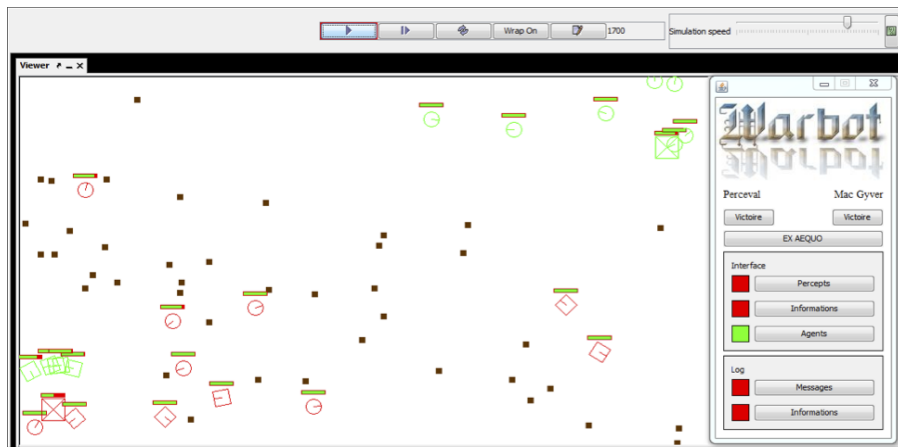


Resp: Jacques Ferber,

TP 7 - Warbot - communications et FSM



proposé par Jacques Ferber - modifs nov-2019



Dans ce TP nous allons développer la manière de réaliser des communications sous Warbot. Voir le cours sur les communications.

Dans Warbot, il y a deux manières d'envoyer des messages, soit directement à l'agent quand on connaît son id, soit à tous les agents d'un certain groupe: la même équipe, d'un certain type, ou tout simplement tous les agents jouant un certain rôle dans un certain groupe (regarder la documentation de Warbot à ce sujet).

Remarques:

- la fonction `getMessages()` retourne une liste de messages. Ceux-ci sont supprimés de la boîte aux lettres.

1. Implémentation de messages

1.a Appel à l'aide de la base

Si la base est attaquée, elle demande à tous les lanceurs de missiles de revenir à la base. Implémentez le système de communication de manière simple: tous les agents retournent à la base.

Cela signifie que, automatiquement, les agents vont se tourner vers l'envoi du message de demande d'aide.

1.b Appel à l'aide de la base avec décision des receveurs

Si la base est attaquée, elle demande à tous les lanceurs de missiles de revenir à la base. Mais maintenant, les lanceurs de missiles ne répondent que s'ils sont à une distance inférieure à un certain rayon, les autres continuent leur vie.

1.c Demande d'attaque venant des explorateurs.

Suite du précédent. Si un robot exploreur voit la base ennemie, il demande à tous les lanceurs de missiles de venir attaquer la base.

Observez ce qui se passe et ensuite modifier le code pour que seuls les lanceurs de missiles qui sont "près" de la base ennemie (inférieurs à une certaine distance) viennent la détruire.

2. Aveugle et paralytique

On veut implémenter le principe de l'aveugle et le paralytique ou un Explorateur dispose de la "vision" tandis que des tireurs (Heavy ou MissileLauncher) disposent de la possibilité d'action. Ces derniers restent à distance sans voir leur objectif tout en lui tirant dessus, leur objectif étant donné par un Explorateur.

Pour cela, dès qu'un explorateur trouve la base, il demande à ce que des tireurs viennent l'aider (comme dans l'exemple précédent), puis il s'arrête se contentant de donner la position de la base, qui est ensuite transposée par les tireurs, en utilisant le triangle trigonométrique (besoin de rafraichir votre trigo? vous utiliserez les principes de trigo vus en cours).

Pour Java sur <https://github.com/rascal222/TER-Warbot-2016/tree/master/src/main/java/edu/warbot/tools/geometry> (les classes se trouvent dans l'archive Java pas besoin de recompiler quoi que ce soit)

Idem pour Python: [Trigo.py](#) (se trouve dans l'archive de Warbot)

3. FSM

Dans cette partie du TP, on vous demande d'écrire le comportement des agents sous la forme de FSM avec réflexes. On a vu la définition générale des FSM dans [le cours sur les architectures réactives](#).

Vous trouverez une [FSM pour Python dans le ce fichier](#) et ici un [exemple de FSM en Java](#).

dans laquelle un explorateur se balade, récupère de la nourriture s'il en voit. Il remplit totalement son sac avant de rentrer à la base. S'il rencontre la base ennemie, il en informe immédiatement la base. (comportement très simple, le travail consiste à programmer la FSM).

a) A partir de ce modèle, créez des attaquants (Heavy ou MissileLaunchers) qui cherchent la base ennemie et attaquent s'ils l'ont trouvé où s'ils ont reçu l'information du lieu de la base par l'intermédiaire d'un explorateur.

b) modifiez la FSM de l'explorateur pour que votre explorateur puisse diriger une attaque s'il rencontre une base ennemie. A ce moment là, il ne recherche plus de nourriture et reste auprès de la base. Ecrivez auparavant la FSM graphiquement sur papier avant de la programmer!