

Résolution de niveaux du Sokoban

PoulpoGaz, darth-mole

15 mai 2023

Candidat n° 012345

Le jeu du Sokoban

Principe de résolution

Réduction de l'espace de recherche

- Analyse statique

- Analyse dynamique

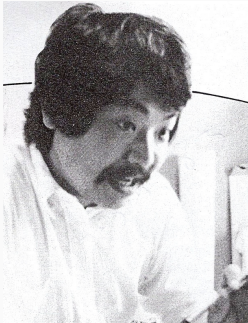
Recherche dirigée par une heuristique

Optimisations

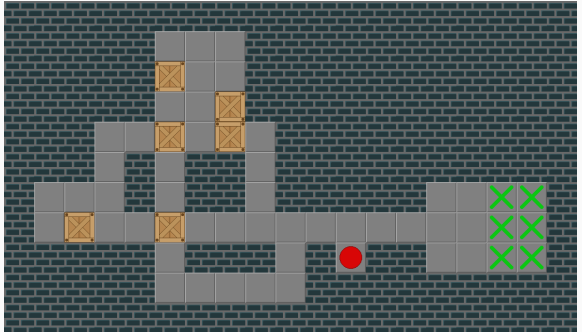
Résultats

Le jeu du Sokoban

Le jeu du Sokoban



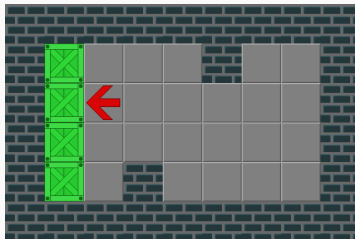
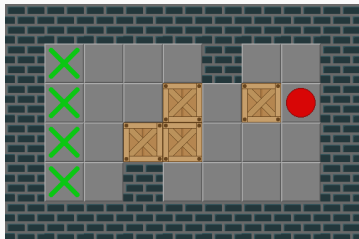
Hiroyuki Imabayashi

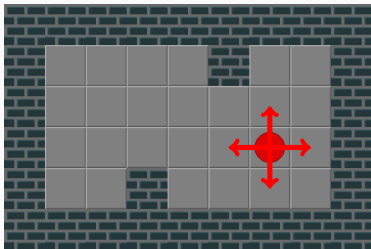


XSokoban

But du jeu

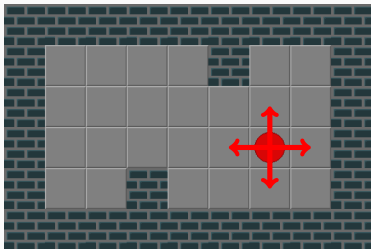
Déplacements



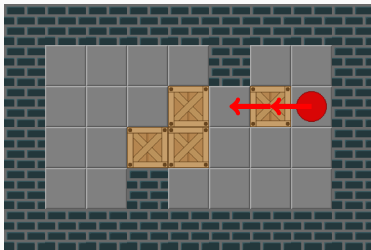


Déplacements autorisés

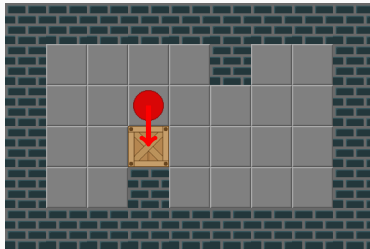
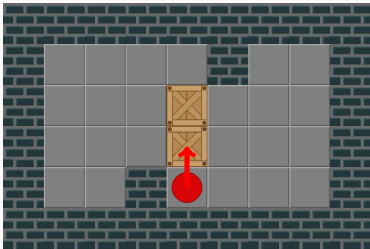
Règles



