Multi-agent Surveillance

GROUP 8:
YANNICK DAMOISEAUX, ROMAN ILIC, JOHANN LOTTERMOSER,
JOAQUIN MONEDERO SIMON, MARIE PICQUET, GIACOMO TERRAGNI

Maastricht University

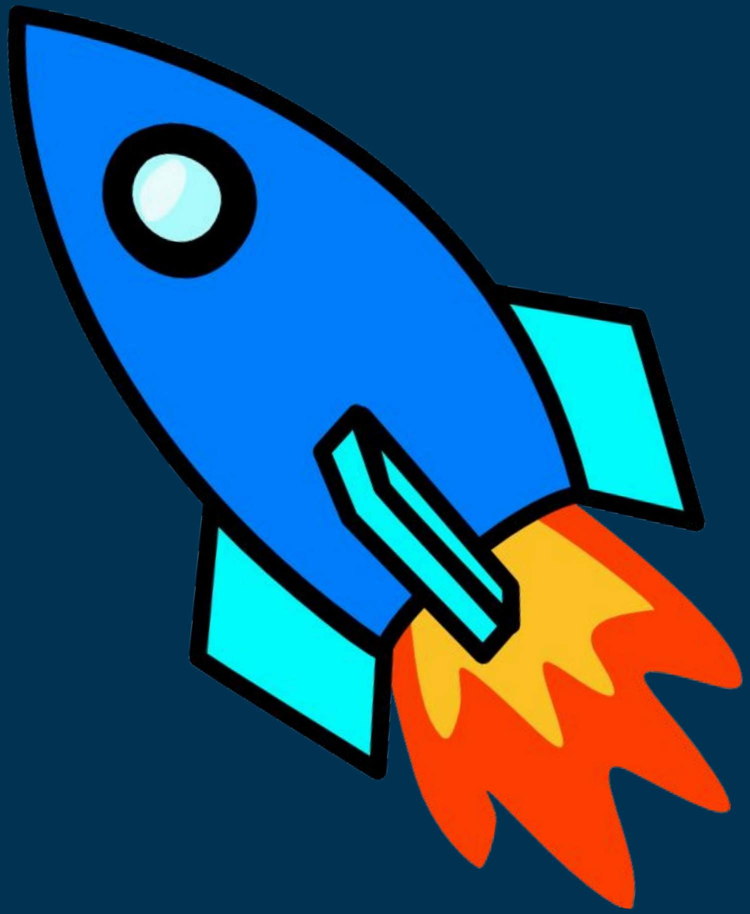
Table of content

Research Questions

- What are the effects of using pheromones on the exploration and therefore coverage of our map?
- How does indirect communication affect the results of our surveillance game?
  - Specifically how does the adding of sound affect the results of our surveillance game?
- How does the number of agents affect the results of our simulation?
- How does a classification approach compare to an algorithmic approach when detecting sounds?
- How does a Deep Reinforcement Learning evasion approach compare to an algorithmic approach?

