

## Vampires contre Loups-Garous

### Introduction

Le but de ce projet est de développer une intelligence artificielle capable de jouer au jeu « Vampires contre Loups-Garous » dont les règles précises sont expliquées ci-après.

Vous devrez vous inspirer des algorithmes vus en cours pour construire votre logiciel. Cependant, rien ne vous empêche d'aller plus loin en implémentant des améliorations à ces algorithmes en vous basant sur la littérature la plus récente sur le sujet.

### Evaluation

La note de TP sera calculée de la façon suivante. Une note sera attribuée en fonction de la qualité du rapport et de la présentation. Le rapport devra être rendu de façon électronique avant la deadline annoncée sur le calendrier du cours. Il se devra d'être synthétique mais d'explicitier les heuristiques et les choix d'implémentation de l'algorithme de jeu uniquement (pas des interfaces graphiques...).

Le jour de la compétition, vous aurez 5 minutes pour présenter à vos adversaires, histoire de faire monter la pression, les tactiques et le tempérament de votre IA. Ensuite, ce même jour, les différentes IA s'affronteront en matchs aller et retour (un coup une équipe sera les vampires, un coup les loups-garous sur une même carte). La compétition est organisée sous forme de pool.

L'IA gagnante se verra rajouter 3 points à sa note de TP, le finaliste malheureux 2 points, et enfin les demi-finalistes malheureux 1 point.

Un serveur sera mis à votre disposition lors de la compétition. Il permettra de chronométrer le jeu, de vérifier le respect des règles et d'afficher le plateau. Si votre logiciel n'arrive pas à communiquer avec le serveur, il sera automatiquement disqualifié. Si le serveur surprend votre logiciel à tricher, vous serez disqualifié et vous perdrez 1 point sur votre note de TP. Dans le cas où vous déclareriez forfait, vous perdrez 2 points sur votre note de module. Enfin, si aucun projet n'est rendu, vous perdrez 4 points sur votre note de module.

### Règles du jeu

Dans un monde lointain, des êtres mortels et ordinaires vivaient une vie paisible. Mais un soir, à la nuit tombée, leurs terres furent le témoin d'une lutte acharnée entre deux espèces : les Vampires et les Loups-Garous. Du coup, les humains s'enfermèrent dans leurs demeures.

Vous incarnerez l'une ou l'autre des espèces : Vampires ou Loups-Garous. Votre but est de devenir l'espèce dominante.

Pour augmenter son effectif, un camp peut faire muter des humains : une morsure de Vampire transforme un humain en Vampire, une griffure de Loup-Garou le transforme en Loup-Garou. Cependant, les humains peuvent se défendre : pour que les Loups-Garous ou les Vampires puissent convertir une chaumière, ils doivent être au moins aussi nombreux que les humains.

De plus, les Loups-Garous et les Vampires peuvent se tuer les uns les autres. Il suffit qu'ils soient au moins 1,5 fois plus nombreux que leurs victimes.

Le jeu se termine quand une des deux espèces s'est éteinte ou lorsque le jour se lève. Le vainqueur est alors l'espèce dominante à ce moment-là.

### Représentation du monde

Le monde est représenté par une grille de dimension  $n \times m$ . Certaines cases sont des chaumières. Il faudra connaître, dans chacune des cases, le nombre d'individus de chaque espèce. L'univers est totalement observable.

### Etat initial

Deux cases sont sélectionnées dans la grille comme case de départ des deux espèces. Dans l'état initial, elles contiennent tous les membres d'une espèce.

### Actions

Les vampires ou les loups garous peuvent se déplacer dans la grille.

A chaque tour, un joueur doit effectuer au moins un déplacement. Cela consiste à déplacer la totalité ou une partie des individus d'une case vers une case adjacente. A chaque tour, le joueur peut déplacer chacun de ses groupes d'une case. Ce nombre de déplacements à effectuer est laissé au choix du groupe en fonction du jeu ou de sa capacité à gérer cette combinatoire. La Figure 1 représente, pour une case donnée, les cases destination possibles.

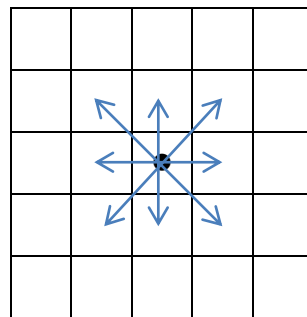


Figure 1: Mouvements possibles

Lors d'un déplacement, si la case cible contient des humains, il faut être au moins aussi nombreux qu'eux pour les convertir. Sinon une bataille a lieu et il en résultera qu'une seule espèce survivra en partie (voir section Bataille aléatoire). Si la case visée contient des monstres, il faut être au moins 1,5 fois plus nombreux qu'eux pour les tuer, sinon, deux cas de figures émergent : soit ils sont 1,5 fois plus nombreux que le joueur, et dans ce cas ils tueront tous leurs assaillants, soit eux-mêmes ne sont pas assez nombreux et il en découlera une bataille aléatoire.

