

Pour conclure cette partie du cours SCI dédiée à la simulation multi-agents, vous allez réaliser un projet basé sur l'extension IODA de NetLogo. Ce projet est calibré pour un travail en binôme.

1 Contexte

Le jeu Worms est un jeu vidéo créé par *Team17* en 1994 dans lequel des équipes composées de plusieurs « vers » s'affrontent dans un combat sans merci. L'équipe gagnante est celle qui a réussi à éliminer tous les vers adverses. Pour ce faire, diverses armes sont disponibles, chacune dotée de ses propres caractéristiques, éventuellement en quantités limitées.

Vous allez dans ce projet mettre en œuvre les méthodes et les techniques de simulation vues en cours afin de compléter les comportements des agents du jeu, notamment ceux du *joueur*, qu'il soit piloté par un humain ou par l'ordinateur, mais aussi les diverses entités du jeu qui seront toutes représentées par des agents. Le jeu sera basé sur la méthode orientée interactions (IODA) et implémenté au moyen de IODA-NetLogo (version 2.3b).

2 Règles du jeu

Les joueurs jouent chacun à leur tour en pilotant un des vers de leur équipe. Celui-ci peut, **dans une certaine limite de temps**, se déplacer, choisir une arme et l'utiliser. Lorsque le temps est écoulé, ou le ver qui joue blessé, ou tous les effets de l'attaque dissipés, la main passe à un ver de l'équipe suivante. Chaque type d'arme est disponible en une certaine quantité pour chaque équipe. En plus des armes, les vers peuvent effectuer certaines actions spéciales comme se téléporter, sauter en parachute, utiliser un « jet pack » pour s'élever dans les airs, etc.

Les agents initialement présents sont les suivants :

- Les équipes (**teams**) composées d'un certain nombre de vers. Chaque équipe peut être pilotée par un joueur humain ou par l'ordinateur. Ces agents sont invisibles mais ils ont bel et bien un comportement.
- Les vers (**worms**) peuvent se déplacer à gauche ou à droite, ou sauter pour atteindre un point plus élevé devant eux. S'ils ne sont pas sur un mur, ils tombent. Lorsque la chute atteint une vitesse trop importante, ils perdent des points de santé. S'ils tombent dans l'eau ils se noient. Lorsque leurs points de santé arrivent à zéro, ils meurent.
- Chaque ver est lié à un agent cible (**target**) dont le **heading** donne l'angle de tir, ainsi qu'à un agent d'information (**head**) qui affiche ses points de santé et apparaît sous forme d'une flèche lorsque c'est au ver de jouer.
- Les murs (**walls**) sont des obstacles infranchissables. Certains sont indestructibles, alors que les autres peuvent être détruits par une explosion ou un outil approprié.
- Un agent **sea** gère l'océan : il perçoit sur tous les patches d'ordonnée négative et retire les noyés.
- Des agents de décor sont également présents : les nuages (**clouds**) qui bougent au gré du vent et les poissons (**fishes**) qui nagent, nagent, nagent...
- D'autres agents sont prévus mais pas encore dotés de leurs comportements, notamment :
 - des mines (**mines**) qui sont posées par le ver et s'arment quand elles détectent la présence d'un ver ; elles explosent au bout d'un certain temps (mais peuvent également « s'éteindre » sans dégât si elles sont défectueuses) ;
 - des **grenades**, qui sont lancées dans la direction de tir avec une certaine force, et explosent après un compte à rebours ;
 - des **missiles**, qui peuvent être lancés par exemple au moyen d'un bazooka et explosent au contact d'un mur ou d'un ver ;

- le feu **fire** projeté par exemple au lance-flammes ou par un cocktail Molotov, et qui subsiste pendant un certain temps avant de s'éteindre ;
- le souffle de l'explosion (**blasts**) qui se propage dans les cases adjacentes avec une force décroissante : les vers touchés perdent des points de santé selon la force du souffle et peuvent même être projetés violemment, et les murs destructibles peuvent être détruits.

3 Travail demandé

Vous partirez pour ce projet du modèle fourni (<http://www.lifl.fr/~picault/netlogo/worms.zip>) qui contient :

- un ensemble d'interactions « de base » ;
 - une matrice d'interactions et de mise à jour donnant un comportement réactif très simple aux agents ;
 - le modèle IODA-NetLogo correspondant avec l'initialisation de l'environnement et la définition des primitives ;
 - un fichier (en mode texte) correspondant à une configuration initiale de l'environnement.
1. **Implémenter des armes de base.** La première étape de votre travail consistera à donner aux vers des armes et compétences de base :
 - déposer une mine ;
 - lancer une grenade ;
 - tirer un missile au bazooka ;
 - se téléporter.
 2. **Compléter les comportements.** Ensuite, vous devrez compléter les comportements des agents de façon à respecter toutes les règles mentionnées ci-dessus et non implémentées dans la version de base (notamment un chronomètre : cf. **timer** en NetLogo). Les chutes doivent être réalistes. Il faut également prévoir l'abandon du jeu lorsque la situation est bloquée. Enfin, lorsqu'un niveau est terminé, on souhaiterait pouvoir passer automatiquement au niveau suivant.
 3. **Automatiser les joueurs.** L'étape suivante consiste à programmer des vers dont le comportement peut être piloté par l'ordinateur, de façon à ce qu'il soit capable au moins dans des situations simples d'affronter un ver adverse sans se mettre trop en danger.
 4. **Ajout d'agents, d'armes, d'événements.** Vous pourrez enfin étendre le jeu en vous inspirant du jeu d'origine ou en ajoutant vos propres fonctionnalités, par exemple :
 - introduire de nouvelles armes ou des effets comme le tremblement de terre ou la montée des eaux ;
 - prendre en compte l'effet du vent dans la trajectoire des missiles ;
 - faire tomber du ciel de temps à autre des caisses de munitions ou d'aide médicale ;
 - rendre les murs sensibles à la pesanteur (ils tombent s'ils ne sont pas soutenus par d'autres murs)...

La conception de ces étapes doit être incrémentale. Elles doivent être intégrées dans un seul programme NetLogo, quitte à prévoir divers fichiers de définition des interactions et de la matrice (cf. par exemple le modèle **colors** dans la partie **tutorials** de IODA-NetLogo).

4 Documents à rendre

L'ensemble du projet est à rendre par mail à sebastien.picault@univ-lille1.fr, au plus tard le **vendredi 19 décembre 2014**, sous la forme d'une archive zip contenant :

- le fichier NetLogo correspondant au(x) modèle(s) réalisé(s), accompagné des fichiers d'interactions, de matrices, et de niveaux de jeu ;
- une description des solutions que vous avez proposées et de leurs résultats, **en complétant l'onglet « Information »** du fichier NetLogo (voir notamment *Model Library* → *Code Examples* → *Info Tab Example*).