

Magellan - JDIS Games 2024

Mise en situation

Imaginez-vous au début du 16e siècle 📅, une époque où les cartes du monde étaient encore remplies de zones inconnues et mystérieuses 🗺️. Au milieu de cette période de découvertes et d'explorations, Ferdinand Magellan, un explorateur portugais, se préparait. Son objectif : trouver une route maritime vers les îles aux épices 🌿 dont les richesses étaient convoitées par toute l'Europe 🌐.

À bord de son navire 🚢, Magellan et son équipage s'aventurèrent dans des eaux inexplorées 🌊, armés de patience et de leurs connaissances en navigation 🧭. Leur principal allié était l'astrolabe, un instrument capable de mesurer l'altitude des étoiles et des planètes ✨. Cet outil était indispensable pour tracer leur chemin à travers les vastes océans, leur permettant de se frayer un chemin vers l'inconnu 🗺️.

Votre objectif

À l'image de Magellan, vous deviendrez des explorateurs naviguant sur des eaux inconnues à la recherche d'un trésor. Vous connaissez sa coordonnée 📍, mais l'emplacement des obstacles reste floue 🗺️. Il y a aura deux phases dans votre aventure : découverte 🔍 et prise 🗝️.

Dans le jeu Magellan, vous contrôlez un agent pouvant se déplacer sur une carte. Vous devrez vous déplacer sur la carte afin de ramasser les pièces 🪙 et les trésors 💰 se trouvant dans votre chemin tout en vous défendant contre les autres agents ⚔️.

Déroulement du jeu

Le jeu est sous format "Long running" ⌚ ce qui signifie qu'il n'arrête jamais. Plusieurs parties auront lieu eu courant de toute la journée 🕒. Vous accumulerez des points **tout au long de la journée** 📊. Vous devez prévoir les meilleurs moments pour déconnecter votre agent pour y mettre une nouvelle version. Il est conseillé de concevoir son bot de manière incrémentale et de faire de l'amélioration continue 🔄.

Un cycle de rafraichissement (tick) dure 300 ms ⌚. Durant un cycle de rafraichissement, le serveur effectuera 10 boucles d'action.



Lors des précédentes expéditions, certaines informations ont été recueillies 📄. Par conséquent, lors d'une partie, l'agent sera placé sur une des carte 🗺️ et recevra des informations sur les [éléments](#) s'y trouvant, bien que l'emplacement exact des murs n'ait pas été recueilli 🧱. Par la suite, l'agent pourra envoyer plusieurs [actions](#) par cycle de rafraîchissement.

À chaque nouvelle partie, tous les murs et les pièces sont placés de manière aléatoire sur la carte 🎲. Une partie est composée de deux phases :

- [Phase 1 : Découverte](#) 🔍
- [Phase 2 : Prise](#) 🗝️

Éléments de la carte

Les éléments suivants se trouveront sur la carte :

Pièce		Placés aléatoirement dans la première phase du jeu, donne 40 points lorsque ramassé.
Trésor		Placé aléatoirement à la deuxième phase du jeu, donne 1200 points lorsque ramassé. Seulement un trésor est présent sur la carte.
Mur		Les murs ne sont pas visibles sur la carte par les agents.

Phase 1

Lors de la première phase de la partie, les agents connectés au serveur ainsi que les pièces seront placés de manière aléatoire sur la carte. 🎲 Ceci est la phase de découverte et d'exploration.

Lorsqu'un agent prend une pièce, celle-ci réapparaît de manière aléatoire sur la carte.

Lors de la fin de la phase, les agents et les pièces sont enlevés de la carte pour passer à la deuxième phase.

Phase 2

Lors de la dernière phase de la partie, un trésor sera placé sur la carte 🗳️, les agents apparaîtront à équidistance de déplacement du trésor. Dans cette phase, il n'y aura pas de pièces sur la carte.

Mort

Lorsque l'agent perd toute sa vie 💀, ce dernier disparaît de la carte et ne reçoit aucune donnée du serveur pendant un temps défini ⌚.

Actions

Plusieurs actions peuvent être envoyées au serveur dans un même cycle de rafraichissement avec certaines contraintes d'utilisation.

Déplacement

La position envoyée est celle vers laquelle l'agent va se déplacer 🏠. L'agent ne peut pas traverser les murs 🧱. Pour vous rendre à une position précise, vous devrez implémenter un algorithme de recherche de chemin (pathfinding).

Cette action n'a pas de contraintes d'utilisation.

Attaque

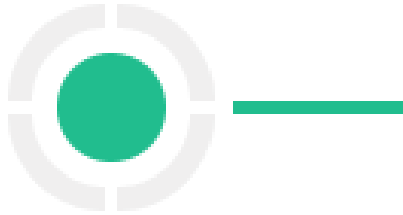
- **Changement d'arme** 🗡️ Chaque arme vous permet d'effectuer une action différente. Pour pouvoir l'effectuer, vous devez vous équiper de l'arme à l'aide du changement d'arme.