En aucun il est possible de rester statique, c'est-à-dire qu'au moins une action doit être effectuée.

Le tableau ci-dessous reprend les règles concernant les mouvements de manière synthétique :

<b>Règle 1</b>	Il faut au moins un mouvement
<b>Règle 2</b>	On ne peut déplacer que ses propres pions
<b>Règle 3</b>	Il faut avoir assez de pions sur une case pour satisfaire tous les mouvements partant de

cette case

**Règle 4** On ne peut se déplacer que sur une des 8 cases adjacentes**Règle 5** Une case ne peut pas être présente à la fois dans les cibles et les sources**Règle 6** Il faut au moins bouger un pion

### Bataille aléatoire

Dans le cas où le joueur se déplace ou attaque une case contenant des humains ou des ennemis et que le nombre d'individus déplacés est inférieur à 1 fois celui des humains ou 1,5 fois le nombre d'ennemis, une bataille aléatoire commence.

L'issue de cette bataille est déterminée par la loi de probabilité suivante, en fonction des espèces concernées. Soit  $E1$  et  $E2$  les effectifs respectifs des deux espèces,  $E1$  étant l'effectif de l'espèce attaquante :

- Si  $E1=E2$ , alors  $P = 0,5$  ;
- Si  $E1 < E2$ , alors  $P$  est donnée par la droite passant par les points  $(0 ; 0)$  et  $(E2 ; 0,5)$ , donc  $P = \frac{E1}{2 \times E2}$  ;
- Sinon  $P$  est donnée par la droite passant par les points  $(C \times E2, 1)$  et  $(E2 ; 0,5)$  donc  $P = \frac{E1}{E2} - 0,5$ .

La probabilité que  $E1$  remporte la victoire est donnée par  $P$ .

Cependant la lutte a été rude et il y a forcément des pertes. Chaque individu restant est donc soumis à la grande faucheuse :

- Si l'attaquant remporte la victoire, chacun de ses individus a une chance  $P$  de survivre. De plus, s'il a remporté le combat contre des humains, chaque humain a une probabilité  $P$  de survivre et d'être converti en l'espèce de l'attaquant.
- Si l'attaquant perd, chaque individu de l'adversaire a une chance  $1-P$  de survivre.

### Communication avec le serveur

Afin de jouer, il est important de communiquer avec le serveur. Il faudra se connecter en TCP via une socket et une connexion permanente. L'adresse et le port du serveur pouvant changer, il faut prévoir un fichier de configuration.

Une trame est composée d'un ordre codé sur 3 octets, puis d'un ensemble d'octets codant l'information correspondant à l'ordre. L'ordre est donné au format ASCII.

#### Communication serveur → joueur

SET	Permet au serveur de communiquer la grille
HUM	Indique les maisons dans la grille
HME	Indique la case de départ
MAP	Indique le contenu de la grille au départ
UPD	Indique les modifications à apporter à la grille
END	Indique que la partie est terminée
BYE	Indique que le serveur va couper la liaison

**Ordre SET**

La trame SET est constituée comme suit :

3 octets	1 octet	1 octet
SET	n	m

Où n et m sont des entiers non signés codés sur 1 octet (byte) et indiquent respectivement le nombre de lignes et de colonnes de la grille.

Elle permet à votre logiciel d'initialiser la grille du jeu.

**Ordre HUM**

3 octets	1 octet	n x 2 octets
HUM	N	Liste des coordonnées

Où N est le nombre de maisons à placer sur la grille. La liste des coordonnées est donnée sous la forme de deux entiers non signés codés sur 1 octet correspondant respectivement à x, y, les coordonnées d'une maison. L'origine des coordonnées de la grille est la case tout en haut à gauche. Les indices sont donnés à partir de 0.

**Ordre HME**

3 octets	1 octet	1 octet
HME	x	y

Indique la case qui est la demeure de votre camp.

**Ordre UPD**

3 octets	1 octet	n x 5 octets
UPD	N	Liste des changements

N est le nombre de quintuplets qui indiquent les changements dans la carte. Un quintuplet est formé de 5 octets représentant :

- X l'abscisse de la case
- Y l'ordonnée de la case
- Le nombre d'humains
- Le nombre de Vampires
- Le nombre de Loups-Garous.

La mise à jour contient à la fois votre tour précédent et le tour de l'adversaire.

**Ordre MAP**

L'ordre MAP est identique à l'ordre UPD sauf que celui-ci est envoyé une seule fois en début de partie pour compléter la formation de la carte.

**Ordres END et BYE**

Les ordres END et BYE n'ont pas d'argument. A la réception de END, vous pouvez supprimer tout ce qui concernait la partie précédente et vous préparer à une nouvelle partie.

A la réception de BYE, vous devez prendre en compte que le serveur a stoppé la connexion.

