Langage C : Puissance 4

Sommaire :

Structure du projet

Le makefile

L’interface utilisateur

Choix sur les IA

-facile

-moyen

-difficile

Choix de la sdl

-« deux page »

-les bouttons

-print des cercles

Remarque sdl :

-SDL\_Delay et SDL\_WaitEventTimeout

Les tests

Structure du projet :

// a recup avec make arbo a la fin

Le makefile

<make install> : Avant d’exécuter la commande « make » on peut exécuter la commande « make install » pour vérifier que les packs nécessaire a la compilation du projet soient bien installer sur la machine.

<make> : Pour compiler le projet. Crée un dossier « obj » où tout les .o seront stocker et le .d correspondant qui liste les dépendances du fichier tout en gardant arborescence du dossier src. Il suffit d’entrer la commande <./bin/exec> pour lancer le programme ou <./bin/execTest> pour les tests.

<make run> : compile le programme avec un <make> et exécute immédiatement l’exécutable du programme.

<make runtest> :compile le programme avec un <make> et exécute immédiatement l’exécutable des tests.

<make arbo> : Donne l’arborescence depuis le dossier Puissance\_4.

<make memory> :qui exécute valgrind sur l’exécutable du programme avec les options qui vont bien.

<make clean> : Supprime le dossier « obj ».

<make mrproper> : Supprime le dossier « bin » ou sont stocker les executable.

<make DEBUG=0> : Pour compiler en débogage.

L’interface utilisateur

Si on suit le chemin Puissance\_4/src/interfaces, on y trouvera un fichier userInterface.h. Ce fichier gère l’interface utilisateur, ce qui est relatif au nombre de ligne/colonne du plateau de jeu. L’énumération Player défini le symbole dans une case du plateau de jeu, qui est vide par default, croix ou rond selon la personne qui joue. La structure Bouton pour les bouton de la SDL, un bouton est un rectangle avec une couleur. La structure userInterface garde la fenêtre, le renderer et les boutons de la SDL. Le choix fait et que si on est en mode texte ces 3 arguments sont nuls, dans nos fonctions communes aux interfaces de jeu SDL ou texte on a juste à regarder si une fenêtre est défini ; si oui on est en mode SDL, sinon texte. Pour finir, la structure Puissance qui gère le plateau de jeu, garde en mémoire le joueur courant, le mode de jeu et la difficulté de l’IA pendant la partie.

Choix sur les IA

-facile

Cette difficulté est comparable à un enfant de 2 ans, il est très difficile … de ne pas gagner contre elle. En effet aucun calcul n’est fait par la machine, elle choisie une colonne au hasard et si elle n’est pas pleine alors elle place son pion

-moyen

Cette fois ci elle ne joue pas sans réfléchir.

Elle suit ces étapes :

- Si elle peut gagner la partie, elle joue le coup gagnant

- Si elle peut bloquer un coup gagnant de l’adversaire, elle joue ce coup

- Sinon elle joue le coup que l’adversaire aurait pu jouer c’est-à-dire un coup qui aurait aligné un grand nombre de pion si l’adversaire l’avais joué en s’assurant que son adversaire ne gagne pas en plaçant un pion au-dessus du sien ensuite. Toute fois si plusieurs choix sont possibles pour aligner beaucoup de pions, elle choisit une colonne aléatoirement entre celle-ci

Exemple : J1 joue dans la colonne 2, l’IA va jouer dans la colonne 1, 2 ou 3

- Dans le cas ou le plateau de jeu est bien rempli il se peut qu’elle refuse de jouer car cela pourrait emmener son adversaire a gagner, ou que là ou son adversaire aurait aligné le plus de pions soit dans une colonne pleine, dans ce elle cherche une colonne ou elle peut jouer pour faire avancer le jeu.

-difficile // a faire

Choix de la sdl

-« deux page »

Le mode de jeu SDL lance une fenetre où les regles du jeu sont affichés et comment fonctionne les boutons de la page suivante afin que l’utilisateur puisse jouer sans chercher comment fonctionne les boutons.

-les boutons

Les boutons de mode de jeu et de difficulté de l’IA sont plus petit que les 2 autres afin de les mettre en valeur s’ils sont sélectionnés. Si on choisie un mode de jeu avec IA, il est nécessaire de choisir la difficulté de l’IA. Le plateau attend l’instruction pour placer le premier pion une fois que les modes sont choisis

-print des cercles

Plutot que de s’embeter avec une surface pour afficher le cercle, gerer le pointeur, faire des acalcul superflux pour l’allocation de mémoire ; on trace simplement des traits de couleur depuis le centre de la case.

Remarque sdl :

-SDL\_Delay et SDL\_WaitEventTimeout

Le probleme de la fonction SDL\_Delay est qu’aucune action de l’utilisateur n’est faite quand le programme lance cette fonction. Pour cela nous avont implémenté une fonction pauseSDL utilisant SDL\_WaitEventTimeout.

\*\*SDL\_WaitEventTimeout return 1 on success or 0 if there was an error while waiting for events; call SDL\_GetError() for more information. This also returns 0 if the timeout elapsed without an event arriving.\*\*

Malgès cela il y a tout de meme un bug venant de cette fonction, en effet meme apres le temps, passé en parametre, attendu ; le programme reste en attente d’un événement au lieu de retourner 0, un mouvement de souris est necessaire pour que l’IA joue.

Les tests : //// a reecrire

Pas de test d’interface car c’est que des printdemande a l’utilisateur (mode text) ou des mise a jour d’affichage (sdl)