

MARIOBOT PROJET G

2013-2014



SIDORENKO KONSTANTIN
GR 1

Sommaire

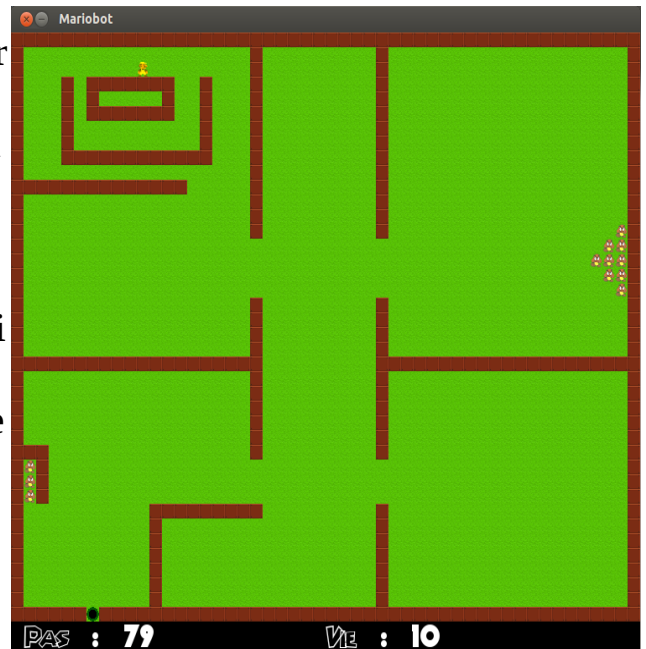
-Introduction.....	Page 3
-Algorithme utilisé(s).....	Page 4
-Difficulté(s) rencontrée(s).....	Page 6
-Conclusion.....	Page 7
-Les suggestions.....	Page 8

Introduction

Le projet de cette année, était de réaliser un robot étant capable de récupérer un extincteur (position connue) situé le long d'un mur puis d'aller éteindre un feu dont la position était inconnue.

Donc il fallait adapter un algorithme qui allait permettre au robot d'accéder à l'extincteur comme ; le parcours « main droite ou gauche », le parcours en largeur, le A*...

Après avoir récupéré l'extincteur, il fallait que le robot recherche le feu pour pouvoir l'éteindre. Cela nécessitait l'utilisation d'un second algorithme.



Tout en prenant en compte le nombre de points de vie du robot, qui diminuait en fonction de la proximité du foyer du feu.

Le programme devait créer le tableau à deux dimensions en fonction de la première ligne du fichier texte qui indiquait la hauteur et la largeur de l'appartement. Et ensuite le remplir à partir du fichier texte.

L'étudiant devait rajouter un paramètre d'exécution, permettant au programme de soit se lancer en mode texte, soit en mode graphique.

Algorithme(s)

Pour trouver l'extincteur :

Vu que la position de l'extincteur était connue, j'ai choisi d'utiliser un algorithme de pathfinding (recherche du chemin le plus court), je me suis tourné vers l'algorithme de Dijkstra. En effet, cette méthode trouve, à coup sûr, le chemin si le but existe (ici un extincteur) même s'il peut être moins efficace que certains algorithmes comme le A* car celui de Dijkstra va analyser une plus grande surface (un peu comme un sonar).

Il fonctionne de la manière suivante :

-Il commence son analyse par les 4 cases entourant la case de départ. (droite, gauche, en bas et en haut pour mon programme) Si la case est valide donc ici un espace ou le feu, alors il l'enregistre dans un tableau de type « coordonnées » donc les positions x, y et le type de la case.

	4	
2	R	1
	3	

-Quand les 4 cases autour ont été vérifiées, alors il repart de la première case enregistrée dans le tableau et vérifie les 4 cases autour de celle-ci, ensuite la seconde position du tableau (ici le numéro 2) et ainsi de suite.

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

			24			
		22	12	17		
	20	10	4	7	15	
18	8	2	R	1	5	13
	19	9	3	6	14	
		21	11	16		
			23			

De cette façon, il est certain que l'algorithme analysera toutes les cases possibles jusqu'à la sortie, si un chemin est possible. Chaque fois qu'il analyse une case, il la remplira d'un point pour éviter de retomber dessus. Une fois qu'il a trouvé le point de sortie, il remonte le chemin qu'il a parcouru jusqu'à la case de départ et cela lui donne le chemin le plus court pour se rendre à la sortie.

Les difficulté(s)

J'ai eu une difficulté majeure liée au fait de m'être trop basé sur le tableau à 2 dimensions. Par exemple ; pour trouver le feu, il me fallait placer des checkpoints mais si le checkpoint était dans le feu :

- soit je l'effaçais en réécrivant le feu dans le tableau pour qu'à l'affichage on voit le feu mais alors, j'effaçais le checkpoint donc on risquait de ne pas tomber dans le feu.

- soit je laissais les checkpoints mais alors, à l'affichage voir une case vide à la place de celle du feu car lors de l'affichage du tableau quand le caractère est un 'h', j'affichais un espace .

J'ai donc enregistré le feu dans un tableau de type « coordonnées » ce qui m'a permis dans un premier temps de cacher les checkpoints à l'affichage. Quand les coordonnées étaient égales à celle d'une des cases du feu alors j'affichais la valeur du feu et ainsi j'augmentais la coordonnée en x de 1 lors de la lecture du tableau pendant l'affichage. Mais aussi pour afficher l'ancienne position du robot s'il était passé dans le feu. Et dans un second temps de pouvoir animer mon feu lors de la partie graphique.

J'ai aussi buté sur la partie où il fallait trouver le feu. Après en avoir parlé avec des élèves bien avancés dans le projet, l'idée des checkpoints m'est parue bonne et pas si difficile à mettre en place. Donc j'ai opté pour celle là.

Conclusion

Ce projet m'a appris en peu de temps beaucoup d'enseignements ; tant au niveau de la programmation, que du temps à gérer et aussi de l'entraide en étudiants.

Pour la partie programmation, c'est surtout les tableaux de structure que j'ai trouvés sur internet, qui sont bien pratiques. Mais aussi les `#if defined` pour pouvoir « adapter » son programme à d'autres systèmes d'exploitation et les `#ifndef XXXX_H` pour pouvoir utiliser différents « header ». Et enfin ce travail m'a permis de parfaire chaque compétence acquise lors des différents cours d'informatique (même si je n'ai pas réellement utilisé d'arbre).

Les suggestions

Au niveau de la programmation, c'est un bon test, qui nous fait revoir toute les compétences vues depuis le début de l'année.

J'aimerais peut-être apporter une petite difficulté, qui serait de rajouter un étage (que j'essaye de réaliser).

Il serait plus intéressant que ce projet puisse être mené à bien en dehors de la période de révision des partiels.

