Emergency Response Simulation Report

Author:
Yosef Bekele

Date:
April 23, 2025

OOP Concepts Used

Abstraction
The `EmergencyUnit` class is an abstract base class that defines a template for all units (Police, Ambulance, Firefighter, etc.) with methods like `CanHandle` and `RespondToIncident`.

Inheritance
Specialized units such as `Police`, `Firefighter`, `Ambulance`, and `SearchAndRescue` inherit from `EmergencyUnit`, extending functionality.

Polymorphism
Different unit types override the `RespondToIncident` and `CanHandle` methods to behave appropriately depending on the incident type.

Encapsulation
Data like `Name`, `Speed`, and `Type` are contained within each class, accessed via constructors and not directly exposed for modification.

Lessons Learned

- Designing for flexibility with polymorphism allowed easy extension (e.g., adding `SearchAndRescue`).
- Simulating response time with `Stopwatch` added realism and scoring complexity.
- Implementing user interaction via CLI helped validate real-time decisions.
- Learned the importance of balancing game mechanics (scoring vs difficulty).

Challenges Faced

- Ensuring correct unit selection logic — sometimes users selected invalid units; added input validation.
- Balancing score calculation with speed vs. difficulty vs. time required tuning.

Class Diagram

EmergencyUnit
(abstract)

- Name: string
- Speed: int

+ CanHandle(type): bool
+ RespondtoIncident(i)

Incident

- Type: string
- Location: string
- Difficulty: int

Police

Overrides:
+ CanHandle
+ RespondInti
dent

Firefighter

Overrides:
+ CanHandle
+ RespondInci
dent

Ambulance

Overrides:
+ CanHandle
+ RespondInc

**SearchA
nRescue**

EmergencySimulation

+ Main()
+ InitializeUnits()
+ GenerateIncident()
+ GetUserSelectedUnit
+ HandleIncident()
+ CalculateScore()