Aquar.iom : Présentation et règles

Présentation générale

Le jeu est constitué d'une carte en deux dimensions dans laquelle évoluent des cellules de forme circulaire, comme du plancton dans un aquarium. Ce jeu est fortement inspiré par agar.io.

La carte est choisie par le serveur avant le début de la partie et définit de nombreux paramètres comme les positions de départ des cellules. Les cellules peuvent se déplacer dans n'importe quelle direction mais ne pourront jamais sortir de la carte.

Les cellules sont de différents types :

- Les **cellules joueuses** sont contrôlées par les joueurs. En plus de se déplacer, elles peuvent également manger d'autres cellules, se scinder, etc. C'est le seul type de cellules qui appartiennent à des joueurs.
    - Les cellules joueuses peuvent être **isolées** ou non. Les cellules isolées ne pourront pas fusionner ensemble (voir le paragraphe sur les actions possibles, section "manger d'autres cellules"). L'isolement est un caractère temporaire : au bout d'un certain nombre de tours, une cellule isolée redeviendra non isolée.
- Les **cellules neutres initiales** sont des cellules présentes depuis le début de la partie et qui ne peuvent pas se déplacer ni manger d'autres cellules. Elles ne font aucune action, et réapparaissent à leur place au bout d'un certain temps après s'être fait manger. Leurs positions sont fournies par le serveur avant le lancement de la partie.
- Les **cellules neutres non initiales** résultent de l'abandon de ses cellules par un joueur. Elles ont les mêmes caractéristiques que les cellules neutres initiales mais ne réapparaissent pas après avoir été mangées.
- Les **virus** sont un type de cellules spécial qui font exploser les cellules joueuses qui les mangent. Une cellule joueuse qui mange un virus va être scindée en plusieurs cellules plus petites (qui appartiennent encore au joueur) dont le nombre dépend de la carte.

Les cellules sont définies à chaque instant par leur type, leur propriétaire (dans le cas des cellules joueuses), la position de leur centre et leur masse. La masse des cellules permet de définir le rayon du disque qui les représente.

Au cours d'une partie, les cellules joueuses vont donc se déplacer en mangeant des cellules neutres, des virus et d'autres cellules joueuses. La vitesse de déplacement des cellules dépend de leur masse : plus une cellule est grosse, plus elle se déplace lentement.
Le score d'un joueur dépend de la masse totale de toutes ses cellules depuis le début de la partie :

$$score_{joueur}(tour\ courant) = \sum_{tour\ initial}^{tour\ courant} masse\ totale_{joueur}$$

D'waprès la formule précédente, le score d'un joueur est donc la somme des masses de ses cellules à chaque tour depuis le premier tour de la partie.
La partie se termine au bout d'un nombre de tours, déterminé par le serveur au début de la partie. Le joueur avec le score le plus élevé est déclaré vainqueur.

Communication entre le serveur et les clients

Puisque aquar.iom est un jeu en réseau, jouer une partie nécessite plusieurs agents communiquants. Un **serveur** est chargé de la gestion du jeu. Différents clients peuvent se connecter au serveur. Tout d'abord, une **visualisation** permet d'afficher comment une partie se déroule. Enfin, les **joueurs** sont des clients qui peuvent décider des actions à effectuer dans la partie. La manière dont les agents communiquent est régie par un protocole qui ne sera pas très détaillé dans le présent document. Une bibliothèque facilitant la communication avec le serveur vous est proposée mais si vous souhaitez implémenter vous-mêmes cette partie, vous pouvez trouver dans le dossier *proto* différents fichiers décrivant comment ce protocole fonctionne. Tout joueur doit effectuer les actions suivantes pour respecter le protocole :

1. Se connecter au serveur,
2. S'identifier en tant que joueur (en donnant son nom),
3. Attendre un message d'accueil du serveur (qui contient les paramètres de la partie),
4. Attendre un message disant que le jeu commence (qui contient le numéro du joueur, la position initiale des différentes cellules),
5. Attendre le début d'un tour de jeu,
6. Réagir au tour de jeu en disant au serveur quelles sont les actions à réaliser, puis revenir à l'étape 5.