Initial Assumptions

- Discrete-time
- Discrete-space
- Agents have no prior knowledge of the map



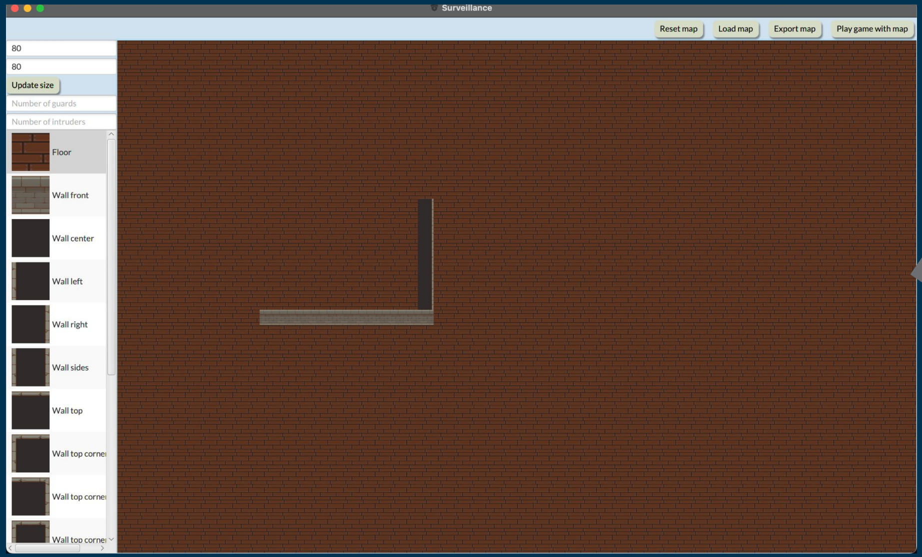
Features

GUI features

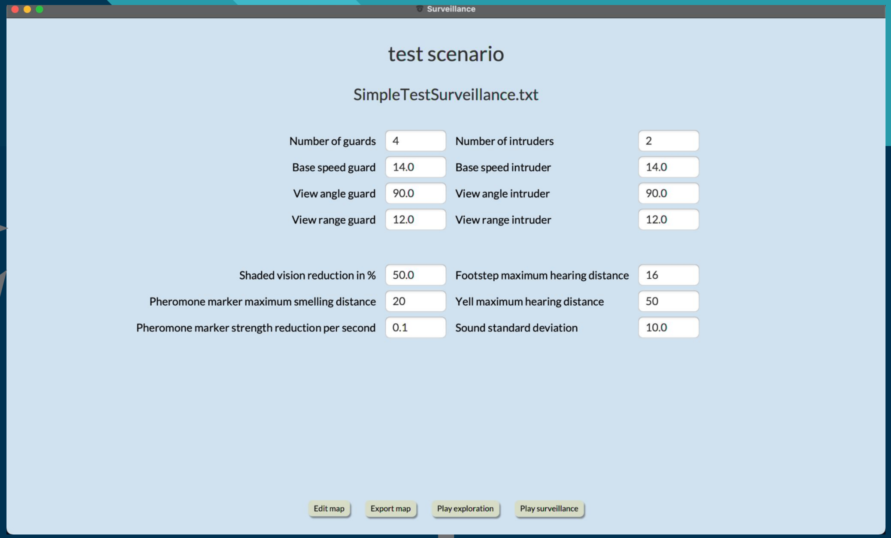
Upload map

or

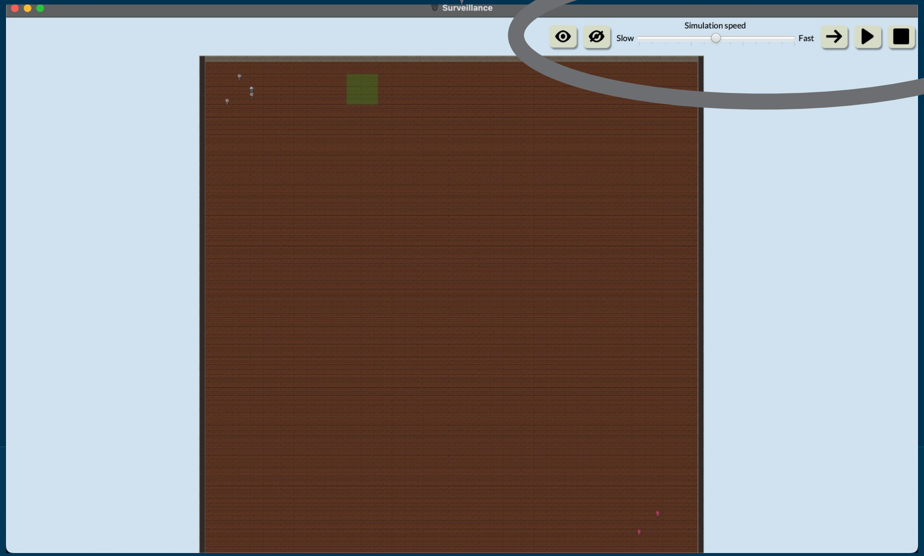
Create map



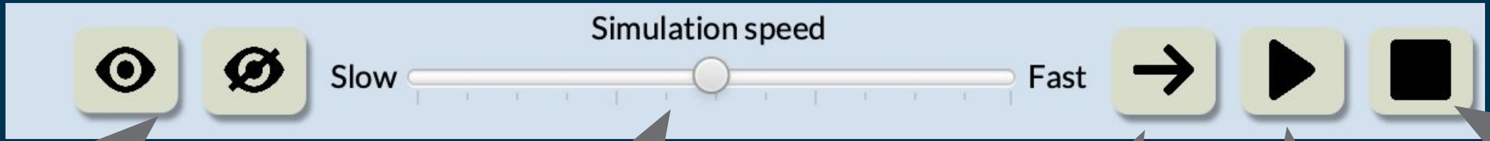
Modify map



Play map



GUI features



Show/Hide vision

Modify the speed of the simulation

Show one simulation step

Play simulation

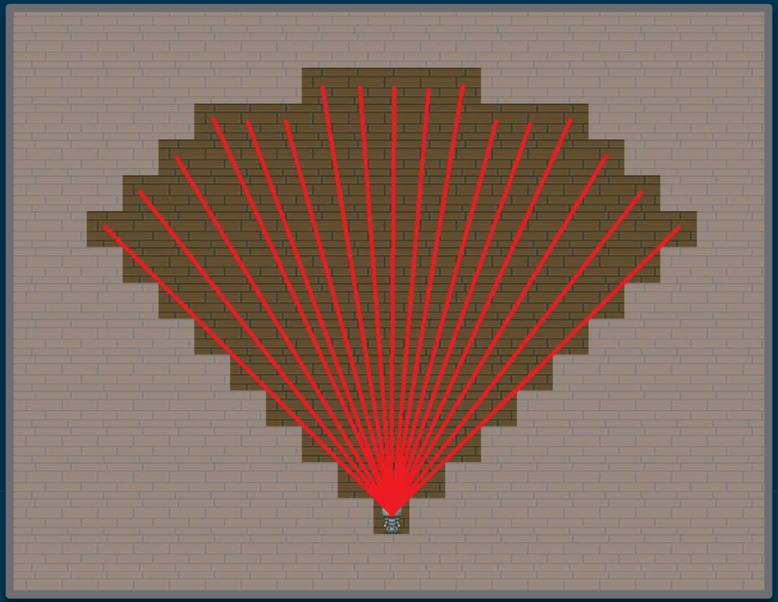
Stop simulation

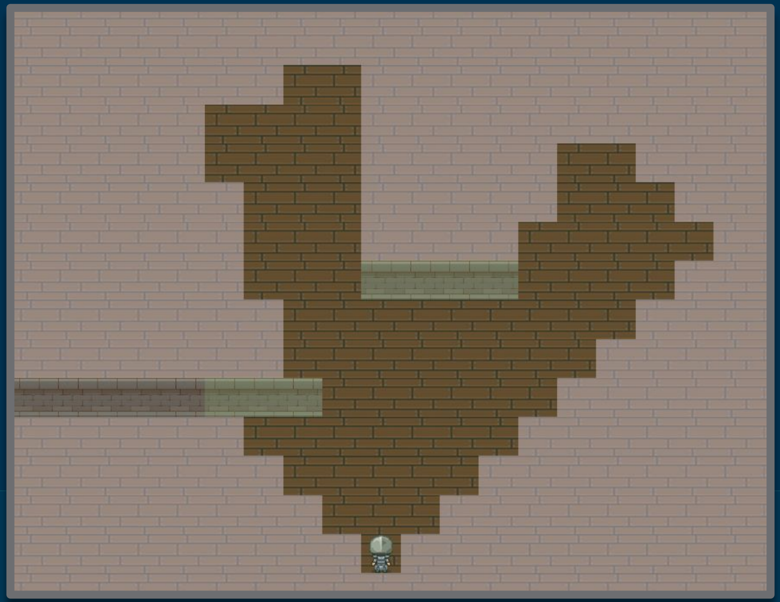
Game features

- A graph is used to represent the map
- Ending conditions → 3 seconds in target area, or all intruders get caught
- Teleports → from one direction, used by both
- Shaded areas → reduction of sight range
- State class
- Agent and controller are independent
- Relative coordinate systems

Vision

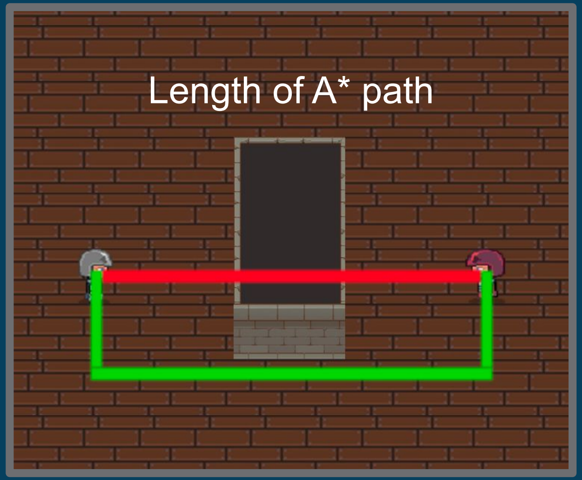
- For vision calculation we use ray-casting
  - If ray is interrupted by a vision blocking element → ray stops
- All agents can see in a cone-shaped area in front of them
- The vision is calculated by the controller based on the “scenarioMap”
  - Only the visible area is passed to the agent
  - The agent has no access to the real map
- If an agent sees another agent a “Vision memory” is created
  - All agents remember other agents last seen position
  - Knows how long ago that was





Sound

- We have different actions that generate sound
  - Walking
  - Rotating / turning
  - Guard yell (cooldown of 1 second)
- All sounds are automatically generated by the controller
- Each soundtype has its own maximum distance it can be heard from
- When agent can hear a sound, it is given an angle [0, 360) together with a loudness [0, 1]
- Normal distributed uncertainty is added to the angle, with $\sigma = 10$



Markers

- Guards have pheromone markers
- Used as indirect communication for every type of exploration and also to match sounds
- The strength of a pheromone marker reduces with time
- The distance pheromone markers can be perceived from depends on the current strength

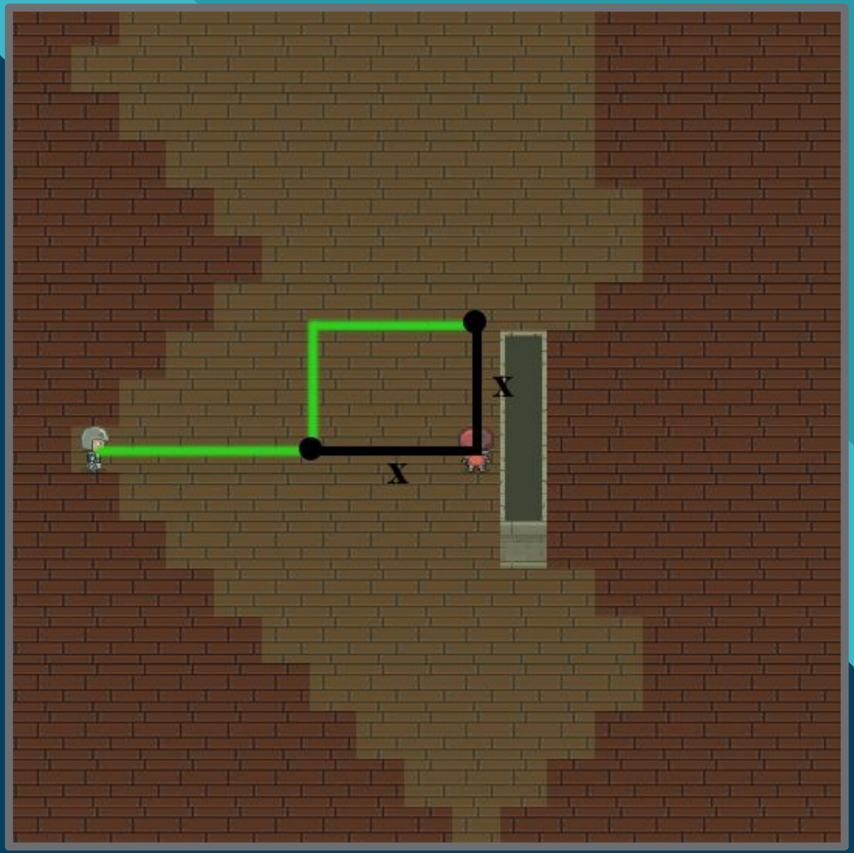
Tasks

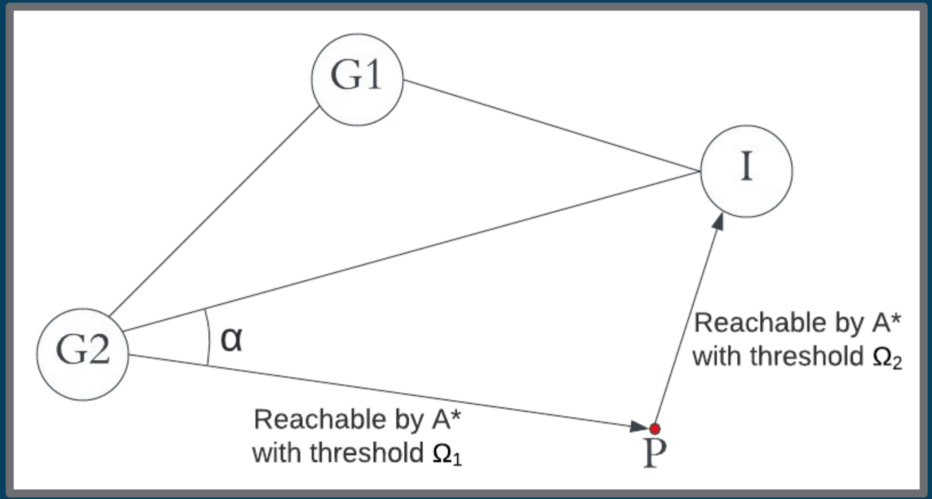
- Task decider which selects a task to perform
- Task to perform is selected based on sound, vision and markers
- Each agent type can perform different tasks
  - Exploration agent
  - Guard
  - intruder
- Each task has a priority

