Résolution de niveaux du Sokoban

PoulpoGaz, darth-mole

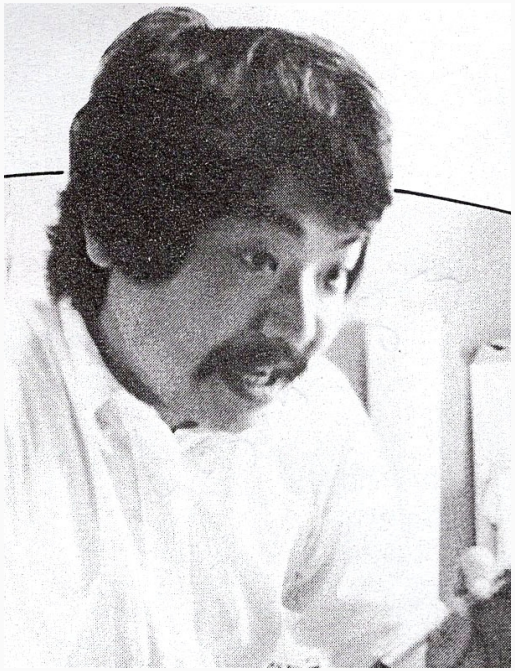
16 mai 2023

Candidat n° 012345

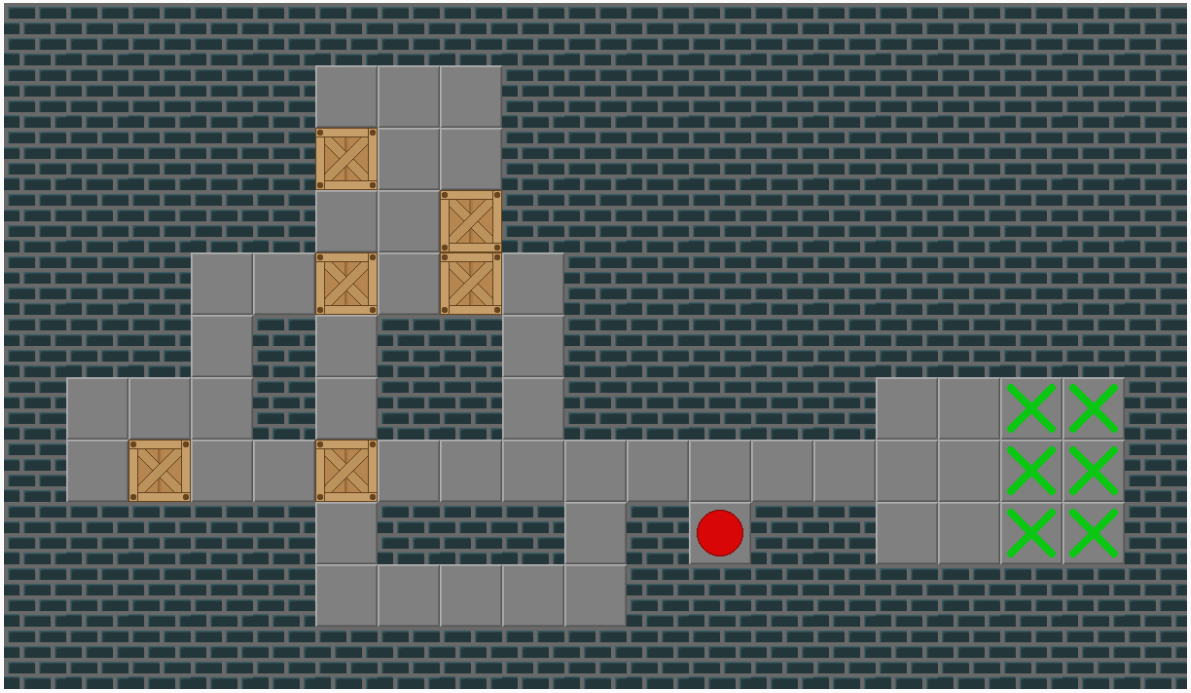
Plan

Le jeu du Sokoban

Le jeu du Sokoban



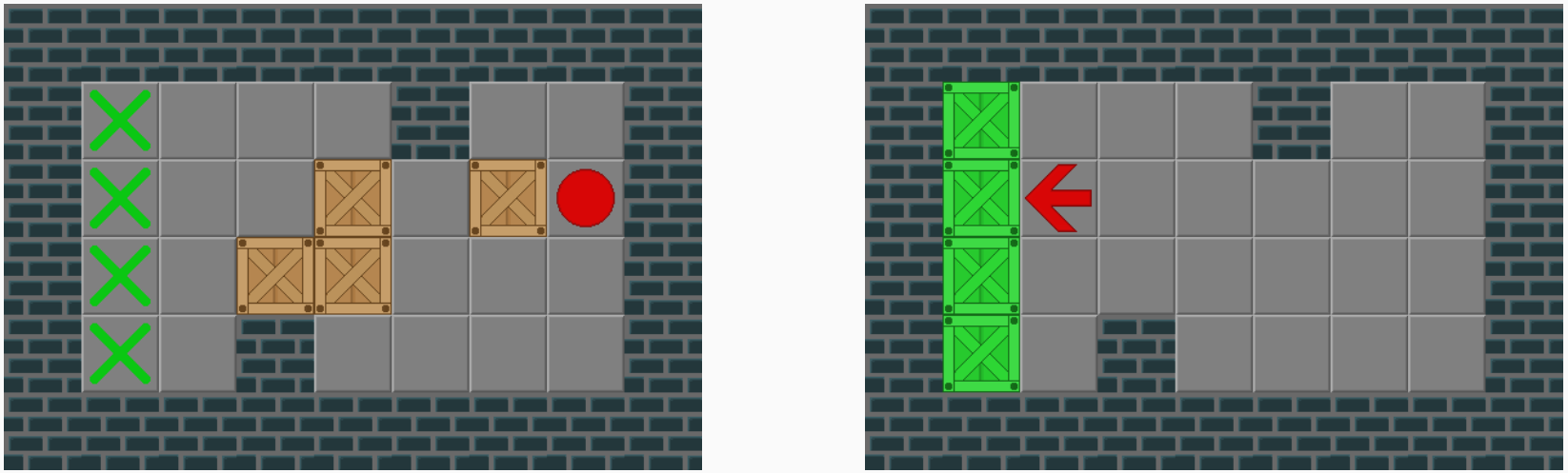
Hiroyuki Imabayashi



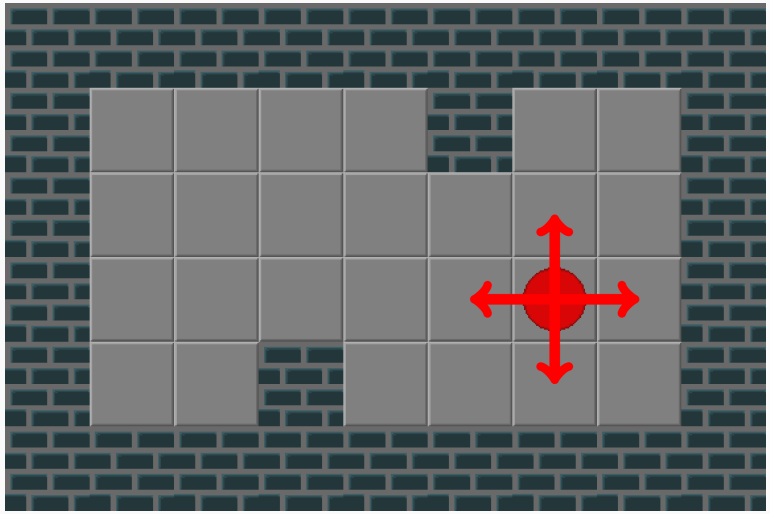
XSokoban

But du jeu

Déplacements

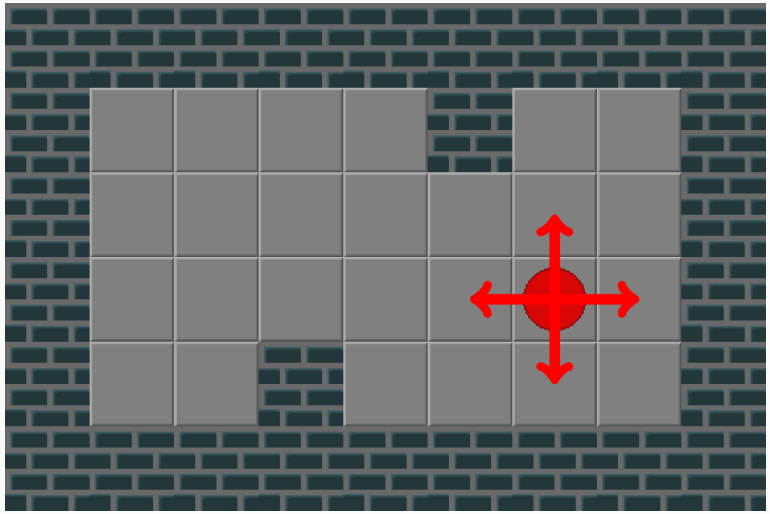


Règles

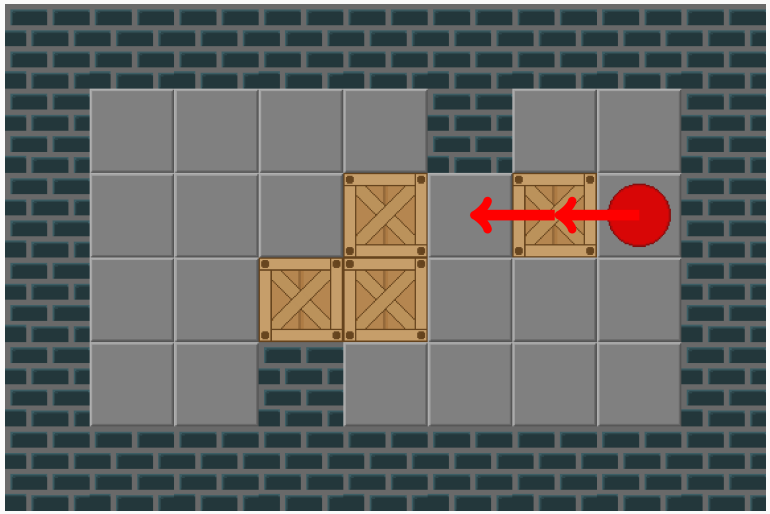


Déplacements autorisés

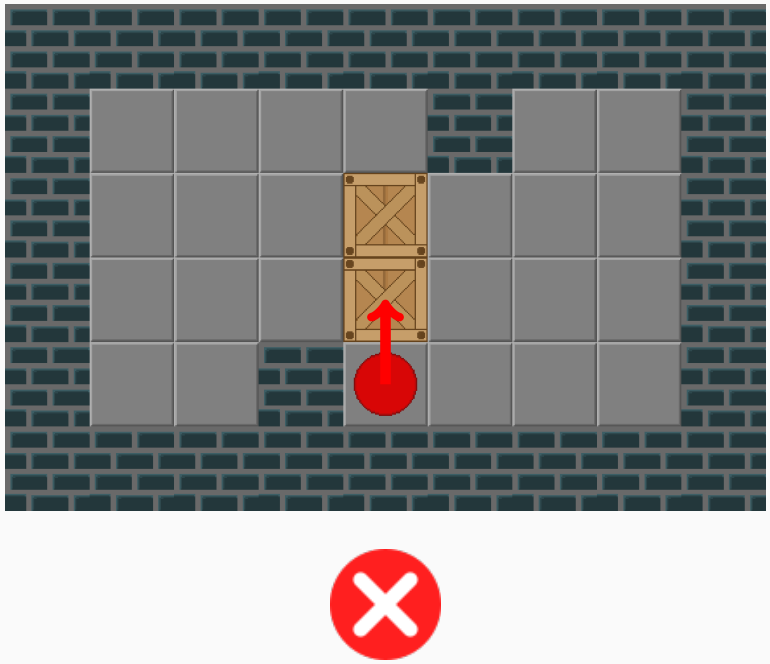
