

Rapport des projets PATIA

Sokoban :

Pour compiler le projet sokoban, placez-vous à la racine du projet (Socoban/) et tapez la commande : `mvn compile`

Pour l'exécuter taper la commande :

```
java --add-opens java.base/java.lang=ALL-UNNAMED \  
-server -Xms2048m -Xmx2048m \  
-cp "$(mvn dependency:build-classpath -Dmdep.outputFile=/dev/stdout \  
-q):target/test-classes/:target/classes" \  
sokoban.SokobanMain
```

Arvand statique et dynamique :

pour compiler le projet Arvand, placez-vous à la racine du projet (ASP/) et tapez la commande :

```
javac -d classes -cp lib/pddl4j-4.0.0.jar src/fr/uga/pddl4j/tutorial/asp/Node.java  
src/fr/uga/pddl4j/tutorial/asp/*.java
```

Pour l'exécuter, tapez la commande :

```
java -cp classes:lib/pddl4j-4.0.0.jar fr.uga.pddl4j.tutorial.asp.Arvand [domaine.pddl]  
[probleme.pddl] -e FAST_FORWARD -w 1.2 -t 1000
```

veuillez remplacer `Arvand` par `ArvandStatique` si vous voulez lancer la version statique du programme et `ArvandDynamique` si vous voulez lancer la version dynamique du programme

Validateurs

Pour valider les plans générés par Arvand statique (ARS) ou Arvand dynamique (ARD) sous un système d'exploitation linux (L), mac (M) ou windows (W), tapez la commande : `python3 validateur.py [ARS, ARD] [Domaine.pddl] [Probleme.pddl] [L, M, W]`

par exemple : `python3 validateur.py ARD pddl/blocksworld/domain.pddl
pddl/blocksworld/p01.pddl L`

Etudes de performance :

Pour visualiser les plots qui correspondent à chaque classe de problème en fonction de la taille des plans générés et du temps d'exécution, tapez la commande : `python3 graphe.py` depuis la racine du projet (ASP/). cette commande permet de lancer le programme graphe qui dessine les graphes qui correspondent à chaque classe de problèmes avec un TimeOut égale à 180s (3min)

Les résultats de l'exécution du programme graphe.py :





