

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 Boulder Dash est un jeu vidéo créé en 1984 dans lequel un personnage doit collecter des diamants dans une grotte, en évitant les avalanches de pierres, les créatures monstrueuses et les explosions. Lorsque tous les diamants ont été ramassés, le joueur doit gagner la sortie. 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 2.3, pour NetLogo 5).

2 Règles du jeu

Les agents initialement présents sont les suivants :

- Le personnage (**hero**) doit collecter les diamants : en se déplaçant, il creuse la terre et crée des espaces vides. Il peut également pousser des pierres. Le héros n'est pas affecté par la pesanteur.
- Les murs (**walls**) sont des obstacles infranchissables. Certains sont indestructibles, alors que d'autres peuvent être détruits par une explosion.
- La sortie (**door**) a l'apparence d'un mur indestructible tant que tous les diamants n'ont pas été ramassés, après quoi elle se révèle et devient franchissable (elle permet de terminer le niveau).
- Les pierres (**rocks**) sont immobiles tant qu'elles reposent sur un support, c'est-à-dire n'importe quel agent (y compris le personnage), mais dès qu'il y a du vide sous une pierre, celle-ci commence à tomber, et continue verticalement jusqu'à ce qu'il y ait à nouveau quelque chose sous elle. Si une pierre tombe sur un monstre ou sur le héros, il se produit une explosion qui, soit détruit tout (sauf les murs indestructibles), soit crée de nouveaux diamants, et ce dans un carré de 3×3 cases autour de l'impact. Si elles tombent au contraire sur d'autres pierres, des murs, la sortie ou de la terre, elles s'arrêtent. Par ailleurs, les pierres peuvent également rouler avant de tomber, si elles sont sur un support mais que la case immédiatement adjacente et celle située dessous sont vides. Enfin, une pierre *immobile* peut être poussée par le joueur vers la droite ou vers la gauche si l'emplacement suivant est vide.
- Les diamants (**diamonds**) sont collectés par le joueur (il suffit de se positionner dessus). À part cela, ils ont le même comportement que les pierres.
- La terre (**dirt**) est très largement présente dans l'environnement. Elle est creusée au fur et à mesure des déplacements du joueur, ce qui peut entraîner des chutes de pierres ou de diamants, mais aussi libérer des monstres.
- Les monstres (**monsters**) effectuent un déplacement simple (par exemple un suivi de mur). S'ils touchent le joueur, se produit une explosion comme indiqué. Les monstres ne sont pas affectés par la pesanteur, mais peuvent être détruits par la chute d'une pierre ou d'un diamant.
- Le souffle de l'explosion (**blast**) se propage dans les cases adjacentes avec une force décroissante. Certaines explosions ont pour conséquence de faire le vide (tous les agents sont tués sauf les murs indestructibles et les portes), tandis que d'autres laissent derrière elles des diamants.

3 Travail demandé

Vous partirez pour ce projet du modèle fourni (<http://cristal.univ-lille.fr/~picault/netlogo/boulder.zip>) qui contient :

- le présent document ;
- un ensemble d'interactions « de base » ;
- une matrice d'interactions et de mise à jour donnant un comportement réactif très simple (et incomplet) aux agents ;
- le modèle IODA-NetLogo correspondant, avec l'initialisation de l'environnement et la définition des primitives ;
- des fichiers (en mode texte) correspondant à divers niveaux de jeu donnés (configuration initiale de l'environnement) fournis à titre d'exemple.

1. **Compléter les comportements.** La première étape de votre travail consistera à compléter les comportements des agents de façon à respecter toutes les règles ci-dessus (en particulier, les éléments soulignés n'ont pas été implémentés dans la version fournie). Il faut prévoir également l'abandon du niveau lorsque la situation est bloquée. Enfin, lorsqu'un niveau est terminé, on souhaiterait pouvoir passer automatiquement au niveau suivant.
2. **Étendre le modèle de jeu.** Votre deuxième tâche consiste à ajouter d'autres agents avec leurs comportements, parmi ceux présents dans le jeu d'origine, par exemples :
 - les « amibes » (**amoeba**) sont immobiles mais croissent à une vitesse aléatoire (dépendante du niveau de difficulté) en s'étendant au détriment des espaces vides ou de la terre dans les cases adjacentes. Elles se comportent comme un obstacle pour le joueur. À leur contact, les monstres explosent. Une grosse quantité d'amibes peut soudainement cristalliser en pierres, et une colonie dont le développement est bloqué (plus d'espace pour s'étendre) peut cristalliser en diamants.
 - Les murs magiques (**magicwalls**) sont perméables aux pierres qui tombent et les transforment en diamants.
 - La dynamite (**explosives**) peut être déposée par le joueur pour provoquer une explosion. D'autres idées peuvent être trouvées sur http://en.wikipedia.org/wiki/Boulder_Dash. Vous veillerez tout particulièrement à fournir pour chaque nouvelle situation un fichier niveau de jeu permettant d'illustrer le fonctionnement de vos ajouts (voyez-le comme une sorte de tutoriel).
3. **Automatiser le héros.** La dernière étape consiste à construire un mode « IA » dans lequel le joueur est contrôlé par l'ordinateur, de façon à ce qu'il soit capable au moins dans des situations simples de collecter tout seul les diamants en évitant les dangers, et de détecter les situations sans issue.

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 18 décembre 2015**, sous la forme d'une archive zip contenant :

- le fichier NetLogo correspondant aux modèles réalisés, accompagné des fichiers d'interactions, de matrices, et de niveaux de jeu ;
- une description des solutions que vous avez proposées et de leurs résultats, ainsi qu'un bref manuel utilisateur, **en complétant l'onglet « Info »** du fichier NetLogo (voir notamment *Model Library* → *Code Examples* → *Info Tab Example*).