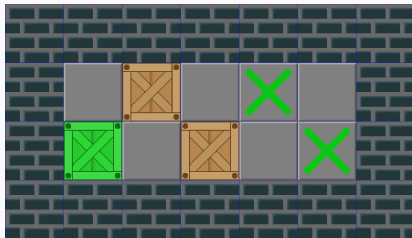
Déplacements autorisés



Règles



Tuiles



Mur



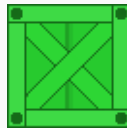
Sol



Caisse

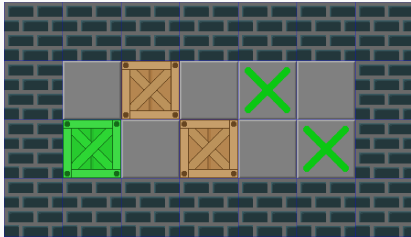


Cible



Caisse sur une cible

Tuiles



Mur



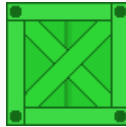
Sol



Caisse



Cible



Caisse sur une cible

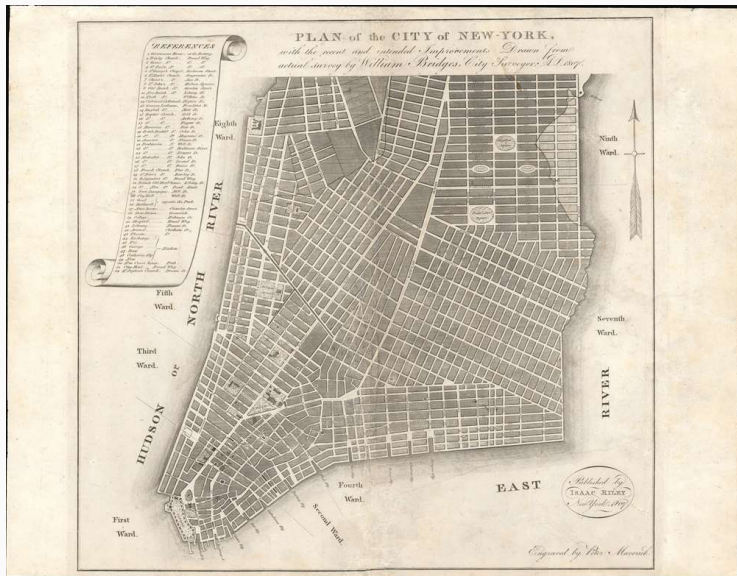
```
enum Tile {WALL, FLOOR, CRATE, TARGET, CRATE_ON_TARGET};
```

```
Tile[] [] map = new Tile[height][width];
```

