Candigit Crush

I. Modifications apportées

Nous avons ajouté au projet minimal plusieurs améliorations pour apporter plusieurs nouvelles fonctionnalités. Parmi elles, nous avons :

(Fonction) **TimerThread**: Procédure asynchrone pour mesurer le temps de jeu.

(*Structure*) **Timer_data** : Structure contenant les variables de score, temps de début de partie, temps de fin de partie et le booléen qui informe si la partie est finie ou non.

(Fonction) LifeCounter: Fonction permettant d'enlever les vies. Le joueur perd une vie après chaque déplacement où il ne marque pas de point.

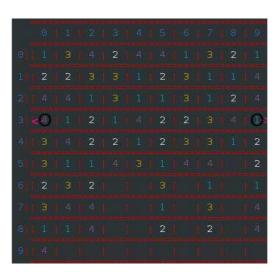
(Fonction) Menu : Attend l'entrée de la ligne, de la colonne et la direction de la case à permuter. Redemande l'entrée de l'utilisateur si celui-ci tape un mouvement incorrect.

(Fonctions) **BonusRow** et BonusCol : Compte à la fin de la partie le nombre de lignes et colonnes vides pour rajouter un bonus de points au joueur.

(Fonction) MakeAMove: Si l'on veut déplacer la première valeur rencontrée sur une ligne vers la gauche elle sera permutée avec la dernière valeur rencontrée sur la ligne. Inversement pour la dernière valeur rencontré qui sera permutée avec la première valeur rencontrer si celle-ci est déplacée à droite.

Les valeurs présentent sur la première ligne, après déplacement vers le haut seront permutées avec la dernière valeur correspondante a la colonne. De même pour la dernière valeur sur une colonne, si déplacée vers le bas elle sera permutée avec la première valeur de la colonne correspondante.

Si le joueur tente de déplacer une valeur sur une case vide, le mouvement ne sera pas effectué.



(Fonction) **ComputeScore** : Fonction permettant de calculer le nouveau score après chaque mouvement.

(*Main*) **ppal**: Initialise la grille de jeu ainsi que les variables compteur de vies (LifeCpt), colonnes vides (EmptyCol), lignes vides (EmptyLine). Une boucle for dans ppal permet de savoir si le temps est écoulé où voir s'il reste ou non des vies. Elle permet aussi de compter le nombre de nouvelles colonnes et lignes vides en s'aidant des fonctions BonusRow et BonusCol.

(Fonction) Color: Active le mode couleur dans le terminal.

(Variable) KBackground: Couleur de fond du jeu (noir).

II. Test des possibles erreurs

Nous avons énuméré les différentes erreurs possibles qu'il pouvait y avoir, pour voir que nos fonctionnalités.

Pour voir si le compteur de vies marchait, nous avons fait exprès de perdre la totalité de nos vies.

Pour voir si le compte à rebours était fonctionnel, nous l'avons laissé s'écouler entièrement. Pour voir si le score (avec notamment les fonctions BonusRow et BonusCol) était fonctionnel, nous avons compté à la main en même temps que la machine comptait à l'aide du programme.

Pour voir si les mouvements étaient fonctionnels, nous avons joué plusieurs parties. Pour les entrées utilisateurs, nous avons tapé toutes les erreurs possibles que pourrait faire l'utilisateur (exemple : entrer une string à la place d'un caractère, un caractère à la place d'un nombre...).