

PROJET : PUISSANCE-4

I. Introduction

L'assembleur MIPS est basé sur une architecture registre-registre, elle ne peut travailler que sur des données dans le banc de registres. MIPS est principalement utilisé dans les consoles de jeux vidéo. C'est pourquoi dans le cadre de notre projet nous avons essayé d'illustrer au mieux les principes propres de l'assembleur MIPS en implémentant le projet Puissance-4.

II. Puissance-4

1- Règles du jeu :

Puissance 4 est un jeu de stratégie combinatoire abstrait pour deux à quatre joueurs. Pour gagner, le joueur doit aligner une suite de 4 jetons de même couleur sur une grille comptant 6 rangées et 7 colonnes.

Chaque joueur dispose de 21 jetons d'une couleur.

Tour à tour, les deux joueurs placent un jeton dans la colonne de leur choix, le jeton coulisse alors jusqu'à la position la plus basse possible dans ladite colonne à la suite de quoi c'est à l'adversaire de jouer.

Le vainqueur est le joueur qui réalise le premier un alignement (horizontal, vertical ou diagonal) consécutif d'au moins quatre jetons de sa couleur. Si, alors que toutes les cases de la grille de jeu sont remplies, aucun des deux joueurs n'a réalisé un tel alignement, la partie est déclarée égalité.

2- Personnalisation du jeu :

Le code initial du jeu nous a été fourni par les enseignants. Celui-ci comprenait les fonctionnalités de base du jeu. Cependant, quelques fonctionnalités supplémentaires devaient être ajoutées afin de perfectionner le jeu, le rendre plus vivant et original.

III. Structure du programme

Au début, nous avons travaillé chacun de notre côté pour toucher à toutes les étapes du projet. Puis, nous avons mis l'ensemble des recherches en commun et choisi les fonctions les plus pertinentes. Par la suite, nous avons défini les parties à traiter par chacun. Setayesh s'est chargée de la partie concernant du relancement du jeu (PalyerWon) ainsi que

La partie concernée aux calculs des positions exactes du jeton (Update Record). King s'est chargé la partie WinCheck et option choix de la couleur.

Initialement, il fallait définir les phases par lesquelles passer pour une amélioration optimale du jeu. Pour ce faire, nous avons commencé par faire des schémas pour avoir une vue du début à la fin des lignes à coder. Puis des tests ont été fait pour avoir un résultat réel et déterminer les améliorations à apporter.

Voici quelques résumés généraux sur des fonctions fournies :

- La fonction UpdateRecord :

Détermine la position exacte où placer le jeton et met à jour l'état du jeu en mémoire, puis renvoi la position de jeton.

Pour cela, il faut vérifier si l'index est entre [1 ...7], puis il faut trouvez la prochaine ligne vide, ensuite, il faut bien ajouter numéros du joueur.

- La fonction WinCheck :

Détermine si le dernier jeton joué a permis de gagner. Pour cela, il faut vérifier des trucs vérifiant la ligne horizontale, la ligne verticale, la diagonale avant, la diagonale arrière et aussi vérifier si tout le plateau est bien rempli.

Pour cela, il faut vérifier soigneusement les positions (i.e verticale, horizontale et diagonale).

- La fonction GameTie :

Affiche l'égalité et arrête de jeu.

Pour cela, on demande à l'utilisateur s'il veut relancer le jeu ou non.

Affiche le gagnant et arrête de jeu.

Pour cela, il faut effacer les jetons du tableau puis relancer le jeu.

IV. Conclusion

Ce projet a été très enrichissant car nous avons dû réaliser un certain nombre de recherches avant même de pouvoir débiter la programmation. Nous avons tout de même rencontré des difficultés. La première était 'sur l'effacement du tableau contenant des jetons pour pouvoir relancer le jeu et le deuxième c'était sur les calculs des positions mais en tout cas, avec des recherches et de la réflexion ensemble, on a bien trouvé des solutions.

PUISSANCE-4

AVEC TOUTES NOS SALUTATIONS,

SEYEDEHSETAYESH GHAMAT ET KING LAU