Règles

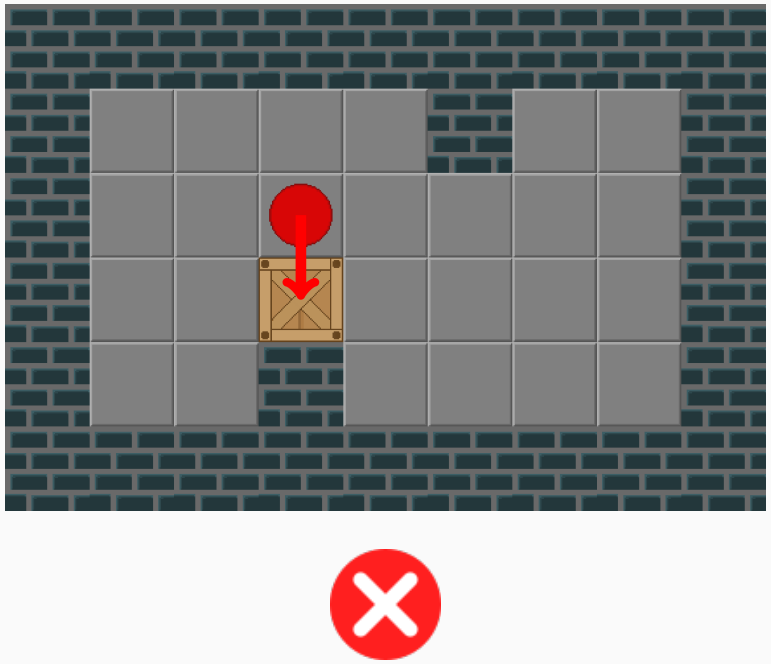


Déplacements autorisés

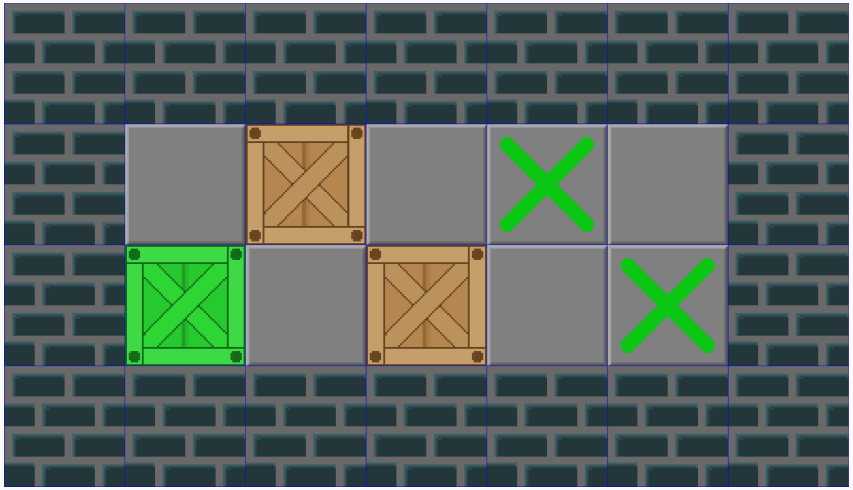


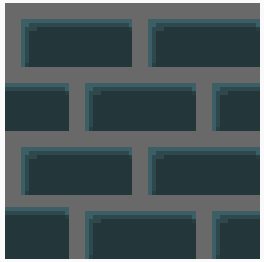
Règles



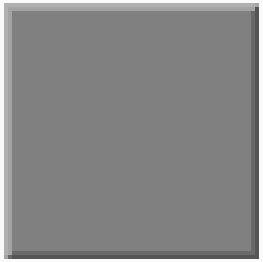


Tuiles

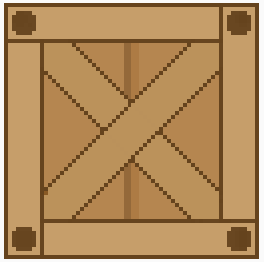




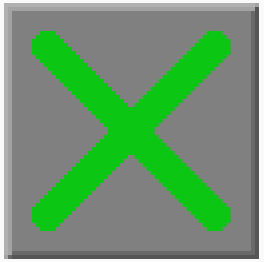
Mur



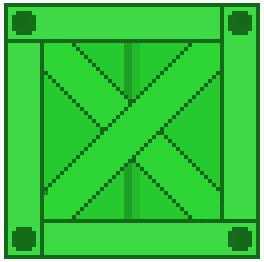
Sol



Caisse

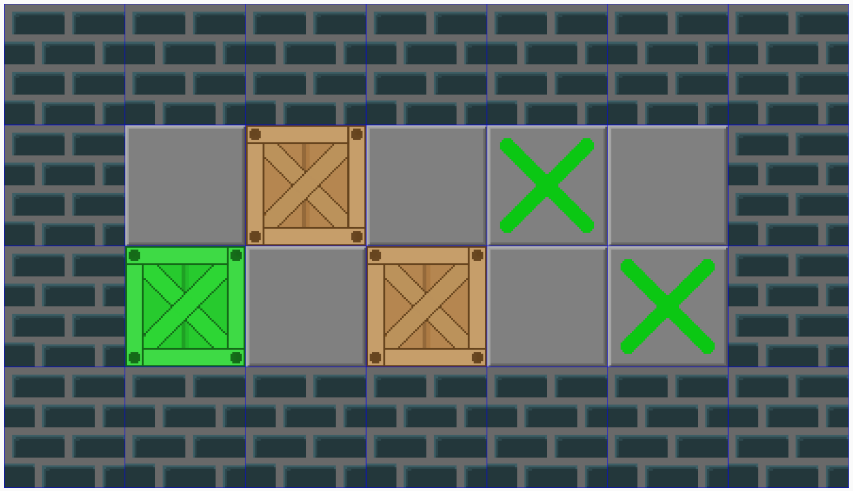


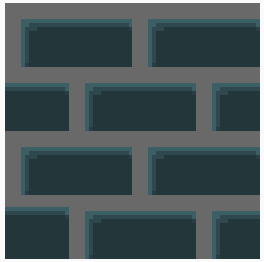
Cible



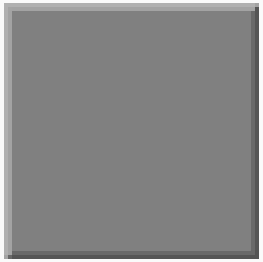
