

Node Runner

DIRO ~ Hackathon 2018

1 Défi

Au cours de la fin de semaine, vous devrez programmer une intelligence artificielle pour un clone maison du jeu *Lode Runner*, un jeu de plateforme datant des années 80.

Pour citer Wikipédia :

Le joueur incarne un personnage évoluant dans un décor en deux dimensions constitué d'échelles, de barres de franchissement, de murs et de passerelles de briques et de pierre. Le but du joueur est de ramasser les lingots disséminés dans le décor (sur les passerelles, en haut des échelles ou suspendus dans le vide) tout en évitant des gardes qui essayent de l'attraper. Une fois que tous les lingots sont récupérés, il doit s'échapper en rejoignant le sommet du décor (éventuellement une échelle apparait pour l'aider lorsque tous les lingots sont récupérés) pour passer au tableau suivant.

2 Directives

La première chose à faire est de vous choisir une clé secrète. Cette clé secrète sera utilisée comme nom de la salle dans laquelle votre partie se déroulera. N'importe qui ayant cette clé pourra donc regarder vos parties.

3 Tester le jeu

Projet : NodeRunnerKeyboard

Fichier : NodeRunnerKeyboard.java

Ouvrez le projet dans le dossier `NodeRunnerKeyboard` fourni dans votre IDE préféré (Netbeans, Eclipse, ...) et modifiez le fichier `NodeRunnerKeyboard.java` : ajoutez votre clé secrète à la ligne 14.

Une fois la clé ajoutée, rendez-vous sur <http://138.197.128.148:7331/> dans votre navigateur web et entrez la clé secrète dans la case prévue à cet effet. C'est sur cette page web que vous verrez vos parties se dérouler.

Vous pouvez lancer le jeu `NodeRunnerKeyboard` : vous devriez voir une fenêtre apparaître qui vous permettra de vous déplacer avec W, A, S, D, creuser à gauche avec Z et à droite avec C.

Jouez un peu au jeu pour vous familiariser avec les règles et les niveaux.

4 Intelligence Artificielle

Projet : `java`

Fichier : `Runner.java`

Ouvrez le projet dans le dossier `java` fourni. Vos modifications devront principalement se faire dans `Runner.java`, mais vous pouvez vous créer d'autres fichiers au besoin.

Ajoutez votre clé secrète et lancez le client pour tester.

Le comportement par défaut de l'IA est de se déplacer dans une direction au hasard... Vous devrez changer ce comportement si vous voulez avoir une chance de gagner !

Notez : si vous préférez coder en `JavaScript` pour la compétition, une base de code utilisable avec une version récente de Node.js vous est fournie dans `javascript`. Le fichier `runner.js` contient sensiblement la même structure de code que celle décrite dans les sections suivantes.

5 Déroulement du jeu

5.1 Début de partie

Au début de la partie, la méthode `start` est appelée. Vous recevez en paramètre dans cette fonction une représentation textuelle de la grille.

Chaque case de la grille peut contenir l'un des éléments suivants :

- Case vide (`espace`) : rien de spécial, on peut se déplacer dedans
- Bloc de pierre (`@`) : on peut marcher dessus
- Bloc de brique (`#`) : on peut marcher dessus et *creuser dedans* au besoin pour créer un chemin
- Corde (`-`) : on peut se déplacer latéralement dessus (gauche/droite), ou se laisser tomber vers le bas
- Échelle (`H`) : on peut monter, descendre, ou aller de côté pour se laisser tomber en bas de l'échelle. Notez qu'il est possible de marcher sur le dessus d'une échelle.
- Lingot d'or (`$`) : on doit ramasser tous les lingots d'or avant de pouvoir aller au niveau suivant
- Runner (`&`) : la case où le Runner se trouve
- Sortie (`S`) : la porte de sortie pour aller au prochain niveau

Notez qu'à partir d'un certain niveau, certains blocs de brique (`#`) sont en réalité des pièges... Lorsqu'on marche dessus, on tombe comme s'il n'y avait pas de bloc.

5.2 À chaque tour du jeu

À chaque tour de jeu, la méthode `next` est appelée. Cette fonction reçoit en paramètre la position actuelle (x, y) du Runner.

Notez que y correspond au numéro de ligne dans la grille et x correspond au numéro de colonne. Le point $(x, y) = (0, 0)$ correspond donc au point en haut à gauche de la grille de jeu.

On peut donc accéder à ce qui se trouve dans la grille à la position du Runner en faisant :

```
char element = grid[y].charAt(x);
```

La fonction `next` est la fonction principale de votre intelligence artificielle : c'est là que vous devrez décider du mouvement vous souhaitez faire à chaque étape du jeu.

La fonction retourne un `Move()` dont le premier paramètre est le type de mouvement (`Event.MOVE` pour se déplacer ou `Event.DIG` pour creuser) et le second paramètre est la direction (`Direction.UP`, `Direction.DOWN`, `Direction.LEFT` ou `Direction.RIGHT`).

Notez qu'il est impossible de creuser en haut ou en bas : on peut seulement creuser à gauche et à droite.

Par exemple, une intelligence artificielle qui essaierait d'aller toujours à droite serait codée comme suit :

```
public Move next(int x, int y) {  
    return new Move(Event.MOVE, Direction.LEFT)  
}
```

5.3 Creuser dans un bloc

Lorsqu'on creuse dans un bloc, le bloc disparaît pendant *8 tours*, puis réapparaît.

5.4 Fin de partie

Le Runner commence avec 5 vies. La partie se termine lorsque le Runner n'a plus de vies.

Le Runner perd une vie si un bloc creusé se reconstruit sur lui.

6 Compétition

La compétition finale se déroulera en temps réel, sous vos yeux ébahis. L'équipe dont l'IA réussira à se rendre le plus loin remportera la compétition.

Notez qu'en cas d'égalité, l'équipe dont l'IA se sera rendue le plus rapidement au dernier niveau atteint sera déclarée gagnante.

Bonne compétition ! :v)