# Projet Fosyma 2022









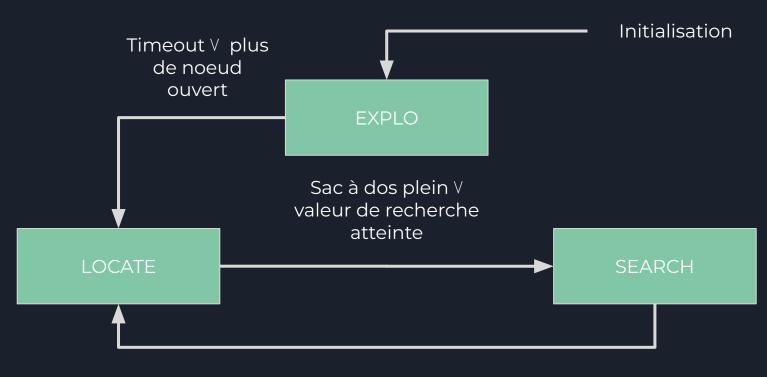




CASSAN Jules - YAO Qingyuan

Github: https://github.com/MRVNY/Fosyma

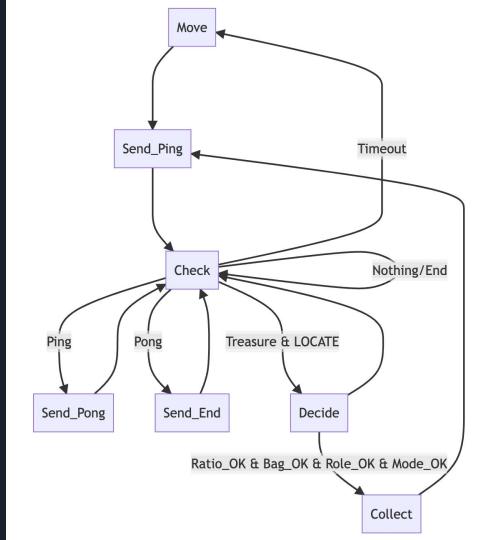
## Mode de Déplacement



Mise à jour de la valeur de recherche

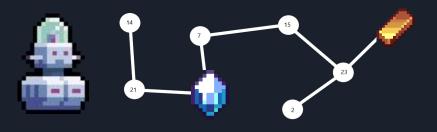
### FSM

- Move État initiale
- Check
  - DEFAULT
  - TIMEOUT
  - o SEND\_PONG
  - SEND\_END
  - o DECIDE
- Decide
  - DEFAUT
  - COLLECT



## Représentation de la carte et des trésors

Les Agents se déplace sur la carte et enregistre les positions des trésors

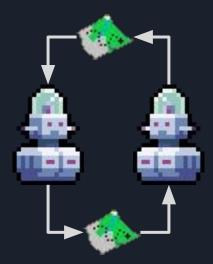


MapRepresentation

Contient les positions des noeud, les méthodes pour choisir le prochain noeud et la fusion des cartes

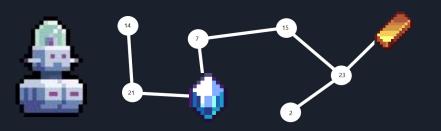
TreasureCollection

Contient les informations de position, quantitée et répartition des trésors sur la carte



## Représentation de la carte et des trésors

Les Agents se déplace sur la carte et enregistre les positions des trésors

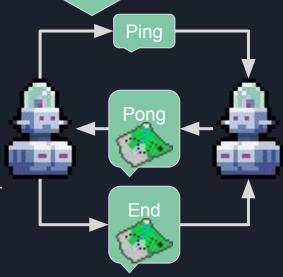


MapRepresentation

Contient les positions des noeud, les méthodes pour choisir le prochain noeud et la fusion des cartes

