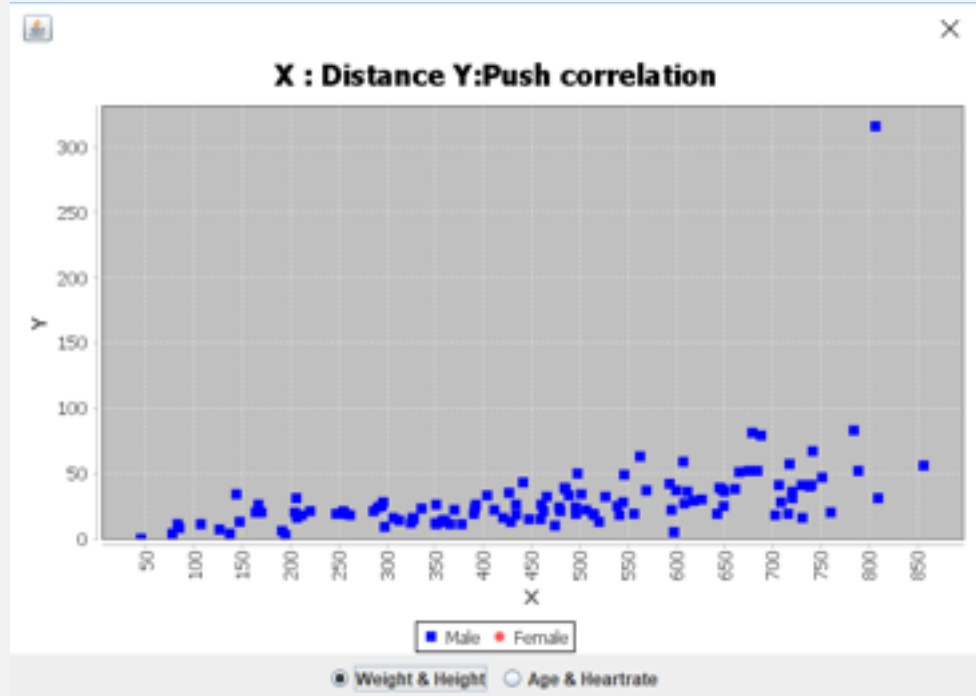


처음 실행하는 것은 SARL의 Manager이다. Manager 에이전트가 생성 및 실행되면 JAVA의 base\_frame 클래스를 실행시켜 GUI를 화면에 띄운다. 그 후 User가 Agent의 수와 탈출구를 설정해주면, 자바에서는 에이전트의 위치를 랜덤으로 설정하고 시각화 한 뒤 에이전트의 X, Y좌표 값과 USER가 생성한 탈출구의 위치 정보를 SARL의 Manager로 보낸다. SARL에서는 여러 Agent와 탈출구까지의 최적 경로를 계산하여 이에 대한 정보를 해당 Agent에게 전달하면, 해당 Agent는 좌표 값을 그에 맞게 변경하여 다시 JAVA의 GUI로 전달하는 방식을 사용한다.



