

Robot Ricochet Solver

SI-MOHAMMED Sonia-Taous

BOUAUD Malik

AIT KHEDDACHE Wissam

SINI Lynda

CUQUEMELLE Mathieu

Encadré par :

M.BONNET Gregory

M.CHATEL Romain

M.SASSI Taoufik

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

1 Introduction

Robot Ricochet Solver, c'est quoi ?
Problème

2 Éléments techniques

Algorithme A*
Tables de transposition

3 Conclusion

Robot Ricochet
Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments
techniques

Algorithme A*

Tables de transposition

Conclusion

1 Introduction

Robot Ricochet Solver, c'est quoi ?
Problème

2 Éléments techniques

3 Conclusion

Description du jeu

Écran Principal

Robot Ricochet
Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments
techniques

Algorithme A*

Tables de transposition

Conclusion

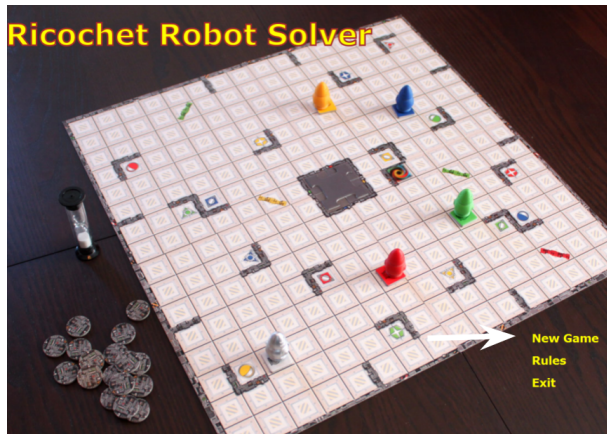


FIGURE – L'écran principal du jeu.

Description du jeu

Règles du jeu

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

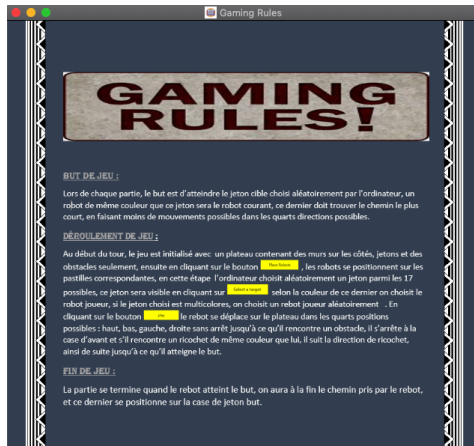


FIGURE – Le clic sur le bouton **Rules**, provoque l’affichage de la fenêtre où on explique les règles du jeu.



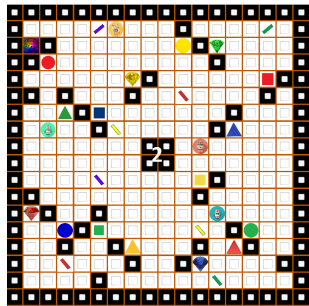


FIGURE – Exemple de deux plateaux possibles.

Remarque

Nous avons un total de 24 plateaux de jeu possible.

Description du jeu

Chargement du plateau de jeu

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

Création du plateau de jeu

Comme vous pouvez le constater ci-dessus, à chaque début de partie le plateau change, cela est du au choix aléatoire des 4 sous-plateaux ,formant le plateau final du jeu.

Remarque

Une rotation des sous-plateaux implique dans deux cas une inversion des ricochets. Ces deux cas représentent la rotation du sous-plateau haut-droit et bas-gauche. En effet, une rotation de 90° , implique le changement de direction de tous les ricochets. En revanche une rotation de 180° , ne change rien à ces derniers.

Description du jeu

But et règles du jeu

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

But du jeu

- Déplacer un robot jusqu'à la case cible choisie, avec un minimum nombre de déplacements.

Règles du jeu

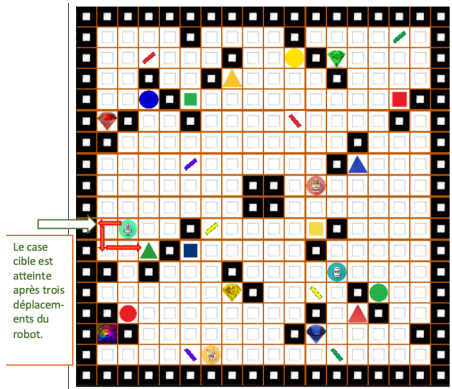
- Les sous-plateaux sont choisis aléatoirement à chaque début de partie.
- Le robot peut se déplacer horizontalement ou verticalement.
- Le robot et la case cible sont de même couleur.
- Les obstacles sont : une case obstacle, ou un autre robot.
- Les ricochets changent la direction du robot.
- Un jeton est choisi aléatoirement.

