Introduction to RoboCup Rescue Simulation

Farshid Faraji
Ali Modaresi

Introduction

- Farshid Faraji
  - Executive Committee (EC) member of International RoboCup Federation in rescue simulation league.
- Ali Modaresi
  - Technical Committee (TC) member of International RoboCup Federation in rescue simulation league.

Content

- Introduction RoboCup
- RoboCup Rescue Simulation
- Research Areas
- Agents Behaviors
- Research Problems

What is RoboCup?

- An attempt to foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined

Why RoboCup is a good option for AI?

RoboCup

- dynamic environment
- real-time decision making and action
- high level of uncertainty and incomplete information
- sensor-acquired information
- distributed control and cooperation

Introduction to RoboCup

- Computer Chess vs. RoboCup

	Chess	RoboCup
Environment	Static	Dynamic
State Change	Turn taking	Real time
Info. accessibility	Complete	Incomplete
Sensor	Symbolic	Non-symbolic
Control	Central	Distributed

Introduction to RoboCup

- The Dream

By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup

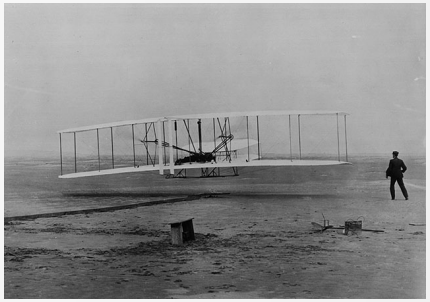
Can we accomplish the goal?

- Apollo Project
- Deep Blue

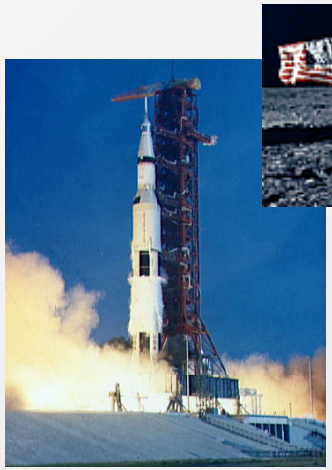
Apollo Project

- Dream
  - Send men to the moon and safely return them to the earth.
- Technologies
  - systems science, electronics, aviation, project management, etc.
- Effects
  - Major impacts to U.S. industries.

1903



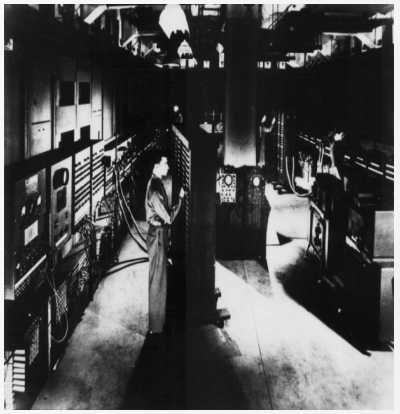
1969



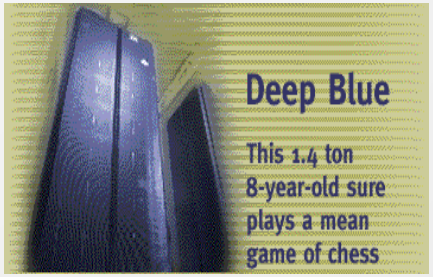
Computer Chess

- Dream:
  - to develop a computer that can beat human chess champion.
- Technologies:
  - Search algorithms, parallel computing, parallel machine architectures, etc.
- Effects:
  - Basic computer algorithms, parallel programming, etc.

ENIAC
1946



Deep
Blue
1997

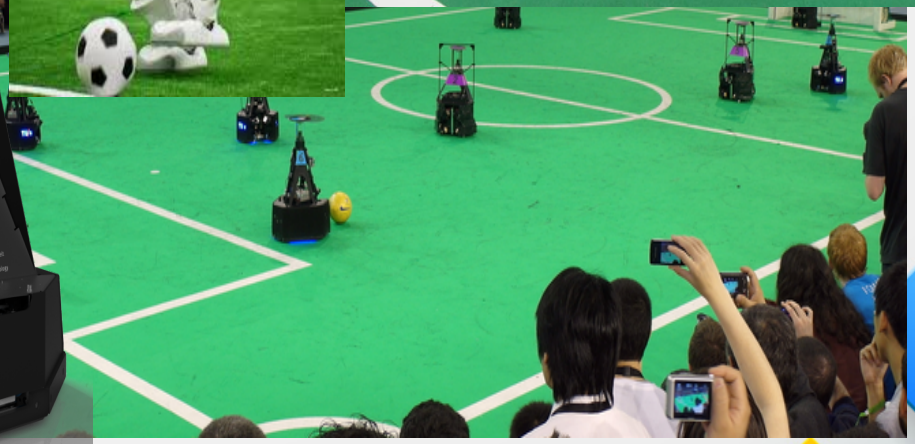


Introduction to RoboCup

- RoboCup major activities
  - RoboCup Soccer
  - RoboCup Rescue
  - RoboCup Junior
  - RoboCup @Home
  - RoboCup Industrial

Introduction to RoboCup

- RoboCupSoccer
  - Humanoid
  - Standard Platform
  - Middle Size
  - Small Size
  - Simulation

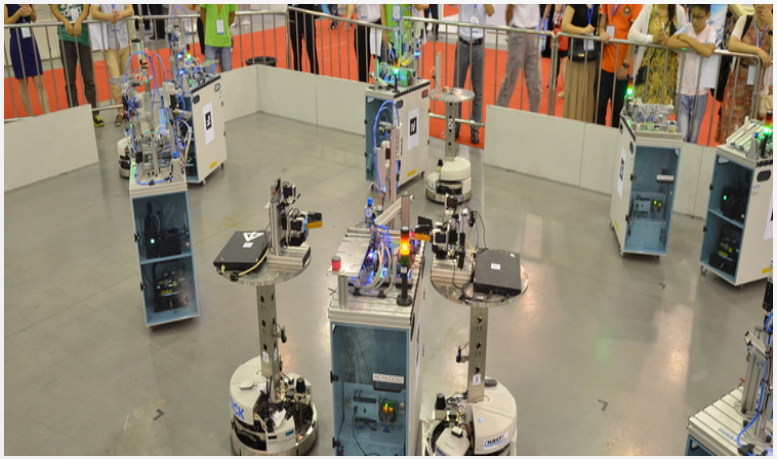


Introduction to RoboCup

- RoboCup@Home
  - Open Platform
  - Domestic Standard Platform
  - Social Standard Platform

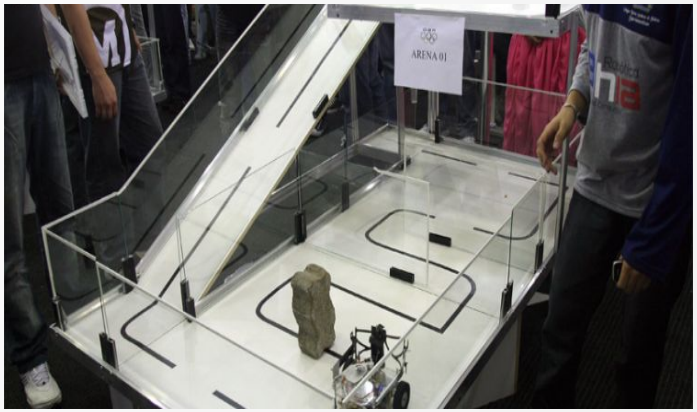
Introduction to RoboCup

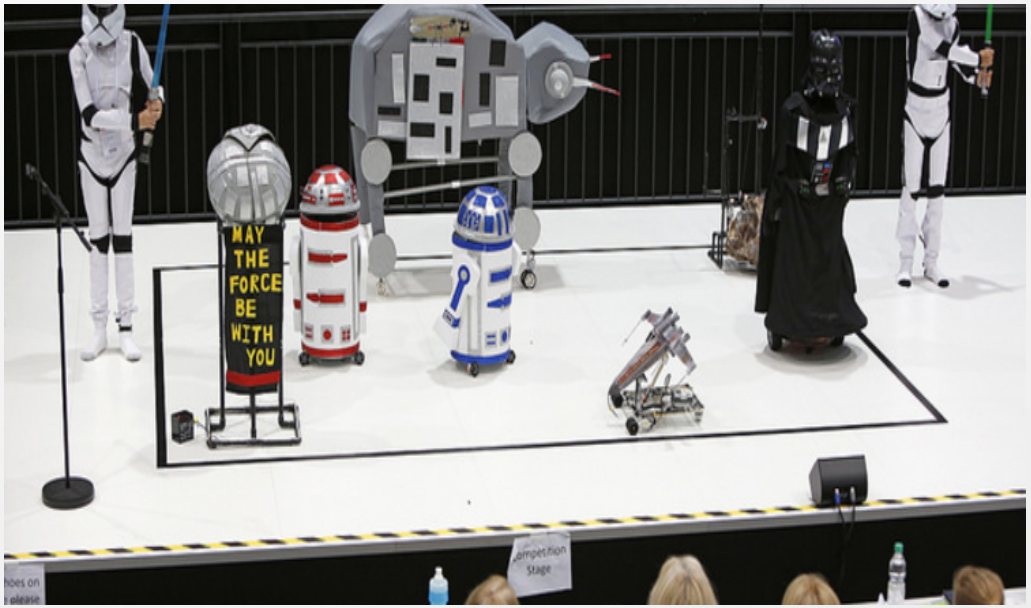
- RoboCupIndustrial
  - RoboCup@Work
  - RoboCupLogistics



Introduction to RoboCup

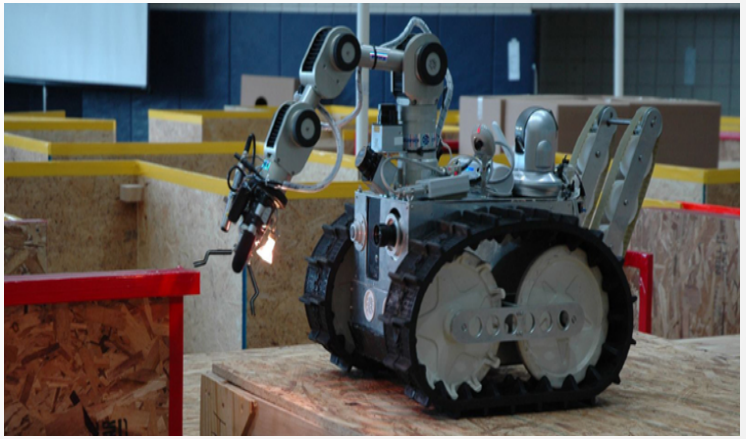
- RoboCupJunior
  - Soccer
  - OnStage
  - Rescue