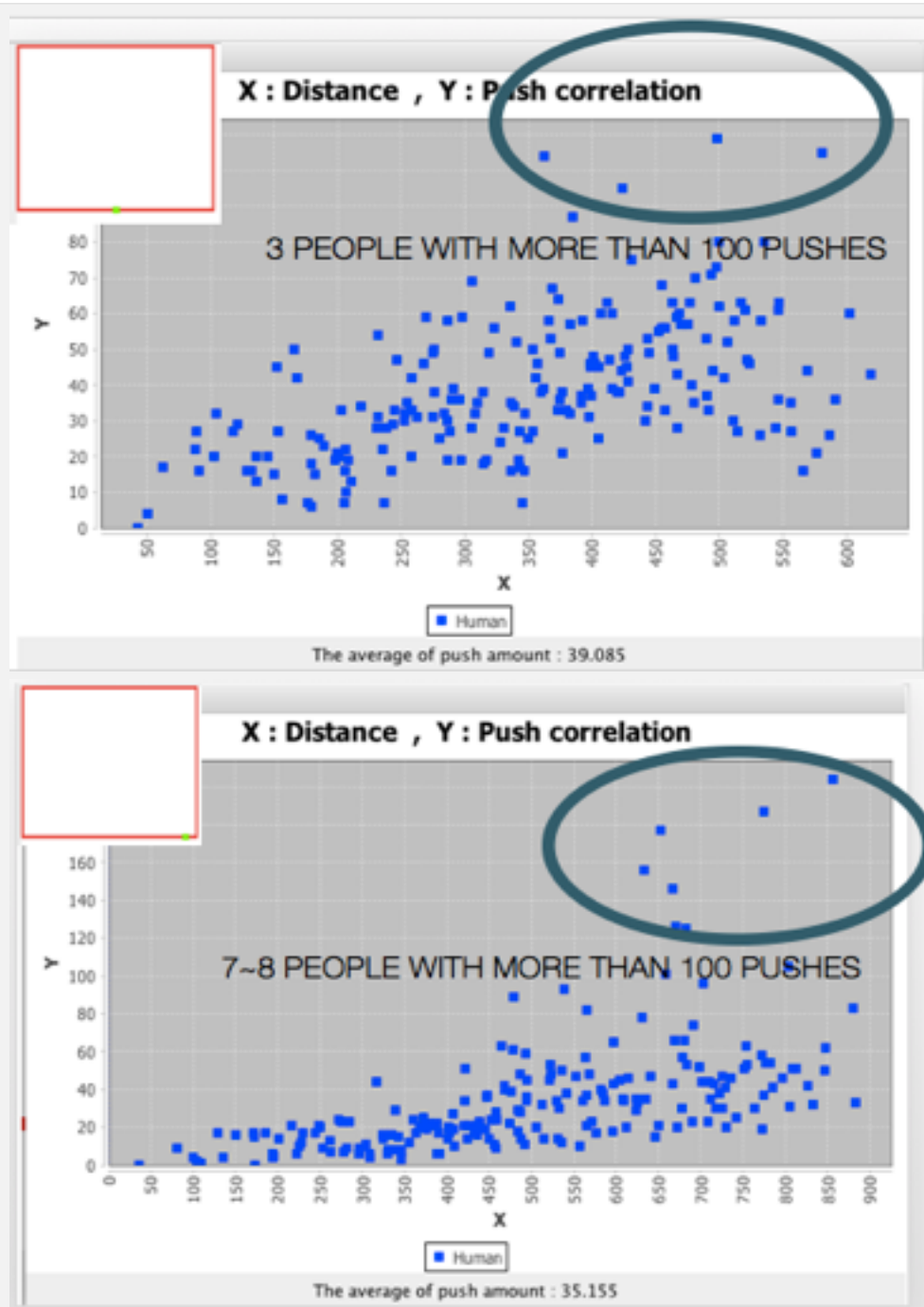
파란색이 Human을 상징하는 에이전트이다. 사람마다 스스로 행동한 뒤 탈출구로 향하여 탈출하는 구조이다. 바닥에서 사람마다 문으로 향하게 실선을 그림으로써 사람들이 어둠 속에서도 자신의 위치에서 가장 가까운 문으로 가게끔 유도하도록 하였다.

그래프는 아래와 같이 정의를 하였다.



위의 그림을 봤을 때 문이 하나일 때는 사람이 문보다 거리가 멀어질 때 다른 사람과 충돌하는 횟수가 증가한다. 여기서 문을 두개로 설정해 준다면 우리가 생각하는 바닥으로 길을 인도하여 고르게 사람들을 문으로 향하게 할 수 있어 그만큼 사람들 간의 충돌 횟수가 적어 들어보다 빠르게 탈출이 가능하게 된다.