The Guards

Tasks of a Guard

- Close pursuit
  - When agent is closest to intruder
  - First try to get *x* blocks near the intruder
  - Second try to go *x* blocks in front of the intruder
- Far pursuit
  - When other agent is closer to intruder
  - Try to circumnavigate to cut the way of intruder
  - Approach intruder from the opposite direction compared to the approaching angle of the other guard





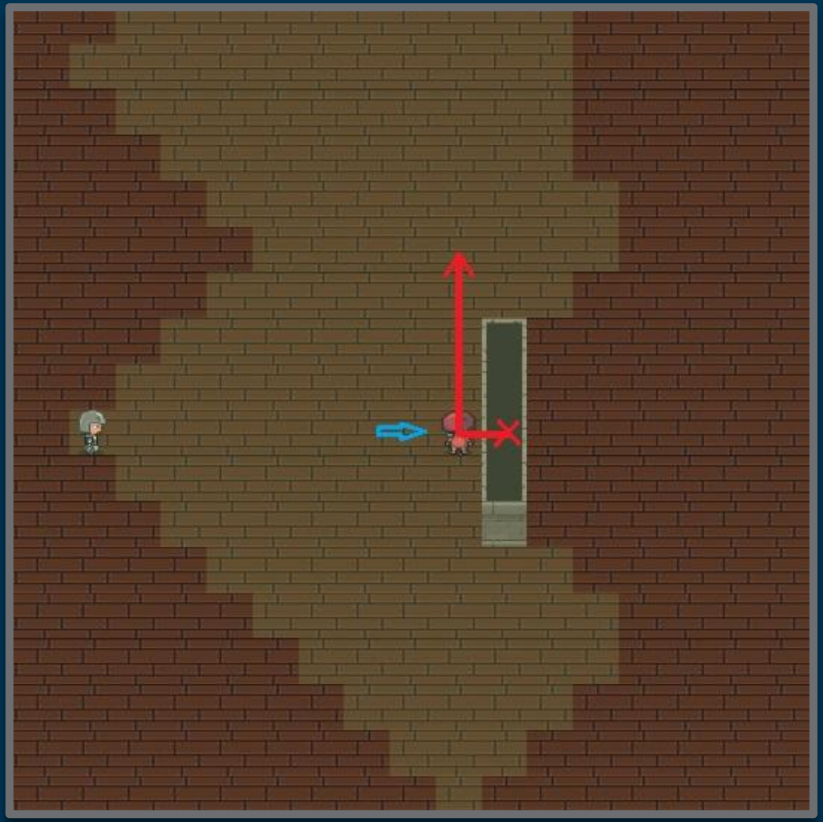
- Finding the source of a guard yell
  - Approximate upper and lower bound of the distance using the loudness
  - Add normal distributed uncertainty to these bounds, with σ changing based on the loudness
  - Find a position in between these bounds in the direction of the angle that can be reached with A*
  - If no position can be reached with A*, perform exploration in direction
- Frontier-based exploration
  - Exploit the graph structure the agent uses to store his map
  - Perform A* to frontier that is:
    - As much in the opposite direction of where the pheromone markers are coming from as possible
    - Closest to the guard, in terms of Manhattan distance
- Visiting the intruders at their last seen position
  - Visit the position of each intruder where the guard has last seen him

- Finding the source of a sound
  - Identical approach as finding a guard yell
  - Two different methods:
    - Algorithmic approach: sounds are ignored when they come from a direction where:
      - Guard pheromone markers are also coming from
      - The Guard saw another guard less than *x* seconds ago
    - Classification using a neural network:
      - Input of the agent his local state (sound details and vision memory)
      - Output is true if sound is matched to vision memory, false if not
      - Accuracy on validation set of 84%

The Intruders

Tasks of an Intruder

- Evasion algorithmic
  - Check if able to go a certain distance in the opposite direction
  - If not try to find an angle where that distance can be reached
  - Always react to vision and sound above a certain threshold
- Evasion Deep Reinforcement Learning
  - Input of the intruder his local state:
    - vision memory
    - the x closest walls
    - the pheromone markers
    - the sounds
    - the orientation of the intruder
  - All intruders use one Neural Network
  - Reward for staying alive during evasion (+1)
  - Reward for finishing evasion (+15)
  - Punishment for getting caught (-25)



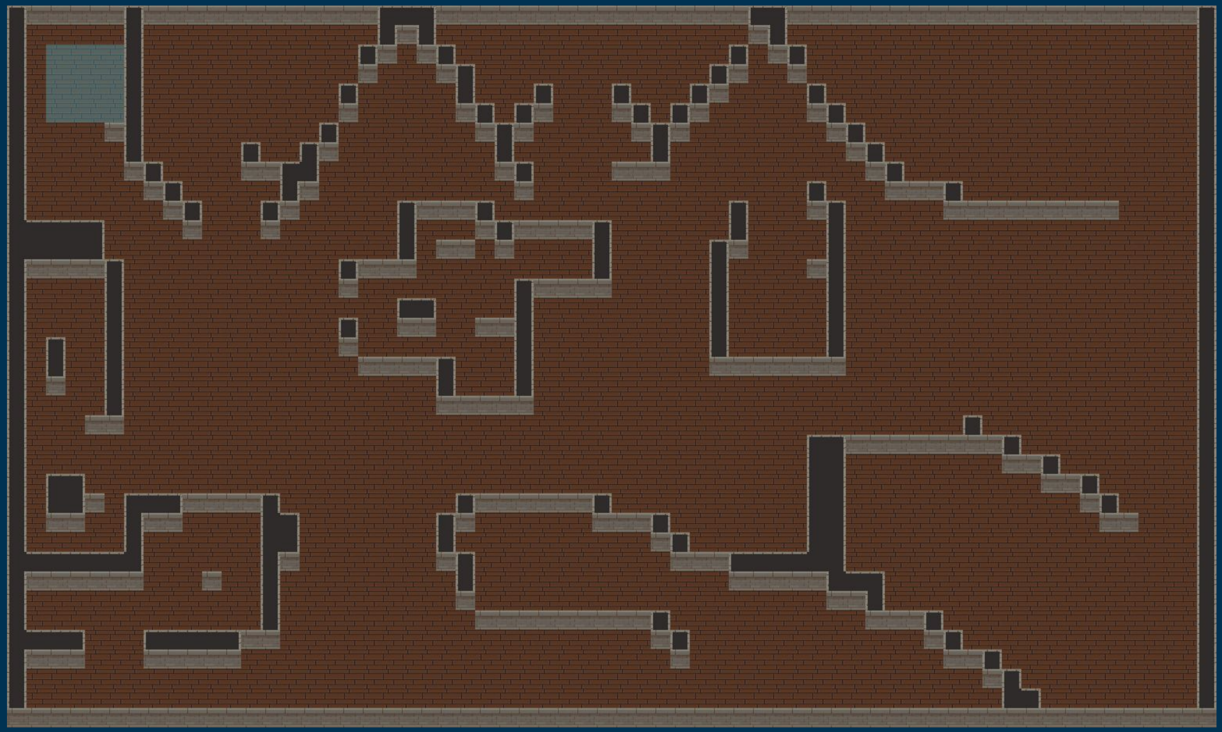
- Going to the target area (if known)
  - Once the target area was seen go to it
- Exploration in direction of the target area
  - Frontier based exploration
  - Set an anticipated goal upon spawning with a certain distance and the angle provided by the controller
  - Frontiers are preferred if they are closer to the anticipated target area
  - Once the anticipated target area is explored, increase the distance and set new goal
- Capturing Target Area

