Organized Crime Fighting Simulation

Ghazi Haj Qassem

May 29, 2025

1 Introduction

This document describes the organized crime fighting simulation project.

2 Objectives

- Simulate real-time crime fighting scenarios.
- Analyze strategies for combating organized crime.
- Provide insights for law enforcement agencies.

3 System Overview

3.1 Processes

- **main:** forks the police, gang and graphics.
- **Police:** one process throughout the simulation.
- **Gang:** multiple processes, each representing a separate gang. No gang manager.
- **Graphics:** one process for the graphical interface.

3.2 Threads

- **Police:** one thread for each police squad.
- **Gang:** one thread for each gang member.
- **Graphics:** one thread for the graphical interface.

3.3 Inter-Process Communication

- **Message Queues:** used for communication between processes.
- **Shared Memory:** used for shared data structures.
- **Semaphores:** used for synchronization between processes.

3.4 Data Structures

- **Police:** contains information about the police force.
- **Gang:** contains information about each gang.
- **Graphics:** handles the graphical representation of the simulation.

4 Implementation

4.1 General

Description	Implementation
Crime Types	ENUM

Table 1: Implementation of the system

4.2 Gang Members

Description	Implementation

Table 2: Implementation of a gang member

5 Game Flow & Strategy

5.1 Gang Member Attributes

Each gang member in the simulation is characterized by a set of diverse attributes that represent their individual skills and tendencies. These attributes play a crucial role in determining the success probability of various criminal operations (targets) attempted by the gang.

The following attributes are defined for each gang member:

- **Smartness**: Ability to plan, strategize, and adapt to changing situations.
- **Stealth**: Skill in avoiding detection by law enforcement or rival gangs.
- **Strength**: Physical power, useful in confrontations or physically demanding tasks.
- **Tech Skills**: Proficiency with technology, including hacking, disabling alarms, or using advanced equipment.
- **Bravery**: Willingness to take risks and face dangerous situations.
- **Negotiation**: Talent for persuasion, bargaining, and manipulating others.
- **Networking**: Ability to build and leverage connections within the criminal underworld.

5.2 Target-Attribute Dependency

Each criminal target (operation) depends on these attributes to varying degrees. The dependency is modeled as a set of weights, one for each attribute, which sum to 1 for each target. These weights represent the relative importance of each attribute for the success of that specific operation.

An example for the dependency weights for all targets are shown in the table below. In practice, these dependencies are stored in config.json.

Target	Smartness	Stealth	Strength	Tech	Bravery	Negotiation	Networking
Bank Robbery	0.10	0.10	0.10	0.40	0.20	0.05	0.05
Jewelry Shop Robbery	0.10	0.45	0.10	0.20	0.10	0.03	0.02
Drug Trafficking	0.10	0.05	0.05	0.05	0.10	0.35	0.30
Art Theft	0.10	0.50	0.05	0.30	0.02	0.02	0.01
Kidnapping	0.05	0.05	0.25	0.05	0.30	0.25	0.05
Blackmail	0.10	0.05	0.02	0.18	0.05	0.40	0.20
Arms Trafficking	0.10	0.05	0.20	0.05	0.10	0.20	0.30

Table 3: Attribute weights for different criminal operations

5.3 Success Rate of a Target

- The success rate of a target should depend on the attributes of each gang member, their rank, and their preparation level.
- This way, when a gang member **dies**, success rate reduces due to the formula below.
- 1.5*1.5 are just a form of success limit

$$success\_rate(T) = \frac{\sum_{i=1}^{|G|} W(T) \cdot A_i}{|G|} * \frac{1 + \frac{rank_i}{|R|}}{1 + \frac{difficulty}{|D|}} * \frac{1 + \frac{prep\_level_i}{|P|}}{1 + \frac{difficulty}{|D|}} * 100\% \quad (1)$$

where:

- T : Type of crime or operation.
- $|G|$: Number of alive gang members.
- $W(T)$: Weight or difficulty factor for crime type T.
- A_i : Attribute vector of gang member i.
- $rank_i$: Rank or gang member i.
- $prep\_level_i$: Preparation level of gang member i for crime T.
- $|R|$: Total number of ranks in the gang.

- $|P|$: Total number of preparation levels in the gang.
- $difficulty$: User-defined constant to measure the success rate (falls into the range [0, $|D|$]).
- $|D|$: Total number of difficulty levels

5.4 Knowledge and Information Spreading

5.4.1 Knowledge Calculation based on Rank

Higher rank gets exponentially more knowledge according to the formula:

$$knowledge = 0.2 + (\frac{rank}{|R|})^2 + random\_normal(0.0, 0.1) \tag{2}$$

5.4.2 Information Packet

The information packet is represented as follows:

```
    typedef struct {
        InfoType type;
        float accuracy;      // 0.0 to 1.0, how accurate the information is
        int source_rank;     // Rank of the member who shared this info
        int timestamp;       // When this info was shared
    } InformationPacket;
```

5.4.3 Member Knowledge and Misinformation Level Update

Information packets are weighted according to (i is the index of the info packet):

- How recent: $w_t = \frac{1.0}{1.0+0.2*i}$
- Source rank: $w_r = 1.0 + src\_rank * 0.1$

5.5 Secret Agents

THIS SECTION IS A MESS, SO TAKE A DEEP BREATH!