Caisse sur une cible

Tuiles

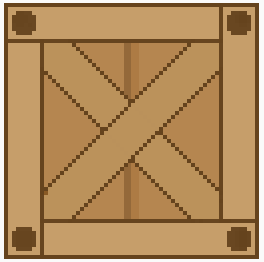




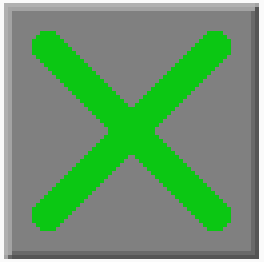
Mur



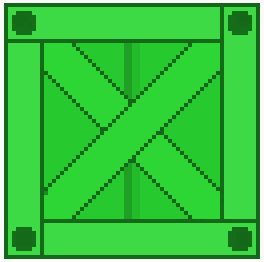
Sol



Caisse



Cible



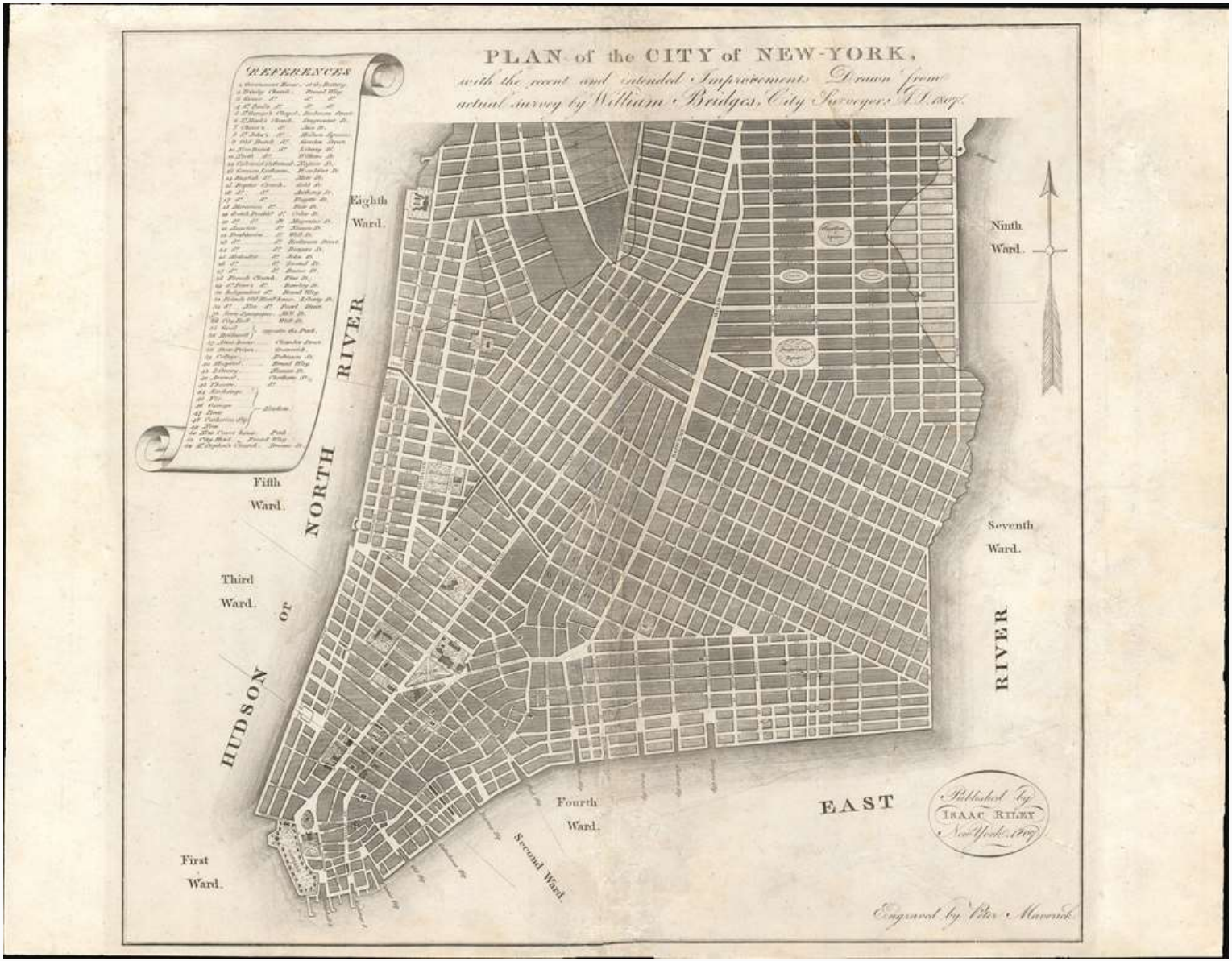
Caisse sur une cible

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

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

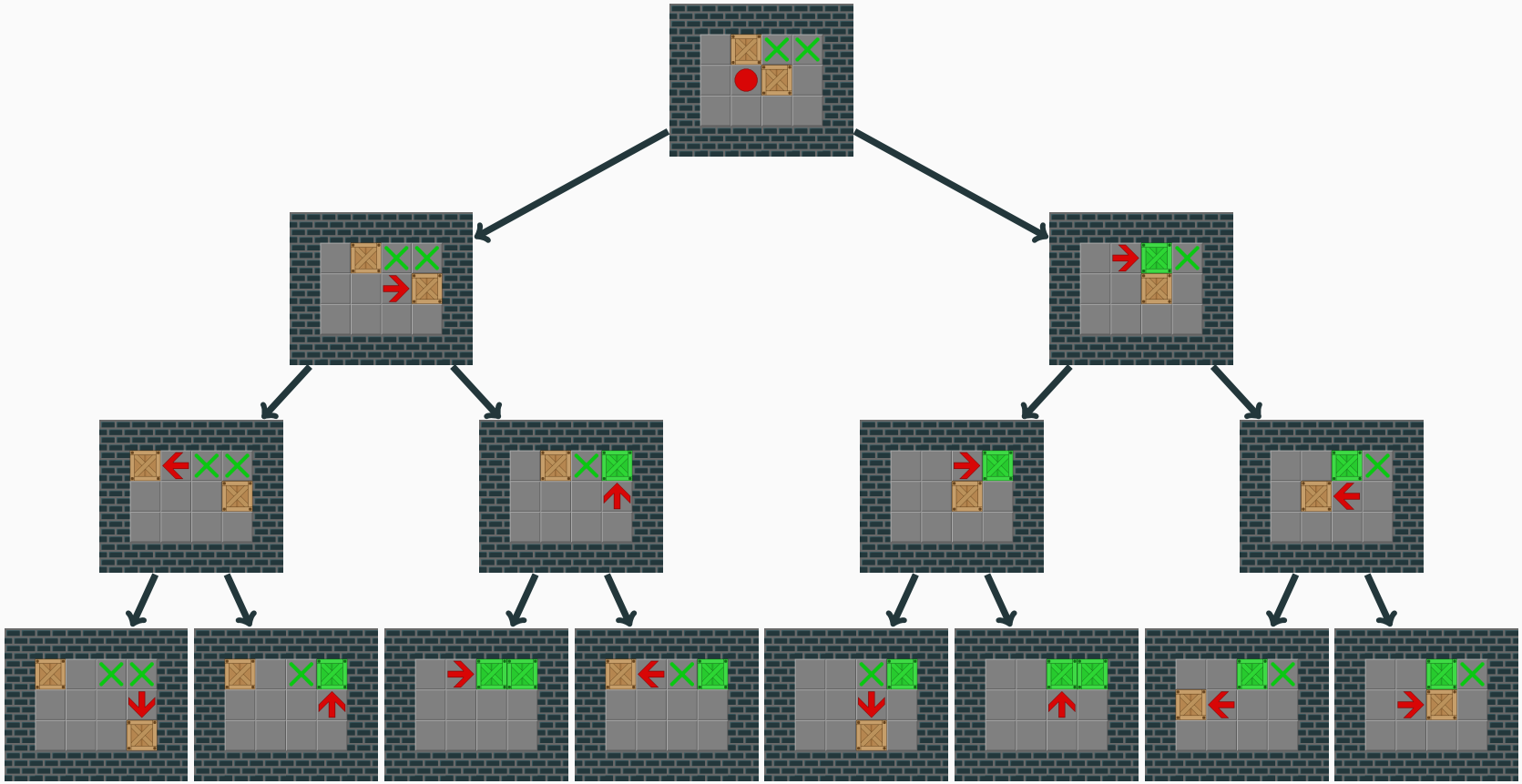
Lien avec le thème de l'année

Lien avec le thème de l'année

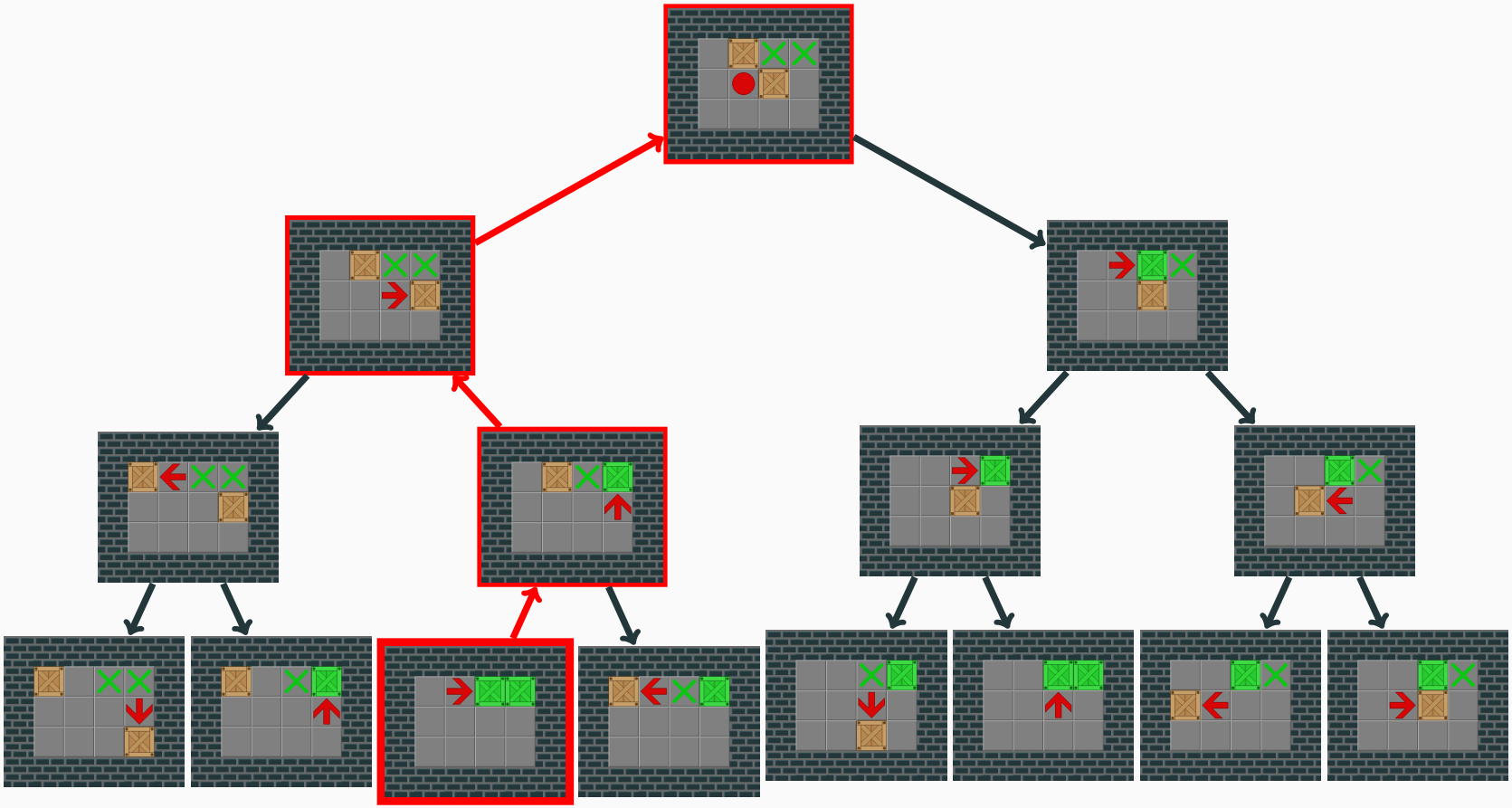