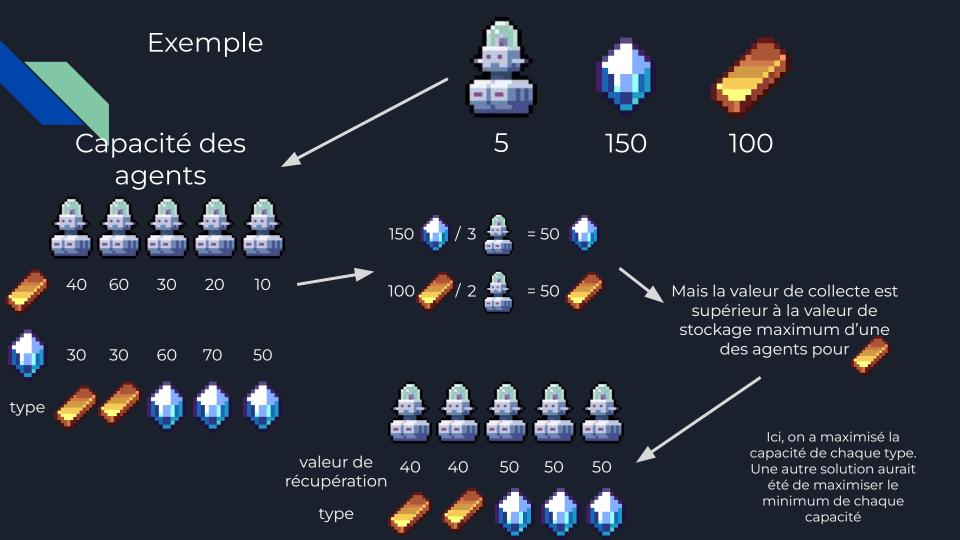
TreasureCollection

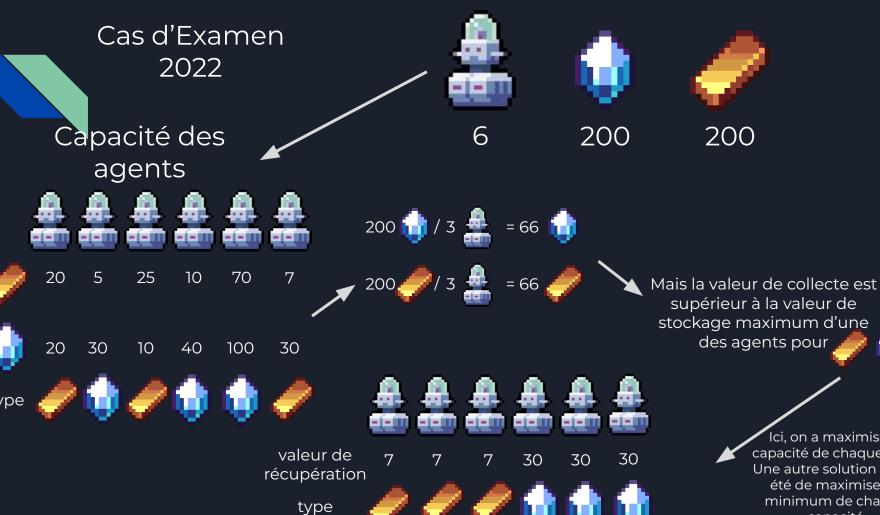
Contient les informations de position, quantitée et répartition des trésors sur la carte class Message
MapRepresentation map
Integer mode
String goal
String nextNode
String position



## Répartition équitable des ressources entre les agents

- Récupération des informations de la carte et des autres agents lors de la phase d'exploration
- Après avoir explorée au maximum la carte, les agents se répartissent les type de ressources à collecter. Cette attribution est définitive
- Chaque agent doit rechercher une quantitée de ressource identique
- La valeur de recherche est dynamique en fonction des nouveaux trésors découvert sur la carte
- Elle est la même pour les agents qui se voient attribuer le même type
- Si un agent à une capacité de collecte plus faible que cette valeur de collecte, on bride la valeur de tous les agents assignés à la ressource.





lci, on a maxim<u>isé la</u> capacité de chaque type. Une autre solution aurait été de maximiser le minimum de chaque capacité

### Coordination

#### Goal & Priorities

- Goal : destination (idéalement) optimale
- Priorities : destinations trié avec critère selon mode

#### Négociation

- Hiérarchie de modes: EXPLORE > LOCATE > SEARCH
- o Conflit:
  - 1. Même goal
    - (myGoal.equals(hisGoal))
  - 2. Blocage
    - (myNextNode == hisNextNode)
    - (myPos == hisNextNode && hisPos == myNextNode)

#### Déblocage

Déclenché si l'agent a pas bougé pendant 10 itérations

Tu blog les mon chemin

Je suis en LOCATE

Je suis en EXPLORE

Je te laisse la place





**Pong**: Voici ma carte et mon état (message)

Ah on se bloque

Mais puisque je suis en LOCATE et tu es en EXPLORE, je garde mon goal

**End**: Voici ma carte et mon état (message)





**End**: Voici ma carte et mon état (message)

Ah on se bloque

Tu es en LOCATE et je suis en EXPLORE, du coup je te laisse la place

Si je vais à 35, je te bloquerais pas





```
'23': Non, c'est le même noeud que lui (myGoal.equals(hisGoal))
'56': Non, mon prochain pas est son prochain pas (myNextNode == hisNextNode)
'37': Non, mon prochain pas est sa position et son prochain pas est ma position (myNextNode==hisPos && hisNextNode==myPos)
'11': Oui, il est bon, je change mon goal à ça alors! (myAdventurer.setGoal(newGoal); break;)
```