Live Demonstration

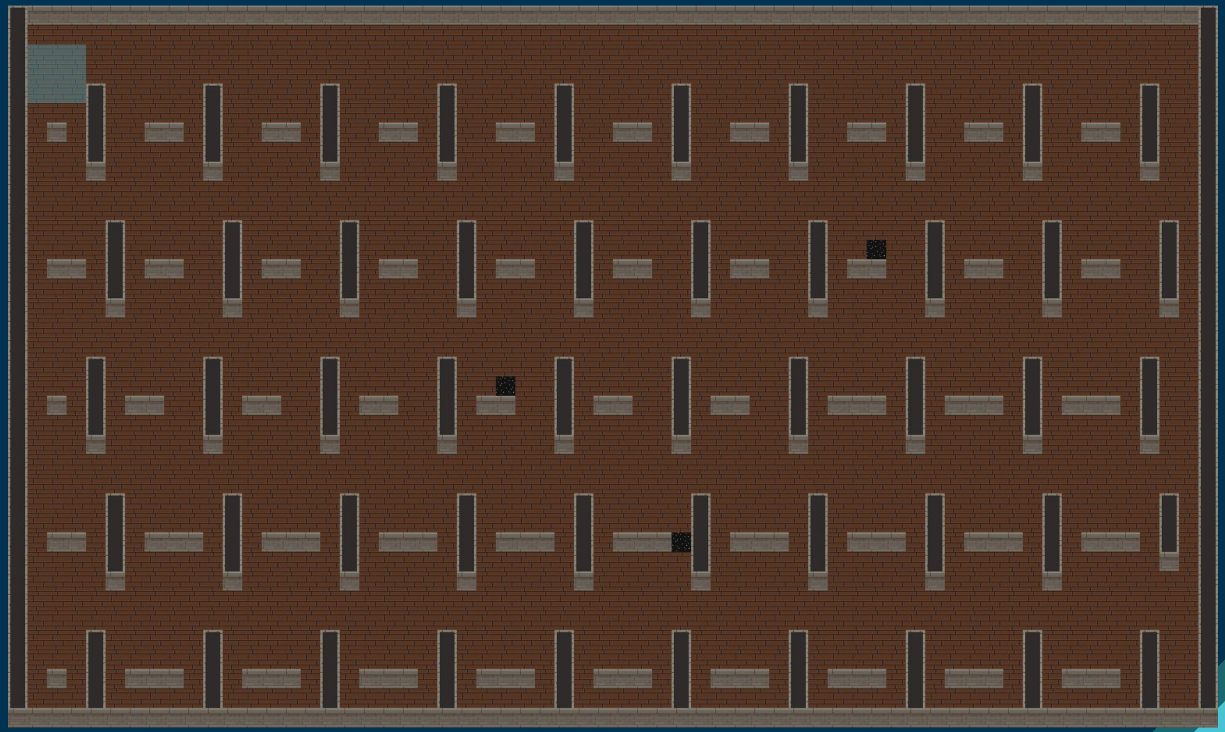
Experiments

Exploration Experiments

Map 1:

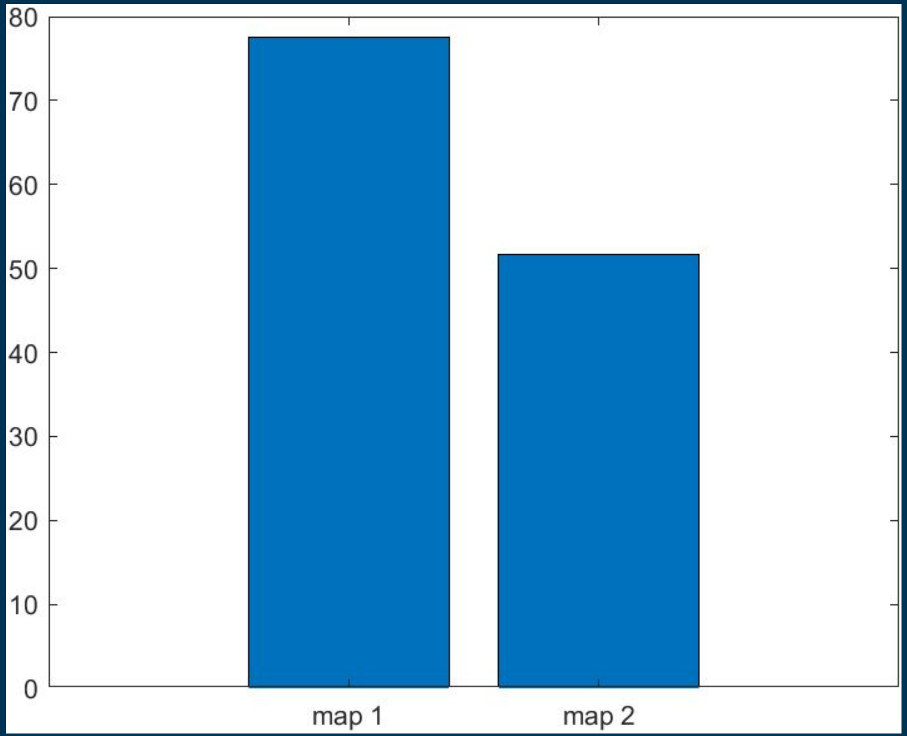


Map 2:



Exploration Experiments

Average time to explore the maps:



Exploration Experiments

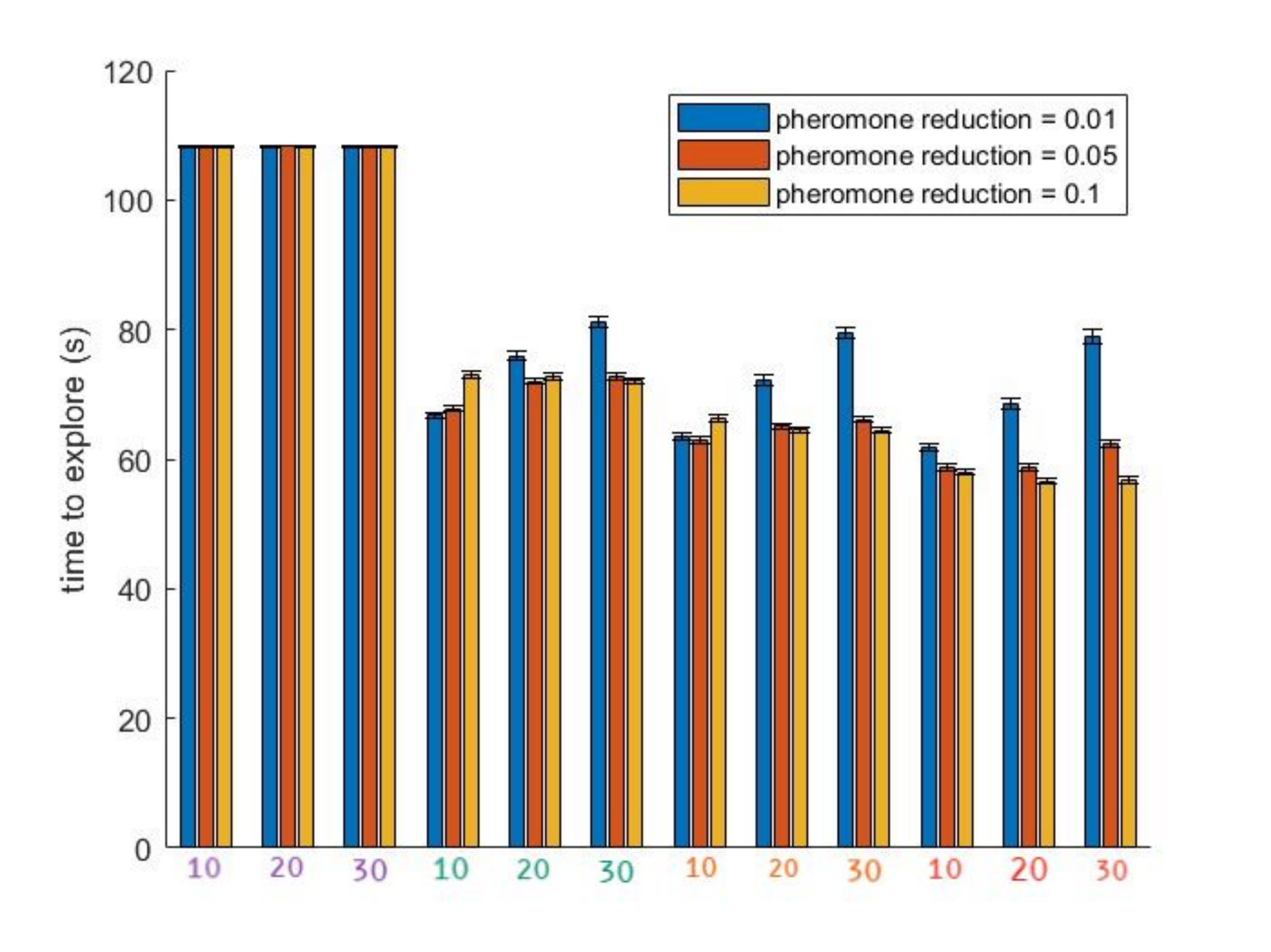
Research Question:

What are the effects of using pheromones on the exploration and therefore coverage of our map?

