

[P22] Projet IA02 : *Helltaker*

À propos du document /changelog

- **Version du sujet** : 1.2.0
- **Auteurs** : Sylvain Lagrue et Khaled Belahcène

Changelog

- v1.0.0 : version initiale
- v1.1.0 : ajout du lien vers le site du développeur
- v1.2.0 : ajouts des liens du gitlab et de l'inscription des trinômes, ainsi que celui de sokoridor

Description du sujet

L'objectif de ce projet est de créer différentes « IA » capables de jouer à la partie puzzle du jeu **Helltaker**. Ce jeu, de la famille de *Sokoban*, est gratuit et disponible sous MacOS, Linux et Windows. L'objectif est d'implémenter différentes méthodes, de les décrire et de les comparer expérimentalement dans un petit rapport final.

Travail demandé

Écriture du problème en STRIPS et en Prolog

Vous devrez donner une représentation en STRIPS (ou l'un de ses dérivés) du problème. Mettre en exergue les principales différences avec *Sokoban*.

Bonus : reprendre le code *Prolog* de Sokoban vu en cours et l'adapter pour Helltaker.

Implémentation

Vous devrez implémenter 2 méthodes différentes parmi les 3 suivantes.

1. Recherche dans un espace d'état (Python)
2. SATPLAN (réécriture en SAT du problème de planification)
3. ASPPLAN (réécriture en ASP du problème de planification)

Rapport

Le rapport, de 10 pages maximum, suivra le plan suivant. Vous pouvez néanmoins l'enrichir à votre guise.

1. Introduction
2. Préliminaires

1. Présentation des règles du jeu
2. Le problème en STRIPS
3. Méthode 1
 1. Représentation du problème
 2. Choix d'implémentation et structures de données
 3. Expérimentations pratiques
4. Méthode 2
 1. Représentation du problème
 2. Choix d'implémentation et structures de données
 3. Expérimentations pratiques
5. Comparaison expérimentale des 2 méthodes

Formation des groupes

- Les groupes sont composés de 3 personnes issues du même groupe de TP
- Si un TP comprend 13 personnes, un groupe de 4 pourra être créé, mais dans ce cas, 3 méthodes devront être implémentées

Vous pouvez inscrire vos trinômes ici : <https://moodle.utc.fr/mod/choicegroup/view.php?id=145491>

Les inscriptions devront être faites au plus tard le **lundi 30 mai à minuit**. Vous devez é

Rendu attendu

- 1 programme Prolog de modélisation
- 2 programmes implémentant les méthodes choisies et respectant les formats d'entrée/sortie imposés
- 1 rapport
- **Date limite de rendu (code + rapport) : mercredi 15 juin (23h59)**

Vos programmes seront testés de façon automatique sur une machine dédiée. Les fichiers seront automatiquement donnés à votre programme et le plan généré sera testé. L'outil de lecture de fichiers et la fonction `main` de votre programme vous seront donnés en python.

- Réponse attendue du programme sur la sortie standard (fin de ligne `\n`) : `hhbgdbbgh`

Ceci correspond à une simplification des actions (`h` = haut, `b` = bas, `d` = droite, `g` = gauche)

Évaluation

Les points suivants seront pris en compte pour l'évaluation :

- Solutions choisies et élégance de celles-ci
- Qualité du code et utilisation des outils dédiés (`black`, `mypy`, `pylint`, etc.)
- Efficacité du code (les programmes seront lancés automatiquement sur une machine dédiée)

Ressources

Quelques liens

- Helltaker sur wikipedia : <https://fr.wikipedia.org/wiki/Helltaker>
- Helltaker sur steam: <https://store.steampowered.com/app/1289310/Helltaker/>
- Télécharger Helltaker sur le site du développeur sans avoir à installer steam : <https://vanripper.itch.io/helltaker>
- Helltaker sur fandom : https://helltaker.fandom.com/wiki/Helltaker_Wiki
- Description SATPLAN : <https://en.wikipedia.org/wiki/Satplan>
- Description ASPPLAN : un exemple pour une version simplifiée de Sokoban <https://moodle.utc.fr/mod/folder/view.php?id=145391>

Code fourni

Le gitlab du projet https://gitlab.utc.fr/lagruesy/helltaker_ia02 fournit :

- un lecteur de fichier générique et main en python
- quelques cartes (dont des mini-cartes de test)