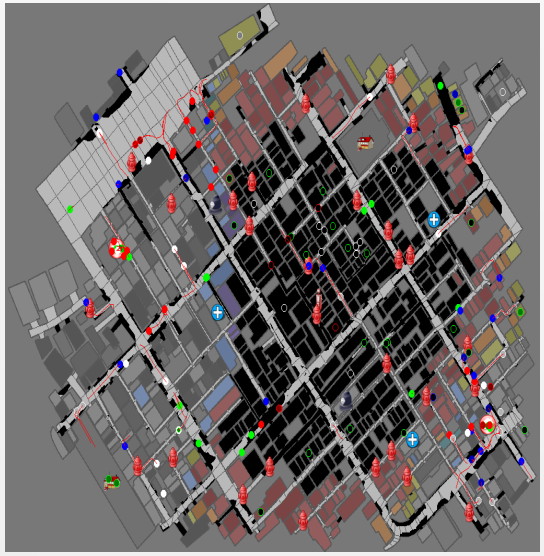


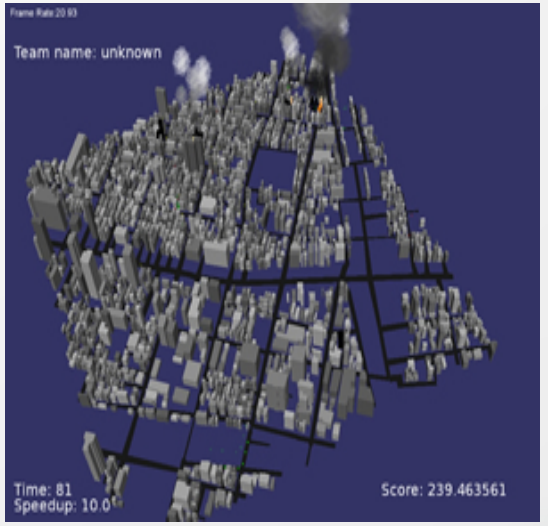


Introduction to RoboCup

- RoboCupRescue
  - Robot
  - Simulation







RoboCup Rescue Simulation

- Why RoboCup Rescue Simulation?
- What is RoboCup Rescue Simulation?
- How does RoboCup Rescue Simulation work?

Why rescue simulation

Earthquake name	Date(AD)`	Magnitude (M_W/M_S)	Description(Deaths, Injury)
The 2011 earthquake off the Pacific coast of Tohoku(2011 Tohoku earthquake)	11 March 2011	9.0 M_W	15,824 deaths - 5,942 injured - 3,847 missing
The 2010 Haiti earthquake	12 January 2010	7.0M_W	316,000 deaths - 300,000 injured
The 2008 Sichuan earthquake (Great Sichuan Earthquake)	12 May 2008	8.0-7.9M_W	69,195 dead - 374,643 injured - 18,392 missing
The 2005 Kashmir earthquake	8 October 2005	7.6M_W	79,000 dead - 106,000+ injured
The 2004 Indian Ocean earthquake	26 December 2004	9.1-9.3M_W	230,000 - 310,000 deaths
The 2003 Bam earthquake	26 December 2003	6.6M_W	26,271 deaths - 30,000 injured
The Great Hanshin earthquake (Kobe earthquake)	17 January 1995	6.8M_W	6,434 death - 300,000 homeless - 100 billion$ damage
The Manjil-Rudbar Earthquake	21 June 1990	7.4M_W	50,000+ deaths – Rudbar , Manjil , Lushan & 700 villages were destroyed - over 300 villages affected
The Tangshan Earthquake	28 July 1976	7.5M_W	242,000 to 779,000 dead
The Damghan Earthquake	22 December 856	7.9M_W	Approx. 200,000

- Roads are blocked.
- Communication lost.
- Aid organizations damaged.
- Fire spreads.
- Lack of precise knowledge of the damage.
- And …

Why rescue simulation

- **Kobe earthquake**
  - January 17, 1995 at 05:46:53 JST
  - Kobe, Japan
  - 6.9 M_w
  - 6,434 people lost their lives
  - 43,792 injured
  - 310,000 displaced
  - Total damage $200 billion USD

Why rescue simulation

- Japanese researchers
  - Developed a simulator that reproduces conditions similar to a urban post-earthquake.
  - Organized the RoboCup Rescue Agent Simulation League

What is Rescue Simulation?

- A system using disaster relief simulation under an inland earthquake in urban area.
- A Large Multi-Agent System which its aim is to manage the disaster when an earthquake happens.
- RoboCup Rescue Simulation System is one of the prominent systems for AI and Multi-Agent researches.

Main purpose

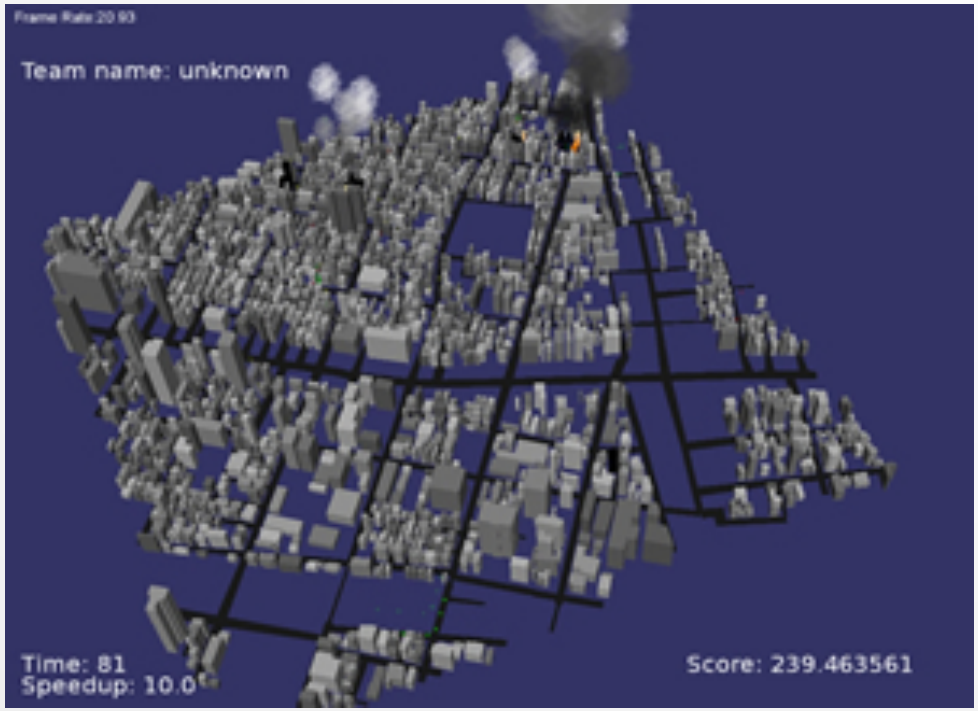
- RoboCup Rescue's main purpose is to provide emergency decision support by integration of disaster information, prediction and planning.

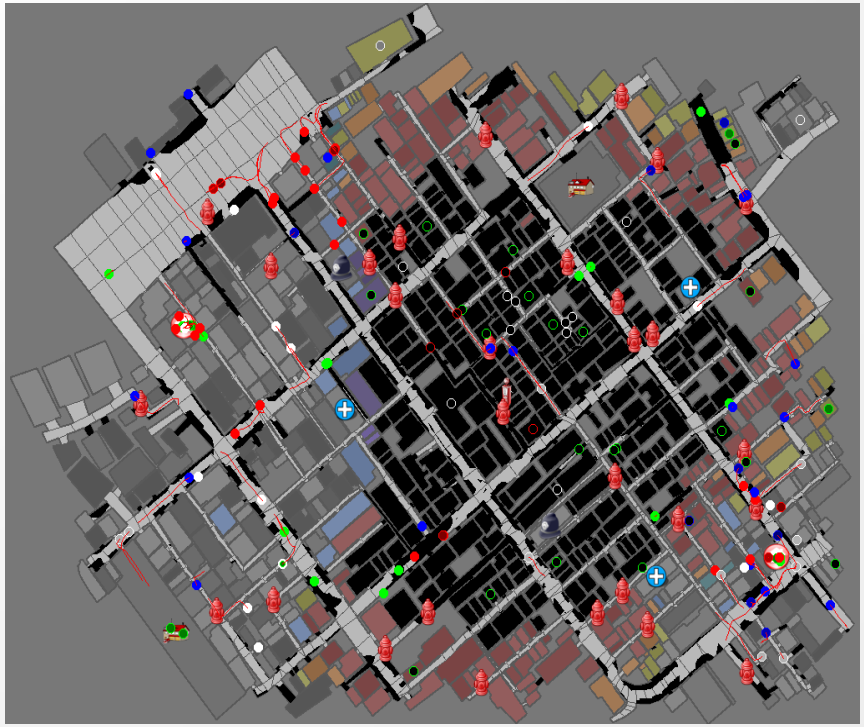
- challenges teams to design virtual robots to solve various challenges, which are evaluated in specially designed rescue simulations.