Description du jeu

Exemple

Exemple

Prenons l'exemple du chemin, que le robot vert doit emprunter, pour arriver jusqu'au triangle vert !



Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

Comment peut-on déplacer le robot jusqu'au but, avec un minimum de coups possibles ?

Robot Ricochet
Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments
techniques

Algorithme A*

Tables de transposition

Conclusion

1 Introduction

2 Éléments techniques
Algorithme A*
Tables de transposition

3 Conclusion

Définition

C'est un calcul de la distance qui sépare chaque nœud du but ,
ceci est pour estimer le meilleur chemin .

Heuristique utilisée

Dans notre implémentation, on a utilisé la distance de Manhattan
comme heuristique.

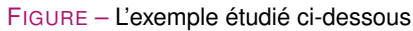
$$H(n) = abs(x) + abs(y) \quad (1)$$

Présentation

L'algorithme A* sert à rechercher un chemin dans un graphe entre deux nœuds : initial et final, en évaluant l'heuristique sur chaque nœud afin de trouver le meilleur chemin.

Exemple

Appliquant A* sur l'exemple donné ci-dessous, en calculant l'heuristique pour chaque nœud.



Heuristique calculée selon la direction du robot				
Positions de robot	Haut	Bas	Gauche	Droite
(10,15)	6	4	2	3
(10,13)	5	/	/	3
(10,16)	6	2	/	/
(11,16)	6	/	0	/

Chemin : Gauche -> Droite -> Bas ->Gauche

On choisit le nœud ayant la plus petite heuristique, les "/"
représente les nœuds déjà visités, donc inutile de recalculer leur
valeurs.

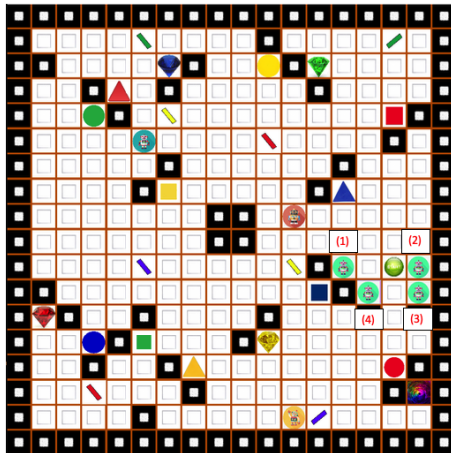


FIGURE – Le chemin obtenu après l'application de A*.

Definition

Une table de transposition est un graphe de positions avec leurs évaluations, ou un noeud déjà visité n'est pas étendu et réévalué.

Utilité

Dans notre implémentation de l'algorithme A*, nous utilisons 2 HashMaps gScore et fScore (current2 startCosts et cheapestCost) qui représentent notre table de transposition permettant de faire gagner en rapidité l'algorithme, puisque ce dernier ne s'appliquera pas sur des nœuds déjà évalués. Et met en priorité les nœuds non visités avec l'heuristique la plus basse, jusqu'à atteindre le but.

Tables de transposition

Déroulement

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

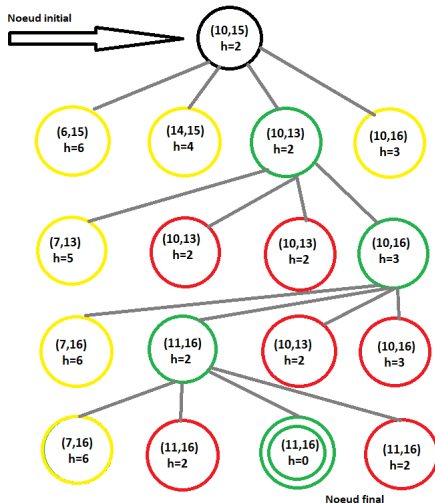
Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion



état de la
priorityQueue:

Première itération

(10,13)
(10,16)
(14,15)
(6,15)

Deuxième itération

(10,16)
(7,13)

Troisième itération

(11,16)
(7,16)



Remarque

Il est important de comprendre que si un nœud déjà visité ne sera pas ajouté à la PriorityQueue *voir l'exemple ci-dessus*, et que chaque itération repose sur le retrait du premier élément de cette structure puisqu'il a l'heuristique la plus intéressante, et donc ce nœud sera évalué à la prochaine itération.

Robot Ricochet Solver

Introduction

Robot Ricochet Solver,
c'est quoi ?

Problème

Éléments techniques

Algorithme A*

Tables de transposition

Conclusion

1 Introduction

2 Éléments techniques

3 Conclusion

Pour conclure

L'objectif du projet était de développer une application "Java" pour le jeu "Ricochet Robot Solver", tout en implémentant l'Algorithme de recherche de chemin A* en utilisant les tables de transposition, nous avons réussi à relever le défi, tout en ajoutant quelques bonus notamment le *Design pattern Singleton* ou bien l'effet sonore. Merci pour votre attention.