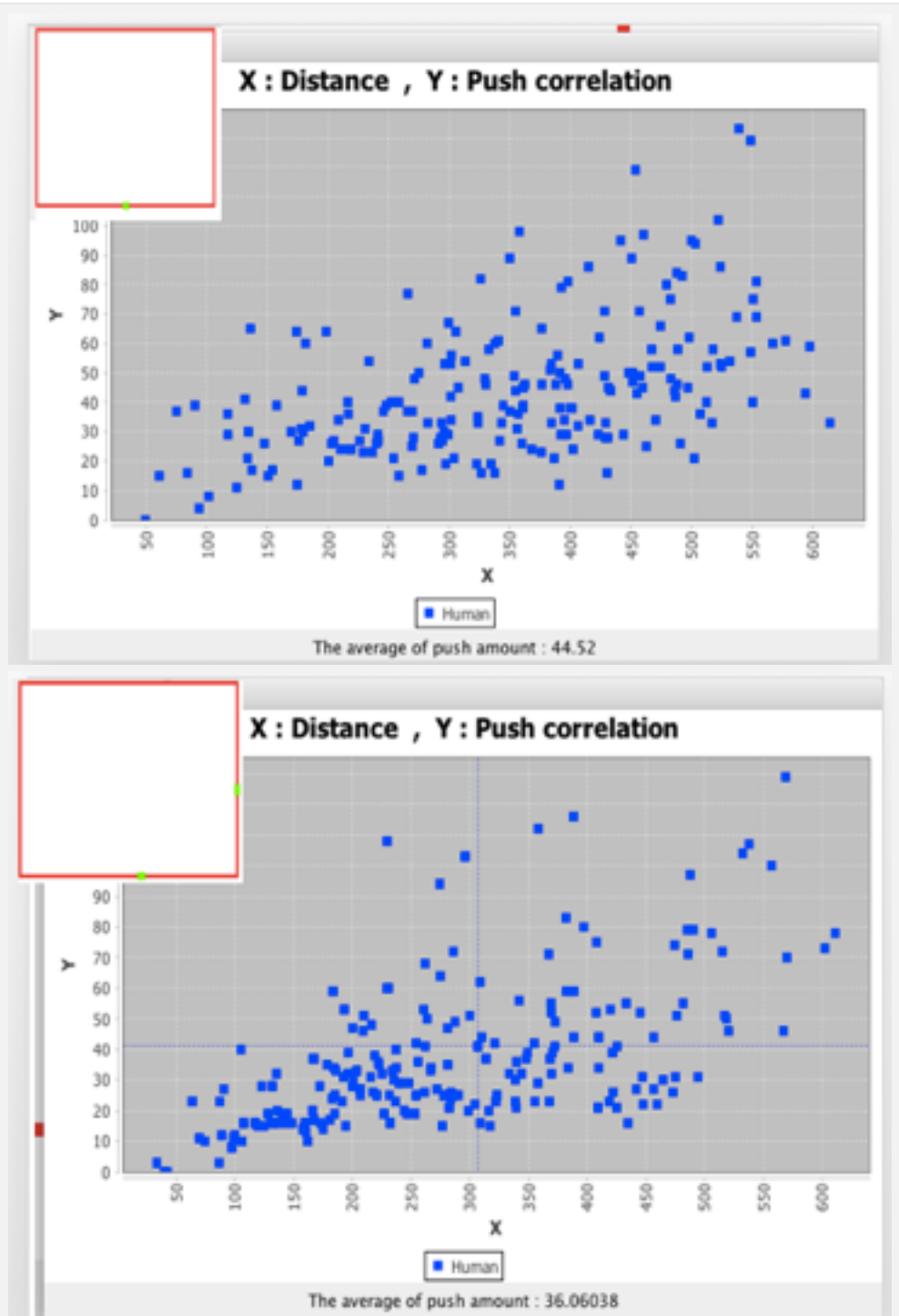
프로그램 개요에서 설계한 실험의 결과는 다음과 같았다.