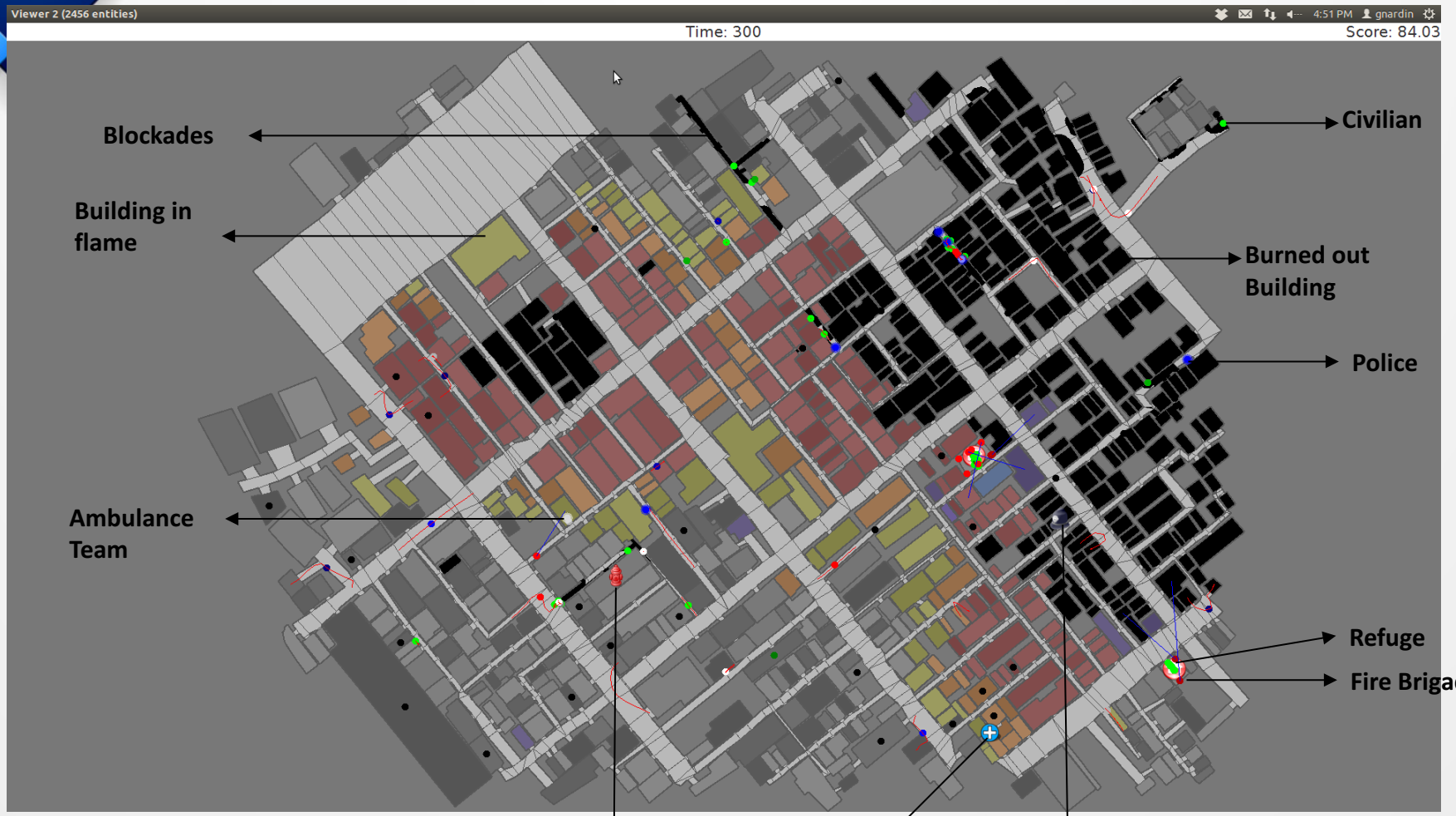
Research Areas

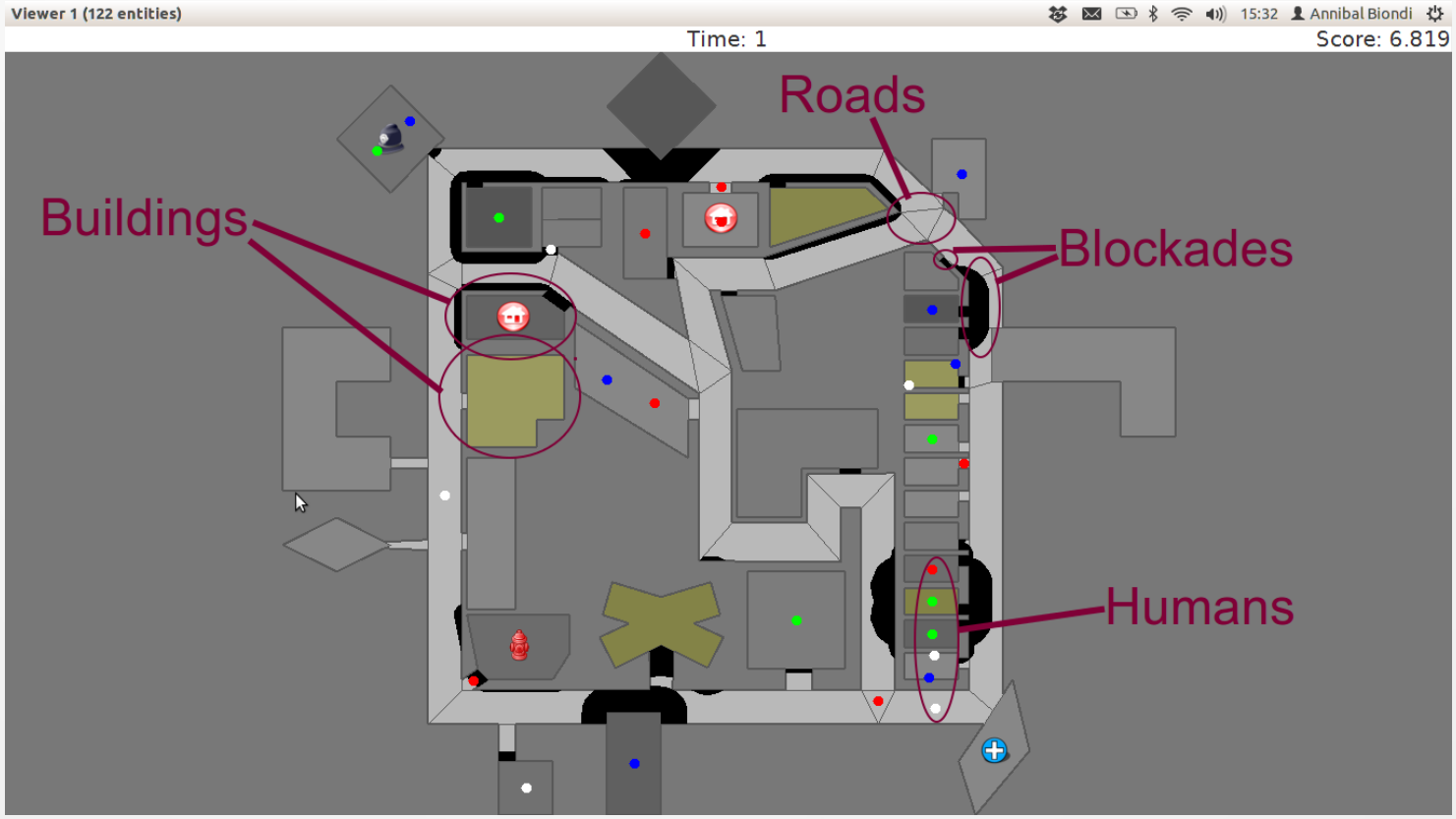
- Large Multi-Agent Systems
- Decision Making Algorithms
- Task Allocation Methods
- Multi-Agent Coordination and Team Formation Methods
- Behavior Modeling
- Path Finding
- Search Algorithms
- Exploration Algorithms
- …

What is rescue simulation?






Viewer 2 (2456 entities)
4:51 PM gnardin
Time: 300
Score: 84.03
Blockades
Building in flame
Ambulance Team
Civilian
Burned out Building
Police
Refuge
Fire Brigade
Fire Station
Ambulance Center
Police Office


Viewer 1 (122 entities)
15:32
Annibal Biondi
Time: 1
Score: 6.819
Roads
Buildings
Blockades
Humans

Agents

- There are two types of agents
  - civilians (Civilian  )
  - Rescue agents

Platoon

  **Ambulance Team**

  **Fire Brigade**

  **Police Force**

Center (Fixed)