Lien avec le thème de l'année

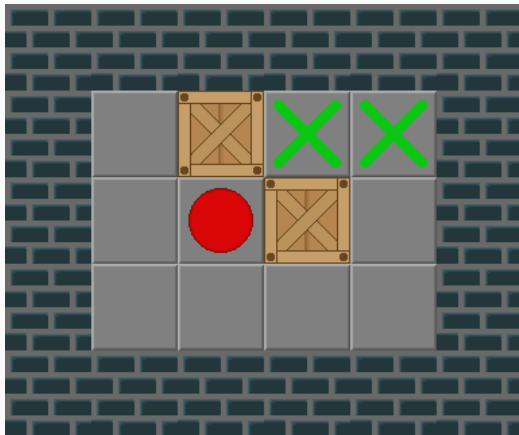


Lien avec le thème de l'année

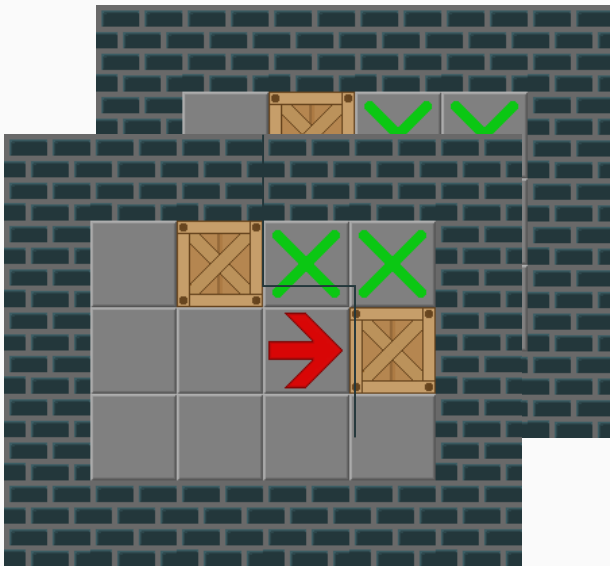


Principe de résolution

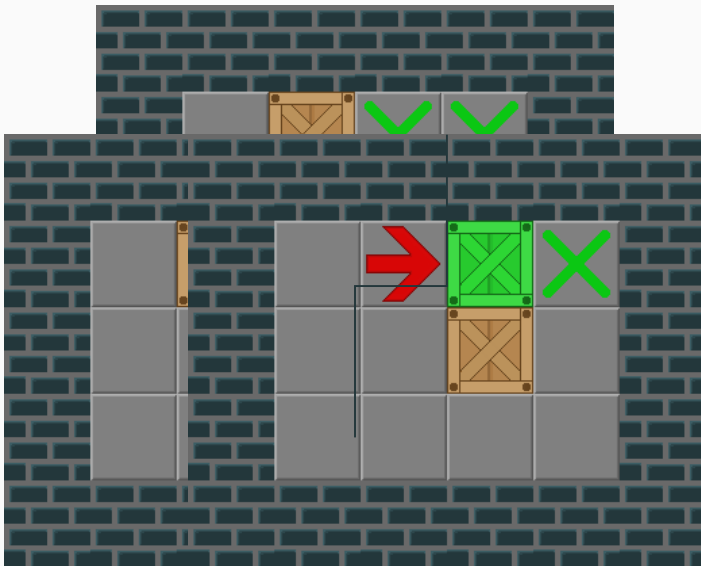
Arbre des états



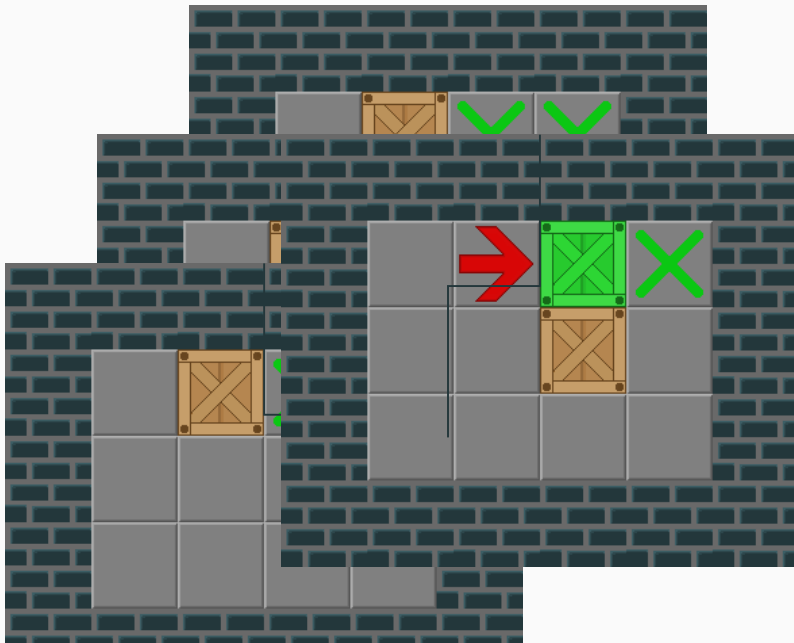