Cette action ne peut pas être accompagnée de l'utilisation d'une arme dans un même cycle de rafraichissement.

- **Pistolet** 🔫 Pour utiliser le pistolet, il faut envoyer la position de destination souhaitée d'un projectile. Le projectile a une portée définie. Lorsqu'un projectile rentre en collision avec un autre agent, ce dernier reçoit du dégât. Le projectile disparaît par la suite.

Cette action ne peut pas être accompagnée de l'équipement d'une arme dans le même cycle de rafraîchissement.

- **Lame** ✂ En début de partie, la lame apparaîtra à 0 radians du joueur comme illustré à l'image suivante.



Pour changer la rotation de la lame, il faut envoyer la nouvelle rotation en radians. Lorsque la lame rentre en collision avec un agent, celui-ci reçoit maximum 40 points de dégât. Le nombre de points de dégâts dépendra de la durée que la lame sera en contact avec l'agent durant un cycle de rafraîchissement.

Cette action ne peut pas être accompagnée de l'équipement d'une arme dans le même cycle de rafraîchissement.

Sauvegarde

Une quantité limitée d'octets pourra être envoyée au serveur 📁. Ces octets seront sauvegardés le temps d'une partie. Ces données seront reçues par le joueur à chaque fois qu'il va se connecter au serveur. Cette action vous permettra donc de sauvegarder de l'information dont vous auriez accès dans une même partie lorsque votre bot sera déconnecté.

Données reçues

État de la carte

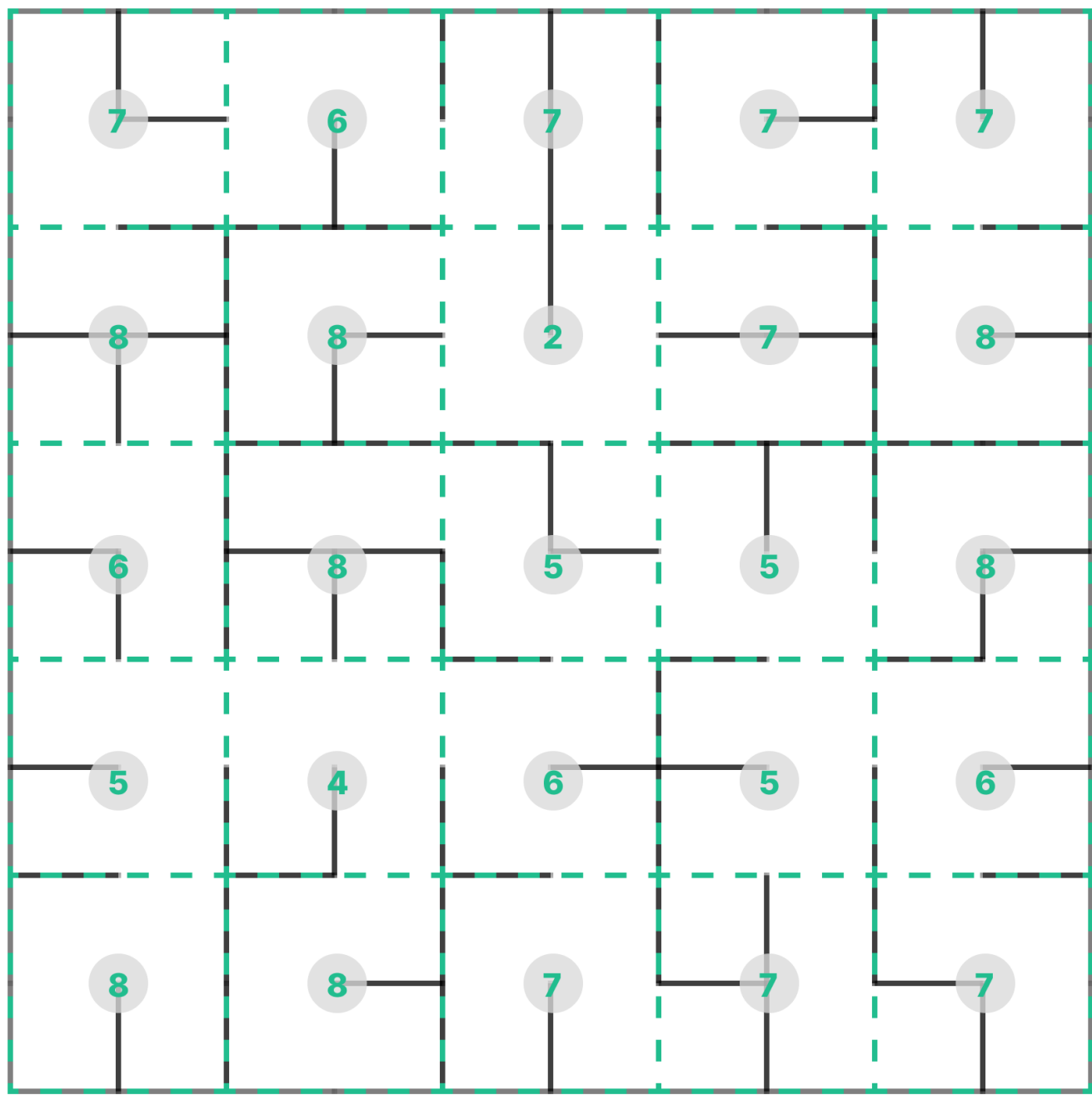
L'agent reçoit l'état de carte lorsqu'il se connecte au serveur 🗺️.

```
MapState {
  DiscretGrid : [][]Integer,      // Grille contenant le nombre de murs par 4
  cases
  size: int,                      // La taille de la carte
  walls: List[Collider],          // La liste des murs, cette liste devrait être
  vide
  save: bytearray                 // La mémoire que vous pouvez sauvegarder dans
  le jeu
}
```

