Fil Rouge - Le Monde des Blocs

Unité d'enseignement : Aide à la Décision et Intelligence Artificielle

Réalisé par :
Léna Rezgui
Mohamed Yassine Lamairi

Enseignant(e) : Amal Mahboubi

21 Novembre 2025

Table des matières

1 Introduction

Ce document a pour objectif de détailler les démonstrations et de justifier les choix d'implémentation effectués pour le projet **"Le monde des Blocs"**. Ce projet, développé en **Java**, constitue le fil rouge de l'unité d'enseignement *Aide à la Décision et Intelligence Artificielle* de notre troisième année de Licence Informatique.

2 Description des démonstrations réalisées

2.1 MainModelling.java (Exercice 5)

L'objectif de cette démonstration est d'évaluer et comparer différents planificateurs en partant d'un état initial vers un but défini.

- Initialisation d'un monde constitué de 4 blocs et 3 piles.
- Récupération des contraintes de base : Unicité (*Uniqueness*), *Fixed*, et *Free*.
- Ajout des contraintes de régularité et de croissance.
- Test des contraintes sur un état valide puis sur un état invalide.

2.2 MainPlanning.java (Exercice 8)

L'objectif est de vérifier la cohérence de la modélisation du *blockworld* ainsi que la validité des contraintes introduites.

- Initialisation (4 blocs, 3 piles) et définition d'un état initial et d'un but.
- Instanciation des différents planeurs et de l'heuristique associée.
- Test d'efficacité de chaque planeur (nombre de nœuds explorés, temps écoulé, etc.).

2.3 MainCSP.java (Exercice 9-10)

Résolution de problèmes de satisfaction de contraintes (CSP) en utilisant différentes contraintes et solveurs.

- Initialisation d'un monde constitué de 6 blocs et 4 piles.
- Ajout des contraintes de croissance et de régularité.
- Test des différents solveurs avec les contraintes générées et affichage graphique de la solution.

2.4 MainMining.java (Exercice 12)

Extraction automatique de motifs fréquents et de règles d'association à partir d'une base de données d'états du monde des blocs.

- Initialisation d'un monde de 5 blocs et 5 piles.
- Test de l'algorithme *Apriori* pour l'extraction de motifs.
- Génération et affichage des règles d'association (fréquence et confiance).

2.5 BlocksWorldDisplay.java

Cette classe permet l'affichage graphique implementant ainsi la bibliothèque donnée en partie 5.

3 Choix d'implémentation

3.1 Variables et contraintes (package modelling)

Exercice	Classe	Rôle et Justification
Exercice 1	`WorldConfig.java`	Configuration du monde ; génère les variables via `getVariables()`.
Exercice 2	`BlockWorld.java`	Génère les contraintes fondamentales (*Uniqueness, Fixed, Free*) via `WorldConfig`.
Exercice 3	`ImplicationGenerator.java`	Génère les contraintes de régularité sous forme d'implications logiques.
Exercice 4	`CroissanceGenerator.java`	Implémente l'interface `Constraint` pour définir la croissance sur les blocs.

3.2 Planification (package planning)

Exercice	Classe	Rôle et Justification
Exercice 6	`BlocksWorldPlanner.java`	Génère l'ensemble des actions de type `BasicAction` pour une configuration donnée.
Exercice 7	`MisplacedGoalHeuristic.java` `BlockFreeingHeuristic.java`	Heuristiques permettant d'estimer le coût d'un état pour optimiser la recherche de solutions.

3.3 Extraction de connaissances (package datamining)

Exercice	Classe	Rôle et Justification
Exercice 11	`BlocksWorldConverter.java`	Permet la conversion d'un état complexe en une transaction binaire traitable par les algorithmes de minage.

4 Conclusion

Ce projet a permis de mettre en œuvre les concepts fondamentaux de l'IA vus en cours. De la modélisation par contraintes à la planification algorithmique, en passant par l'extraction de connaissances, nous avons abouti à une solution robuste capable de manipuler le monde des blocs de manière efficace et intelligente.