 **Ambulance Center**

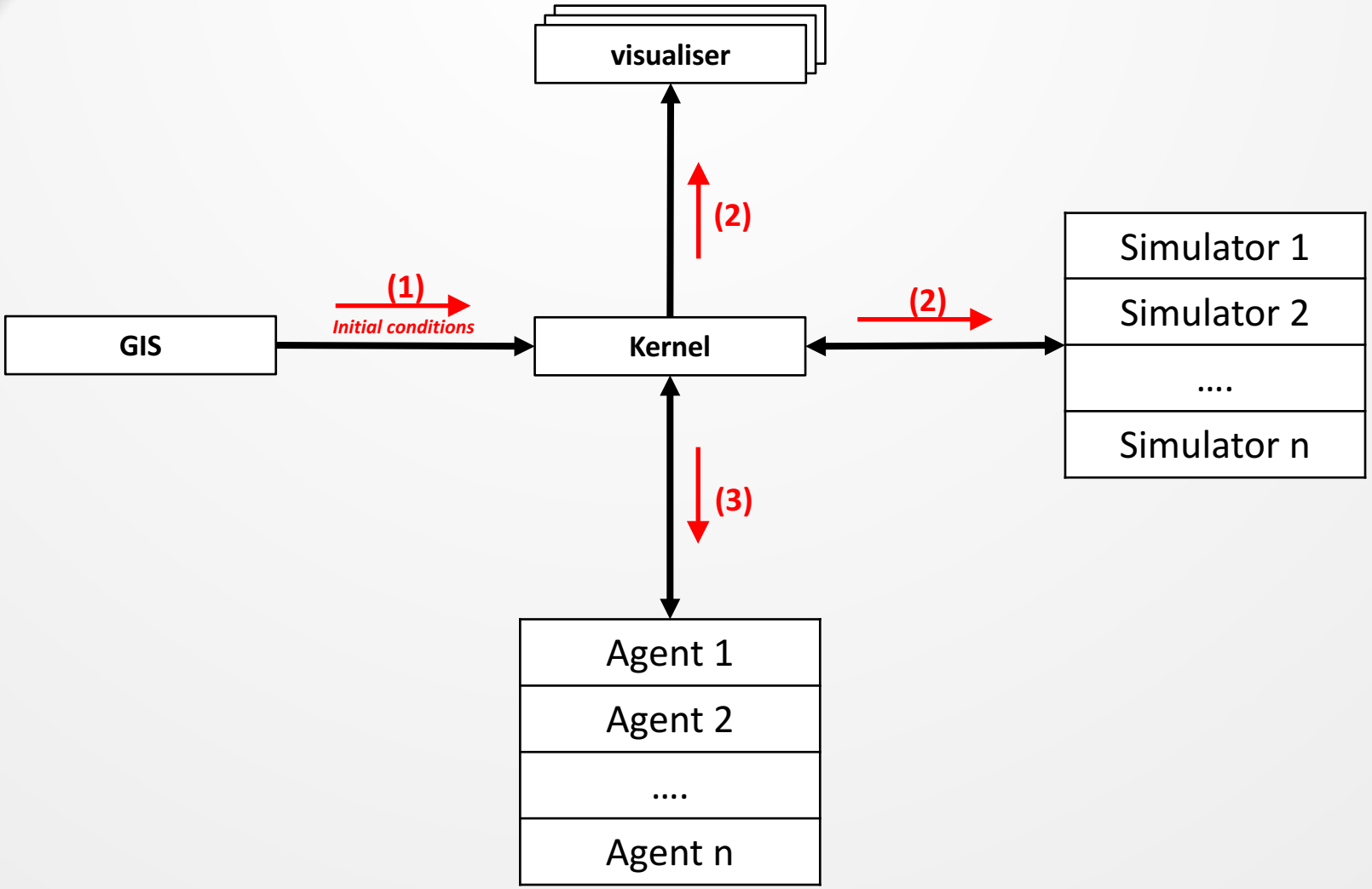
 **Fire Station**

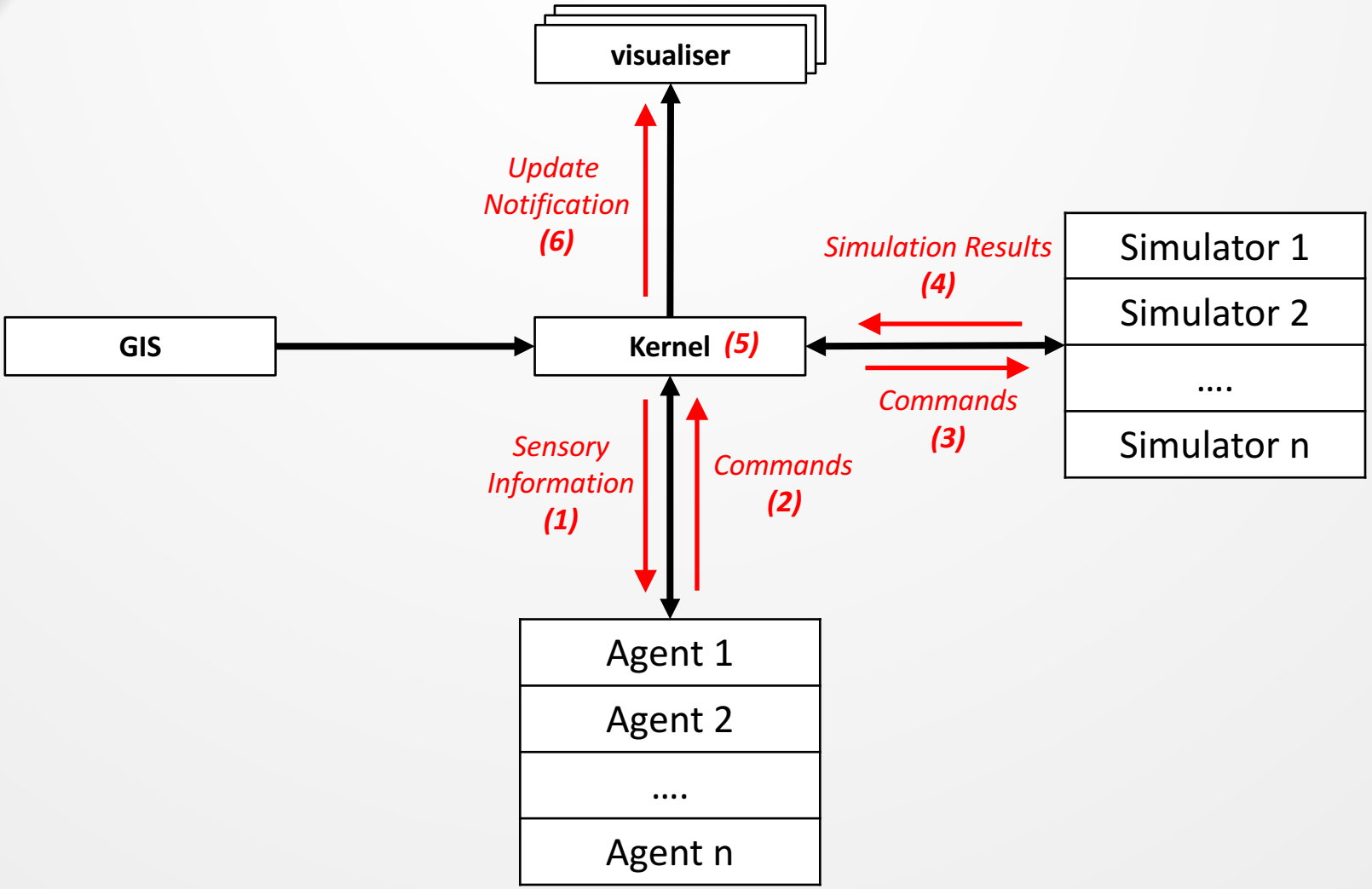
 **Police Office**

Simulators

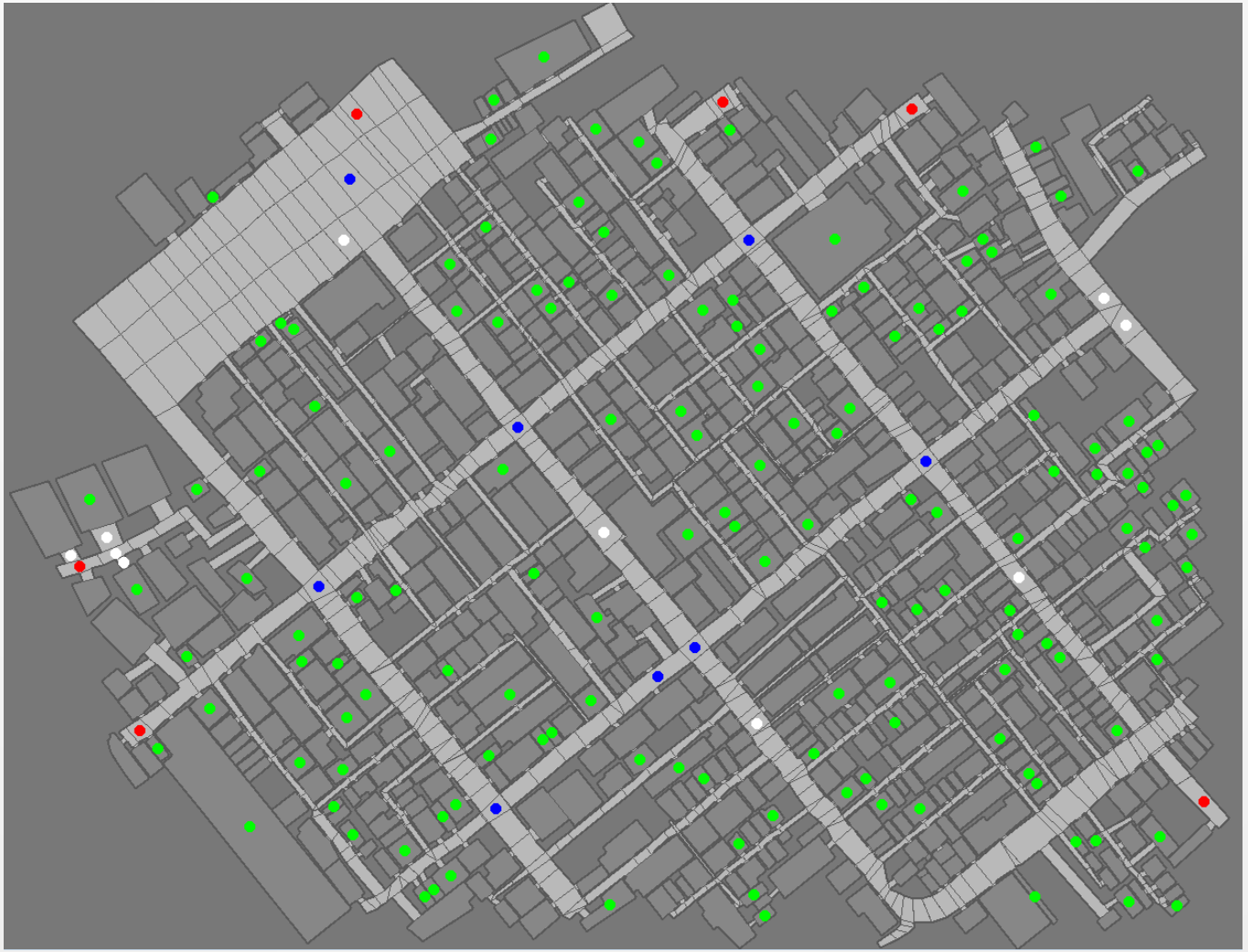
- Clear simulator
- Collapse simulator
- Ignition simulator
- Misc. simulator
- Fire simulator
- Traffic simulator

How Does Rescue Simulation Work?

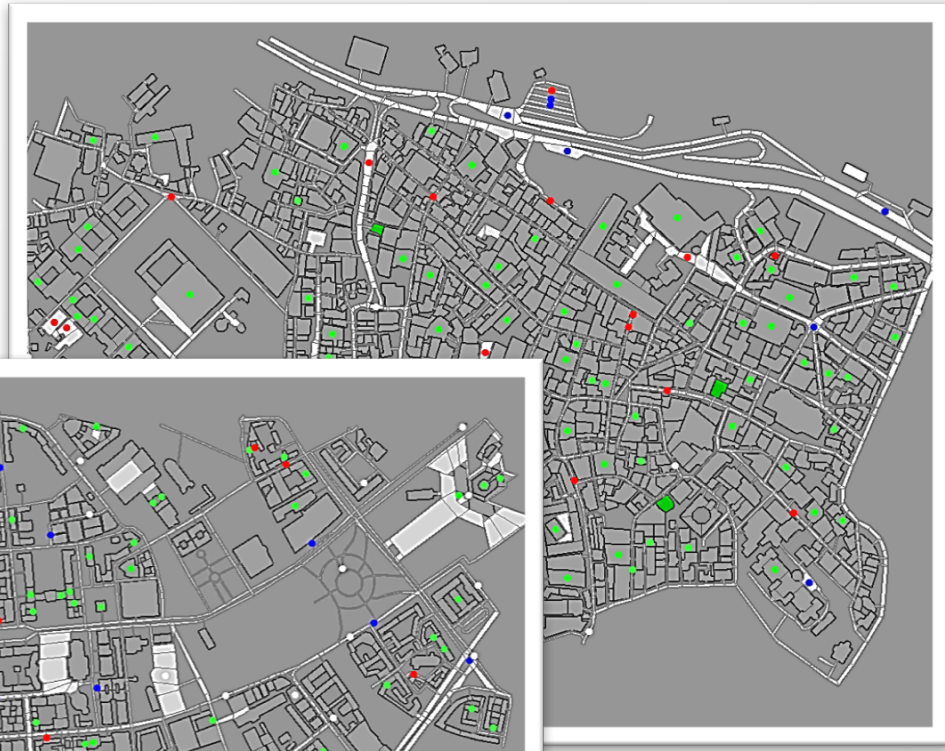

visualiser
(2)
(1)
Initial conditions
GIS
Kernel
(2)
Simulator 1
Simulator 2
....
Simulator n
(3)
Agent 1
Agent 2
....
Agent n

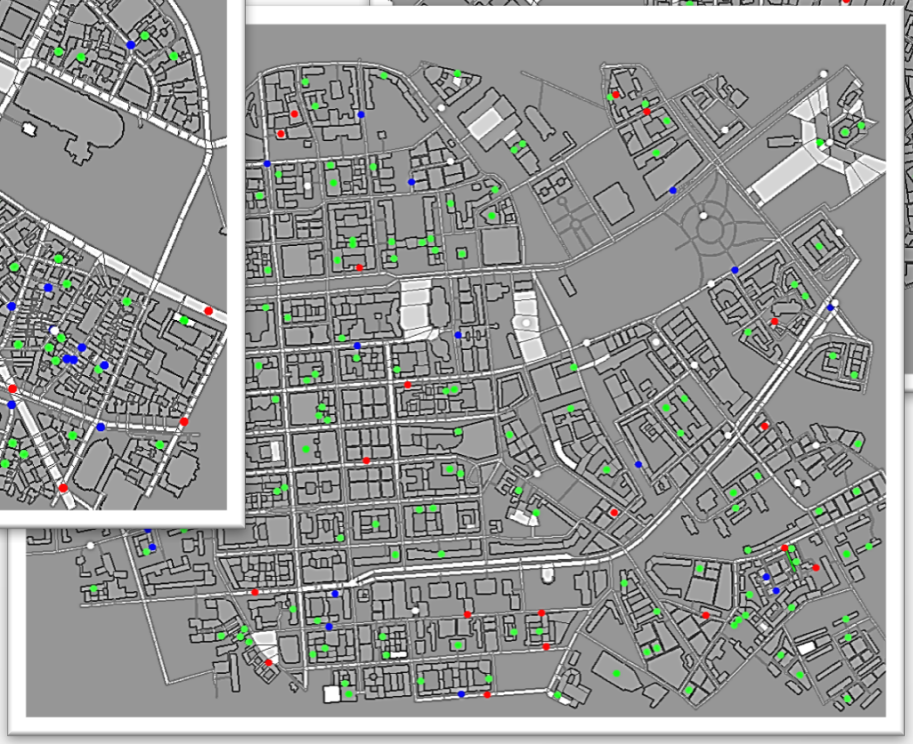

visualiser
Update Notification (6)
GIS
Kernel (5)
Simulation Results (4)
Commands (3)
Simulator 1
Simulator 2
....
Simulator n
Sensory Information (1)
Commands (2)
Agent 1
Agent 2
....
Agent n

