

Proiect POO: HiveMind - Autonomous Logistics Simulator

1. CONTEXT ȘI OBIECTIVE

Suntem în anul 2030. Logistica urbană este gestionată de flote autonome. Voi aveți rolul de **Software Architect** pentru sistemul "HiveMind", un motor de simulare și decizie care coordonează o flotă eterogenă de roboți (Drone, Roboți Tereștri, Scutere).

Obiectivul Principal:

Crearea unei aplicații C++ modulare care:

1. **Generează procedural** hărți de oraș (simulări de mediu).
2. **Simulează fizica și ciclul de viață** al agentilor.
3. **Optimizează profitul** printr-un algoritm propriu de alocare a task-urilor (HiveMind), demonstrând gândire algoritmică "out of the box".

2. DESCRIEREA LUMII (MEDIUL DE SIMULARE)

2.1. Harta (The Grid)

Orașul este reprezentat ca o matrice de dimensiuni MxN (citite din fișier). Fiecare celulă are un rol specific:

- [.] **Drum (Space)**: Zonă accesibilă deplasării.
- [#] **Zid (Wall)**: Obstacol. Nu poate fi traversat de unitățile terestre.
- [B] **Bază (Hub)**:
 - Punctul central de depozitare.
 - Locul de spawn al tuturor agentilor și al pachetelor.
 - Funcționează implicit ca **Stație de Încărcare**.
- [S] **Stație de Încărcare (Station)**: Locații suplimentare unde roboții își încarcă bateria.
- [D] **Client (Destination)**: Locații fixe (case/firme) unde trebuie livrate pachetele.

2.2. Modulul "Genesis" (Generarea Hărții)

Nu veți lucra pe o singură hartă hardcodată. Trebuie să implementați un mecanism flexibil de generare.

Cerințe OOP:

- Utilizarea **Design Pattern-ului Strategy** (sau Builder) prin interfața IMapGenerator.
- Implementarea a două strategii:
 1. FileMapLoader: Încarcă harta dintr-un fișier text (pentru debug).
 2. ProceduralMapGenerator: Generează o hartă random, plasând ziduri, stații și clienți

conform parametrilor din configurare.

Validarea Hărții (Algoritmica):

După generare, harta trebuie validată automat (folosind Flood Fill sau BFS). Trebuie garantat că există drum de la Bază (B) către toți Clienții (D) și toate Stațiile (S). Dacă harta e invalidă, se regenerează.

3. FLOTA ȘI FIZICA SIMULĂRII

3.1. Specificatiile Agentilor (Clase Polimorfice)

Trebuie să implementați o ierarhie de clase derivate din Agent. Fiecare tip are avantaje și dezavantaje economice.

Tip Agent	Simbol	Mediu Deplasare	Viteză (pasi sau celule/tick)	Baterie Max	Consum (bat/tick)	Cost (\$/tick)	Capacitate (pacăte)
Drona	^	Aer (Ignoră #)	3	100	10	15	1
Robot	R	Sol (Ocolește #)	1	300	2	1	4
Scuter	S	Sol (Ocolește #)	2	200	5	4	2

- Viteză:** Câte celule parcurge într-un singur Tick.
- Cost (\$/tick):** Câți bani se scad din profit la fiecare tick de funcționare (menenanță).
- Agentii pot ocupa aceeași celulă simultan (nu se ciocnesc între ei)

3.2. Ciclul de Viață (State Machine)

La fiecare pas al simulării (Tick), fiecare agent se află într-o stare:

- IDLE:** Așteaptă în Bază (se încarcă automat).
- MOVING:** Se deplasează spre destinație. Consumă baterie.
- CHARGING:** Dacă e pe o celulă S sau B, bateria crește cu 25% pe tick.
- DEAD:** Dacă bateria ajunge la 0 în afara unei stații, agentul este pierdut definitiv.

4. LOGICA DE BUSINESS: CLIENTI ȘI PACHETE

Este esențial să distingeți între locații (statice) și comenzi (dinamice).