Arbre des états



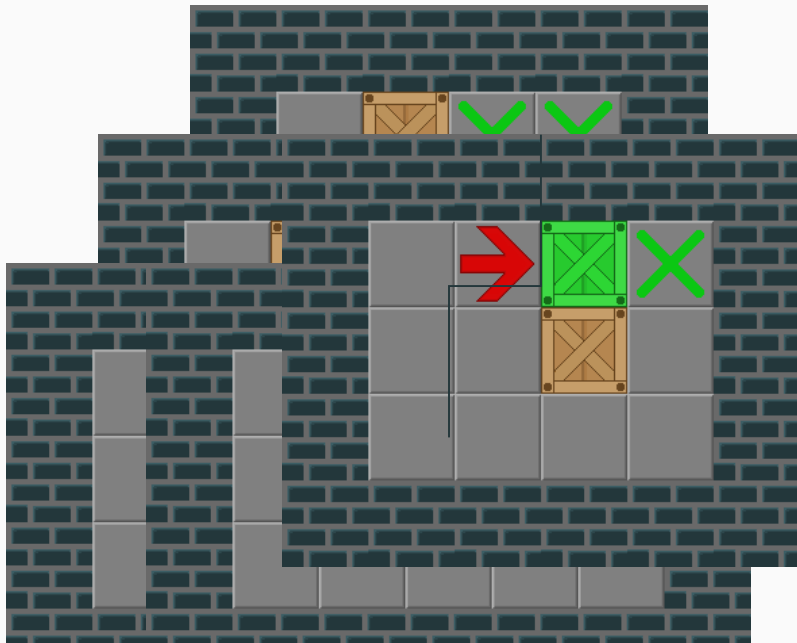
Arbre des états



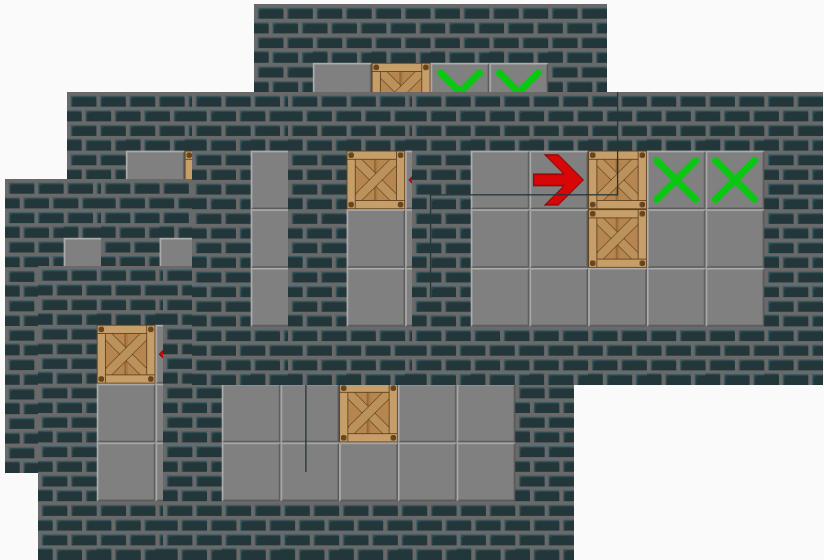
Arbre des états



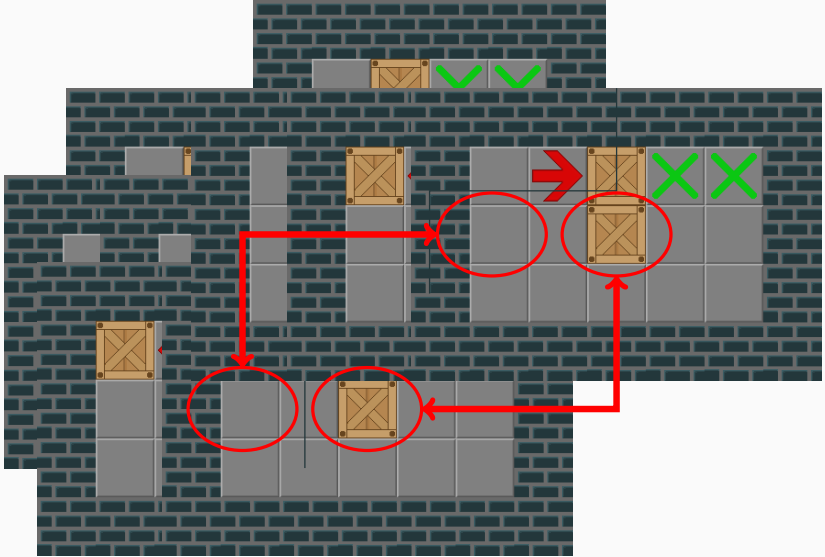
Arbre des états



Exemple développé



Un graphe vu comme un arbre

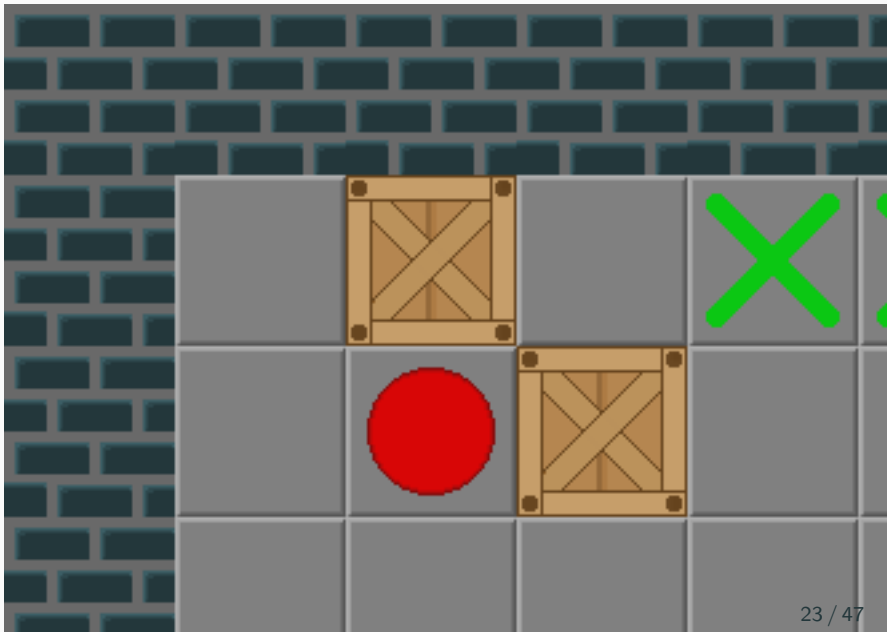