Initial state

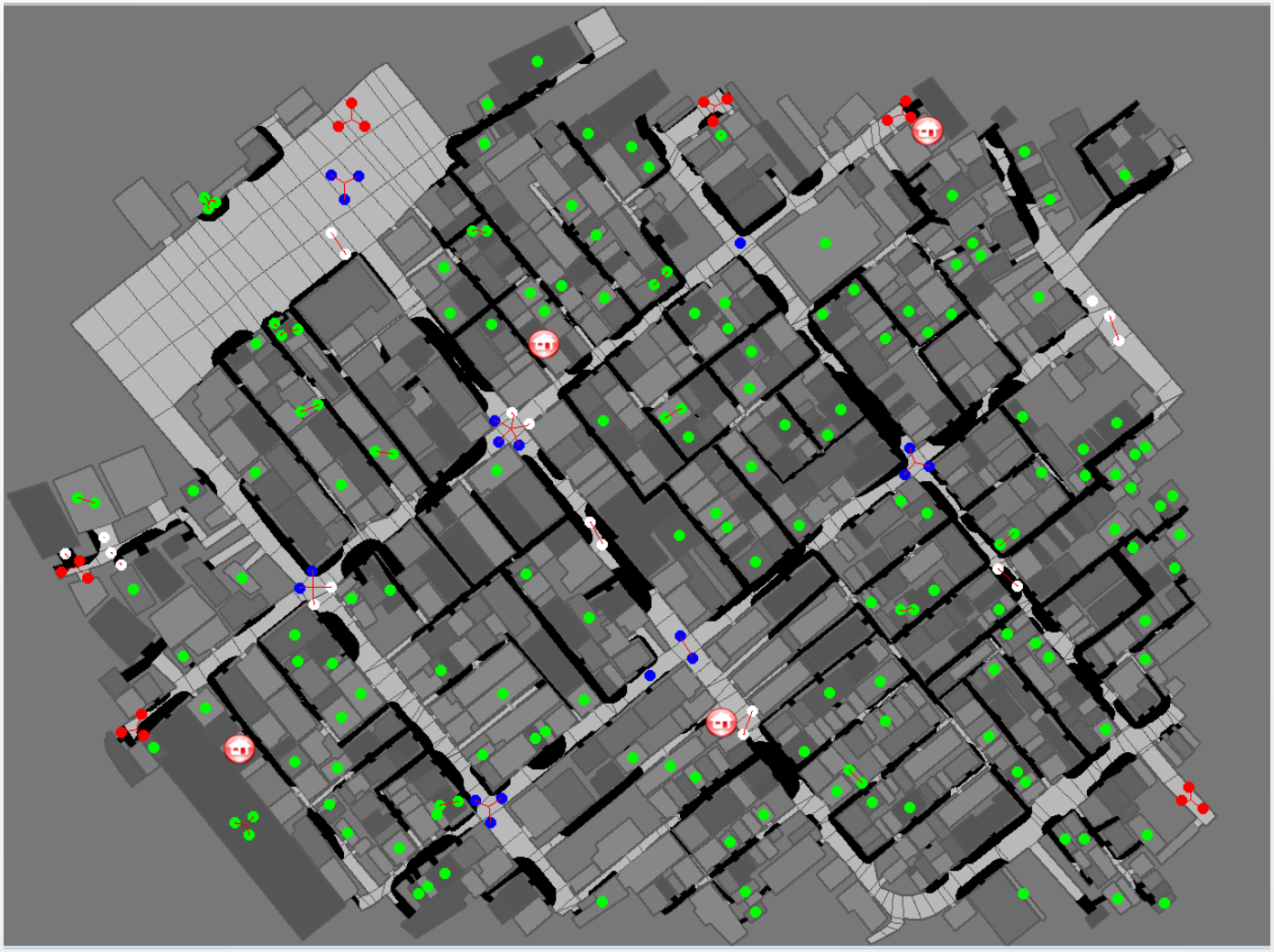


Maps and scenarios

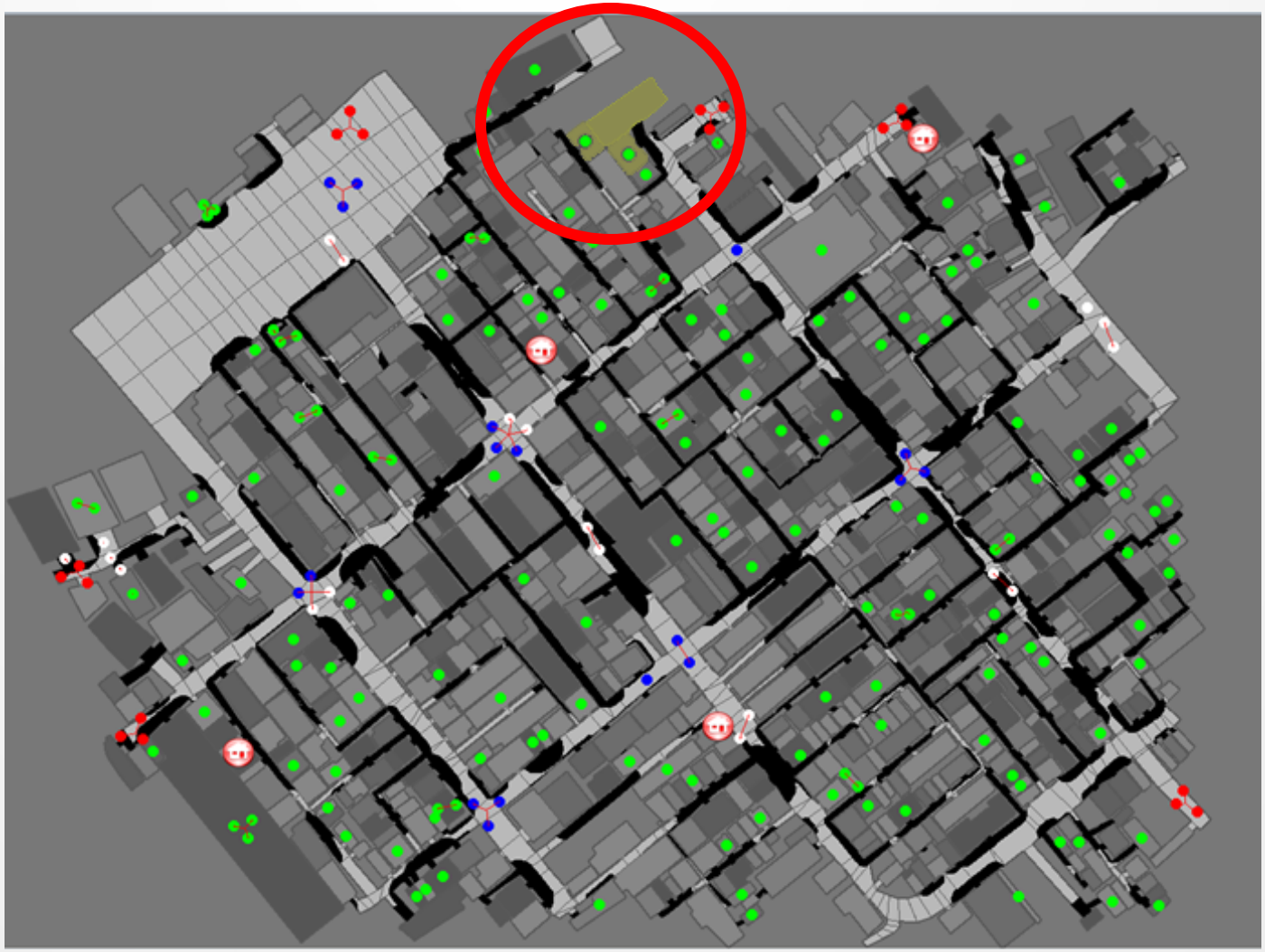




Cycle 1:Earthquake happened



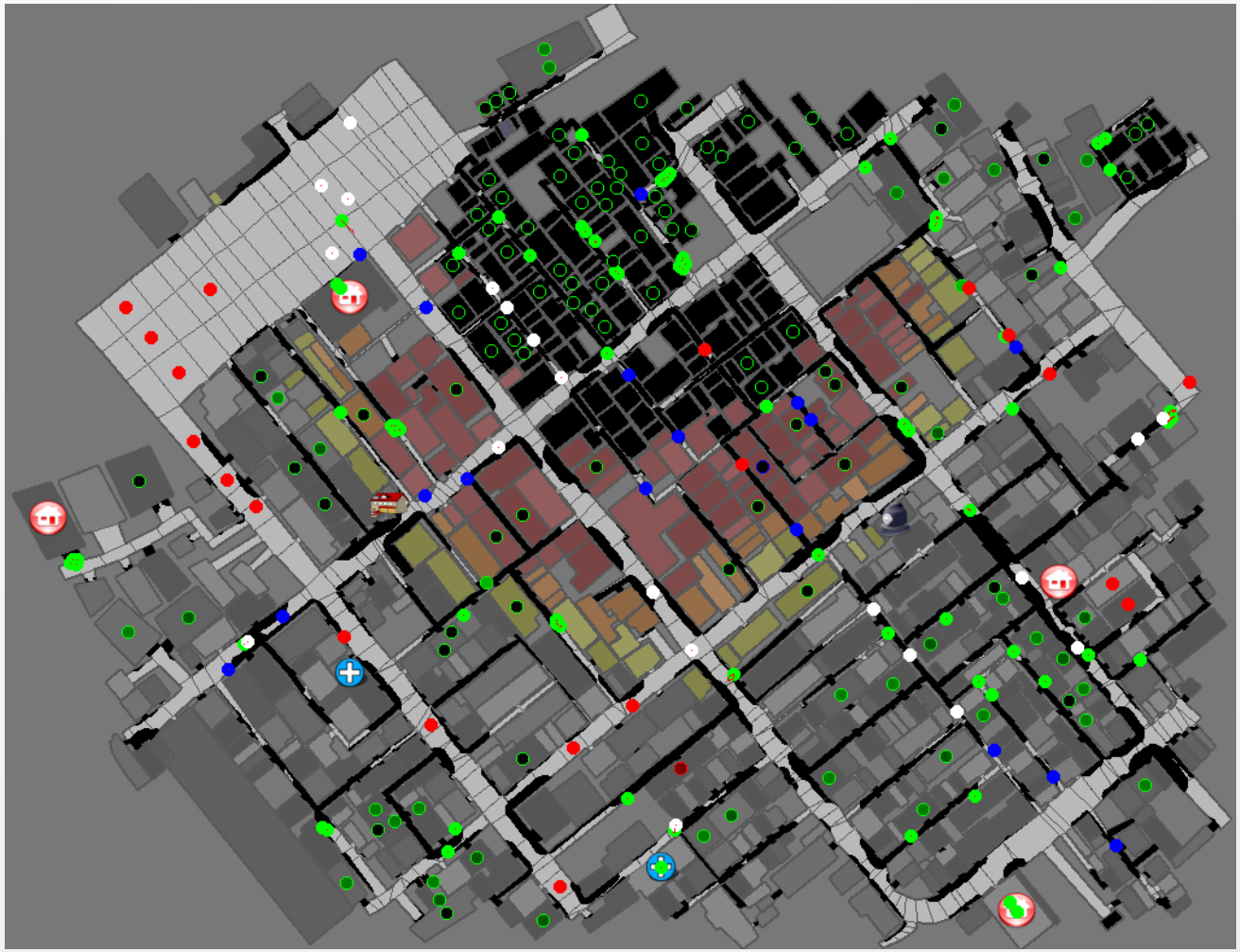
Cycle 2: Ignition started



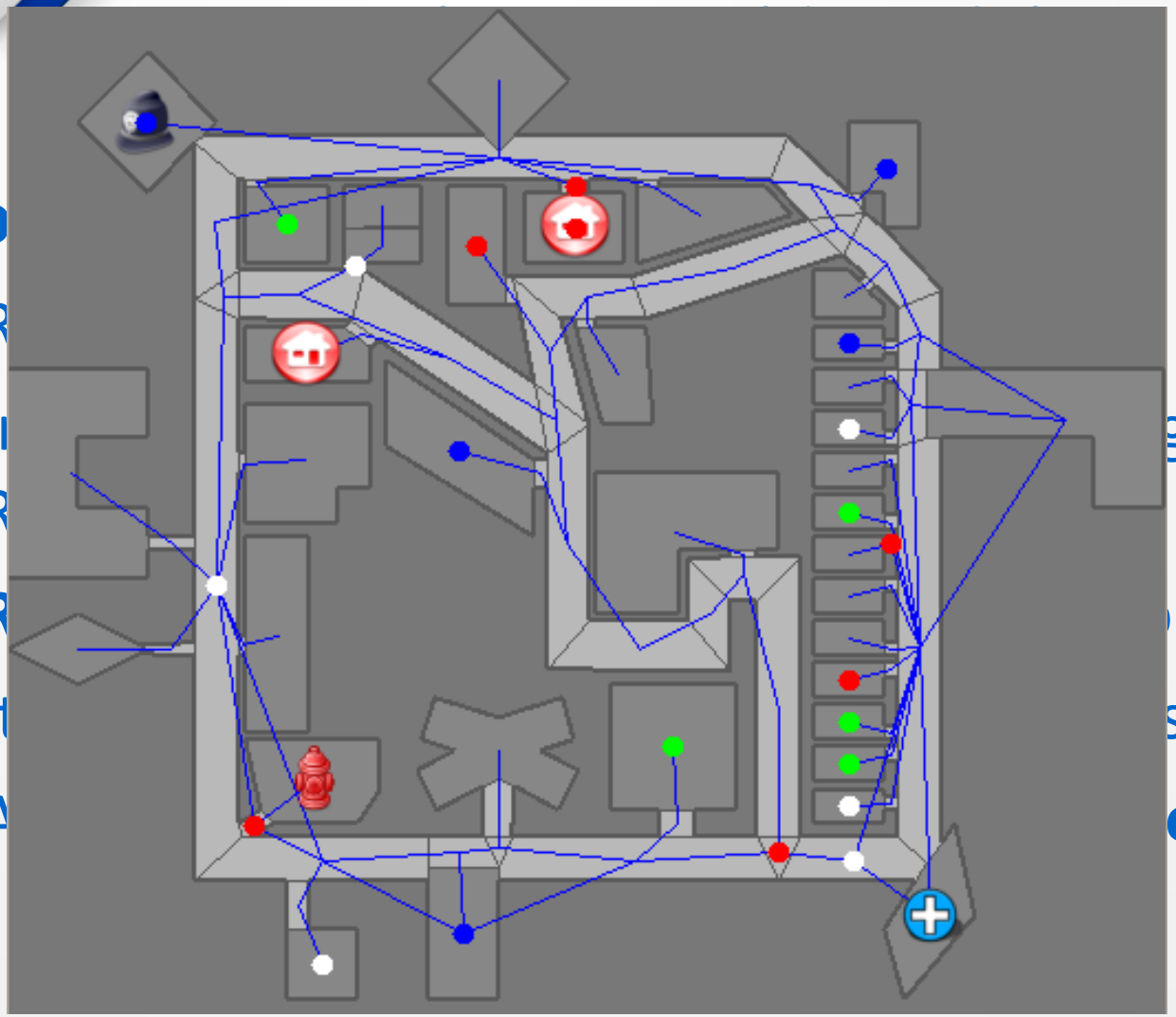
Cycle 3: starting agents mission

- Now agents start their mission which is mitigate the impact of the disaster by cooperation.
- **Police Forces** are responsible for **opening** blocked roads.
- **Ambulance teams** are responsible for **rescuing** buried humans.
- **Fire brigades** are responsible for **extinguishing** fiery buildings.

