

# Cops and Thieves

Project for the Agent Systems course

Tomasz Kawiak, Mateusz Mazur

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, AGH

Field of study: Computer Science and Intelligent Systems

Specialization: Artificial Intelligence and Data Analysis

March 18, 2025

## 1 Introduction

# Introduction

**Cops And Thieves** (*cops and robbers*) is a strategic pursuit-and-evasion game where two opposing agent types operate in a shared environment. Thieves aim to evade capture, while cops patrol, chase, and arrest thieves to maintain order. The game mechanics involve agent coordination, pathfinding, and adaptive decision-making.

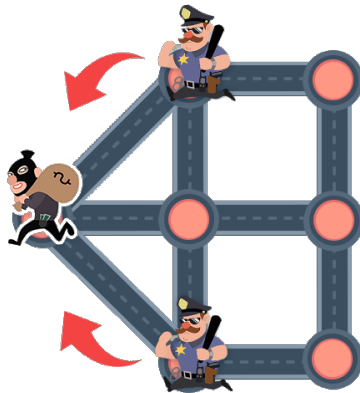


Figure 1: *Cops and Thieves* game depiction.  
Source: *Catch The Thief: Help Police* by MicroEra

Comparison of other approach for a problem considered previously on engineering studies course *Development Workshop*. Our project – *Chase model* – was also implementation of the cops and thieves game. This project aims to hopefully improve our earlier attempt.

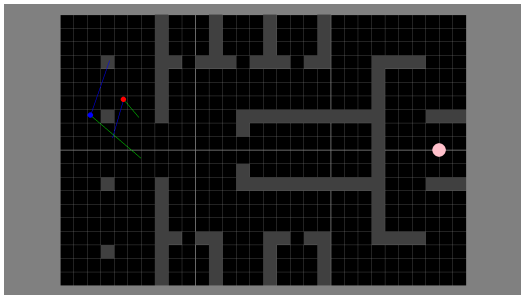


Figure 2: *Chase model* – game area.

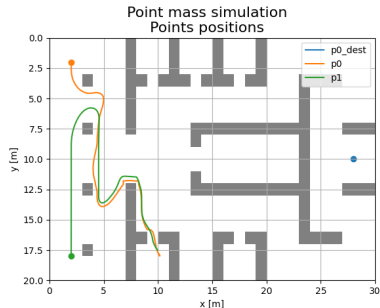


Figure 3: *Chase model* – movement chart.



**WE ARE**  
**BACK**

Figure 4: We are back

As depicted in fig. 4, we are back to face the challenge of the cops and thieves problem, but this time with a different approach.

The main goal of this project is to train *environment-agnostic* agents:

- *Cops*: search and chase the thief.
- *Thief*: hide and try not to get caught.

A key aspect for both agents will be their ability to analyze their surroundings and act based on past observations. For our project we'll initially focus on only 2 agents (1 cop and 1 thief), with possibility of increasing the number of cop agents to observe cooperative behavior and more sophisticated search patterns.

We expect to achieve the following behaviors from agents:

Agent Type	Expected Description
<i>Cop</i>	Search and chase (if more cops, cooperation)
<i>Thief</i>	Evade capture and hide efficiently

In our project we intend to use the following frameworks:

- skrl (fig. 5) for MARL implementation.
- PettingZoo (fig. 6) to guarantee MARL environment standards.
- pymunk (fig. 7) as 2D physics engine, complemented by:
  - ▶ pygame (fig. 8) for visualization.



Figure 5: skrl logo



Figure 6: PettingZoo logo



Figure 7: pymunk logo



Figure 8: pygame logo



Thank you for your attention

Questions?