Déroulement d'un tour de jeu

Aquar.iom est un jeu de type tour par tour. Ainsi, le déroulement d'une partie peut se résumer à un enchaînement de tours de jeu. Chaque tour de jeu se déroule de la manière suivante :

1. Le serveur envoie l'état courant du jeu à tous les clients,
2. Les joueurs prennent des décisions concernant leurs cellules et les envoient au serveur,
3. Une fois un certain pas de temps dépassé, le serveur traite les décisions reçues et actualise l'état courant du jeu.

La manière dont l'état du jeu est actualisé est détaillée ci-dessous :

1. Les cellules joueuses sont divisées (dans l'ordre des messages reçus),
2. Les virus sont créés (dans l'ordre des messages reçus),
3. Les cellules joueuses se déplacent (dans l'ordre des messages reçus),
4. Les abandons de cellules par les joueurs sont effectués (dans l'ordre des messages reçus),
5. Les cellules joueuses perdent de la masse,
6. Pour chaque cellule joueuse, le nombre de tours d'isolation restants est mis à jour,

7. Le nombre de tours avant l'apparition des cellules neutres initiales ayant été absorbées est mis à jour,
8. Les collisions entre les différentes cellules sont effectuées. Cette phase de collision permet de savoir quelles cellules sont absorbées, quelles cellules ont fusionné...

Manger d'autres cellules

Lorsqu'une cellule joueuse rencontre une autre cellule plus petite, elle va la manger si toutes les conditions suivantes sont remplies :

- Le **centre** de la petite cellule doit se trouver dans le rayon de la grosse cellule.
- Si la petite cellule appartient au même joueur que la grosse, aucune des deux cellules ne doit être **isolée**.
- Si la petite cellule appartient à un joueur différent de celui à qui appartient la grosse cellule, le **ratio de leurs masses** doit être supérieur à une certaine valeur qui dépend de la carte (reçue dans les paramètres de la partie).

Note : Lorsque la cellule rencontrée est une cellule neutre, la contrainte sur le ratio des masses ne s'applique pas : les cellules neutres peuvent se faire absorber par des cellules joueuses plus petites (une cellule joueuse minuscule peut absorber une cellule neutre très massique).

La cellule absorbante voit alors sa masse changer :

- Si la cellule absorbée **appartient au même joueur**, la nouvelle masse est égale à la somme des masses des deux cellules. On dit que les deux cellules ont fusionné.
- Si la cellule absorbée est une cellule n'appartenant pas au même joueur et **n'est pas un virus**, la masse de la cellule absorbante augmente d'un pourcentage de la masse de la cellule absorbée (défini dans les paramètres de la partie). Par exemple, $nouvelle\ masse = masse(cellule\ absorbante) + 80\% * masse(cellule\ absorbée)$.
- Si la cellule absorbée est un **virus**, on soustrait la masse du virus à la masse de la cellule, puis on divise la masse de la cellule par 2. Ensuite, des petites cellules satellites appartenant au joueur sont créées.
    - Le **nombre de satellites** créé est limité par un paramètre du jeu. Le joueur ne peut pas non plus posséder plus d'un certain nombre de cellules (paramètre de la partie) ; le nombre maximum de cellules par joueur limite donc également le nombre de satellites qui sont créés.

      $$nb\ satellites = min[nb\ satellites\ max, nb\ cell\ max/joueur - nombre\ de\ cell\ du\ joueur]$$

      Si la cellule se trouve trop près du bord de la carte, il est possible que moins de satellites soient créés (voir la puce "position des satellites" ci-dessous).
    - La **masse des satellites** dépend de la masse ce la cellule initiale, de la masse du virus et du nombre maximum de satellites :

      $$masse\ satellites = \frac{masse\ cellule\ mère - masse\ virus}{2 * nombre\ max\ de\ satellites}.$$