Principe de résolution

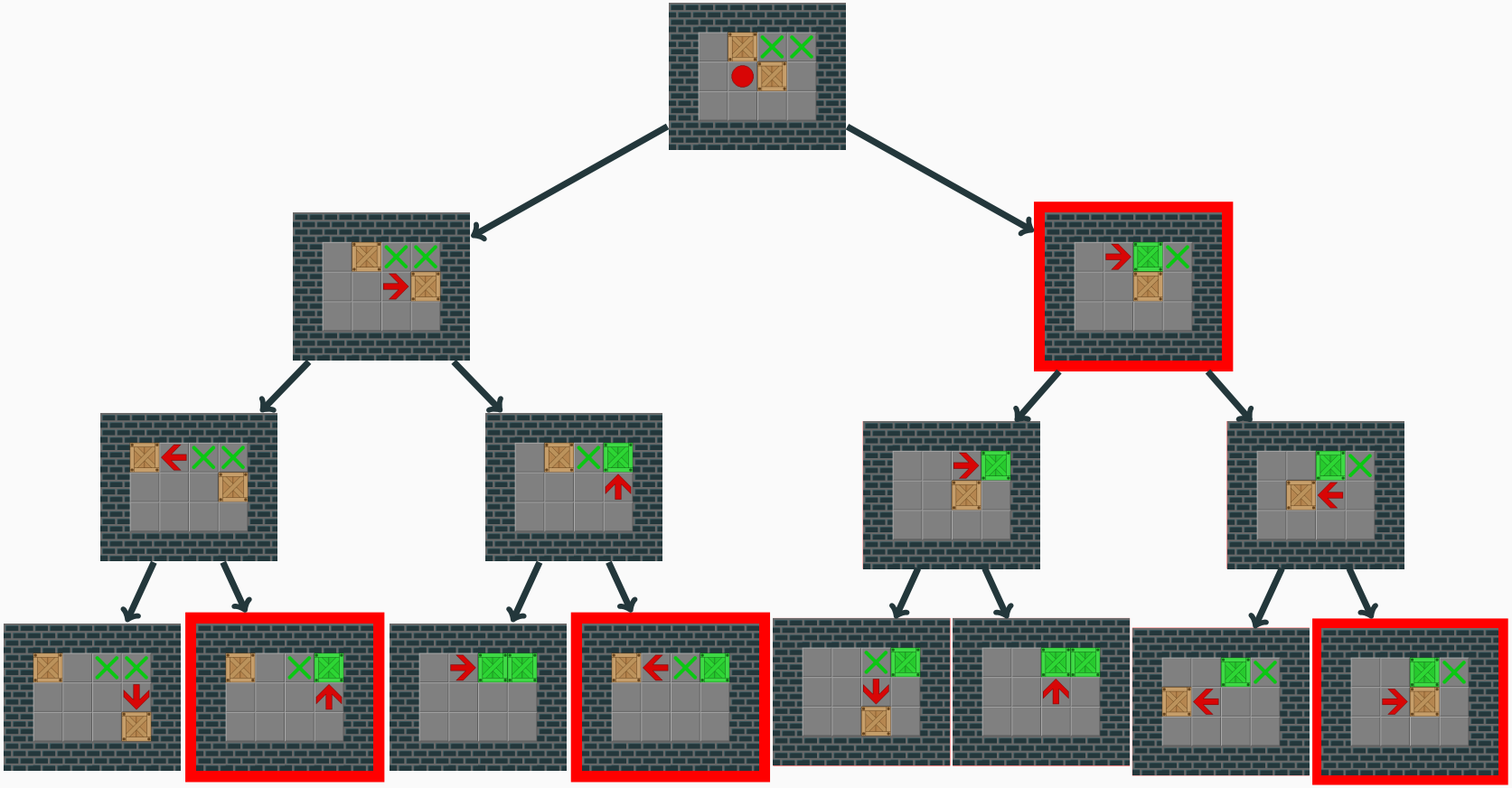
Arbre des états



Arbre des états



Arbre des états



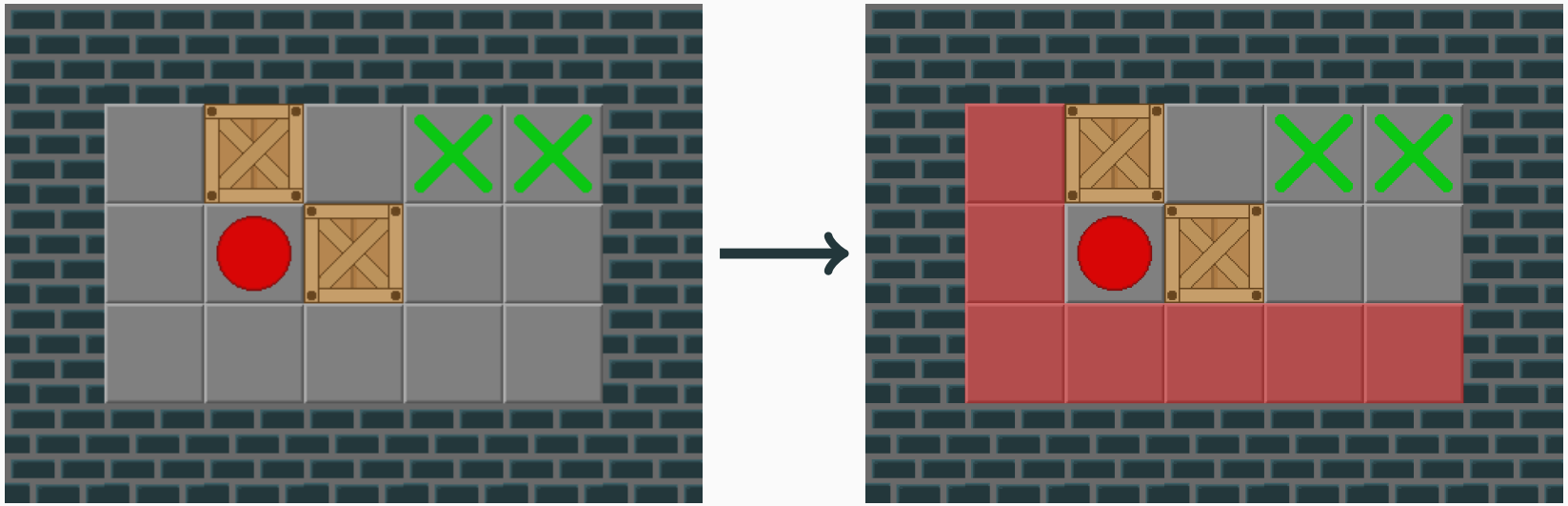
Calcul du *hash* d'un état

Réduction de l'espace de recherche

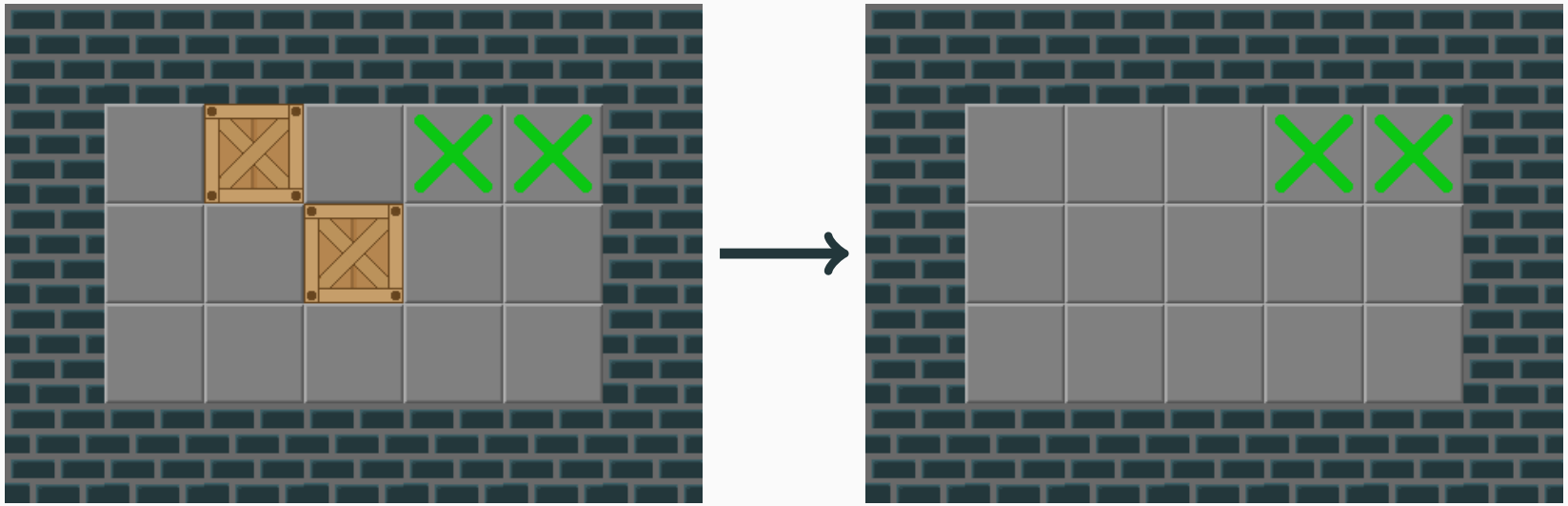
Réduction de l'espace de recherche

Analyse statique

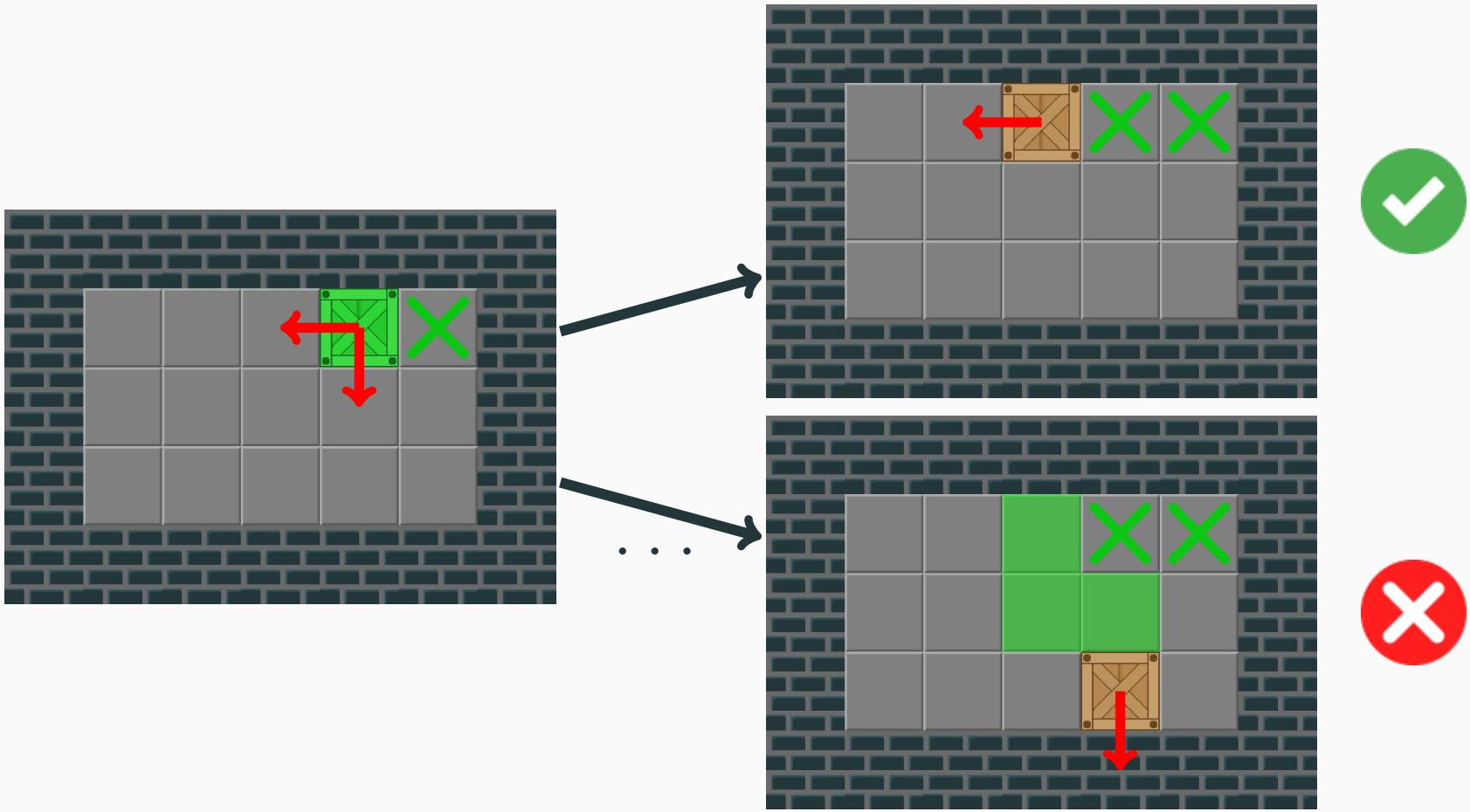
Détection des positions mortes *(dead positions)*



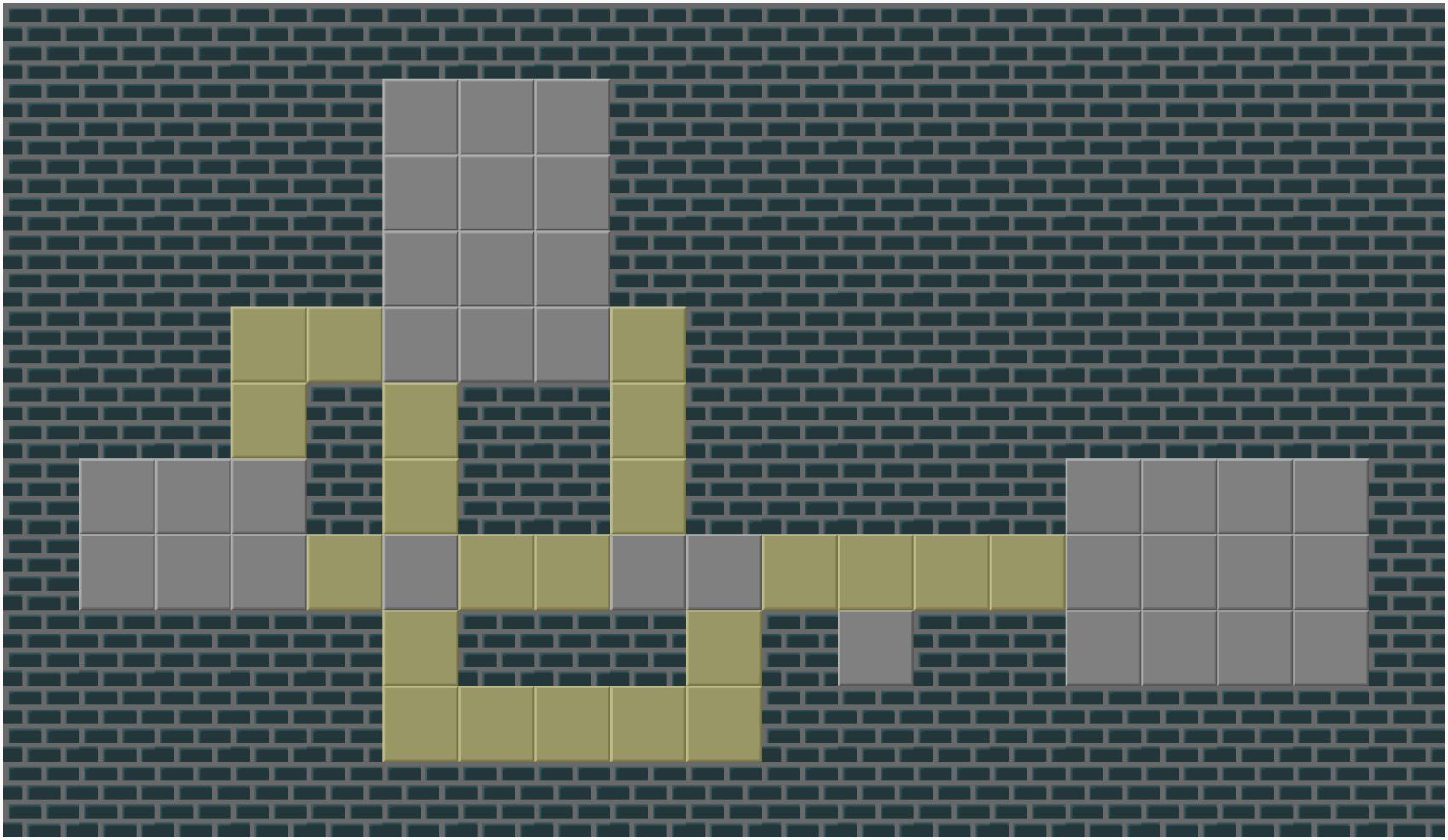
Détection des positions mortes *(dead positions)*