5.5.1 Attributes

Attributes that are related to secret agents:

- **knowledge**: knowledge level about the current plan that is affected by the following factors:
  - the highest gang ranking sometimes spreads false information among the lower gang ranks.
  - gangs spread information among its ranks on on-need basis, meaning as you go down with the ranking, the less you know about the planning of future evil activities.
- **suspicion**: a measure of how suspicious the gang member to the gang. It should depend on an attribute called **discretion** that is used by secret agents when asking other members of the gang.
- **faithfulness**: a measure that can act as a probability of turning against the gang and be used as an agent by the police. Set to be random for simplicity for now.

5.5.2 Actions

- A **secret agent** should ask lower rank gang members after a **random amount** of time about their gang plans, raising suspicion that is dependent on **discretion**. Higher rank of secret agents means more experience so it hides suspicion more.

$$suspicion+ = shrewdness * (1 - \frac{rank}{|R|} * discretion) \tag{3}$$

  Of course, not all info they get from the gang members are correct, since some are spread falsely, so:

$$knowledge(secret\_agent)+ = shrewdness * (knowledge(gang\_member) - 0.5) \tag{4}$$

  and if the gang member asked is an agent, the knowledge would be affected adversely:

$$knowledge(secret\_agent)- = shrewdness * (knowledge(gang\_member) - 0.5) \tag{5}$$

  the 0.5 value acts as barrier between true knowledge and false knowledge.

- After a plan is thwarted, an internal investigation is conducted by highest gang member h, so each gang member should store a **LIST OF WHO ASKED**, and the **TRUTHFULNESS** of the interrogated gang member g is dependent on their **SUSPICION** attribute as follows, for every i in the **LIST OF WHO ASKED**:

$$suspicion(secret\_agent_i)+ = shrewdness_h * (1 - suspicion_g) \tag{6}$$

  If the asked member is a secret agent a, then they would remain silent and say don't know, but suspicion rises according to:

$$suspicion_a+ = shrewdness_h * suspicion_a \tag{7}$$

  Once the investigation is concluded, each secret agent whose suspicion level rises above the user-defined threshold is executed.

- After the attribute **knowledge** rises above a user-defined threshold, the secret agent will send a message via a **shared message queue** with the police about the gang being complicit.
- If the attribute **knowledge** stays below the user-defined threshold, then a periodic communication (make it long enough) would be established between the secret agent and the police, and the secret agent would send their current knowledge level.
- The **secret agent** should be ready to receive a message through the **shared message queue** requesting them to share their current knowledge on the gang.
- When a secret agent **dies**:
  - The police will try to plant one to refill their position.
  - The death of the secret agent is of no use to the gang's investigation.

5.6 Police

Multiple threads (squads) shall undertake the task of uncovering the mysteries. To simplify it, each squad is responsible for one gang.

5.6.1 Actions

Phase 1: Secret Agent Interaction Check secret agent logic for more details, but here is a recap:

- Police awaits a message from the secret agents. Beware of the message's knowledge field though, check secret agent logic for more info.
  - knowledge > threshold: gang involvement in criminal activities.
  - knowledge < threshold: current knowledge about gang. (sent periodically)
- If knowledge < threshold, police will send a message to another secret agent for info.

Phase 2: Gang Uncovering Send a message to the gang process conveying the jail period.

5.7 Police-Agent Communication

These two entities shall use message queues as a communication mechanism.

Handshaking If the police plants the agent successfully, the agent should know which police thread planted them so as to establish communcation. This is accomplished via the message (mode=0) sent in the first place to plant the agent in the gang. The police will send a message to the gang main thread to plant the agent, and will include its ID in the message. Luckily, the main threads ids are predetermined as below.

mtype Field The mtype field should be filled by the sender to specify the addressed entity. The recipient checks if the ID matches theirs, so they can receive the correct message. Recall that the communcating entities are:

- Gang main thread
- Secret agent
- Police thread (squad)

An mtype-allocation mechanism should be determined prior to communication. It is illustrated as follows:
*Note all mtypes are shifted 1 since mtype of 0 means wildcard.
*Note *_ids fall into the range of [0, MAX-1] inclusive.

$$agent_{mtype} = (MAX\_AGENTS + 1) * gang\_id + agent\_id + 1 \tag{8}$$

$$gang\_main_{mtype} = MAX\_AGENTS * gang\_id + MAX\_AGENTS + 1 \tag{9}$$

$$police_{mtype} = (MAX\_AGENTS + 1) * NUM\_GANGS + police\_id + 1 \tag{10}$$

Secret Agent Death The police thread needs to know which secret agent died at at time, so the main thread of the gang process of that agent send, once the secret agent dies, a message conveying their death to the police thread (mode=1).

Police Request When an agent does not provide enough information about the gang, the police needs to ask the other secret agent, but to ask them, you need a request message (mode=2).

Police Report The report should contain the knowledge level of the sending secret agent's knowledge level (mode=3).

```
typedef struct {
    long mtype;
    uint_8 mode;
    union {
        float knowledge;
        int agent_id;
    } message_content;
} message;
```

Listing 1: Message Struct