Réduction de l'espace de recherche

Réduction de l'espace de recherche

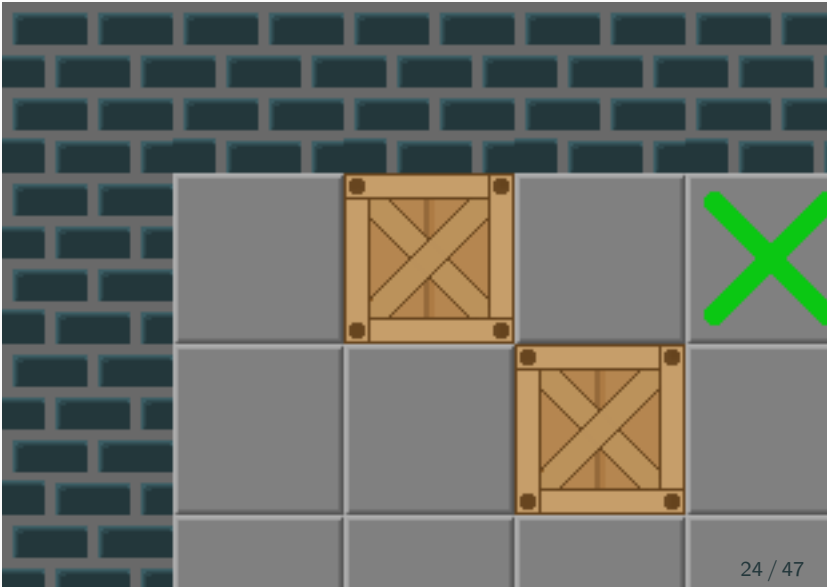
Analyse statique

Détection des positions mortes (*dead positions*)



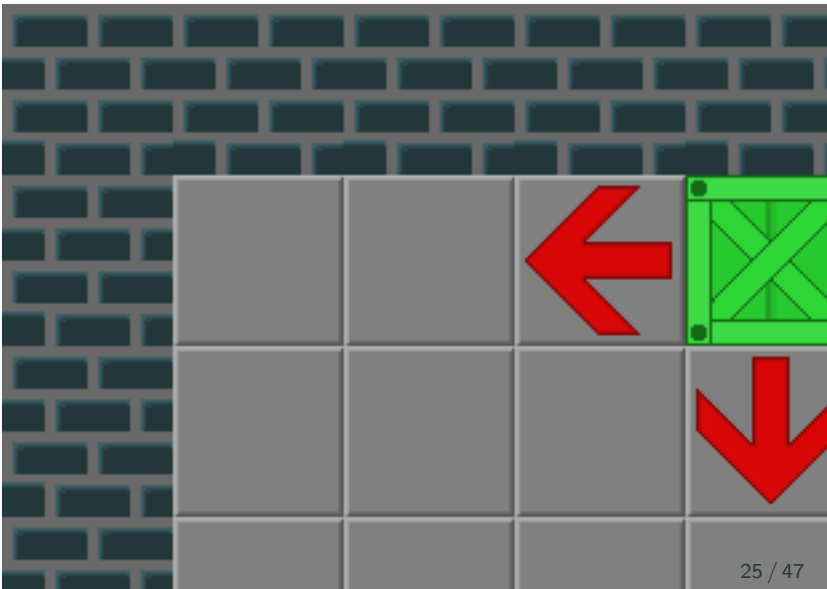
Détection des positions mortes (*dead positions*)

1.