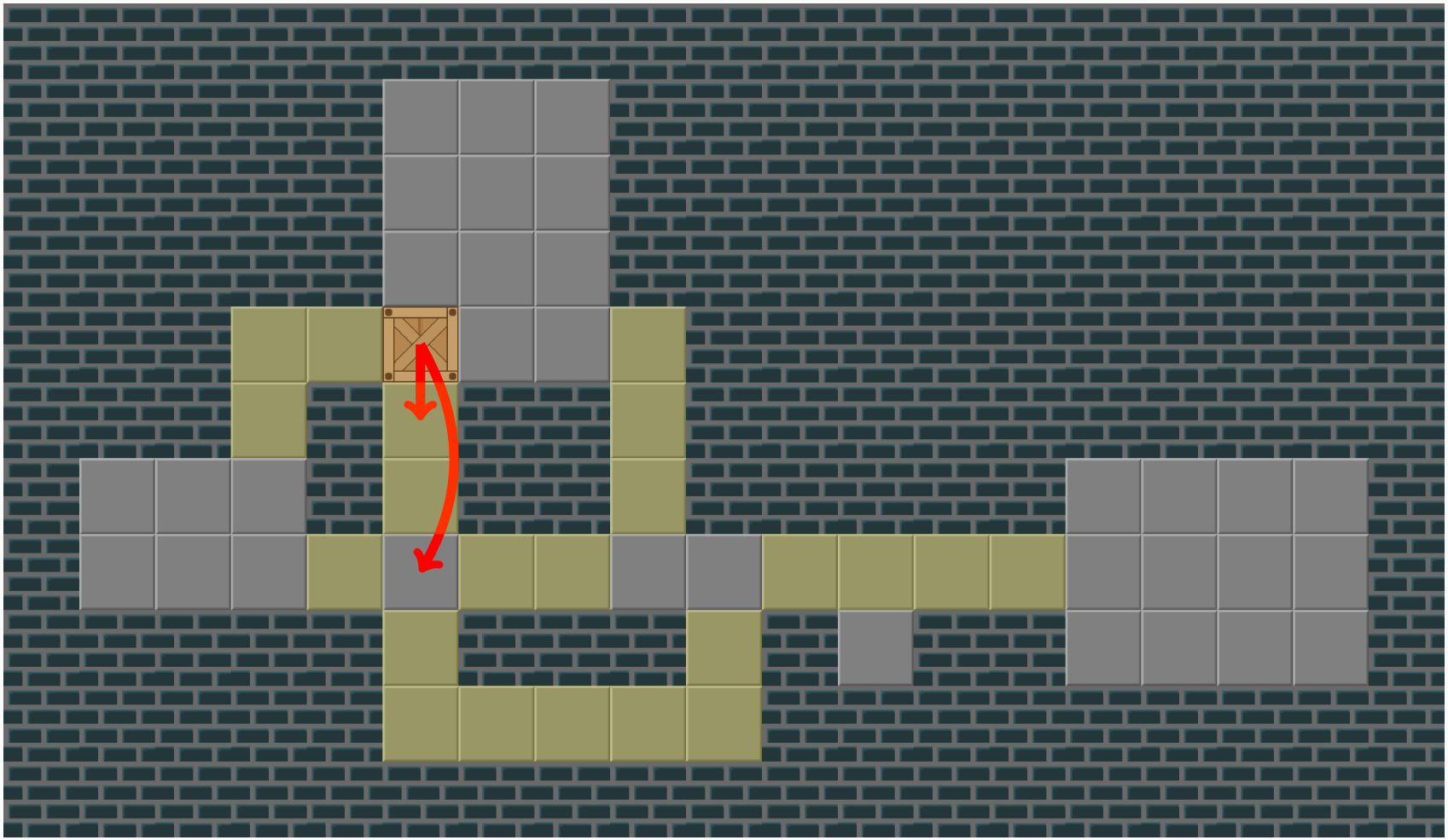
Détection des positions mortes *(dead positions)*