Final Cycle



- **Wo**
  - R
  - I
    R
  - **R**
  - It
  - A



gs,

h.

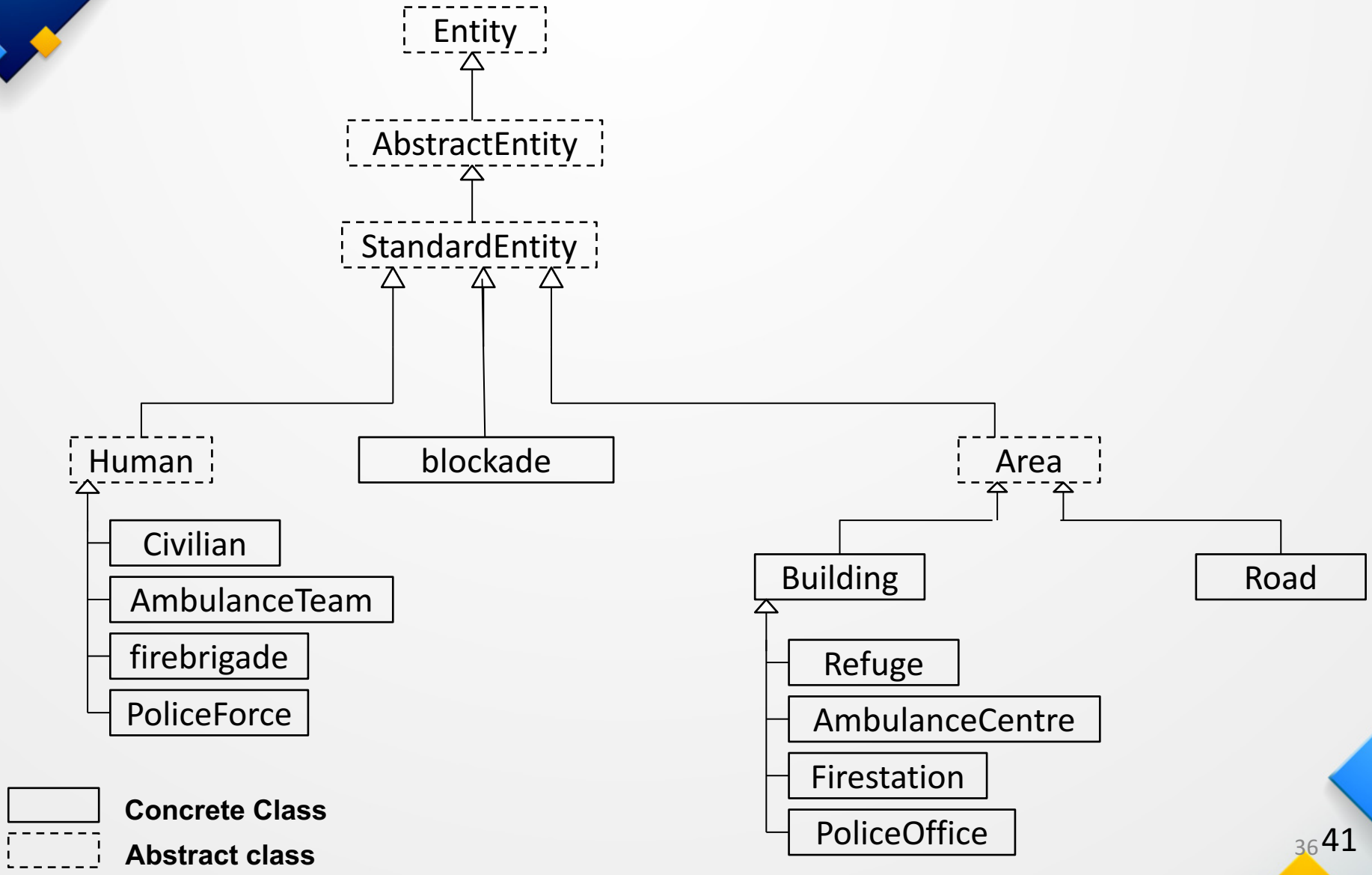
ss.

odel

useful Methods of StandardWorldModel

- **getEntities()**
  - Returns all world model entities
- **getEntitiesOfType()**
  - Returns entities of one or more particular type
- **getObjectsInRange()**
  - Returns entities within the radius range
- **getDistance()**
  - Returns the Euclidean distance between the two entities

Class Hierarchy
World model entities



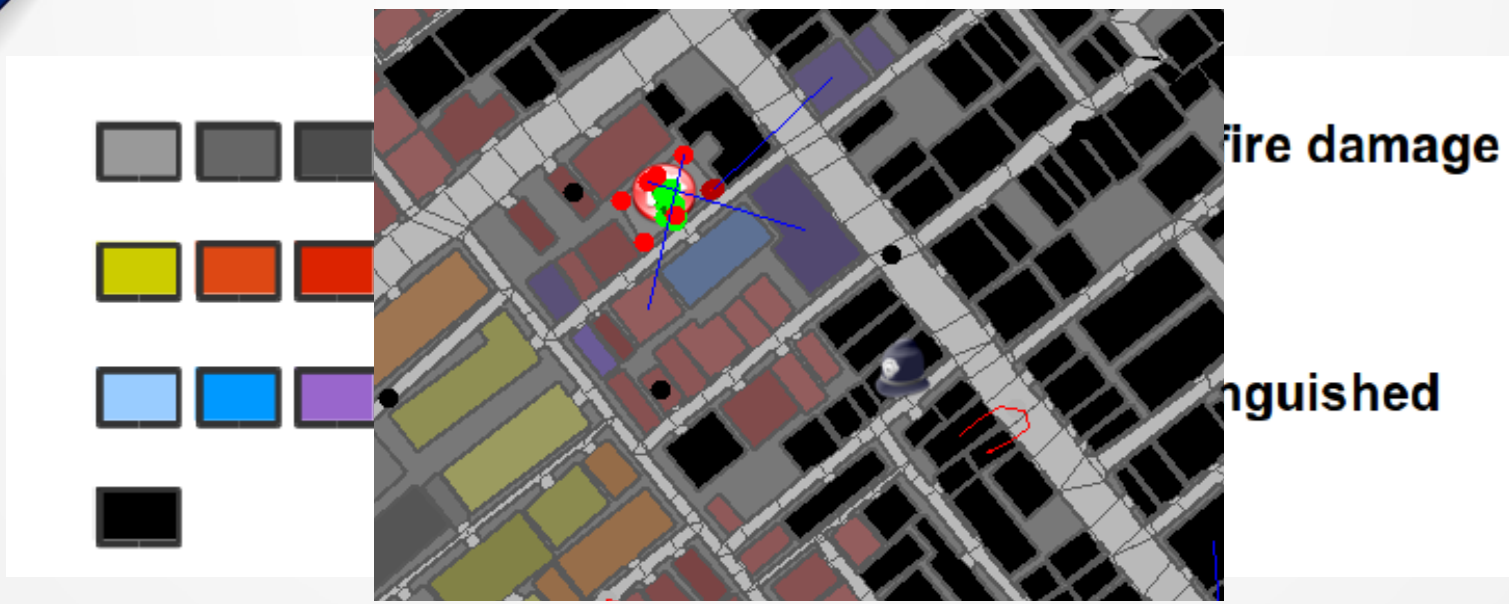
What is Building?

