

Emergency Response Simulator – Project Report

1. Introduction

The **Emergency Response Simulator** is a console-based simulation game where players manage emergency units (Police, Firefighters, and Ambulance) in response to different incidents like crimes, fires, and medical emergencies. The player's objective is to select the right unit for the given incident and maximize their score based on the unit's response time and the difficulty of the incident.

2. OOP Concepts Applied

The project is designed using Object-Oriented Programming (OOP) principles. The following key OOP concepts were used:

-

Abstraction: We created an abstract class `EmergencyUnit`, which provides a blueprint for the different types of emergency units (Police, Firefighters, Ambulance). This class defines common properties (like `Name` and `Speed`) and abstract methods (`CanHandle` and `RespondToIncident`) that must be implemented by derived classes.

-

-

Inheritance: The `Police`, `Firefighter`, and `Ambulance` classes inherit from the `EmergencyUnit` class, each implementing the abstract methods in a way specific to their responsibilities. For example, only the `Police` unit can handle `Crime` incidents, while the `Firefighter` unit responds to `Fire` incidents.

-

-

Polymorphism: The `CanHandle` method is overridden in each subclass, allowing each unit to check if it can handle a particular type of incident. The `RespondToIncident` method is also overridden to provide a specific response for each unit based on the incident type.

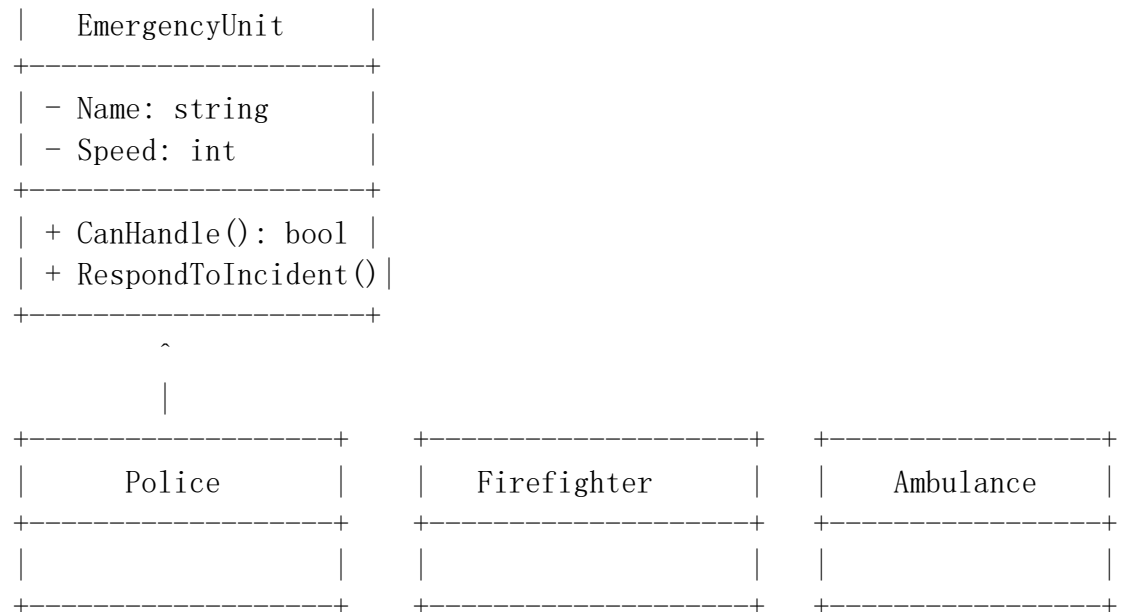
-

3. Class Diagram (Text-Based)

pgsql

CopyEdit

+-----+



4. Gameplay Overview

The game simulates a busy city with random incidents occurring at various locations. The incidents include:

-

Crime: Handled by the `Police` unit.

-
-

Fire: Handled by the `Firefighter` unit.

-
-

Medical Emergency: Handled by the `Ambulance` unit.

-

In each round, the game:

-

Generates a random incident type and difficulty level (Easy, Medium, Hard).

-
-

Prompts the player for the location of the incident.

-
-

The player selects which unit should respond based on the incident type.

-
-

Points are awarded for correct responses, and penalties are given for incorrect responses or missed incidents.

-

5. Scoring System

The game uses a scoring system based on the following factors:

-

Base Points: Based on the difficulty of the incident (Easy = 10, Medium = 20, Hard = 30).

-
-

Response Time: Each unit has a speed rating, which affects how quickly it can respond to incidents. A faster unit results in a shorter response time and more points.

-
-

Bonus Points: Each turn automatically adds 10 bonus points to the score.

-

If the player selects the correct unit for the incident type, they receive points based on the unit's response time and the incident's difficulty. If the wrong unit is selected, they lose points.

6. Challenges Faced

-

Handling User Input: Ensuring that the user inputs were properly validated (e.g., unit selection and location) was challenging. Handling invalid inputs gracefully was key to maintaining a smooth user experience.

-
-

Scalability: The project could be expanded by adding more unit types, incident types, and scoring mechanics, but the initial design provided a solid foundation for further enhancements.

-

7. Conclusion

The **Emergency Response Simulator** demonstrates the use of OOP principles to create a simulation that models the real-time decision-making process of emergency response units. It is a fun and interactive game that combines strategic thinking with quick decision-making, allowing players to experience the challenges faced by emergency responders in a dynamic city environment.