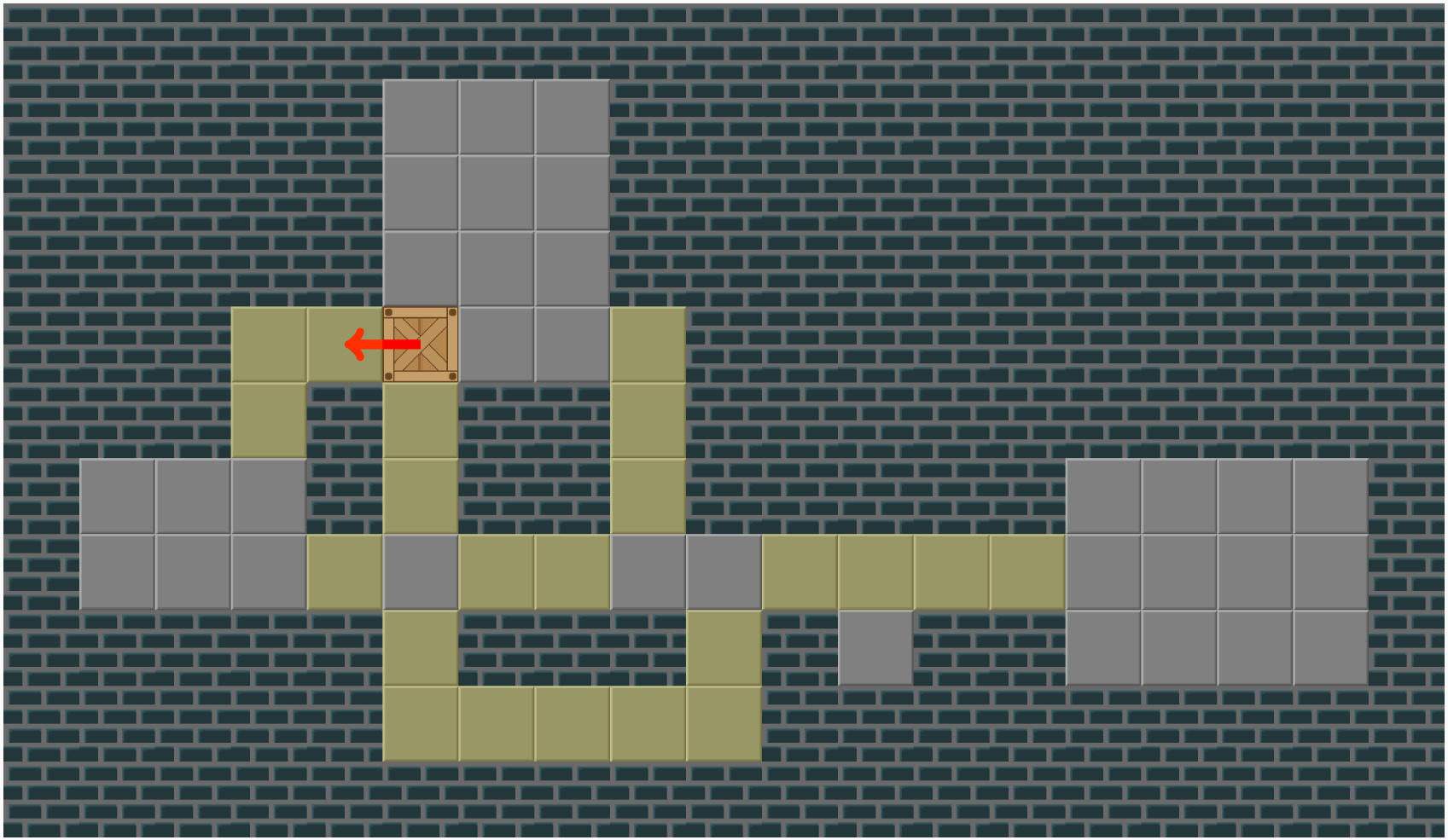
Détection de tunnels



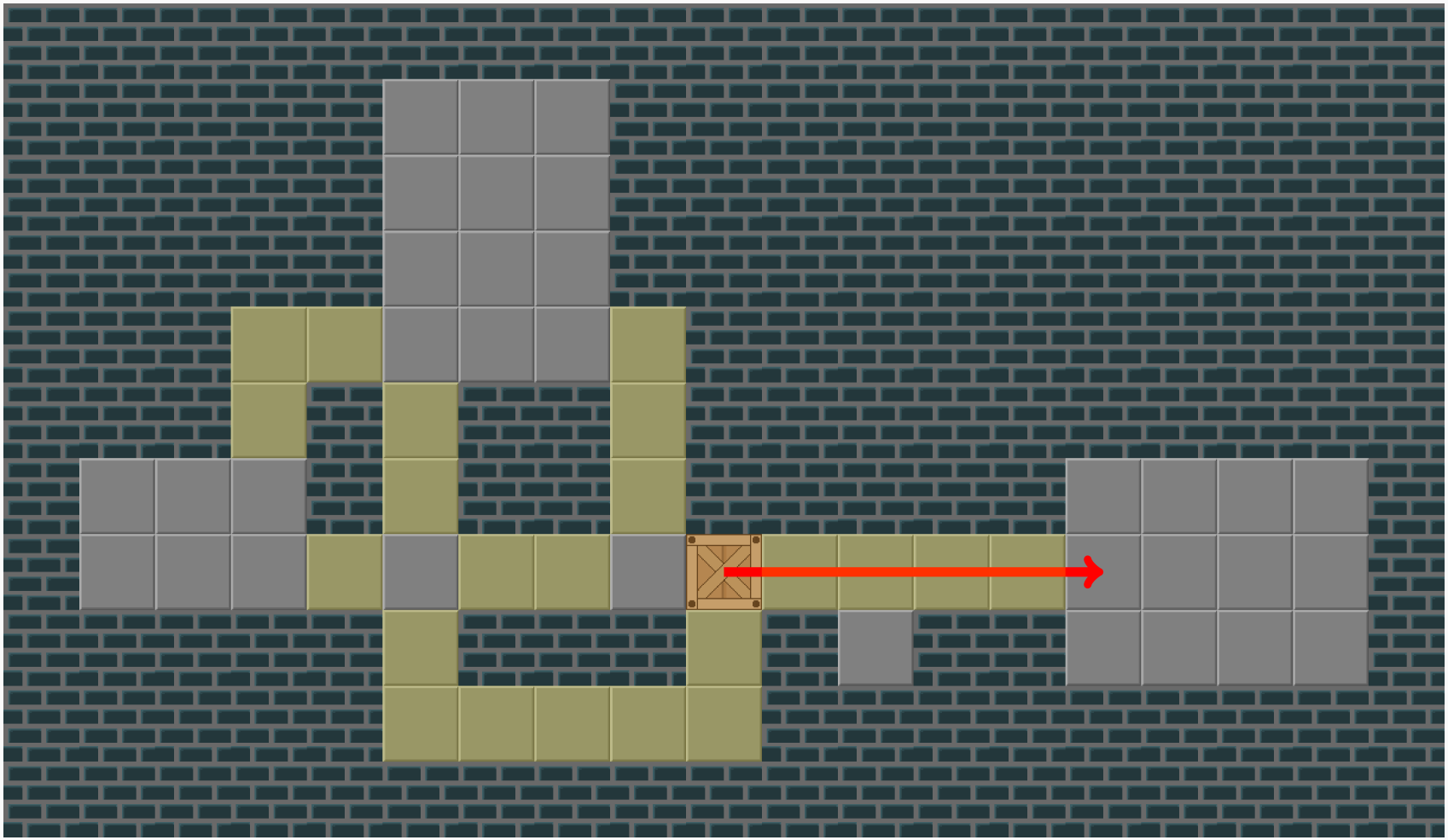
Détection de tunnels



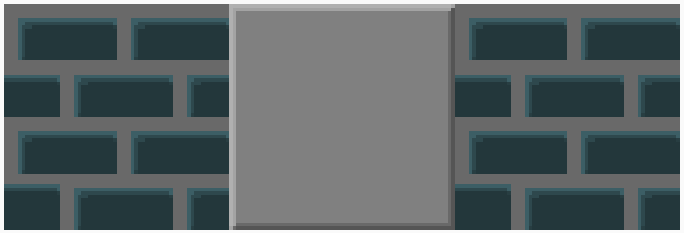
Détection de tunnels

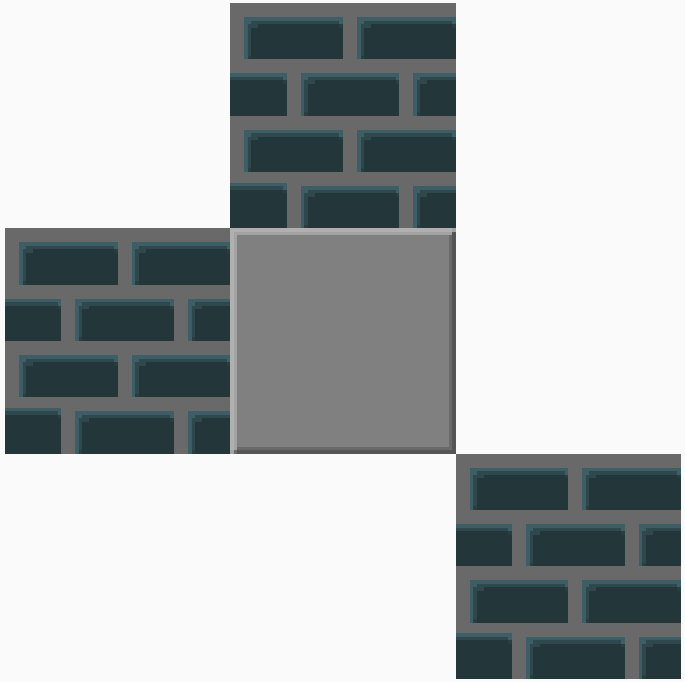


Détection de tunnels



Détection de tunnels



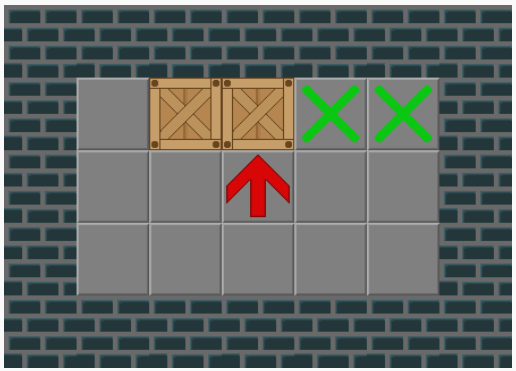


Calcul d'un ordre de rangement *(packing order)*

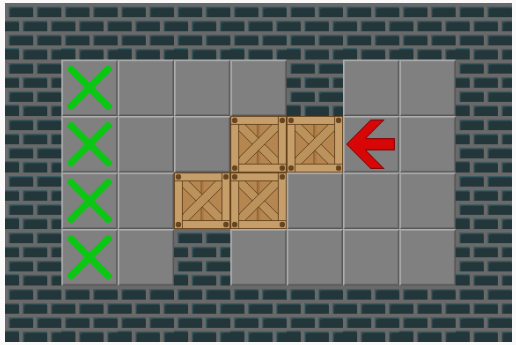
Réduction de l'espace de recherche

Analyse dynamique

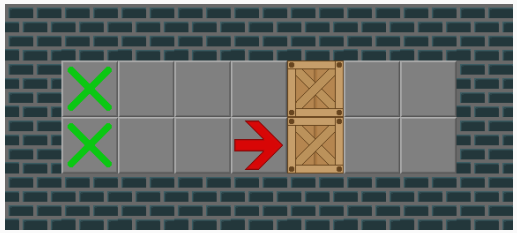
Détection d'impasses *(deadlocks)*



(a) *Freeze deadlock n°1*



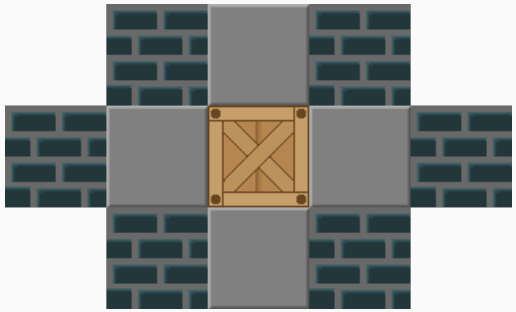
(b) *Freeze deadlock n°2*