      Attention : comme le nombre de satellites effectivement créés n'est pas forcément le nombre maximum autorisé, il risque d'y avoir une perte de masse supplémentaire.
    - **Position des satellites** : tous les satellites créés sont tangents à la cellule initiale. Ils sont répartis équitablement sur le tour de la cellule (par exemple, si on crée quatre satellites, ils seront positionnés tous les 90°). Cependant, si la

cellule est trop **proche du bord de la carte**, et qu'un satellite devrait apparaître en dehors de la carte, ce satellite n'est pas créé (perte de masse supplémentaire).

Enfin, toutes les cellules (cellule initiale et ensemble des satellites) sont **isolées** pour un certain nombre de tours, lui aussi défini dans les paramètres de la carte.

Les actions possibles

A chaque tour, chaque joueur peut effectuer une action pour chaque cellule qu'il possède. De plus, un joueur peut également décider d'abandonner ses cellules actuelles pour recommencer avec de nouvelles cellules. Un joueur peut donc, en un tour, effectuer autant d'actions qu'il possède de cellules et abandonner ses cellules actuelles s'il le souhaite. Les différentes actions possibles sont décrites ci-dessous. Les paramètres et les contraintes de chaque action sont également décrits. Si les contraintes d'une action ne sont pas respectées, l'action est ignorée mais le joueur n'est pas exclu de la partie pour autant.

- **Se déplacer** : permet de déplacer une cellule vers une position souhaitée. Les paramètres de cette action sont :
  - le numéro unique de la cellule devant être déplacée,
  - la position vers laquelle la cellule doit se déplacer.

  Le serveur va essayer de déplacer la cellule vers la destination choisie. Si la destination est assez proche pour être atteinte en un tour, la cellule sera déplacée directement dessus. Sinon, la cellule sera placée le plus loin possible, en ligne droite, vers la destination choisie. Voici les contraintes sur cette action :
  - la cellule doit exister et appartenir au joueur qui souhaite la déplacer,
  - les coordonnées de la position souhaitée doivent être finies,
  - la cellule ne doit pas avoir déjà agi pendant ce tour.

  Attention : la position d'une cellule est définie par la position de son centre.
  Attention : la cellule se déplace en ligne droite et n'évite pas les obstacles.

- **Se scinder** : permet de diviser une cellule en deux : une cellule *mère* (l'ancienne cellule) et une cellule *fille* (la cellule créée). Les paramètres de cette action sont :
  - le numéro unique de la cellule *mère* (celle devant être scindée),
  - la position vers laquelle on souhaite créer la cellule *fille*,
  - la masse de la cellule *fille*.

  Le serveur va essayer d'ôter la masse de la cellule *fille* à la cellule *mère*, réduire le rayon de la cellule *mère* en conséquence puis faire apparaître la cellule *fille* tangente à la cellule *mère* dans la direction souhaitée (si la direction souhaitée et le centre de la cellule *mère* sont confondus, la cellule *fille* est créée au-dessus de la cellule *mère*).
  Attention, plus de contraintes sont fixées sur cette action :
  - la cellule doit exister et appartenir au joueur qui souhaite la scinder,
  - les coordonnées de la position souhaitée doivent être finies,
  - la cellule ne doit pas avoir déjà agi à ce tour,
  - la masse de la cellule *fille* doit être inférieure ou égale à la moitié de la masse de la cellule *mère*,

- la masse de la cellule *mère* doit être supérieure ou égale à deux fois la masse minimale autorisée pour les cellules joueuses (voir paramètres de la partie),
- la masse de la cellule *fille* ne doit pas être trop petite (voir paramètres de la partie).