### 실험 1. 문 위치에 따른 방의 안전도 실험

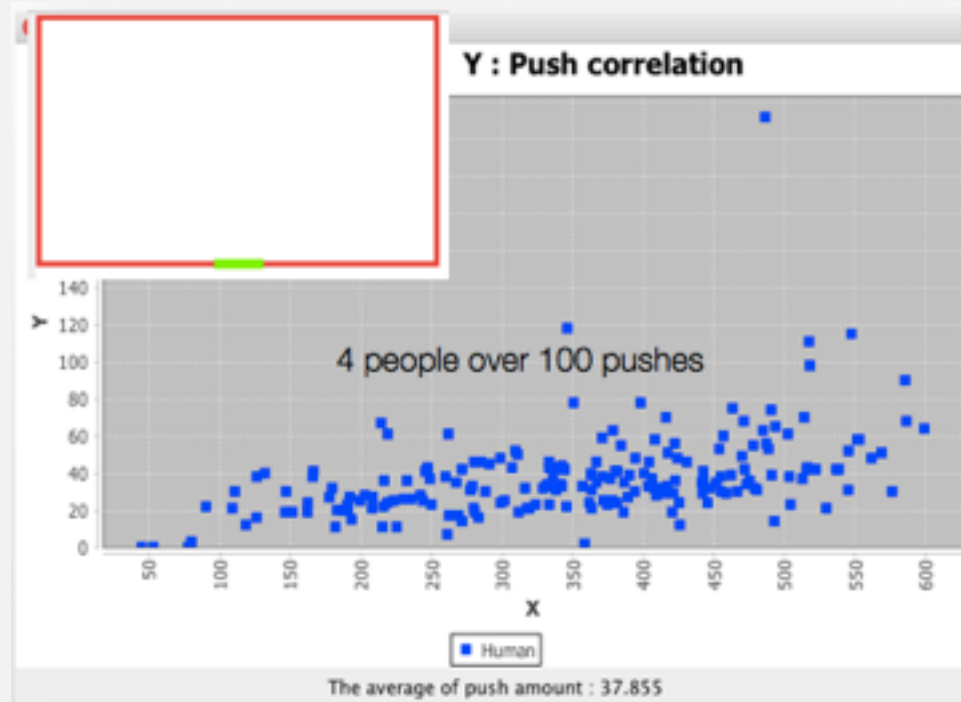
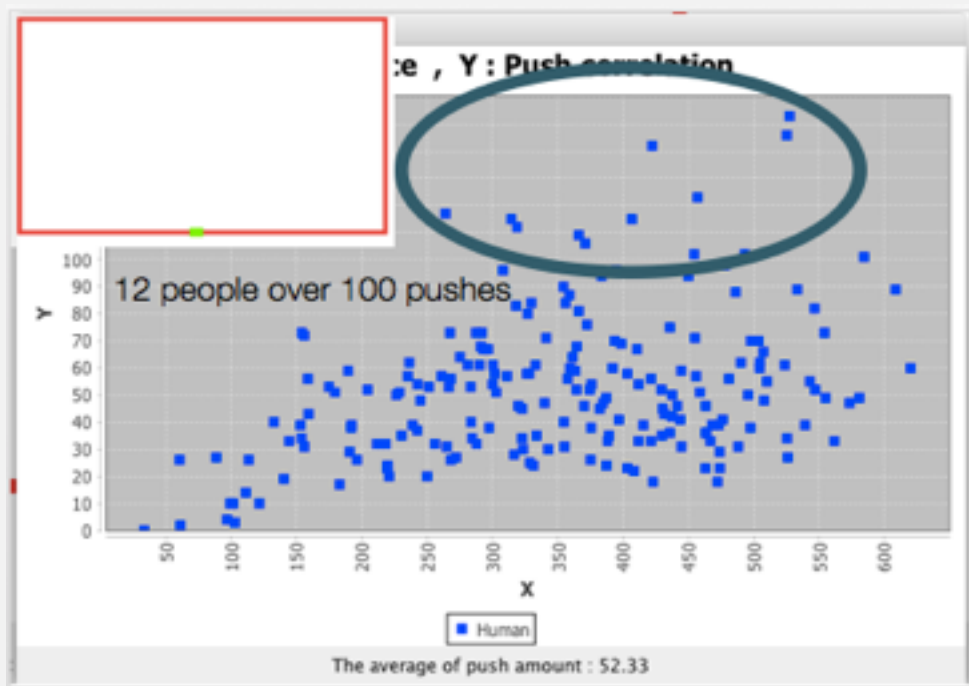
첫 번째 사진과 같이 문이 벽 가운데에 있는 경우에는 3명에서 4명의 사람들이 100이 넘는 충돌을 받은 것을 확인할 수 있었다. 하지만 두 번째 사진과 같이 문이 벽 구석에 있는 경우에는 무려 7명에서 8명의 사람들이 100이 넘는 충돌을 보였다. 충돌 횟수(push amount)이 100을 넘는다는 것은 실제로 압사로 인하여 사상을 입을 확률이 매우 높다는 것을 뜻하므로, 문을 구석에서 만드는 것보다 가운데에 만드는 것이 탈출하는데 있어서 더 효과적이라는 것을 입증한다.





