ENSSAT

LANNION

Compte rendu de projet : Algorithme 1

Lode Runner

Julien BOURDET

ENSSAT

1ère année - Informatique

Lannion, November 2024

Table des matières

1 Introduction **2**

2 Préliminaires **3**

2.1 Tas-Min 3

2.2 Algorithme A* 3

3 Stratégie **4**

4 Conclusion **5**

1

Introduction

Dans le cadre du module Algorithme 1, nous avons pu appliquer les concepts étudiés en cours en réalisant un projet. Ce projet consiste à réaliser une IA capable de jouer au jeu Lode Runner. C'est un jeu d'arcade qui se déroule sur une carte en deux dimensions où le joueur doit récupérer des bonus tout en évitant des ennemis.

L'objectif du projet est de développer une IA capable de pouvoir terminer n'importe quel niveau du jeu. Cependant, des limitations ont été imposées : l'IA ne connaît pas les actions futures des ennemis et ne peut pas mémoriser des informations d'un tour à l'autre. Ce cadre impose une stratégie approfondie sur la position actuelle afin de ne pas faire un mouvement qui serait mortel plus tard.

Au cours du développement, j'ai rencontré plusieurs défis . L'absence de mémorisation empêche d'utiliser une stratégie qui prend trop de temps de calcul ou de pouvoir créer une stratégie sur plusieurs tours, tandis que ne pas connaître les prochains coups des ennemis empêche d'explorer un "arbre des positions". Par ailleurs, l'implémentation en C, avec ses contraintes de gestion manuelle de la mémoire, a ajouté une dimension technique non négligeable.

Malgré ces contraintes, j'ai pu élaborer une IA fonctionnelle et performante (au moins sur les niveaux disponibles).

Ce compte rendu présente ma démarche de développement, la stratégie utilisée, et les résultats obtenus.

2

Préliminaires

Avant de commencer à détailler la stratégie utilisée, il est nécessaire de présenter certaines notions sur lesquelles elle repose.

2.1 Tas-Min

Un tas-min est une structure de données qui permet de stocker un ensemble d'éléments et de les récupérer dans un ordre particulier. C'est un cas particulier d'une file de priorité, où l'élément le plus petit est toujours en tête de file.
Notre tas-min est implémenté sous forme d'un tableau d'entiers, où chaque élément est un nœud de l'arbre binaire représentant le tas. Les indices des éléments sont choisis de manière à ce que le fils gauche de l'élément à l'indice i soit à l'indice $2i + 1$ et le fils droit à l'indice $2i + 2$.
Pour maintenir la propriété de tas-min, nous avons besoin de deux fonctions : `heapify` et `insert`. La première fonction permet de rétablir la propriété de tas-min après une suppression d'élément, tandis que la seconde permet d'ajouter un élément au tas.

2.2 Algorithme A*

L'algorithme A* est un algorithme de recherche de chemin dans un graphe pondéré. Il est basé sur l'algorithme de Dijkstra, mais utilise une heuristique pour guider la recherche. L'algorithme A* est utilisé pour trouver le chemin le plus court entre un nœud de départ et un nœud d'arrivée dans un graphe.

3

Stratégie

4

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

This page intentionally left blank.