- Performed experiments on different kinds of parameters for pheromones
  - Pheromone maximum distance (10 / 20 / 30)
  - Pheromone reduction (0.01 / 0.05 / 0.1)
- 1/3/4/6 Agents are used
- 4000 iterations (games) for each experiment

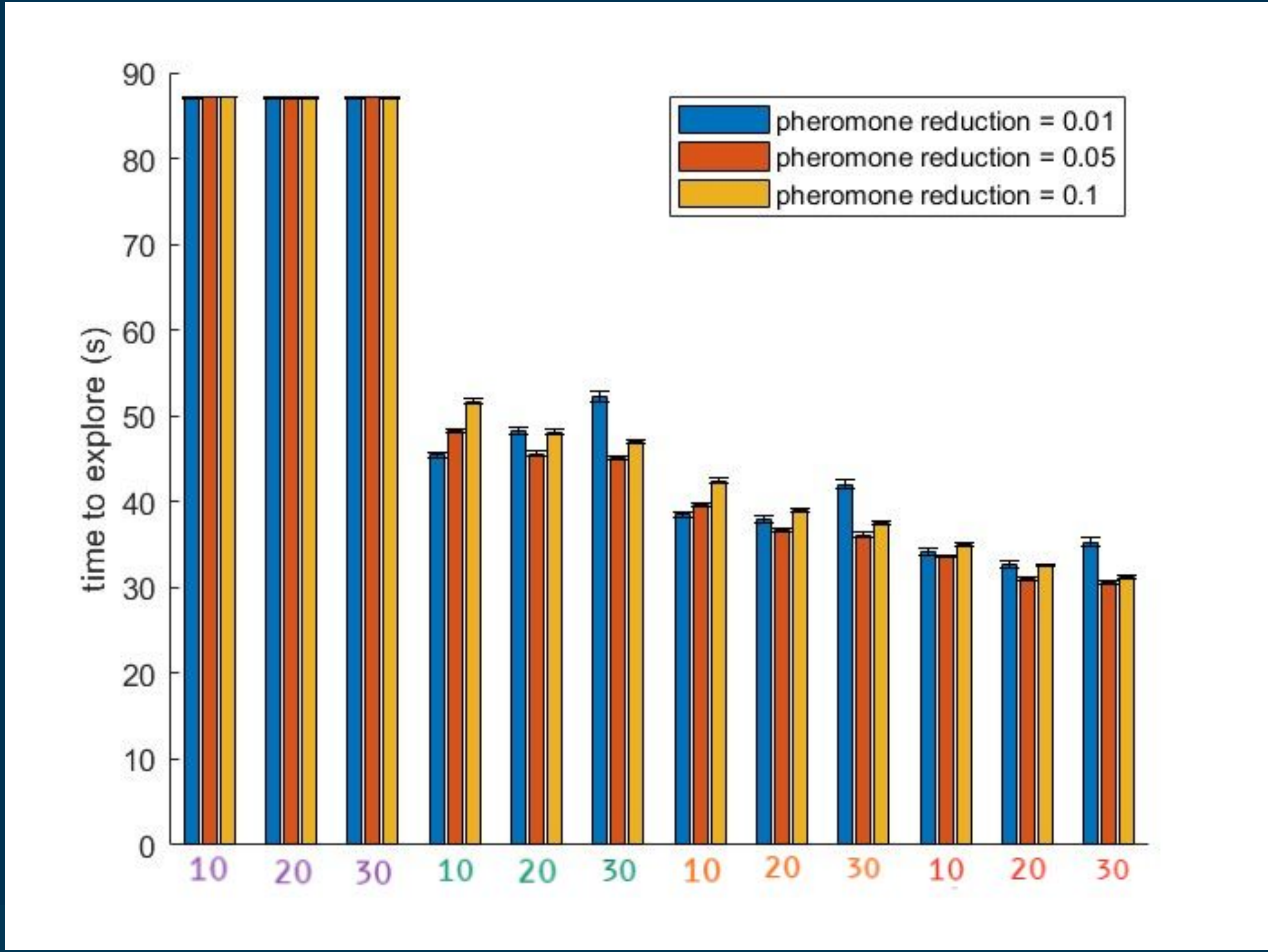
Exploration Experiments

Map 1:

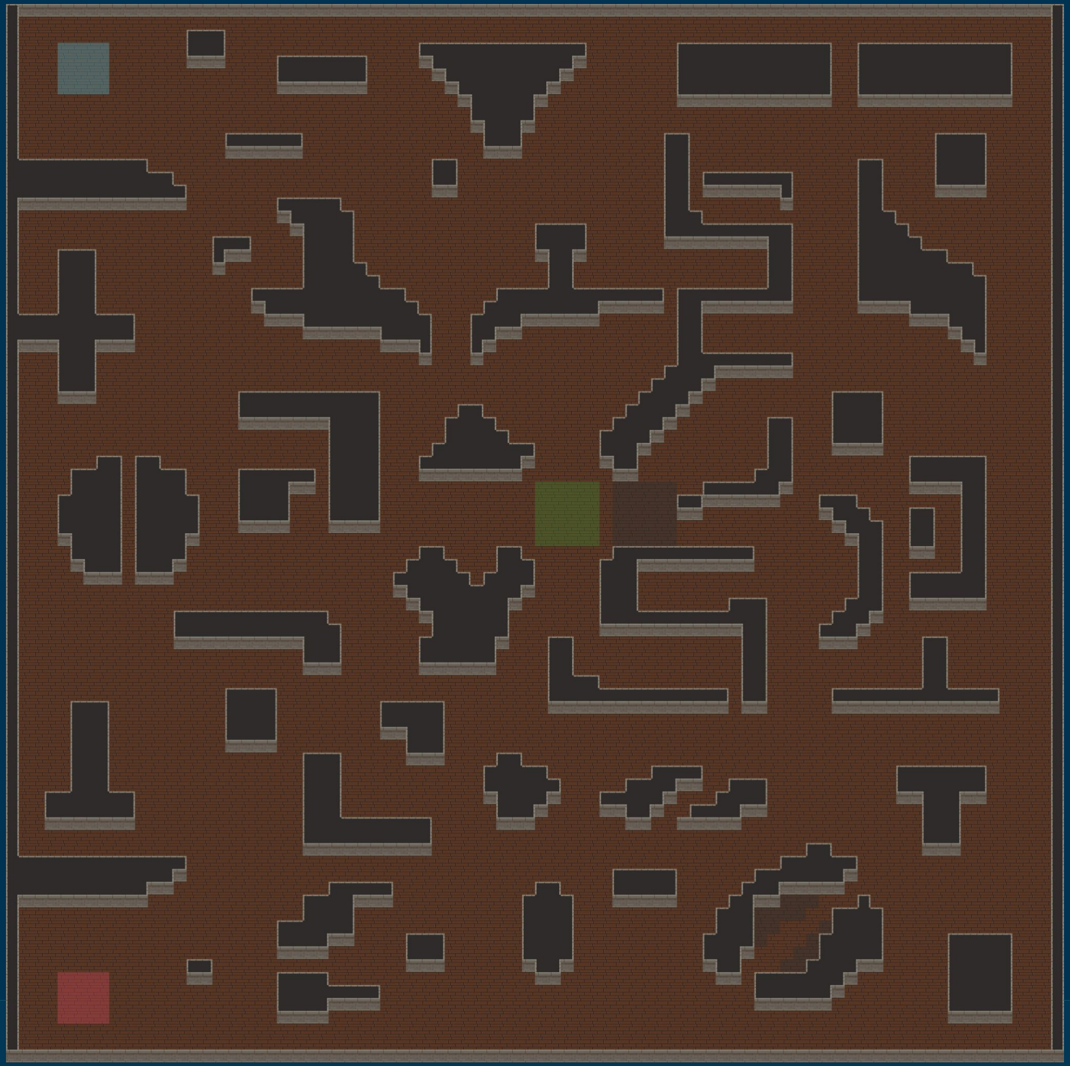


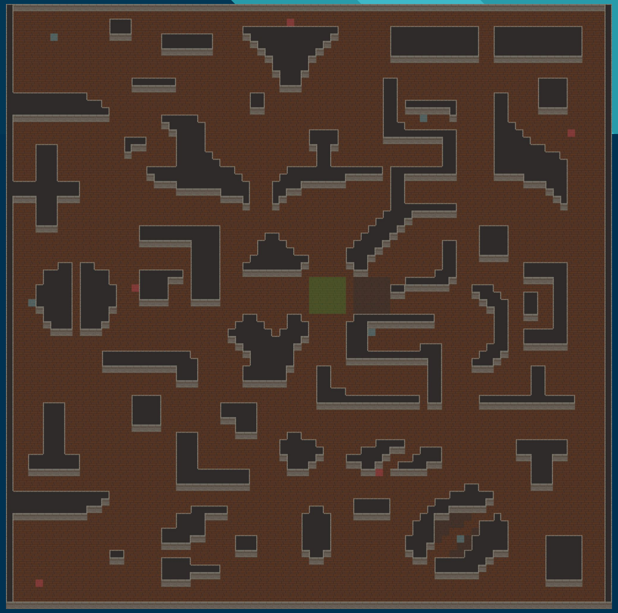
Exploration Experiments

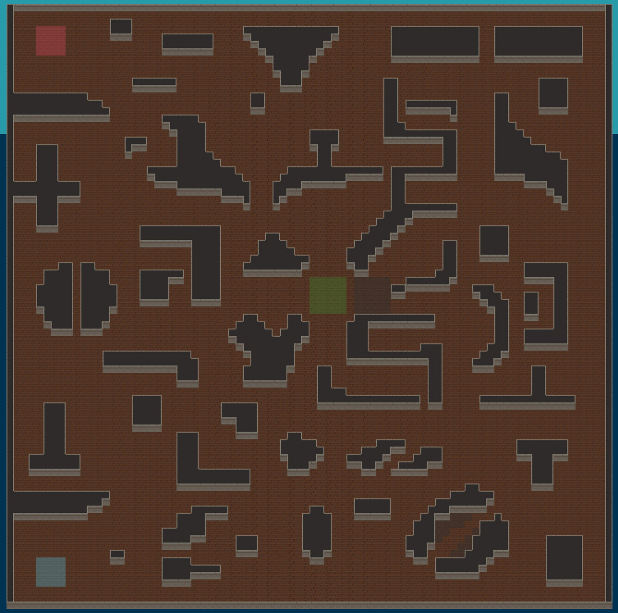
Map 2:

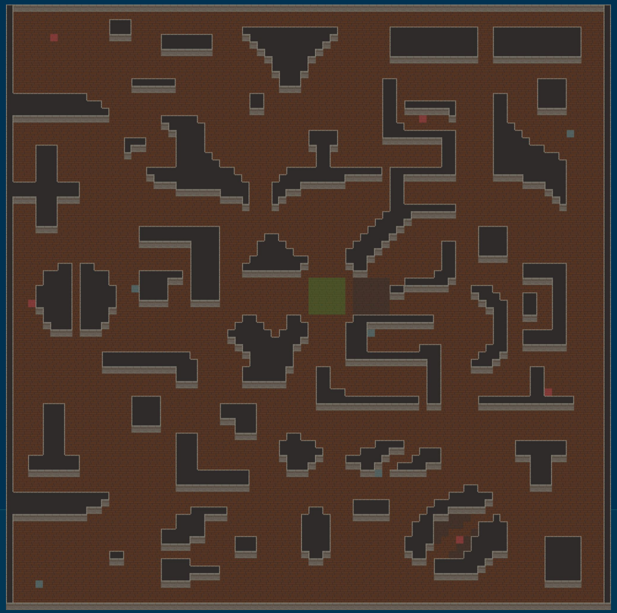


Surveillance Experiments

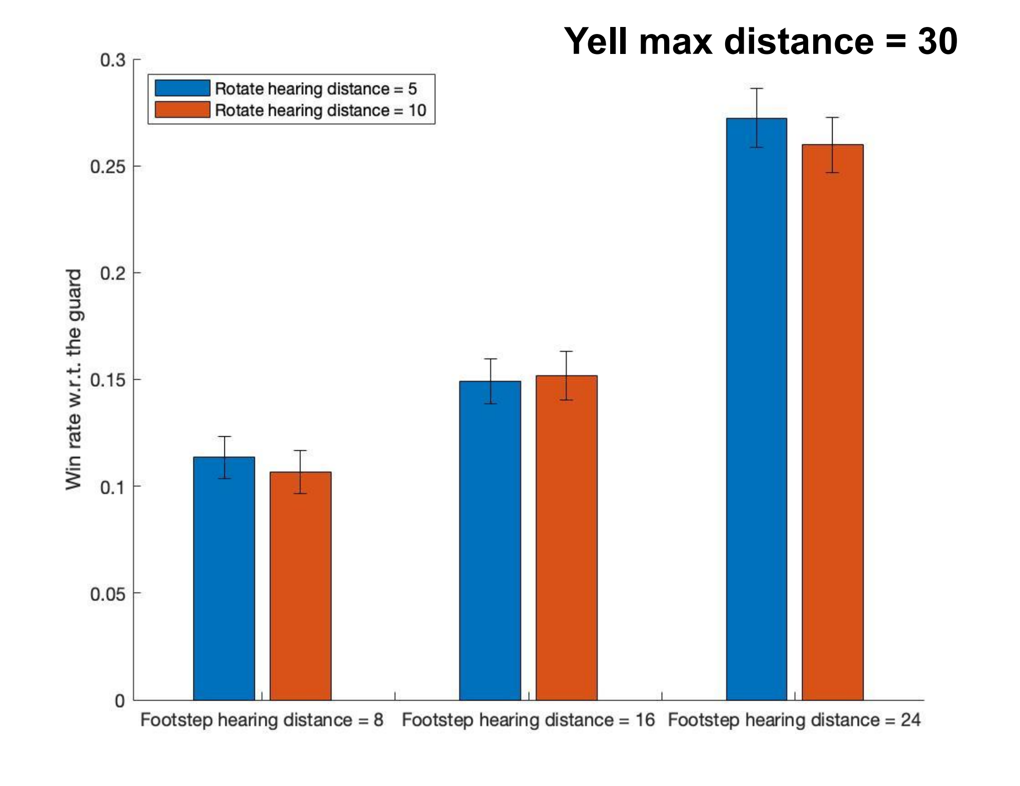








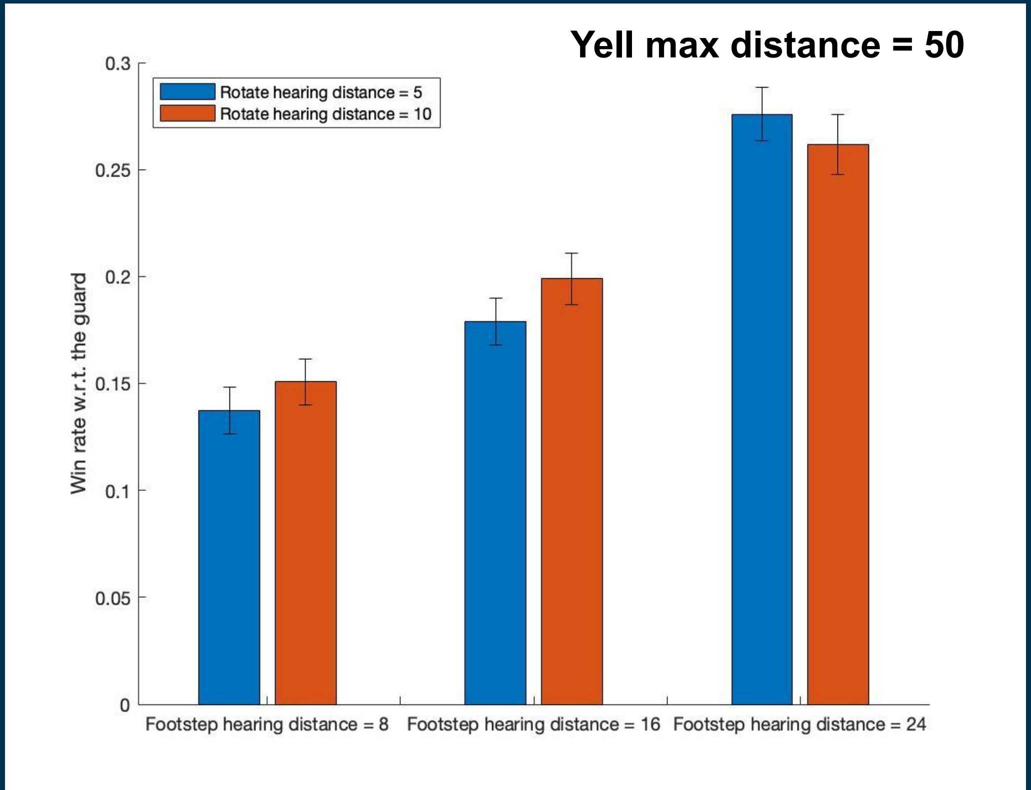
Surveillance Experiments

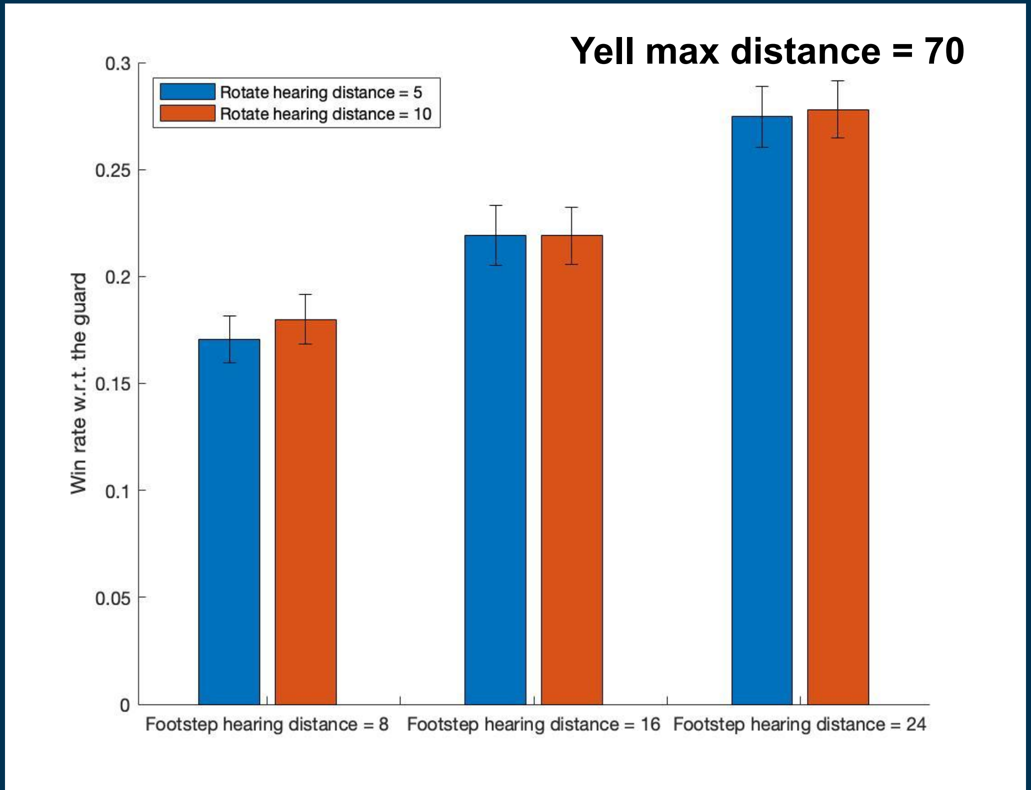


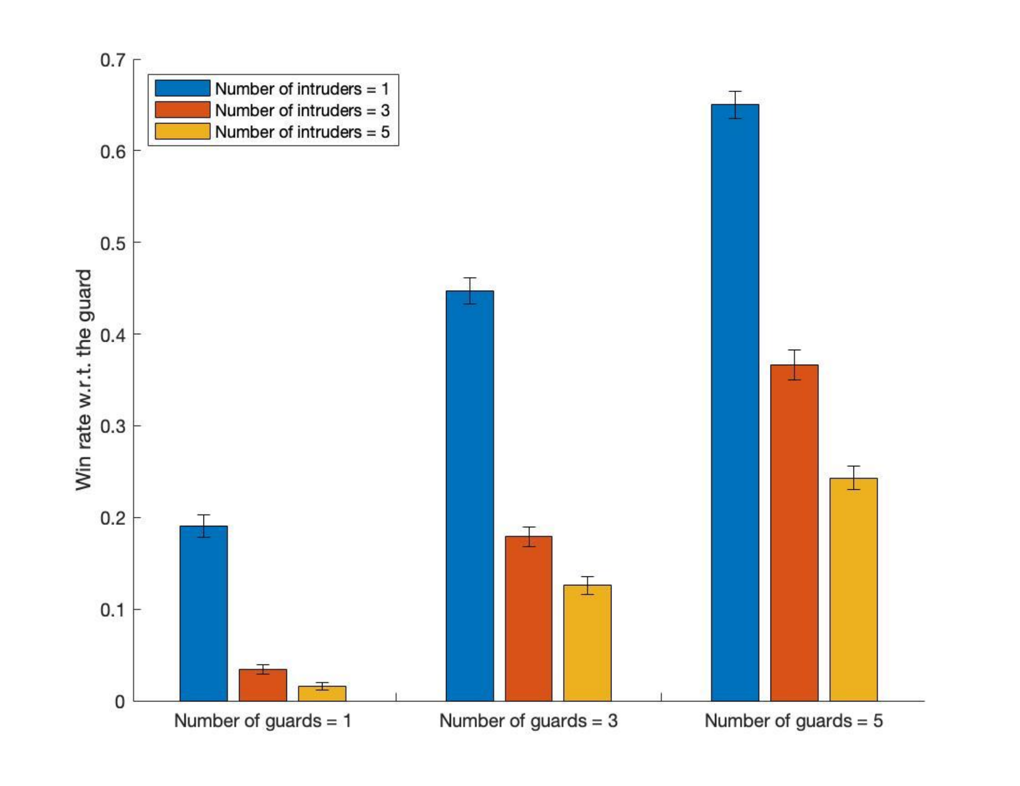
Research Question:

- How does indirect communication affect the results of our surveillance game?
- Specifically how does the adding of sound affect the results of our surveillance game?

- Performed experiments on different kinds of parameters for sound
  - Footstep maximum hearing distance (8 / 16 / 24)
  - Rotation maximum hearing distance (5 / 10)
  - Guard yelling maximum hearing distance (30 / 50 / 70)
- All combinations with these values
- 3 Guards, 3 Intruders
- Averaged over 4 maps, 1000 games per map, total of 4000


Yell max distance = 50
Rotate hearing distance = 5
Rotate hearing distance = 10
Win rate w.r.t. the guard
0
0.05
0.1
0.15
0.2
0.25
0.3
Footstep hearing distance = 8
Footstep hearing distance = 16
Footstep hearing distance = 24

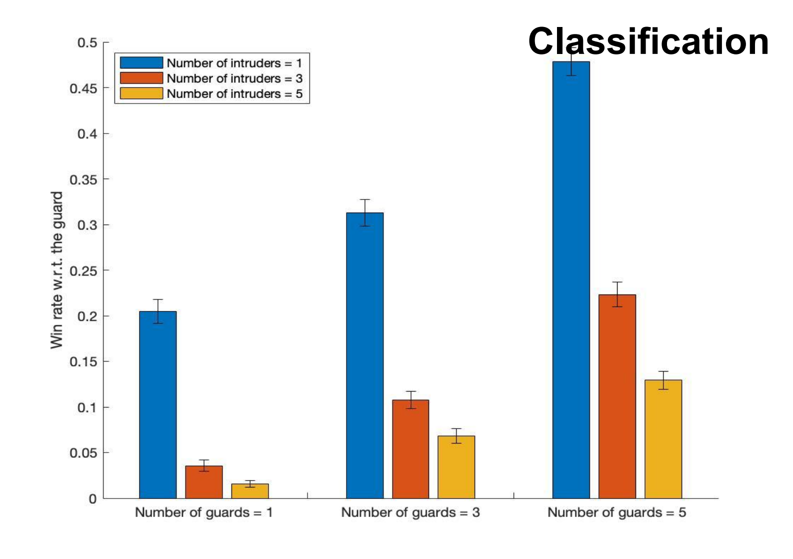

Yell max distance = 70
Rotate hearing distance = 5
Rotate hearing distance = 10
Win rate w.r.t. the guard
0
0.05
0.1
0.15
0.2
0.25
0.3
Footstep hearing distance = 8
Footstep hearing distance = 16
Footstep hearing distance = 24

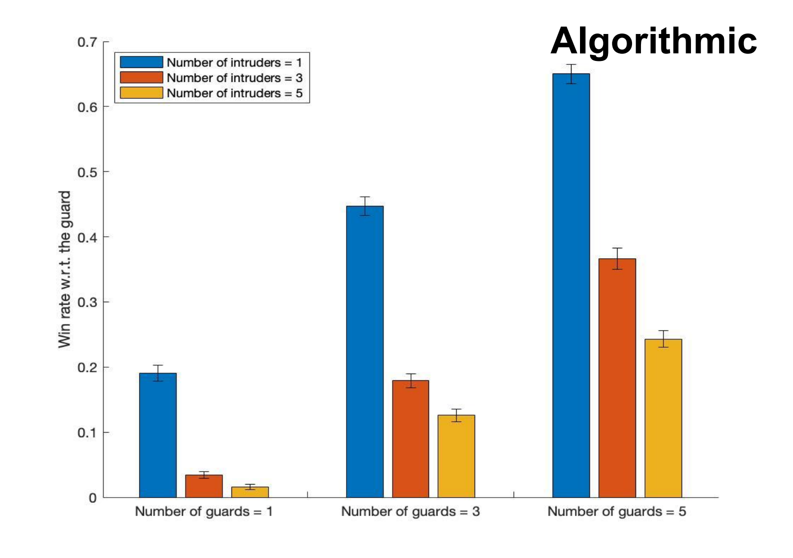


Research Question:

How does the number of agents affect the results of our simulation?