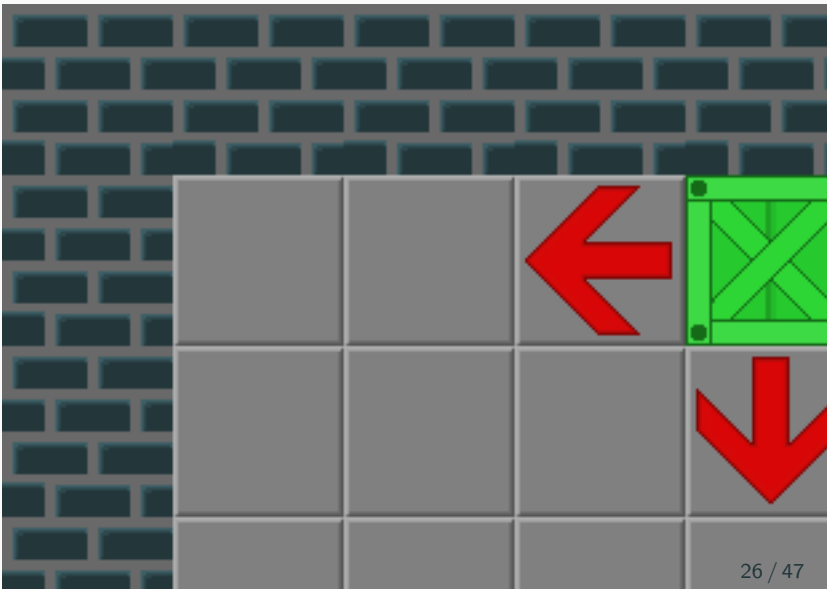
Détection des positions mortes (*dead positions*)

2.

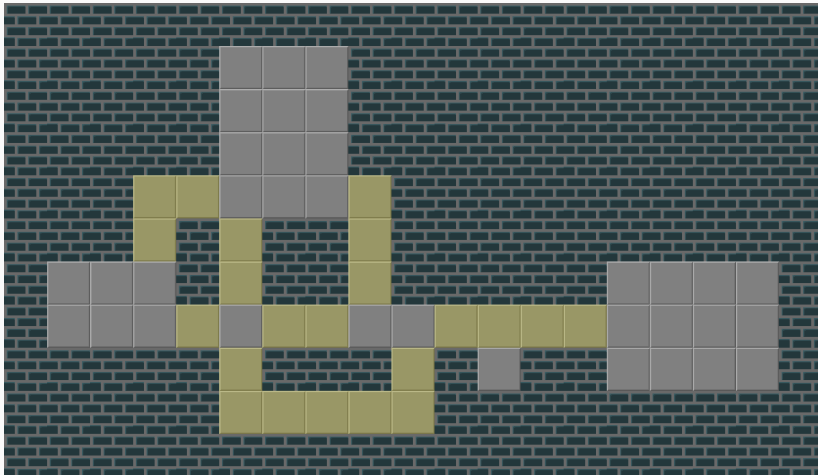


Détection des positions mortes (*dead positions*)

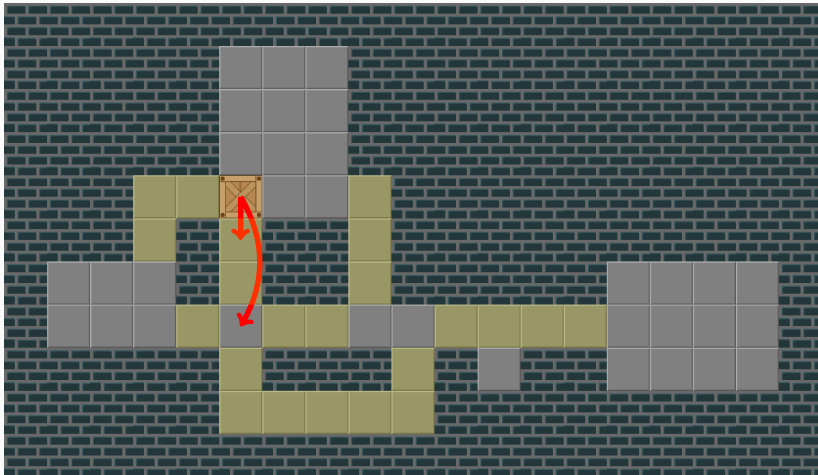
2.