Extends from class **Area**

properties

Property	Description
ID	building identification number
brokenness	Indicates how the construction is damaged
Fieryness	Indicates the intensity of fire
Temperature	Indicates the temperature of building
Ground Area	Building area
Floors	Number of floors in the building
Type	Type of the building (wooden, steal, conceret)
Edges	Represent the edges of the Building shape

Building



0 → Unburnt	5 → Minor damage
1 → Heating	6 → Moderate damage
2 → Burning	7 → Severe damage
3 → Hell	8 → Burnt out
4 → Water damage	

Building

BuildingCode:

the type of construction material code

Code	Kind	Transmission Rate(default)
0	wood	1.8
1	Steel	1.8
2	Concrete	1.0

Road

- **Road**
  - Represents the ways
  - Extends from class **Area**
- properties

Property	Description
ID	The identification number
Blockades	List of existing blockades on the road
Edges	Represent the edges of the road shape

Road

- Methods useful class **Road**

  List<EntityID> getBlockades()

  Returns the block list in the track area

  List<EntityID> getNeighbours()

  Returns the list of neighboring objects to such a route

Blockade

- **Blockade**
  - Is a barrier on the road

- properties

Property	Description
ID	identification number
Position	on which road the blockade is positioned
Repair cost	blockade repair cost
Edges	Represent the edges of the blockade shape

Humans

- Is the real human in the real world.
  - Civilian, Ambulance Team, Police Force, Fire Brigades
- Properties

Property	Description
ID	agent identification number
X	X coordinate on the map
Y	Y coordinate on the map
Buriedness	indicates the amount of burieds
HP	It indicates how much the agent is healthy
Damage	Rate of decreasing HP value
Position	Entity over which the agent is positioned

Behavior of agents

- Capabilities
- Police Force
- Ambulance Team
- Fire Brigade

Behavior of agents

capabilities

Kind	capabilities
Civilian	Rest, Hear, Say, Move
Ambulance Team	Rest, Hear, Say, Move, Communicate via Radio, Rescue, Load, Unload
Fire Brigade	Rest, Hear, Say, Move, Communicate via Radio, Extinguish, Fill The Tank
Police Force	Rest, Hear, Say, Move, Communicate via Radio, Clean
Ambulance Center	Hear, Communicate via Radio with higher bandwidth
Fire Station	Hear, Communicate via Radio with higher bandwidth
Police Office	Hear, Communicate via Radio with higher bandwidth

Ambulance Team Agent

- Search for civilian
- Choose best civilian
- Move to civilian
- Rescue Civilian
- Load Civilian
- Carry to Refuge
- Unload

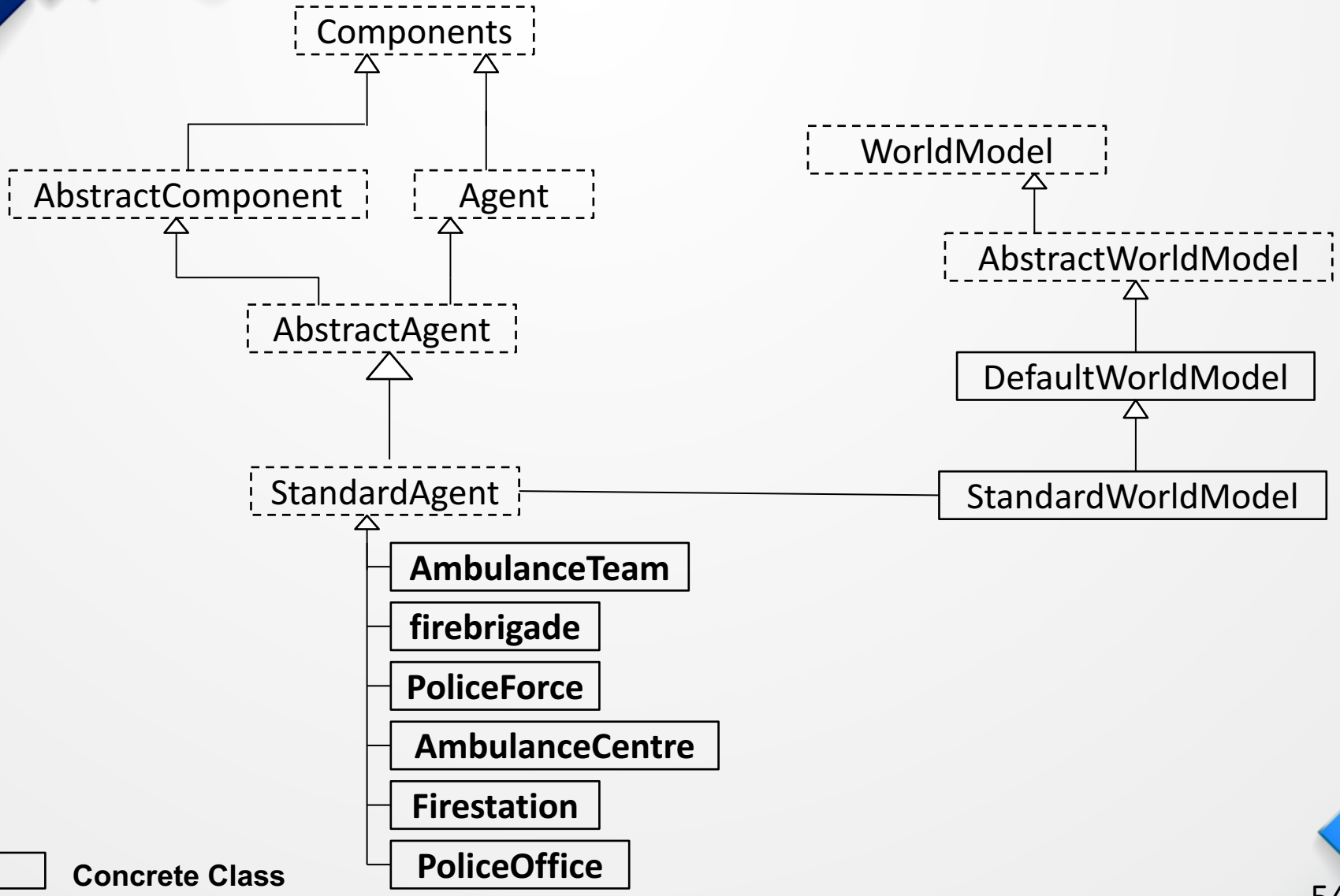
Fire Brigade

- Search for Fires
- Move to fiery buildings
- Extinguishing Fire
- Refill water tank

Police Force

- Search (blocked roads, Fiery building and civilians)
- Choose best blockade
- Clear the blockade

Agent Class Hierarchy



Behavior of agents

- How to implement an agent?

```
public class [NOME_CLASSE_AGENTE] extends StandardAgent<[StandardEntity]> {

    @override
    protected EnumSet<StandardEntityURN> getRequestedEntityURNsEnum(){
          return EnumSet.of(StandardEntityURN. [StandardEntityURN]);
    }

    @override
    protected void postConnect(){
    }

    @override
    protected void think(int time, ChangeSet changed, Collection<Command>
    heard){
    }
}
```

Behavior of agents

Introduction

- **`protected EnumSet<StandardEntityURN> getRequestedEntityURNsEnum()`**

  Method that returns the entity type implemented by that class

- **`protected void postConnect()`**

  Method performes only once after the connection to the simulation platform *kernel* and before the start of the simulation.

  Used to perform **pre-processing** of information received from the simulator before the simulation

  All agents have to complete the execution of the method **postConnect** after a timeout (default 2 minutes).

Behavior of agents

- **`protected void think(int time, ChangeSet changed, Collection<Command> heard)`**

  Method that implements the operation of the agent and is called by each simulation cycle.

  This method has a time to run (default 1 second).

Behavior of agents

- Accessing the parameter values of the configuration files

  this.config.GetIntValue([key])

  at where, **[key]** is the name of the specified parameter in a configuration file

- Example

  this.config.GetIntValue("perception.them.MAX_DISTANCE")

  this.config.GetIntValue("fire.extinguish.MAX_DISTANCE")

Behavior of agents

- Sample agent **AmbulanceTeam**

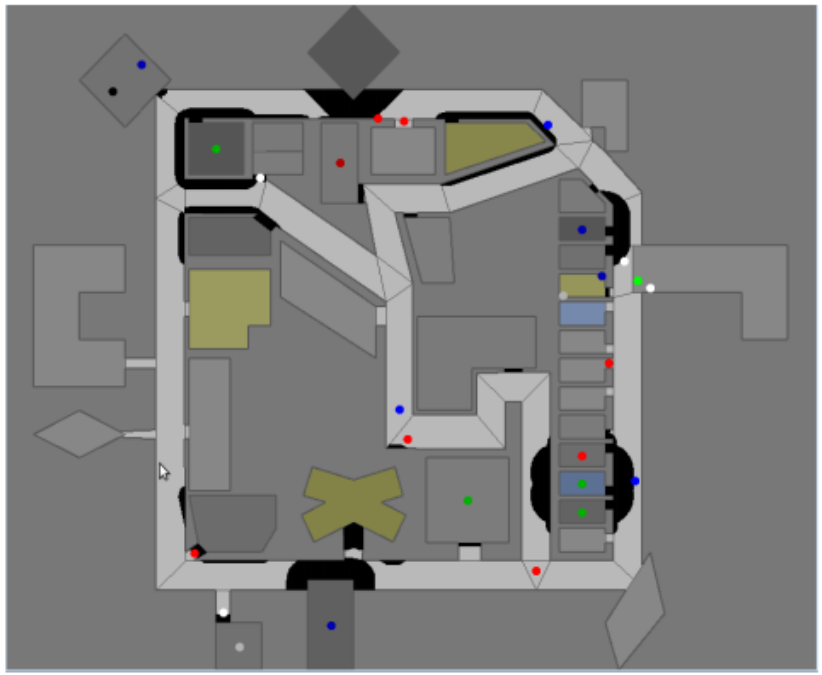
```
public class ExemploAT extends StandardAgent<AmbulanceTeam> {

    @override
    protected EnumSet<StandardEntityURN> getRequestedEntityURNsEnum(){
          return EnumSet.of(StandardEntityURN.AMBULANCE_TEAM);
    }

    @override
    protected void postConnect(){
          int MAX_DISTANCE =
                    this.config.GetIntValue( "perception.them.MAX_DISTANCE");
          ...
    }

    @override
    protected void think(int time, ChangeSet changed, Collection<Command>
    heard){
          ...
    }
}
```