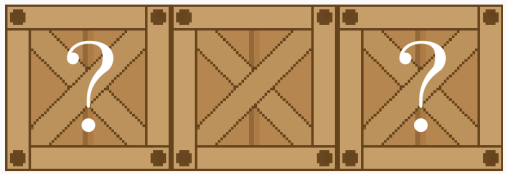
(c) *PI Corral deadlock*

Détection de *freeze deadlocks*

(a) *Règle n°1*

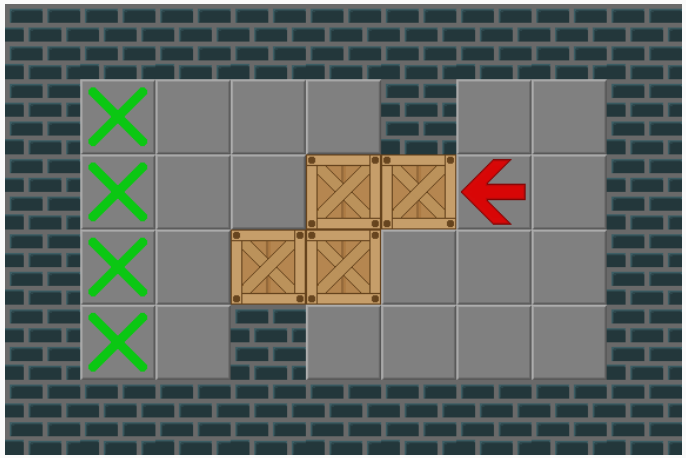


(b) *Règle n°2*

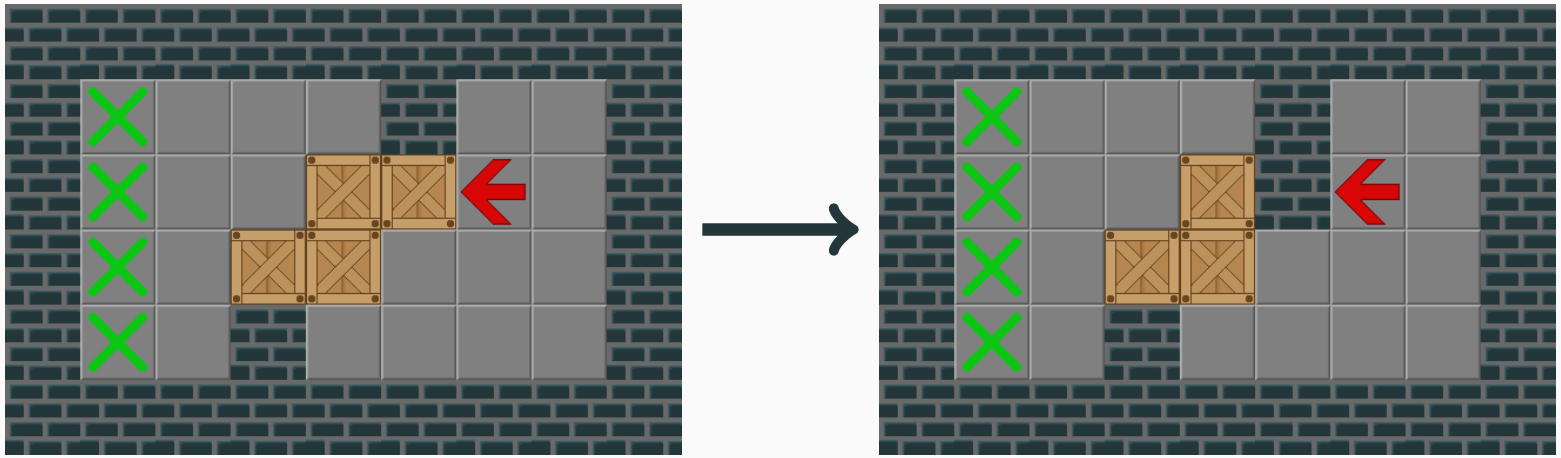


(c) *Règle n°3*

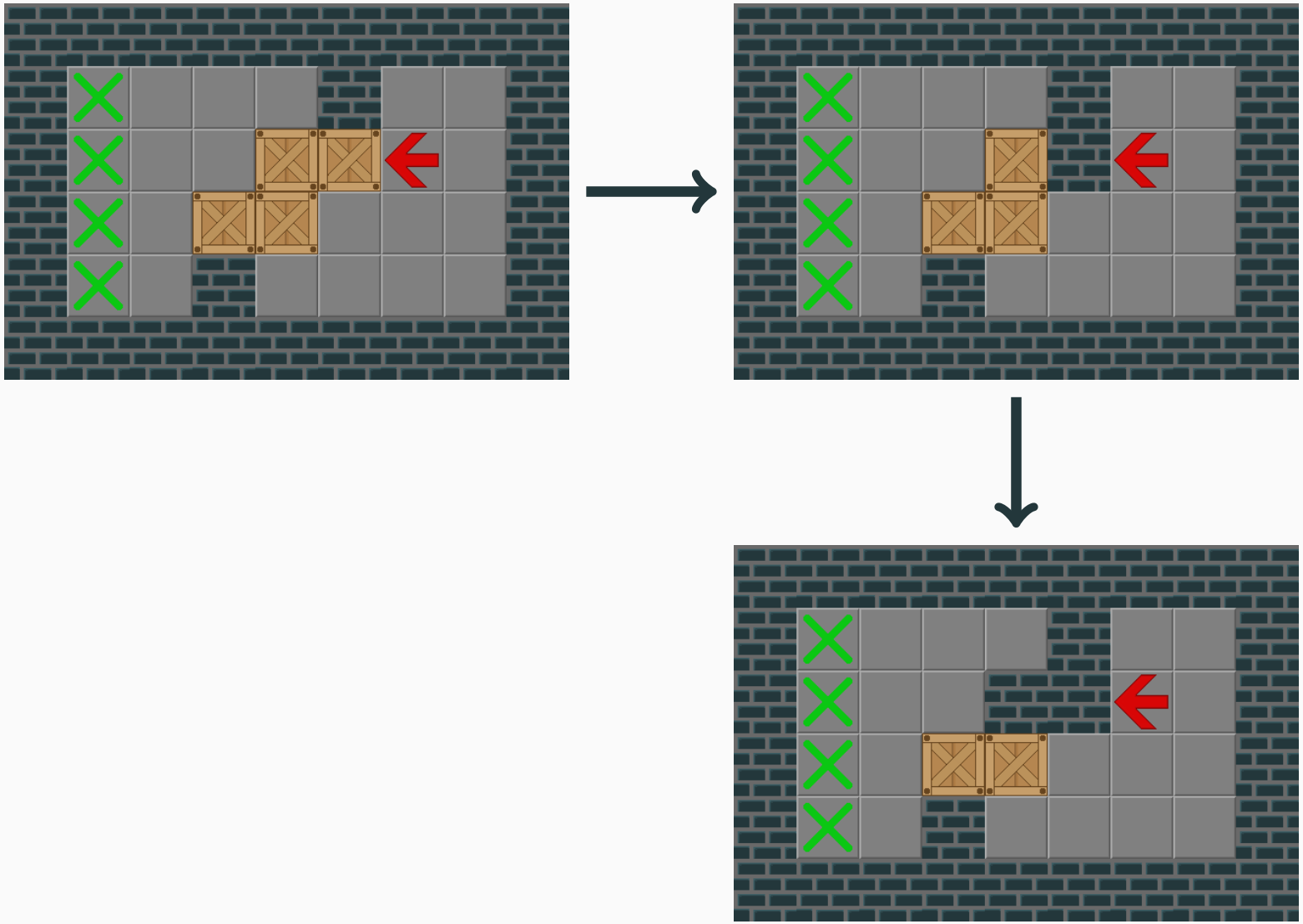
Détection de *freeze deadlocks*



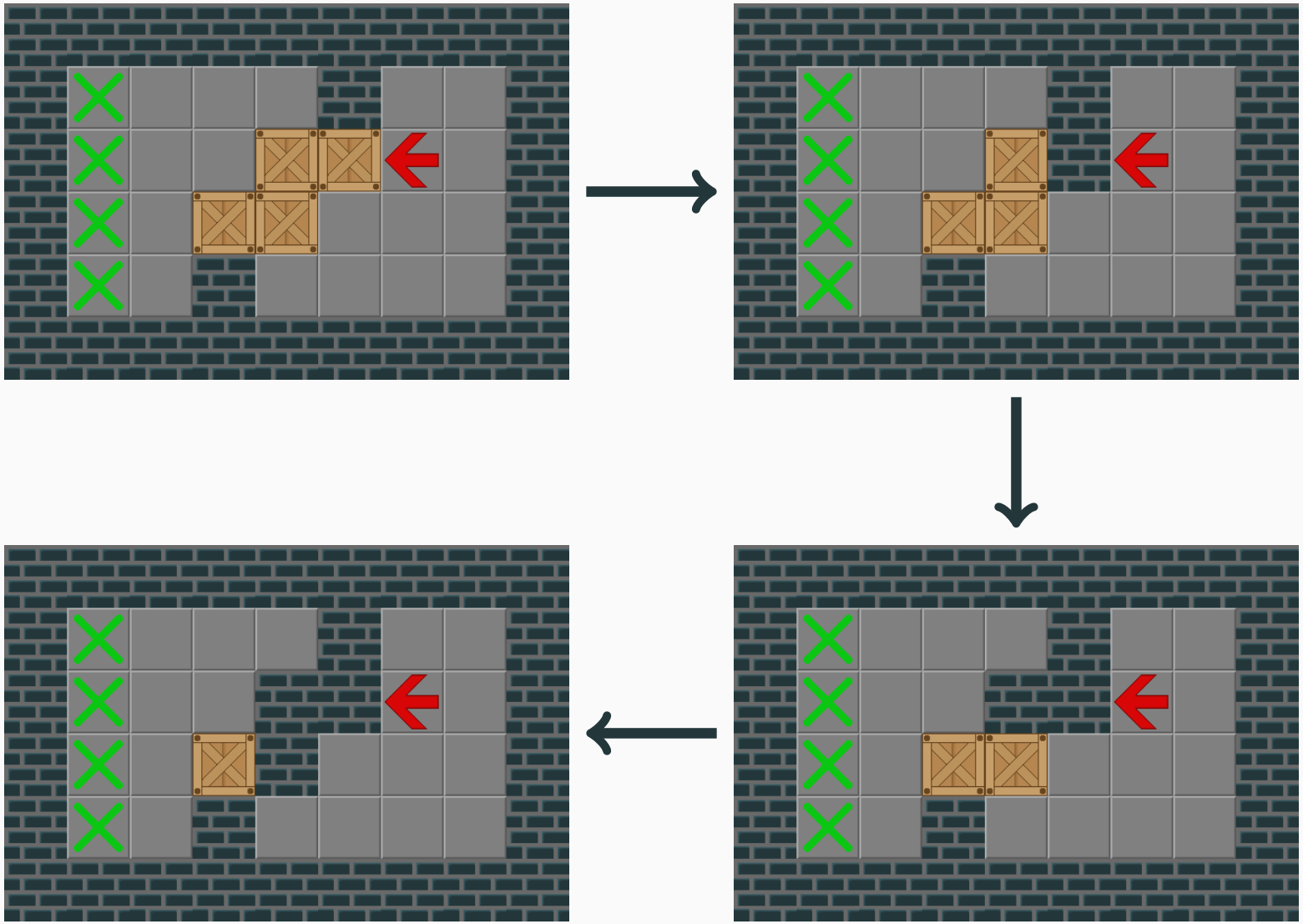
Détection de *freeze deadlocks*



Détection de *freeze deadlocks*



Détection de *freeze deadlocks*



Gelée!

Détection de *PI Corral deadlocks*

Table de *deadlocks*

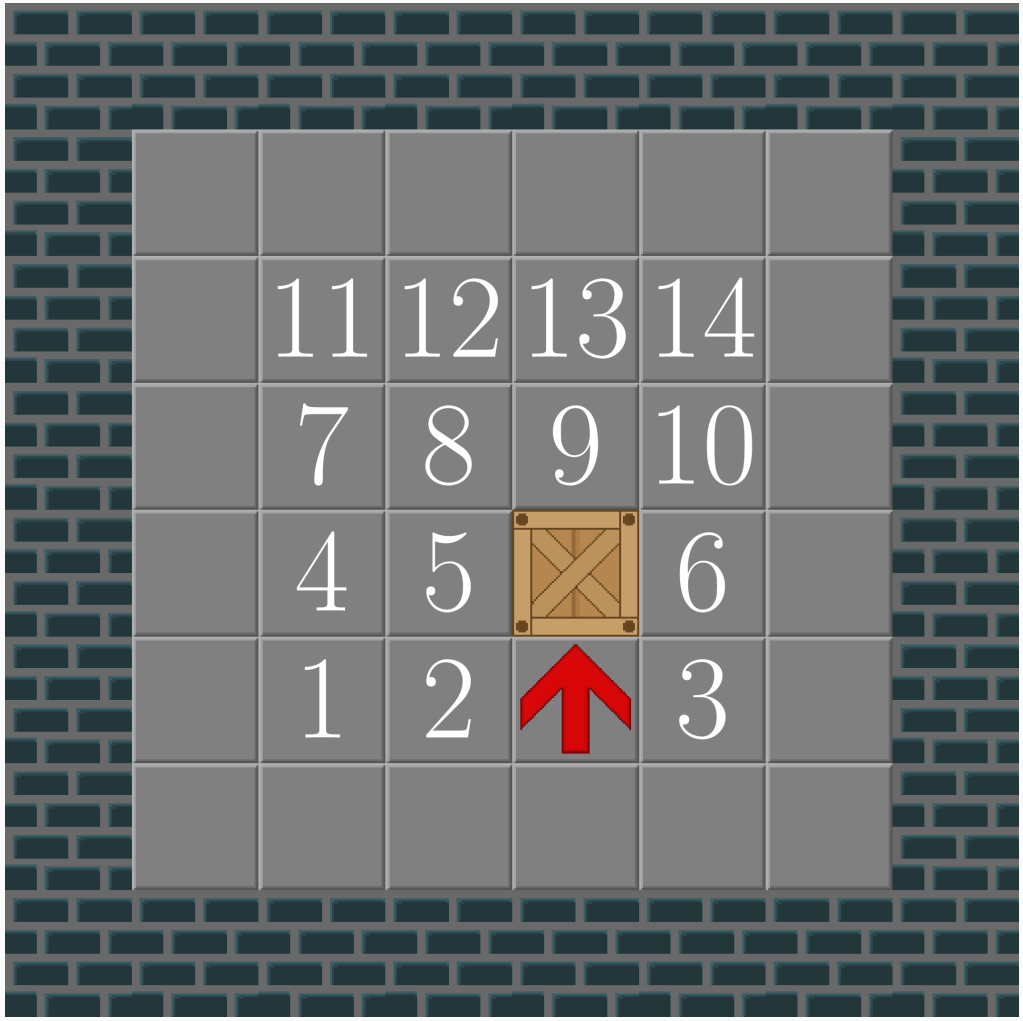
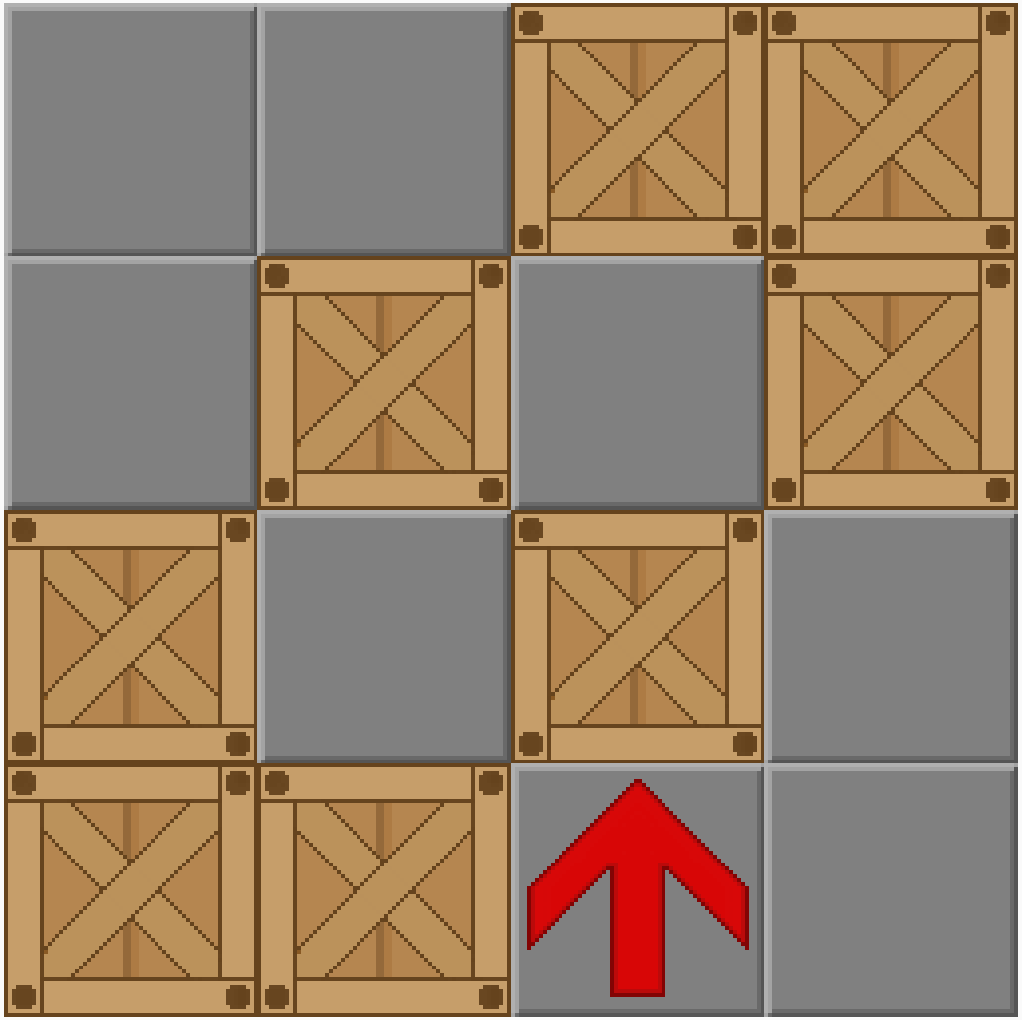
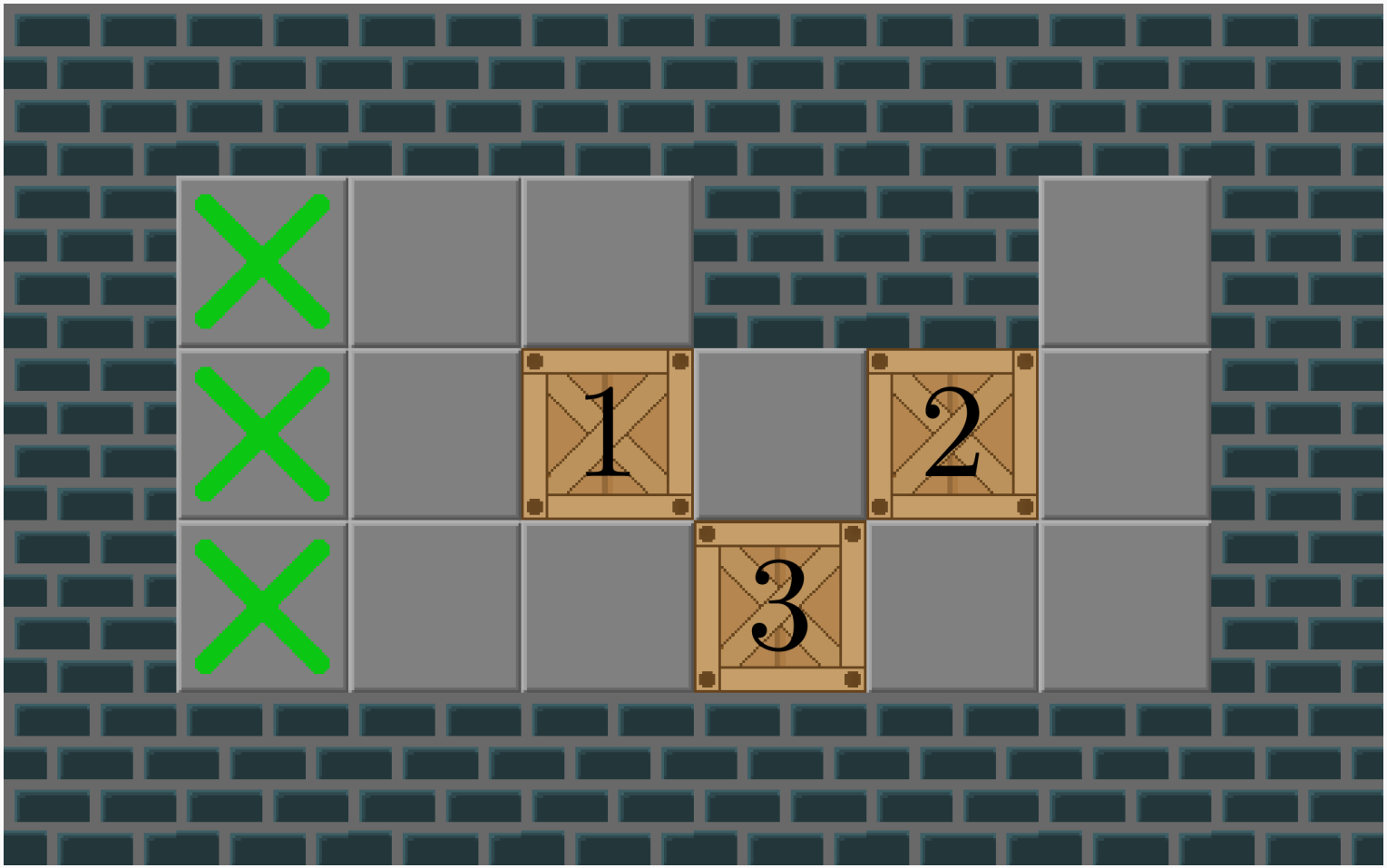


