

Sujet: A la guerre comme à la guerre

Contexte

Le sujet a été écrit bien avant le début de la guerre en Ukraine. Finalement, il se retrouve (malheureusement) d'actualité... Vous êtes chef d'une armée et devez gagner la bataille !

Règles du jeu

Le jeu se déroule sous forme de jeu tour par tour. Le plateau est constitué d'une grille carrée de taille qui est donné par le serveur au lancement de la partie. Chaque équipe commence avec un groupe de N unités. A chaque tour, le joueur peut effectuer une action. (**Voir API**)

La subtilité du jeu se trouve dans le système de "combat" entre groupe d'unités. Lorsqu'une équipe déplace son groupe sur un unité adverse, alors il ne reste sur la case que la soustraction du nombre des deux unités.

Ex: L'équipe A déplace son groupe de 8 unités sur une case où 5 unités de l'équipe B sont présents. Il ne reste alors que 3 unités de l'équipe A sur la case (et l'équipe B perd ses unités). Si égalité, les deux groupes disparaissent.

La partie se termine quand il ne reste plus qu'une seule équipe sur la plateau ou que le nombre de tour maximum est atteint. De plus, à chaque tour, des événements aléatoires peuvent avoir lieu (**Voir Events**). Ces événements peuvent être des bonus ou des malus.

Objectif

Ce jeu n'est pas à jouer en tant réel. L'objectif est de coder une IA afin que le jeu puisse jouer tout seul. Le client échange avec le serveur mais jamais le serveur renvoie son état. C'est-à-dire que c'est au client de garder une trace des déplacements de ses adversaires, d'où ont eu lieu les événements, etc...

Il faut donc coder une IA plus intelligente que son adversaire afin de gagner la partie. Bien entendu, lorsque l'on parle d'IA, on ne parle pas de machine learning ou de technique avancée. Il s'agit simplement d'analyse simple de l'état de la partie afin de pouvoir prendre une décision.

Le serveur est fournis, vous devez uniquement coder le client (pas d'interface graphique, juste des échanges d'informations à travers de l'API).

Le serveur est disponible à cette adresse : `Serveur.py`

Attention : Il faut au moins Python3.7 et avoir pygame pour faire fonctionner le serveur

Echange entre le serveur et le client

Le serveur est celui qui s'occupe de relier toutes les équipes entre elles, ainsi que d'être le maître de la partie. Il possède toute l'autorité et vérifie que les actions faites ne sont pas illégales d'après les règles du jeu. Tout passage des règles signifie la défaite de l'équipe.

Afin de pouvoir échanger des informations avec le serveur, il faut utiliser des **Socket** (indication importante). Les échanges se font à travers des commandes définies dans la section API

API

Toutes les commandes se présentent sous la forme suivante : **COMMANDE|PARAM1|PARAM2|...** que ça soit pour les messages à envoyer ou les messages reçu.

Vers serveur

Commande	Arguments	Description
JOIN	NOM	Rejoins la partie avec ce nom d'équipe
MOVE	GROUPE, NOMBRE, DIR	Déplace NOMBRE unités du groupe GROUPE dans la direction DIR (N,S,E,W)
STAY	/	Ne rien faire pendant ce tour

Depuis serveur

Commande	Arguments	Description
NEWGAME	N,K,I,X,Y	Indique qu'une nouvelle partie avec un tableau de taille $n*n$ et k équipes va commencer. L'équipe commence sur la case (x,y) et possède l'id i
NEWTURN	N	Indique à l'équipe que c'est à son tour de jouer. De plus, indique que N nouveaux événements sont à traiter
EVENT	TYPE, X,Y, PARAMS	Description d'un nouveau événement qui a lieu sur la case (x,y) . Voir la section Event pour plus de détails sur les types.
ERROR	MSG	Une erreur a eu lieu, avec sa raison

Les types d'événement

En plus des arguments présent dans ce tableau. Chaque événement est aussi relié à une case (x,y) donné dans les deux premiers arguments.

ID	Arguments	Description
0	ID, GROUPE, NOMBRE, DIR	L'équipe ID a déplacé NOMBRE unité de son groupe GROUPE dans la direction DIR (N, S, E, W) <i>Attention:</i> N'utilise pas les arguments de position (x,y)
1	ID	Le joueur avec l'ID i n'a pas bougé
2	/	Une case qui double le nombre d'unité qui lui marche dessus en premier
3	A,B	Crée un téléporteur qui téléporte relie la case (X,Y) et (A,B)
4	TEMPS	Cette case devient inaccessible pendant TEMPS tours
5	/	Si un joueur marche sur cette case, il ne pourra pas jouer au prochain tour
6	/	Si un joueur marche sur cette case, son prochain adversaire passe un tour
7	/	Si un joueur marche sur cette case, le nombre d'unité est divisé par 2 (ne peut pas tuer)

Note : l'id des nouvelles unités est auto-incrémenté. C'est-à-dire que chaque nouvelle unité aura un id supérieur de 1 au dernier créé même si d'autres unités ont disparu entre temps.

Déroulement d'une partie

Connexion par socket

Envoyer le nom d'équipe

Recevoir le numéro joueur, et de la position de départ

TANT QUE la partie n'est pas terminée

 Attendre NEWTURN

 Traiter les nouveaux événements

 Préparer son coup

 Envoyer son coup (Un seul coup chaque tour)

FIN TANT QUE

Partie hacking

Une partie de hacking est proposée. N'étant pas l'objectif principal de l'événement, cette partie sert juste de bonus. Les fichiers sont disponibles à l'adresse suivante : <http://sotalite.me/hacking.zip>.

Note : Les exercices 2 et 3 ne peuvent fonctionner que Linux x86. Les sources sont disponibles, si besoin il est possible de compiler le programme sur un Windows ou Mac.

Chaque solution rapporte un peu de points suivant ce modèle :

NOM	Description	Points
hack1	Pourquoi mon pdf ne s'ouvre pas ??	25
hack2	Où est ce mot de passe ??	50
hack3	Si seulement je pouvais voir le code !	100
hack4	Je vois tout rouge..	50