Behavior of agents
capabilities

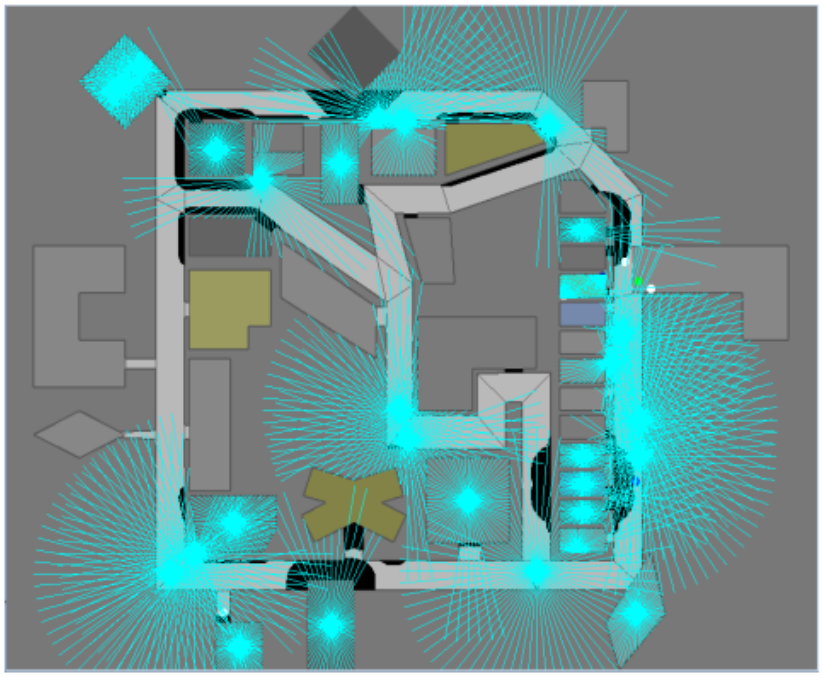
- **Sense**

  This capability enables the agent perceive the environment delimited by his view. These perceptions are received by the agent through the parameter **ChangeSet** in **think** method.

  The key **perception.them.MAX_DISTANCE** defining the range of vision of the agents specified in the file **perception.cfg**



(a) No LoS



(b) LoS

Behavior of agents

capablities

- **Hear**

  This capability allows the agent to receive messages from other agents by means of communication. The messages are received by the agent as a whole by the parameter **heard in** method **think**.

Behavior of agents
capabilities

- **Say**

  This capability enables the agent to transmit a short voice message.

- **Communicate via radio (Speak)**

  This capability enables the agent to send a message by radio communication.

Behavior of agents

capacities

- **Move**

  This ability allows the agent to move in the environment.

  Method / Command

  void sendMove(int time, List<

  Command used to mo
  interconnected entitie

  void sendMove(int time, List<EntityID> Path, int destX, int destY)

  Command used to move the agent by a sequence of interconnected entities in the graph and to move to the final X and Y coordinates.

Behavior of agents

Police Force

- **Clean**

Th a

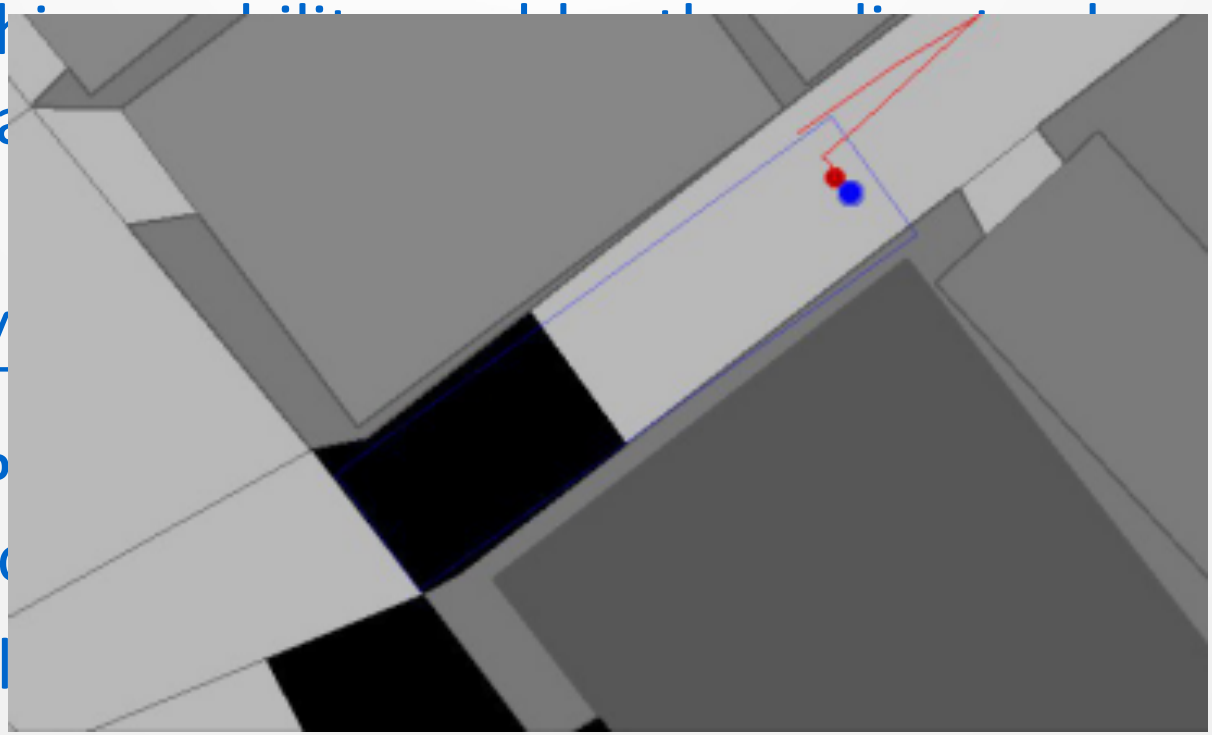
pa

M

vo

C

bl



Behavior of agents

Fire Brigade

- **Extinguish**

  This ca ut water on a bu

  Metho

  void send)

  Comma of water (**power**

Behavior of agents

Fire Brigade

- **Fill the tank**

  This capability allows the firefighters to refill their water tank.

  Method / Command

  void sendRest(int time)

  Command used to the agent standing on the refuge while filling your water tank

Behavior of agents

Ambulance Team

- **Res**

  This eam to
  une

  Me

  void

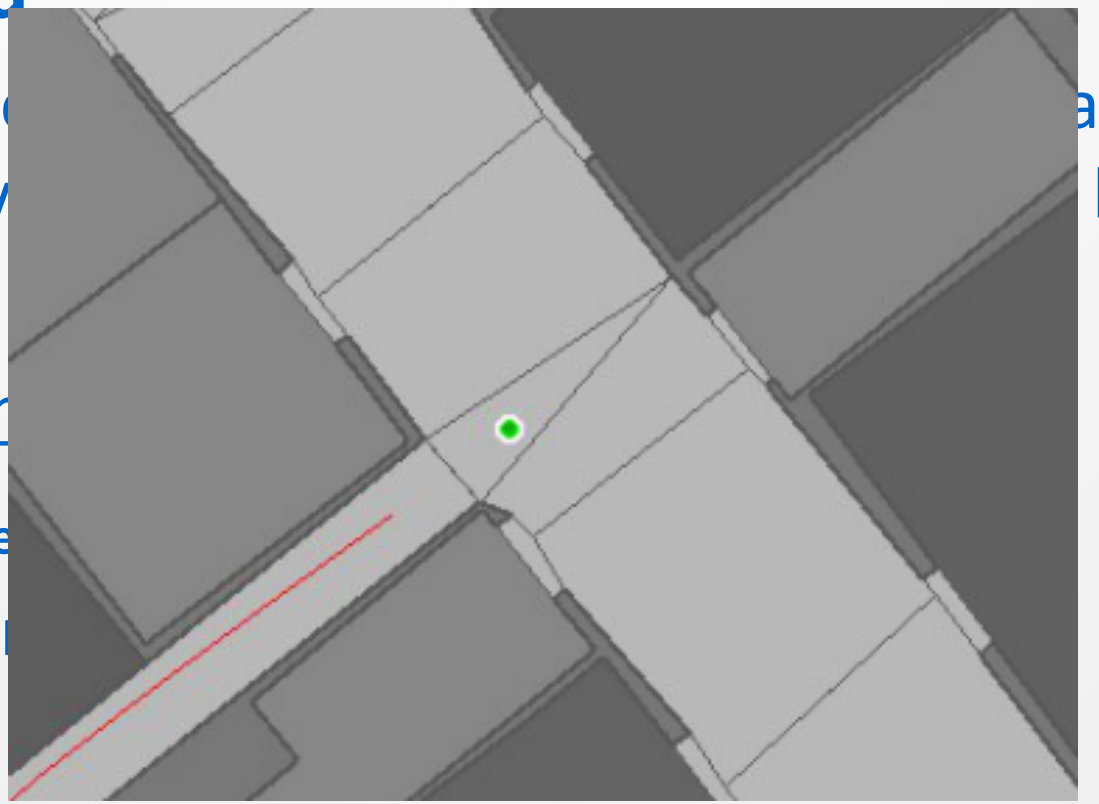
  Cor th buried
  human (**target**)

Behavior of agents

Ambulance Team

- **Load**

  This (am to carry location.

  Meth

  void se

  Com

Behavior of agents

Ambulance Team

- **Unload**

  This capability enables an ambulance team to unload the human which is carrying.

  Method / Command

  void sendUnload(int time)

  Command used to unload the human who is carrying

Research Problems

- There are many multi-agent research problems that can be investigated using the RoboCupRescue simulation package. Researchers can choose which aspects of the system they are interested in investigating.

- Task allocation with uncertainty
- Coalition formation
- Co-operation
- Distributed vs. centralized control
- Communication

Research Problems

- Task allocation with uncertainty

A core part of the standard scenario is allocating tasks to multi-agents. At any point in time there will be a number of fires, injured or buried civilians and blocked roads. The agents will know about some subset of these tasks but probably not all. Decisions must therefore be made about whether to search for new tasks, and how to allocate tasks given that new tasks may appear at any time.

Research Problems

- Coalition formation

A GROUP of agents are generally required for efficient allocation in the earthquake domain. Civilians trapped in building rubble will generally require the combined efforts of several ambulances to be rescued before they die of their injuries.

Research Problems

- Co-operation

The earthquake scenario generally requires different types of agents to cooperate. For example, the roads leading to an injured civilian or the hospital may be blocked and must be cleared by police before an ambulance can get to the target. Similarly, decisions about which fires to extinguish first may depend on the presence of nearby injured civilians and ambulances.

Research Problems

- Distributed vs. centralized control

Because communication is limited there will be a trade-off between centralized and distributed control. A centralized controller may have a more complete picture of the whole situation, but with unreliable communication it may not be able to send commands to remote agents.

Research Problems

- Communication

With the radio channel model of communication it is possible for agents to choose their own communication structure, possibly even changing it on the fly. Researchers can also implement their own communication models if desired.

Thank You