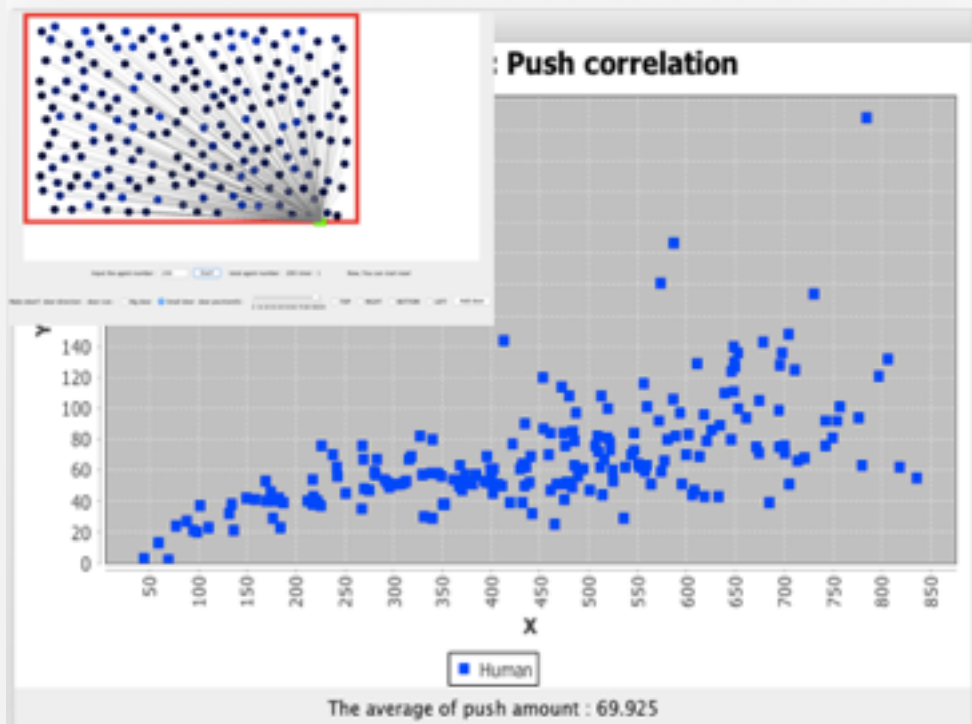
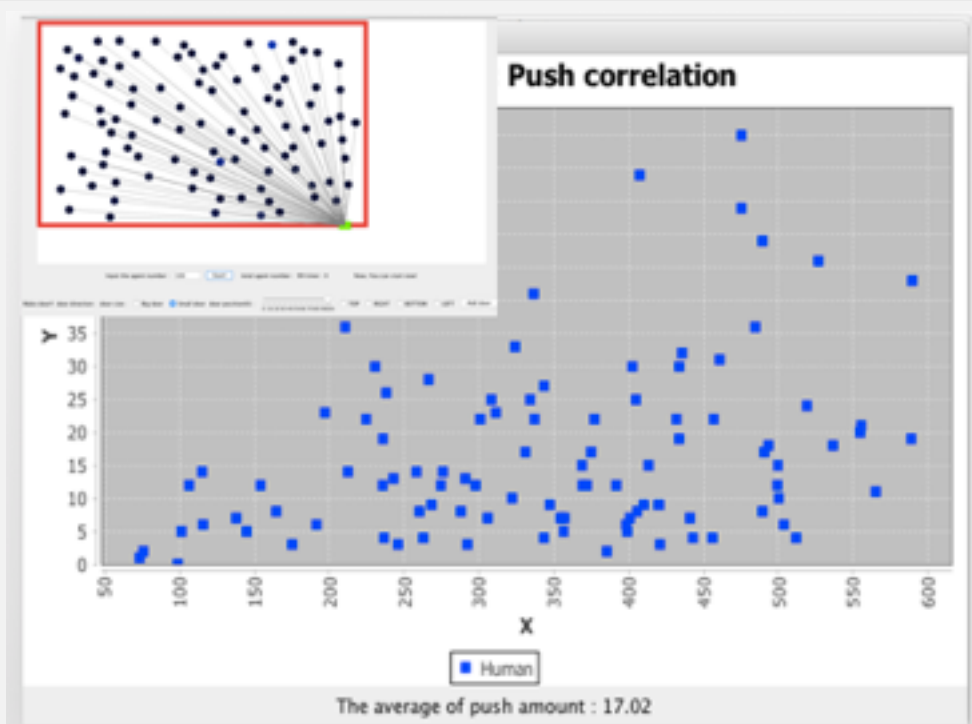
## 실험 2. 문의 수에 따른 방의 안전도 실험