1. **Clienții (D):** Sunt plasați pe hartă la generarea lumii. Numărul lor este fix (ex: 10 clienți).
2. **Pachetele (Comenzi):**
 - Nu există pe hartă de la început.
 - Sunt generate dinamic în Bază (B) la intervale regulate de timp.
 - Fiecărui pachet nou i se asociază **aleatoriu** unul dintre clienții existenți (D) ca destinație.
 - Fiecare pachet are un reward intre 200 si 800 de credite, alocat random. Iar fiecare pachet are un deadline de livrare intre 10-20 ticks alocat tot random (utilizat pentru a calcula intarzirea).

5. ALGORITMUL "HIVEMIND" ȘI SCORING

Obiectivul vostru este maximizarea **Profitului Net**. Nu este suficient să livrați pachetele; trebuie să o faceți eficient.

5.1. Formula de Profit

Sistemul va calcula la final:

$$Profit = \sum(Rewards) - \sum(CosturiOperare) - \sum(Penalizări)$$

- **Cost Operare:** Suma costurilor pe tick pentru toți agenții activi. (Dronele sunt scumpe!).
- **Penalizări:**
 - **-500 credite** pentru fiecare Agent Mort (Battery Depletion).
 - **-200 / 50 credite** pentru fiecare Pachet Nelivrăt / Întârziat.

5.2. Strategia (Algoritmul Vostru)

HiveMind trebuie să decidă la fiecare tick:

- *Dispatching:* Aloc pachetul X robotului ieftin (dar lent) sau dronei scumpe (dar rapide)?
- *Routing:* Trimit robotul direct la client sau îl deviez pe la stația de încărcare?

6. CONFIGURARE (INPUT)

Aplicația nu va avea valori hardcodate. La rulare, va citi fișierul simulation_setup.txt:

Plaintext

```

MAP_SIZE: 20 20      // Dimensiuni harta
MAX_TICKS: 1000     // Durata simularii
MAX_STATIONS: 3     // Nr statii extra
CLIENTS_COUNT: 10   // Nr puncte de livrare (D)

// Flota de start
DRONES: 3
ROBOTS: 2
SCOOTERS: 1

// Dinamica
TOTAL_PACKAGES: 50
SPAWN_FREQUENCY: 10 // Apare un pachet la fiecare 10 ticks

```

7. LIVRABILE ȘI EVALUARE

Livrabile (Arhivă ZIP)

1. **Cod Sursă C++:** Organizat modular.
2. **Raport Simulare (simulation.txt):** Fișier generat automat la finalul rulării, conținând statistici (Pachete livrate, Profit Final, Agenți Pierduți).
3. **Documentație (PDF):**
 - Diagrama UML a claselor.
 - Descrierea algoritmului HiveMind și a strategiei de generare a hărții.
4. **Demo:** video de până la 3 minute max în care să fie prezentată tema și funcționalitățile.

Grila de Notare (20% din nota finală)

- **[25%] Arhitectură Software:** Utilizarea corectă a OOP (Încapsulare, Polimorfism, Moștenire, Abstractizare), Design Patterns (Strategy, Singleton, Factory), Managementul Excepțiilor, Teste Unitare. (comentați-le în cod unde le folosiți)
- **[25%] Funcționalitate Core:** Generare procedurală validă, pathfinding funcțional (fără trecere prin ziduri pentru roboți), respectarea fizicii bateriei.
- **[10%] Performanță Algoritmică:**
 - Algoritmul obține profit pozitiv?
 - Gestionează corect situațiile critice (baterie scăzută)?
- **[10%] Calitate și Stil:** Cod curat, tratarea excepțiilor, documentație clară.
- **[30%] Cunoștințe teoretice:** răspuns la întrebări legate de materie și de proiect la coloviu.

Termen Limită

8 ianuarie 2025, Ora 23:59. Încărcare pe Moodle.