**Communication joueur → serveur**

NME	Indique le nom du joueur
-----	--------------------------

MOV	Déplace des individus d'une case à l'autre
-----	--

**Ordre NME**

L'ordre NME est formulé comme suit :

3 octets	1 Octet	T octets
NME	T	Nom du joueur

T représente la taille de la chaîne de caractères représentant le nom du joueur, codée en ASCII.

**Ordre MOV**

L'ordre MOV est formulé comme suit :

3 octets	1 octet	N x 5 octets
MOV	N	Quintuplets

Cette trame indique que vous allez effectuer N déplacements, avec  $0 \leq N \leq \text{nombre de groupes}$ , chaque déplacement étant caractérisé par 5 octets comme suit :

2 octets	1 octet	2 octets
Coordonnées de départ	Nombre d'individus	Coordonnées d'arrivée

Le quintuplet indique que vous allez déplacer un certain nombre d'individus d'une case à une autre.

Attention, chaque case de départ de chacun de vos déplacements doit contenir le bon nombre d'individus à déplacer.

Si des individus humains ou ennemis sont présents dans la case cible, une attaque aura lieu.

**Déroulement d'une partie**

Votre logiciel doit se connecter au serveur. Une fois la connexion établie, il doit envoyer la commande NME. En retour, le serveur vous enverra cette séquence de commandes : SET, HUM, HME, MAP.

Lorsque les deux joueurs sont connectés et informés, ils reçoivent tour à tour la commande UPD. A la réception de la commande UPD, c'est à votre tour de jouer. Vous avez 10 secondes pour envoyer soit la commande MOV, soit la commande ATK. Ainsi le premier joueur recevra d'abord la commande MAP, puis UPD avec 0 changement pour lui donner la main.

A la fin de la partie, vous recevrez la commande END. En cas d'égalité, une nouvelle partie peut débuter. Vous recevrez alors la commande SET, puis le reste de l'initialisation. Si il y a un vainqueur, vous recevrez la commande BYE pour vous dire que vous pouvez mettre fin à l'exécution de votre programme, ou du moins à ses connexions avec le serveur.

## Limites de temps

A chaque tour, le joueur a 2 secondes pour que son ordre soit reçu par le serveur. Au bout de ce temps, si aucune réponse n'est reçue par le serveur, ce dernier donne la main à l'autre joueur.

Si entre temps, l'ordre du joueur précédent arrive, il est simplement effacé.

Une partie a une durée limitée soit en nombre de coups, soit en temps. Si le temps maximum d'une partie est atteint, le serveur vérifie si chaque joueur a eu le même nombre de tours et arrête la partie.

## Utilisation de l'interface graphique

L'interface graphique communique avec les IA clientes au moyen du protocole spécifié précédemment et d'une connexion TCP non sécurisée persistante.

L'interface est paramétrable via son fichier de configuration et des cartes de jeu à fournir.

## Fichier de configuration

Dans le fichier de configuration, vous devez absolument renseigner les champs suivants :

- Ip : l'adresse ip du serveur
- Port : le port qui doit être écouté pour la connexion des clients.

Les autres champs sont des options :

- Trace à True permet d'afficher un log (un fichier sera créé dans le répertoire d'exécution)
- Map permet de spécifier une carte pour le jeu. Si map n'est pas spécifiée, le serveur vous en demandera une.

## Fichiers de description d'une carte

Il est possible de changer la carte utilisée par l'interface. Pour cela, on utilise un fichier xml. Si le fichier est renseigné dans le fichier de configuration, il sera automatiquement chargé. Attention, la syntaxe est sensible à la casse.

La syntaxe du fichier commence par un élément map dont il faut spécifier les deux attributs :

- Rows qui indique le nombre de lignes de la carte
- Columns qui indique le nombre de colonnes.

A l'intérieur de cet élément, on peut trouver les éléments suivants :

- Humans, avec les attributs X, Y et Count qui indiquent respectivement la position de la maison et le nombre d'habitants. Il peut y avoir autant d'éléments Humans que l'on souhaite ;
- Vampires indique l'unique maison des vampires en spécifiant les coordonnées et le nombre
- Werewolves indique l'unique maison des loups-garous en spécifiant les coordonnées et le nombre.

## Normalisation du démarrage de votre logiciel

Afin que le démarrage de vos logiciels puisse être automatisé, il est nécessaire d'établir une convention. Votre logiciel doit être callable en ligne de commande et se fermer (proprement) après une partie.

A la fin de la ligne de commande, deux paramètres seront automatiquement ajoutés :

- XXX.XXX.XXX.XXX est l'adresse IP du serveur sur lequel vous devrez vous connecter ;
- XXXX est le numéro du port sur lequel la communication sera établie.

Par exemple :

Python monexe.py --mesparamspersos 123.123.3.5 6666

Java monexe.java --mesparamspersos 123.123.3.5 6666

Monexe.exe --mesparamspersos 123.123.3.5 6666