첫 번째 사진과 같이 문이 한 개만 있는 경우 평균 충돌 횟수가 약 44이다. 하지만 두 번째 사진과 같이 문이 두 개인 경우 평균 충돌 횟수가 약 36에 이른다. 이 실험은 문을 1개에서 2개로 늘렸을 때, 약 10 정도의 충돌 횟수 수치가 감소한 효과를 확인할 수 있었다.



### 실험 3. 문의 크기에 따른 안전도 실험

첫 번째 사진과 같이 문의 크기가 작을 때는 12명의 사람들이 100이 넘는 총돌 수치를 보여줬다. 또한 평균 총돌 양도 약 52정도인 것을 확인할 수 있었다. 두 번째 사진과 같이 문의 크기가 클 때는 오직 4명의 사람들이 100이 넘는 총돌 양을 보였고, 평균 총돌 양도 약 37로 문의 크기가 작을 때 보인 수치보다도 약 15 정도 줄어든 수치를 얻을 수 있었다. 따라서 문의 크기 또한 위급 상황에서 사람들이 탈출하는 데 중요한 요소에 속한다는 것을 알 수 있었다.



#### 실험 4. 인구수에 따른 방의 안전도 실험 (100명 / 200명)

전 실험과는 달리 좀 더 흥미로운 주제를 가지고 방의 안전도 실험을 해보았다. 바로 방에 있는 인구 수가 탈출하는 데 있어서 얼마나 방해요소가 되는지에 관한 것이다. 첫 번째 사진과 같이 방안에 100명의 사람이 있을 경우 평균 충돌 횟수가 약 17 정도를 보여줬다. 하지만 두 번째 사진과 같이 방 안에 200명의 사람이 있을 경우에는 전 수치보다 약 4배 정도 높은 약 70이라는 의외의 수치를 확인할 수 있었다. 이로 인해서 건물이나 방에 최대 허용되는 인원수를 설정해 놓는 것이 위급상황에서 탈출할 때 얼마나 중요한 요소가 되는지 확인할 수 있었다.

## 응용 방안

이 프로젝트의 아이디어는 실제 건물에서 화재가 발생할 경우 단순히 경보음이 발생하는 것이 아닌 Sensor가 사람을 인식하여 사람과 탈출구까지의 거리를 계산하여 최적의 경로를 계산함으로써 탈출구를 찾는데 소비되는 시간이나 병목 현상이 발생하지 않도록 최적의 경로를 계산하여 바닥에 있는 화면을 통해 사람이 화살표를 보고 탈출구로 이끄는 아이디어를 생각했다.

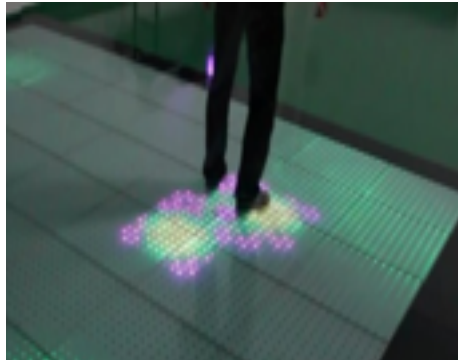


그림 2-1 바닥LED 패널



그림 3-2 사람인식센서

이 아이디어를 프로젝트에서 구현할 탈출 시뮬레이션을 통해 적용해보고자 한다. 또한 최적의 문 위치를 알 수 있게 만듦으로써 발전 가능성이 높다고 보았다.

## 적용기술

Sarl, Java, Multi-agent modeling