PROJET SOKOBAN

**GRILLE:**

creaGrille : list

*(Creation de liste imbriquées)*

affiche : print(str)

*(Affiche proprement la grille)*

*affiche\_level : print(str)*

*(Affiche proprement un niveau)*

dimension :int, int

*(Renvoie la taille de la grille, longueur/largeur)*

coords : int, int

*(Renvoi les coordonnées de la case suivante)*

numCase : -

*(Transofrme les i/j en cases)*

trouveNumCase: int

*(Renvoie le numéro de la case sélectionnée)*

val\_case: char/int

*(Renvoie la valeur de la case sélectionnée par son numéro de case)*

identite : bool

*(retourne vrai si la case correspond au caractère selectionné)*

count : int

*(renvoie la quantité du symbole sélectionné dans la grille)*

trouve\_ligne : int

*(renvoie le numéro de la ligne sélectionnée en fonction de son numéro de case)*

trouve\_colonne : int

*(renvoie le numéro de la colonne sélectionnée en fonction de son numéro de case)*

trouve\_symb : int, int

*(renvoie les numéros de cases du symbole sélectionné)*

retourneSymb : char/int

*(retourne le contenu d'une case de la grille en fonction de ses coordonnées)*

**LEVEL** *(hérite de Grill*e):

import : -

*(importe le fichier .sok pour pouvoir le lire [with open("data.txt", "r") as fichier])*

conversion : list (listes imbriquées)

*(convertit le contenu du fichier .sok en liste imbriquée)*

isPlayer : bool

*(renvoie vrai si la case séléctionnée est le joueur, en fonction de son numéro de case)*

isBox : bool

*(renvoie vrai si la case séléctionnée est une boite, en fonction de son numéro de case)*

isWall : bool

*(renvoie vrai si la case séléctionnée est un mur, en fonction de son numéro de case)*

isFree : bool

*(renvoie vrai si la case séléctionnée est vide, en fonction de son numéro de case)*

isActive : bool

*(renvoie vrai si la case séléctionnée est un emplacement de boite, en fonction de son numéro de case)*

isActivated : -

*(transforme les différentes variables de victoire en 0 ou 1 en fonction de la présence ou non d'une caisse sur un emplacement. Peut être vérifié à l'aide de Level.isActive)*

victoire : bool

*(renvoie vrai si toutes les conditions de victoire sont remplies, donc toutes les variables de victoire à 1)*

mouvPers : -

*(hérite de Grille.coords et Level.isFree, determine la direction dans laquelle le personnage se déplace, donc quelle numéro de case de la grille est désormais occupé par le symbole personnage et comment l'ancien emplacement redevient vide.)*

mouvCaisses : -

*(hérite de Grille.coords et Level.isFree, determine la direction dans laquelle le boite se déplace, donc quelle numéro de case de la grille est désormais occupé par le symbole boite et comment l'ancien emplacement devient occupé par le personnage.)*

ajouter une fonction de retour à la position precedente

ajouter une autre fonction pour reset le niveau

solve : -

*(algorithme de résolution du niveau)*

**GESTION:**

level : list

*(renvoie une liste de tous les fichiers .sok disponibles)*

score : int

*(renvoie le score du joueur; aka nombre de niveaux complétés*Une image contenant table

Description générée automatiquement*)*

*http://informatique.umons.ac.be/ftp\_infofs/2008/Hoste2008-memoire.pdf*

Mars :

- Score du jeu :

Un score qui s’incrémente en fonction des mouvements du joueur, le but étant de réussir la suite de niveaux avec le moins de déplacements possibles

- Sons du jeu :

Une musique d’ambiance et des sons en fonction des actions du joueur (un son pour la validation d’un bouton ?), bouton pour couper le son

- Ecran de fin du jeu :

Une fois au bout de la liste des niveaux le jeu ne crash plus mais renvoi un ecran affichant le score du joueur et la possibilité de revenir au menu ou quitter le jeu

- Animation de victoire (par niveau?)

Une lègere transition entre chaque niveau

- Mise en forme du jeu

Soigner les phrases en jeu comme les tips ou le titre, améliorer le menu pour qu’il paraisse moins vide

- Mise en forme du code

Factorisation du code et du fichier de travail, cohérence du typage (soit uneVariable, soit une\_variable, pas les deux)

- Algo de résolution

- Rapport

- Soutenance