Grille discrète

La carte est reçue sous forme de grille discrète. La grille discrète contient seulement le nombre de murs par 4 cases. Les murs extérieurs délimitant la carte sont aussi comptés. La grille est envoyée à chaque début de partie dans l'état de la carte. La grille ne change pas au long d'une partie, mais une nouvelle carte est générée à chaque début de partie. Les murs d'une grille sont placés de manière aléatoire à chaque nouvelle partie 🎲.

Voici la représentttion d'une grille discrète pouvant être reçu pour une carte donnée :



Pointage

Vous accumulez des points tout au long de la journée à l'aide de ses items :

Item	Points
Pistolet	15
Lame	40
Pièce	40
Trésor	1200

État du jeu

L'agent reçoit l'état du jeu à chaque cycle de rafraichissement.

```
GameState {
  currentTick : Integer, // Cycle de rafraichissement
  courant
  currentRound : Integer, // Tour courant
  Players : [ // Liste des informations des
    joueurs
      name : String, // Nom du joueur
      color : String, // Couleur du joueur sur la
    carte
      health : Integer, // Quantité de vie du joueur
      score : Integer, // Nombre de points du joueur
      position : { x : Float, y : Float }, // Position actuelle
      destination : { x : Float, y : Float }, // Position de destination
      playerWeapon : Integer, // Type d'arme équipée,
    aucune(0), pistolet(1), épée(2)
    Projectiles: [ // Informations des
      projectiles du pistolet
        uid : String, // Identifiant du projectile
        position : { x : Float, y : Float }, // Position actuelle
        destination : { x : Float, y : Float }, // Position de destination
      ],
      Blade: { // Informations sur l'épée
        start : { x : Float, y : Float }, // Position du début de l'épée
        end : { x : Float, y : Float }, // Position de fin de l'épée
        rotation : Integer // Angle en radians
      }
    ],
    Coins : [ // Liste des pièces ou trésor
      uid : String, // Identifiant de la pièce
      value : Integer, // Valeur de la pièce ou du
    trésor
      position : { x : Float, y : Float } // Position de la pièce ou du
    trésor
  ]
}
```








Constantes

Constantes de la carte	
Largeur de la carte en nombre de cellule	10
Hauteur de la carte en nombre de cellule	10
Largeur de chaque cellule	10.0
Hauteur de chaque cellule	10.0
Constantes d'un agent	

Constantes d'un agent	
Taille d'un bot	1.0
Vitesse du joueur (par seconde)	1.15
Vie maximale	100
Temps avant de réapparaître apres une mort (seconde)	5.0
Constantes d'un projectile	
Taille du projectile	0.35
Vitesse du projectile (en seconde)	3.0
Dégats infligés par le projectile	30
Durée de vie du projectile (en seconde)	5.0
Points	15
Constantes d'une lame	
Longueur de la lame (à partir du centre d'un agent)	2.0
Épaisseur de la lame	0.25
Dégats infligés par la lame	4 - 40
Points	4 - 40
Constantes d'une pièce	
Taille d'une pièce	0.5
Quantité	30
Points	40
Constantes d'un trésor	
Taille du trésor	4.0
Points	1200

Interaction avec la plateforme

Comment m'inscrire ?

1.  Rendez-vous sur la page <http://jdis-ia.dinf.fsci.usherbrooke.ca>
2.  Cliquez sur le bouton en bas à gauche pour accéder au formulaire d'inscription.
3.  Dans le formulaire, inscrivez le nom de votre Bot (3 à 16 caractères).
4.  Une fois le nom du bot entré, cliquez sur le bouton pour vous enregistrer.
5.  Une fois enregistré, vous devriez recevoir un jeton d'authentification au bas droit de la page.
6.  Assurez-vous de prendre en note le jeton d'authentification, vous en aurez besoin pour connecter votre agent.
7.  Si jamais vous avez oublié de le noter, allez voir les organisateurs, ils vous aideront.

8. 🔑 Chaque nom d'équipe doit être unique.

C'est tout ! Vous êtes prêt.e à participer ! 🏆