Table de *deadlocks*

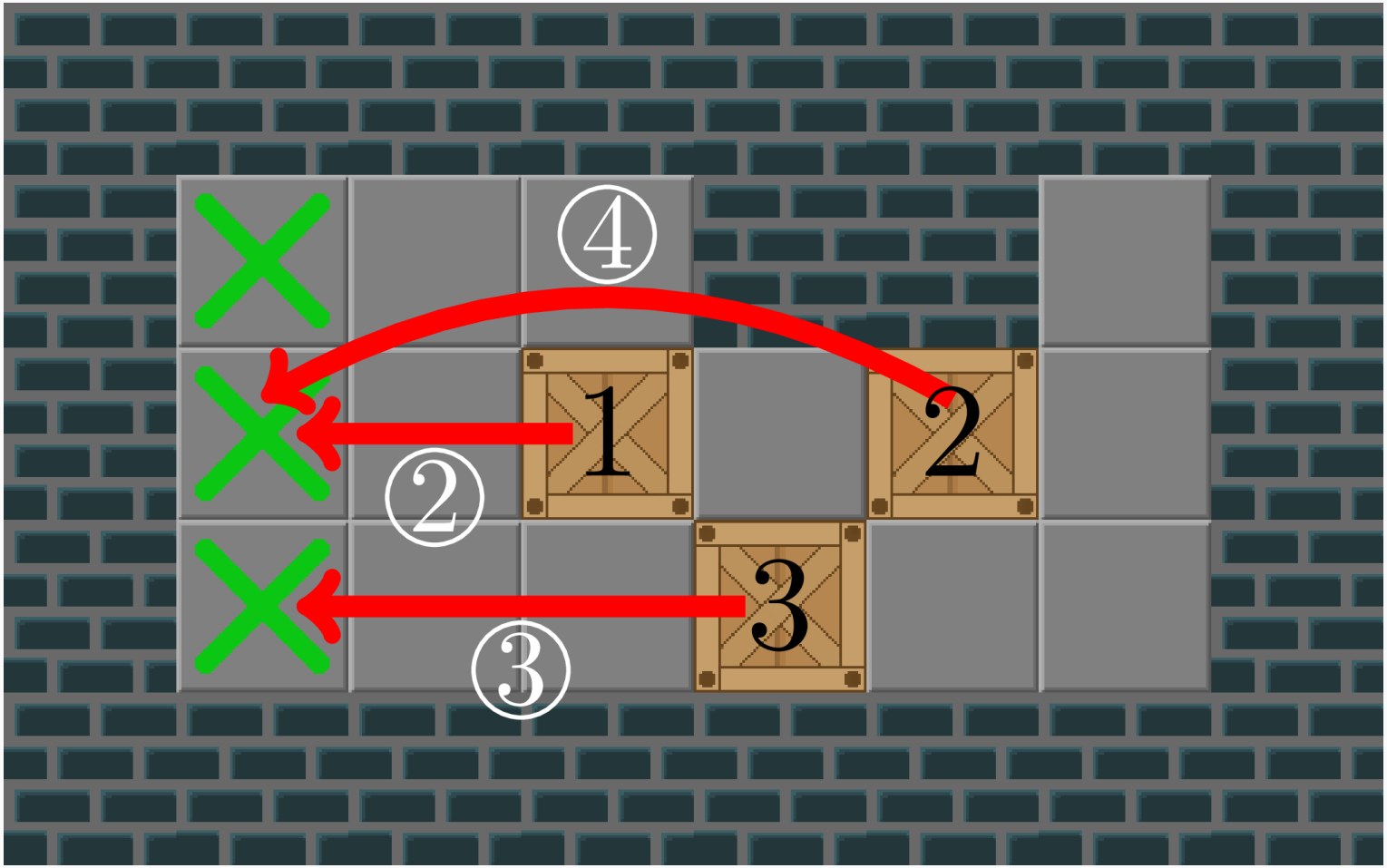


Recherche dirigée par une heuristique

Heuristique simple *(Simple Lower Bound)*

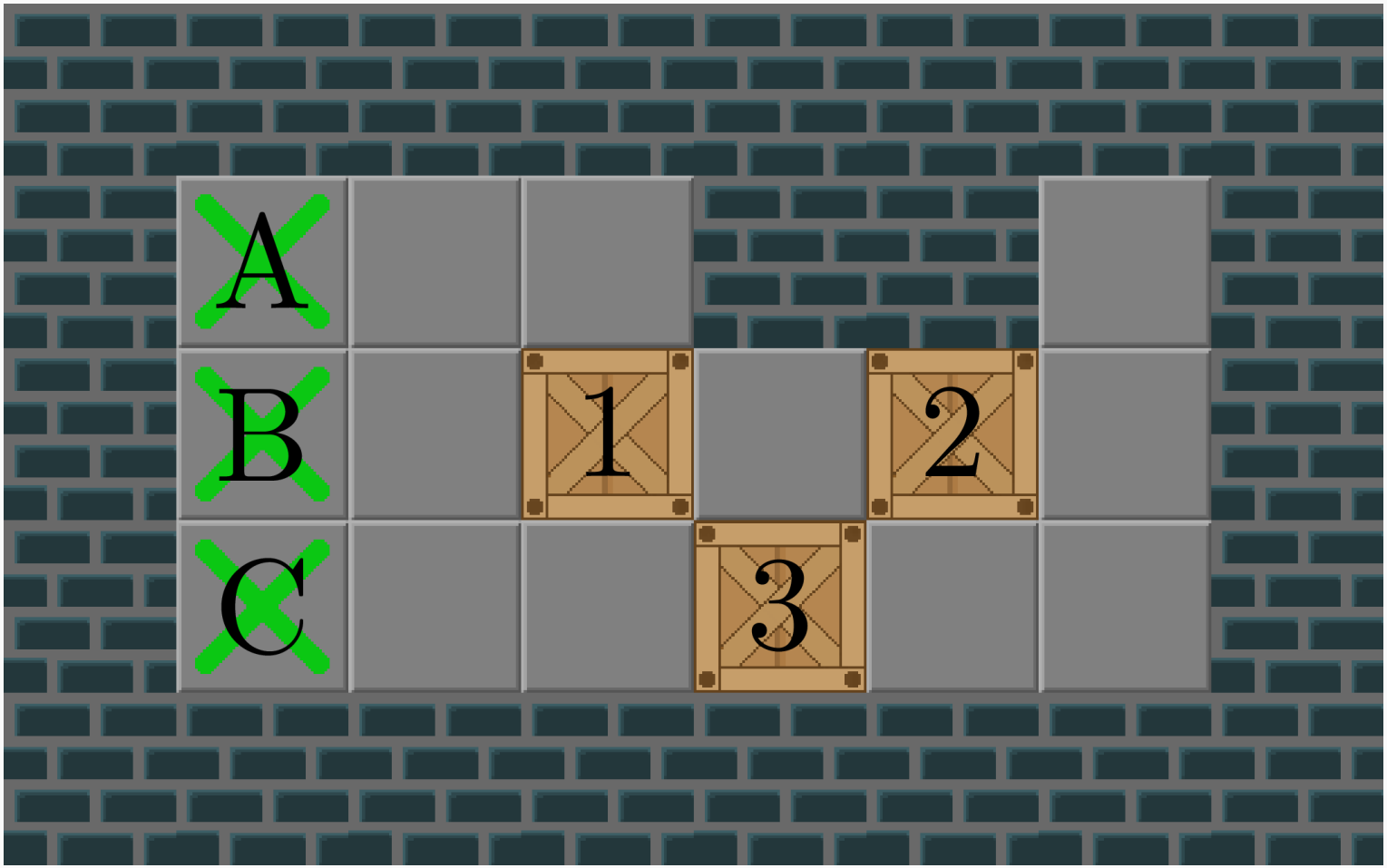


Heuristique simple *(Simple Lower Bound)*

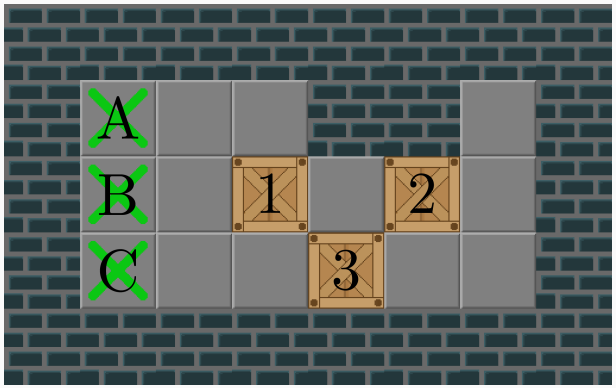


$$\boxed{2 + 4 + 3 = \mathbf{9}}$$

Heuristique gloutonne *(Greedy Lower Bound)*



Heuristique gloutonne *(Greedy Lower Bound)*

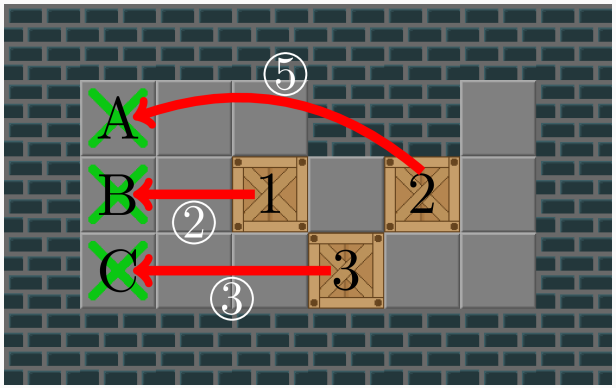


$1 \to A$	3
$1 \to B$	2
$1 \to C$	3
$2 \to A$	4
$2 \to B$	4
$2 \to C$	5
$3 \to A$	5
$3 \to B$	4
$3 \to C$	3

Tri →

$1 \to B$	**2**
$1 \to A$	3
$1 \to C$	3
$3 \to C$	**3**
$2 \to B$	4
$3 \to B$	4
$2 \to A$	5
$2 \to C$	5
$3 \to A$	**5**

Heuristique gloutonne *(Greedy Lower Bound)*



$$2 + 3 + 5 = \mathbf{10}$$

$1 \to A$	3
$1 \to B$	2
$1 \to C$	3
$2 \to A$	4
$2 \to B$	4
$2 \to C$	5
$3 \to A$	5
$3 \to B$	4
$3 \to C$	3

Tri →

$\mathbf{1 \to B}$	**2**
$1 \to A$	3
$1 \to C$	3
$\mathbf{3 \to C}$	**3**
$2 \to B$	4
$3 \to B$	4
$2 \to A$	5
$2 \to C$	5
$\mathbf{3 \to A}$	**5**

Optimisations

Résultats