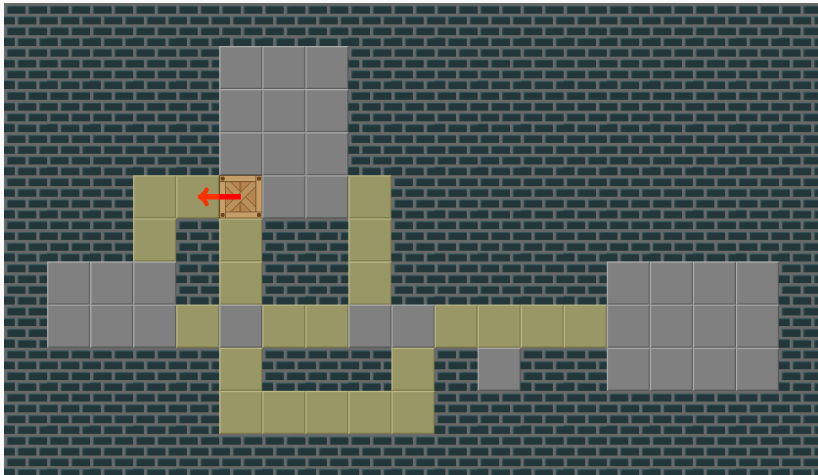
Détection de tunnels



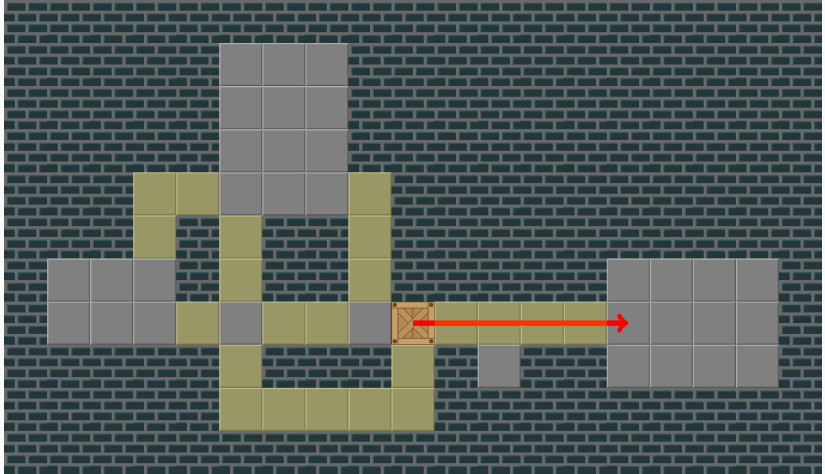
Détection de tunnels



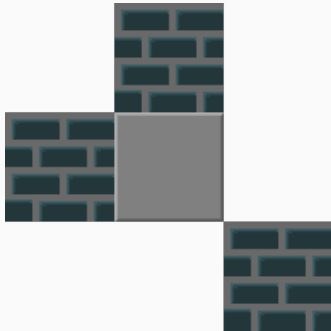
Détection de tunnels



Détection de tunnels



Détection de tunnels



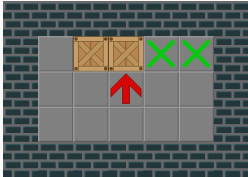
Calcul d'un ordre de rangement (*packing order*)

Réduction de l'espace de recherche

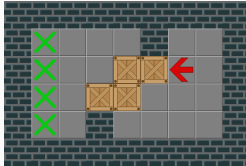


Analyse dynamique

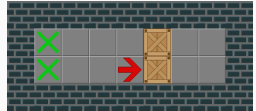
Détection d'impasses (*deadlocks*)