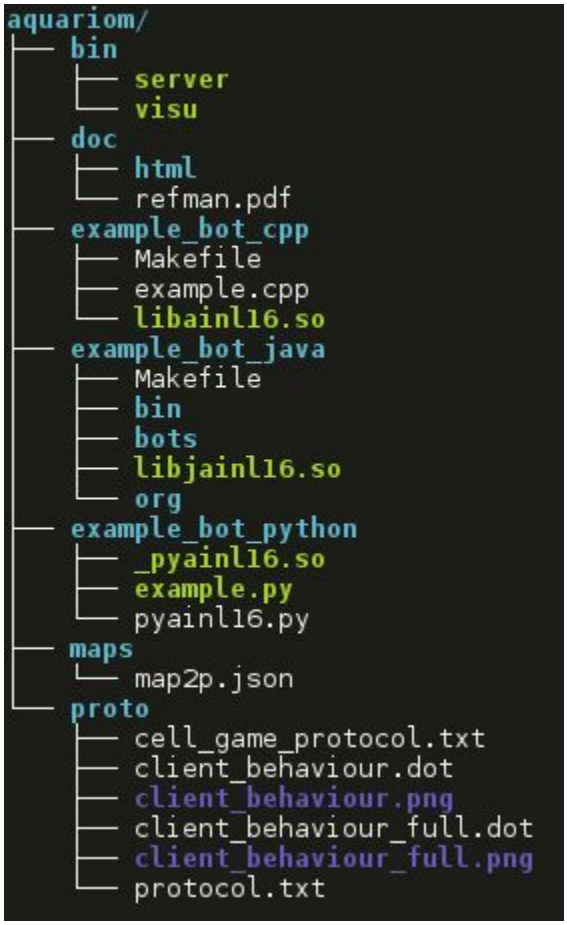
Attention : si la direction choisie est telle que la cellule fille devrait apparaître en dehors de la carte, la cellule fille ne sera pas créée et sa masse sera perdue.

- **Créer un virus** : permet à une cellule de donner naissance à un virus. Les paramètres de cette action sont :
  - Le numéro unique de la cellule devant créer le virus,
  - La position vers laquelle on souhaite créer le virus.

  Le virus aura une masse fixée par les paramètres du jeu. La cellule joueuse va perdre de la masse : la masse du virus qu'elle a créé ainsi qu'un pourcentage de sa masse initiale seront soustraits à sa masse initiale. La position du virus est déterminée de la même manière que celle d'une cellule *fille* dans le cas de la scission : le virus sera tangent à la cellule *mère*, dans la direction choisie par le joueur.

- **Abandonner** : permet à un joueur d'abandonner ses cellules, afin de recommencer avec de nouvelles cellules. Cette action n'a aucun paramètre et aucune contrainte. Toutes les cellules abandonnées deviendront des cellules neutres et resteront à la position qu'elles occupaient lors du tour où le joueur a abandonné. Le joueur se verra attribuer de nouvelles cellules à des positions aléatoires (mais jamais trop près d'autres cellules joueuses). Le nombre de cellules crées ainsi que leur masse sont les mêmes que lors de la création des cellules joueuses initiales (voir paramètres de la partie).

Fichiers fournis



Un ensemble de fichiers vous sont fournis. Ces fichiers comportent le serveur, la visualisation, la bibliothèque client, la documentation de ladite bibliothèque et des exemples de clients joueur dans chaque langage.

La bibliothèque client permet de simplifier l'implémentation d'un joueur communiquant avec le serveur. Elle est écrite en C++ mais des bindings Java et Python sont également générés via SWIG. La documentation de l'API C++ est également fournie.

Une description détaillée du protocole se trouve dans le dossier proto. Cette description devrait être suffisante pour vous permettre d'implémenter vous-mêmes la partie réseau du bot si vous le souhaitez. Cela peut notamment être utile si vous n'aimez pas l'API fournie ou que vous souhaitez coder dans un autre langage tel FORTRAN, MarioLANG ou en Prolog.

Les exemples de code fournis montrent les différents paramètres du jeu ainsi que la manière dont la bibliothèque doit être utilisée. Bien lire les exemples est conseillé puisque cela vous fera gagner du temps *in fine* (notamment en vous évitant de rater certaines informations).