- Changing the number of guards and intruders
- Footstep maximum hearing distance = 16
- Rotation maximum hearing distance = 5
- Guard yelling maximum hearing distance = 50
- Averaged over 4 maps, 1000 games per map, total of 4000

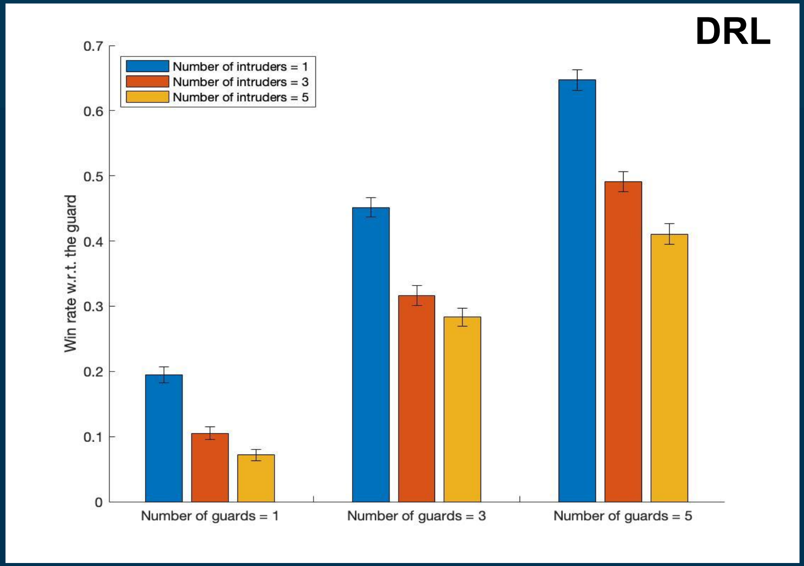


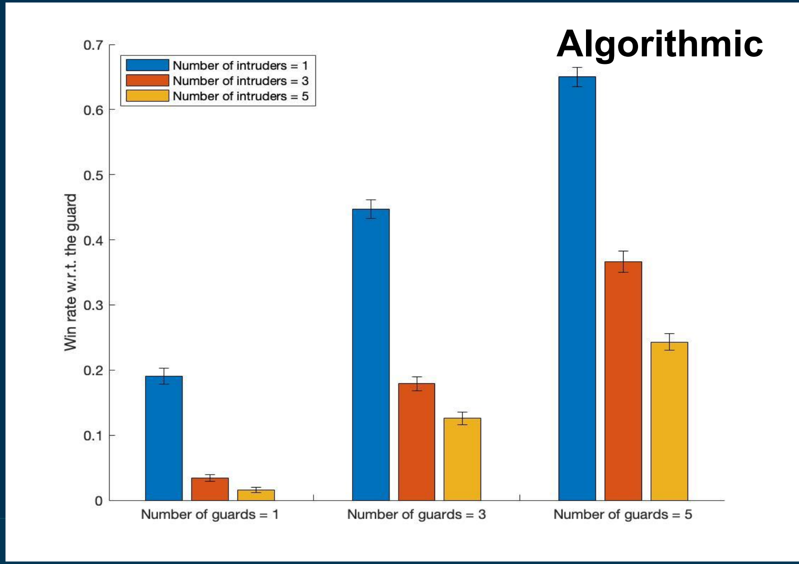


Research Question:

- How does a classification approach compare to an algorithmic approach when detecting sounds?

- Changing the number of guards and intruders
- Footstep maximum hearing distance = 16
- Rotation maximum hearing distance = 5
- Guard yelling maximum hearing distance = 50
- Averaged over 4 maps, 1000 games per map, total of 4000





Research Question:

- How does a Deep Reinforcement Learning evasion approach compare to an algorithmic approach?

- Changing the number of guards and intruders
- Footstep maximum hearing distance = 16
- Rotation maximum hearing distance = 5
- Guard yelling maximum hearing distance = 50
- Averaged over 4 maps, 1000 games per map, total of 4000

Conclusion

Conclusion

- Increasing pheromone marker properties decreases exploration time up to a certain point
- In the surveillance game mode, increasing sound distances helps the intruder to win more
  - Intruder can react sooner to threats (before vision)
- Increasing the number of guards increases winning rate for the guards more than it does for the intruders

Overview of who did what

Name	Coding Tasks Phase 1	Coding Tasks Phase 2	Coding Tasks Phase 3
Christian	Home screen (25%), GUI styling (50%), Map creator (20%), Game screen (20%)	//	//
Giacomo	Frontier agent (30%), Random agent (33.3%), Graph (33.3%), Experiments (25%)	Experiment Code (80%),	Guard yell when intruder caught (50%)
Marie	Frontier agent (30%), Random agent (33.3%), Graph (33.3%), Controller (16.6%), Experiments(25%)	EndingScreens(100%), Game screen surveillance (50%), Zooming (100%)	Guard yell when intruder caught (50%)
Joaquin	Frontier agent (30%), Random agent (33.3%), Graph (33.3%), Controller (16.6%), Experiments (25%)	Ending conditions (100%), Experiment Code (20%)	Capture Target Area Task (100%), Experiments code (40%)
Johann	Vision (100%), Controller (16.6%), Frontier agent (10%), ScenarioMap (50%), Pheromone markers (30%), HashMap(50%)	Task Decider Intruder (100%), Tasks (50%), A-Star optimizations (100%), Shaded (100%), Markers in exploration (50%), Sounds (25%)	Guard Far Pursuit (60%), Reinforcement Learning Evasion (50%)
Roman	Controller (16.6%), EndingExplorationMap (100%), ScenarioMap (50%), Experiments (25%)	Generic Hashmap (100%), Agents Seen (80%), Pursuit Markers (100%)	Intruder Markers (100%)
Yannick	GUI styling (50%), Homescreen (75%), Pause menu (100%), Scenario menu (100%), Map creator (80%), Game screen (80%), Controller (16.6%), Pheromone markers (70%), HashMap (50%)	Game screen surveillance (50%), Sounds (75%), Multithreading (100%), Markers in exploration (50%), Tasks (50%), Guard decider (100%), States (100%), Agents Seen (20%)	Guard far pursuit (40%), Sound deciding model (100%), Reinforcement Learning evasion (50%), Experiments code (60%)
High-level Task			
Coding	Johann (14.29%), Yannick (14.29%), Roman (14.29%), Giacomo (14.29%), Christian (14.29%), Marie (14.29%), Joaquin (14.29%)	Johann (25%), Yannick(25%), Roman (12.5%), Giacomo (12.5%), Marie (12.5%), Joaquin (12.5%)	Johann (25%), Yannick (25%), Roman (12.5%), Giacomo (12.5%), Marie (12.5%), Joaquin (12.5%)
Presentation	Johann (14.29%), Yannick (14.29%), Roman (14.29%), Giacomo (14.29%), Cristian (14.29%), Marie (14.29%), Joaquin (14.29%)	Johann (15%), Yannick(15%), Roman (15%), Giacomo (15%), Marie (25%), Joaquin (15%)	Johann (16.66%), Yannick (16.66%), Roman (16.66%), Giacomo (16.66%), Marie (16.66%), Joaquin (16.66%)
Report	//	Johann (16.66%), Yannick(16.66%), Roman (16.66%), Giacomo (16.66%), Marie (16.66%), Joaquin (16.66%)	Johann (16.66%), Yannick (16.66%), Roman (16.66%), Giacomo (16.66%), Marie (16.66%), Joaquin (16.66%)
Project management	Johann (14.29%), Yannick (14.29%), Roman (14.29%), Giacomo (14.29%), Cristian (14.29%), Marie (14.29%), Joaquin (14.29%)	Johann (16.66%), Yannick (16.66%), Roman (16.66%), Giacomo (16.66%), Marie (16.66%), Joaquin (16.66%)	Johann (16.66%), Yannick (16.66%), Roman (16.66%), Giacomo (16.66%), Marie (16.66%), Joaquin (16.66%)

Literature

- Hoen, Pieter & Tuyls, Karl & Panait, Liviu & Luke, Sean & Poutré, Johannes. (2005). An Overview of Cooperative and Competitive Multiagent Learning.. 1-46
- Zhang, K., Yang, Z., & Başar, T. (2019). Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *ArXiv, abs/1911.10635*.
- C. de Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov and D. Kulić, "Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4552-4559, July 2021, doi: 10.1109/LRA.2021.3068952.
- Nguyen, Austin. (2021). Scalable, Decentralized Multi-Agent Reinforcement Learning Methods Inspired by Stigmergy and Ant Colonies.
- Rossi, F., Bandyopadhyay, S., Wolf, M., & Pavone, M. (2018). Review of multi-agent algorithms for collective behavior: a structural taxonomy. *IFAC-PapersOnLine*, *51*(12), 112-117.
- Quinn, M. (2001). Evolving communication without dedicated communication channels. In European conference on artificial life (pp. 357–366)
- Pramanik, M. I., Lau, R. Y., Yue, W. T., Ye, Y., & Li, C. (2017). Big data analytics for security and criminal investigations.Wiley interdisciplinary reviews: data mining and knowledge discovery, 7(4), e1208.

Thank you