(a) *Freeze deadlock n°1*



(b) *Freeze deadlock n°2*

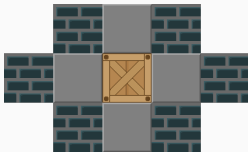


(c) *PI Corral deadlock*

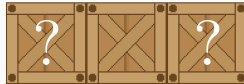
Détection de *freeze deadlocks*



(a) Règle n°1

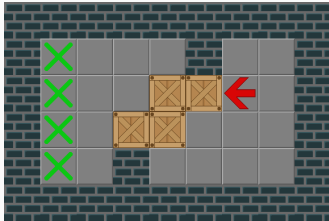


(b) Règle n°2

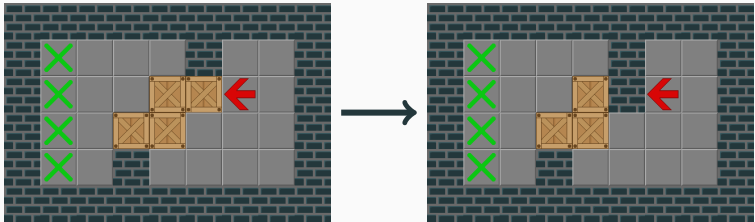


(c) Règle n°3

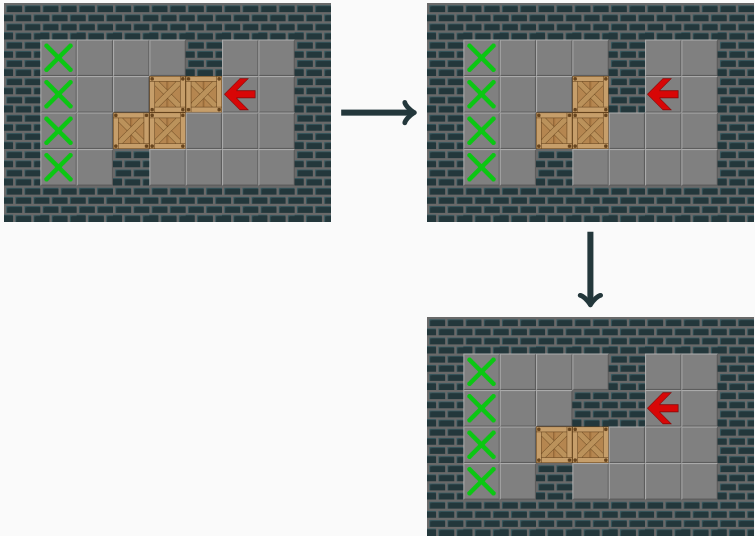
Détection de *freeze deadlocks*



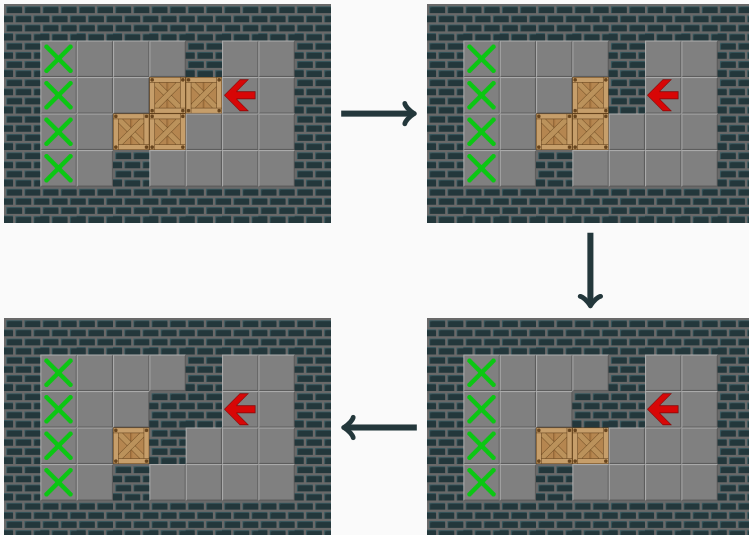
Détection de *freeze deadlocks*



Détection de *freeze deadlocks*



Détection de *freeze deadlocks*



Gelée!

Table de *deadlocks*



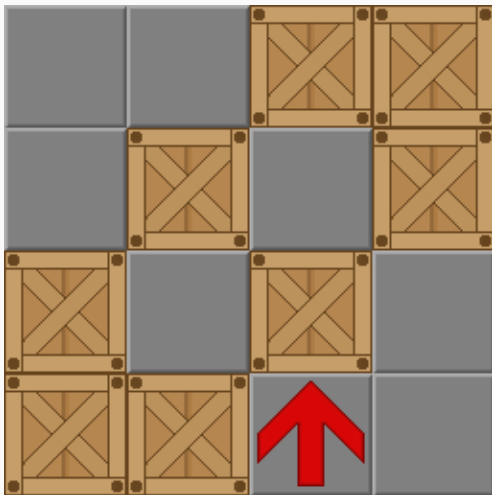
	11	12	13	14	
	7	8	9	10	
	4	5		6	
	1	2		3	

Table de *deadlocks*



Recherche dirigée par une heuristique



Heuristique simple (*Simple Lower Bound*)

Heuristique gloutonne (*Greedy Lower Bound*)

Optimisations

Résultats
