

Descrierea algoritmului HiveMind

1. Strategia de generare a hărții

Generarea hărții a fost făcută printr-un Design Pattern numit Strategy. Acesta permite utilizatorului să aleagă între o hartă fixă sau una generată aleatoriu, la rulare.

- **Generarea Procedurală (ProceduralMapGenerator)**

Această strategie creează un mediu de testare dinamic la fiecare rulare:

- ❖ Distribuția elementelor: Generatorul plasează aleatoriu Baza ('B'), zidurile ('#'), clienții ('D') și stațiile de încărcare ('S') pe baza parametrilor din ConfigMap.
- ❖ Zidurile sunt configurate să ocupe aproximativ 25% din suprafața totală pentru a crea un mediu complex, dar navigabil și, totodată, fără a face livrarea pachetelor de către agenți extrem de dificilă.
- ❖ Algoritmul de validare: Orice hartă generată este supusă unui test riguros de validare prin metoda isValidMap. Folosind un algoritm BFS (Breadth-First Search), sistemul verifică dacă toate punctele de destinație ('D') și toate stațiile ('S') sunt accesibile din Baza ('B'). Dacă un punct este izolat de ziduri, harta este respinsă și generarea reîncepe.

- **Încărcarea din fișier (FileMapLoader)**

Această metodă permite utilizarea de scenarii predefinite pentru benchmark-uri:

- ❖ Parsare și mapare: Citește un fișier text (ex: map_file.txt) linie cu linie, convertind caracterele în matricea internă a clasei Map.
- ❖ Integritate: Include verificări pentru dimensiunea corectă a liniilor și aplică același filtru de validare BFS pentru a asigura că harta încărcată este funcțională din punct de vedere al navigației.

2. Algoritmul HiveMind

Algoritmul HiveMind reprezintă creierul central al simulării, fiind implementat sub forma unui Design Pattern numit Singleton, pentru a asigura o coordonare unică și centralizată a întregii flote de agenți.

❖ Strategia de alocare (Economic Dispatch)

- ❖ Procesul de decizie se bazează pe maximizarea profitului net prin metoda assignPackages. Algoritmul funcționează astfel:
- ❖ Evaluarea profitabilității: Pentru fiecare pereche agent–pachet disponibil, se calculează un „Profit Estimat” folosind formula:
$$\text{Recompensă} - \text{Cost Operare} - \text{Penalizare Întârziere}.$$
- ❖ Simularea resurselor (Battery Awareness): Înainte de alocare, HiveMind calculează distanța totală (ridicare + livrare + întoarcere la bază) și verifică dacă bateria curentă a agentului, minus consumul specific (ex: 10/tick pentru dronă, 2/tick pentru robot), permite finalizarea misiunii în siguranță.
- ❖ Gestionarea încărcării: Dacă bateria este insuficientă pentru un drum direct, algoritmul simulează o rută care include o oprire la cea mai apropiată stație de încărcare ('S'), recalculând profitabilitatea, luând în calcul timpul de încărcare și costurile suplimentare de deplasare.

- ❖ Prioritizarea prin sortare: Toate ofertele valide sunt stocate în obiecte de tip Offer și sortate descrescător după profit. Această abordare garantează că pachetele cele mai valoroase sunt preluate de agenții cei mai eficienți (ex: roboți lenți, dar ieftini vs. drone rapide, dar scumpe).
- ❖ Capacitatea agenților: Fiecare agent are o capacitate maximă de pachete. La generarea unui pachet nou, algoritmul determină dacă acesta este profitabil pentru respectivul agent, moment în care agentul îl „rezervă”. După rezervare și livrarea pachetului pe care îl deține, agentul se întoarce în bază și preia și celelalte pachete rezervate, calculându-se costurile pentru toată această operație.

- **Execuția și monitorizarea (Move Agents)**

- ❖ Funcția moveAgents administrează fizica și stările flotei la fiecare unitate de timp (tick):
 - ❖ Rutare inteligentă: Se utilizează clasa PathFinder pentru a decide următoarea mișcare. Dronele folosesc distanța Manhattan (trec prin pereții '#'), în timp ce agenții tereștri folosesc BFS (Real Distance) pentru a ocoli obstacolele.
 - ❖ Siguranță și redirectionare: Algoritmul monitorizează pragul critic de baterie. Dacă un agent riscă să rămână fără energie, acesta este redirectionat automat către bază sau o stație, prioritatea fiind evitarea penalizării de -500 credite pentru pierderea agentului.
 - ❖ Sistemul de penalizări: HiveMind gestionează automat bugetul, scăzând -200 credite pentru fiecare pachet pierdut și -50 credite pentru livrările efectuate după expirarea duratei de viață (deadline) a pachetului.

Diagrama UML a claselor:

