**DÉCOUPAGE FONCTIONEL : UN LABYRINTHE (SUJET1)**

Contributeurs : Truong Son NGO Tien Thanh NGUYEN

Thanh Nguyen PHAM Guokuang DENG

1. **LISTE DE FONCTION ESSENTIEL :**
2. **FAIRE GUI AVEC TKINTER :**

* **Tkinter (Tk interface)** est un module intégré à la bibliothèque standard de Python, permettant de créer des interfaces graphiques. Grâce à ce module, on peut créer aisément des fenêtres, des widgets tel que les boutons, les labels, les sous-titres, etc
* Les widgets et les fonctions que l’on utilise dans le code:
* **Button(), Label(), Frame()** permettent de créer un interface du jeu.
* **bind()**: permet de lier une fonction *callback* à un événement sur le widget. Dans le jeu, on utilise cette fonction pour mettre à jour la liste de données, supporter des joueurs à marquer un chemin, ouvrir d’autres zones d’affichage.
* **Canvas()** permet de dessiner le labyrinthe et de tracer le chemin entre 2 points marqués sur la zone d’affichage.

1. **FAIRE UN TIMER :**

* **after() :** un attribute de tkinter qui permet de faire appel une fonction call-back après un quantité du temps, donc, aide à lancer un chronomètre en jouant
* **time :** un bibliothèque avec des fonctions utiles comme Time() pour recevoir des informations de temps actuelle.

1. **LIRE ET EXPORTER DES DONNEES :**

* **csv :** un bibliothèque aide à lire des fichiers csv où toutes les informations relatives aux labyrinthes sont stockées.
* **panda :** une autre bibliothèque qui peut gérer des fichiers csv plus avancement, donc permet de stocker les informations relatives à la progression du joueur, telles que les scores les plus élevés, le nombre de fois où ils ont essayé chaque labyrinthe, etc.

1. **L’ALGORITHME : TROUVER LE PLUS COURT CHEMIN**

* Il permet de trouver le chemin le plus efficace entre deux points. Il peut être utilisé pour résoudre des problèmes dans divers domaines, tels que trouver le chemin en évitant des pièges donnés à la main ou par la fonction de random
* Il peut être appliqué à différents types de graphes, tels que les graphes non orientés et orientés, pondérés ou non pondérés.
* Il existe plusieurs algorithmes efficaces pour trouver le plus court chemin, tels que Dijkstra, BFS,...

1. **AMELIORER DES EXPERIENCES DES JOUEURS :** 
   * **Pygame :** une bibliothèque multiplateforme qui facilite le développement du jeu avec langage de programmation et on l’utilise pour jouer de la musique. Le module très utile pour le faire est **mixer().**
   * **Pillow :** une bibliothèque nécessaire de gérer les images permet de améliorer le qualité de notre jeux
2. **D’AUTRES FONCTIONS / BIBLIOTHEQUES :** 
   * **Random :** une bibliothèque avec un nombreuse de fonction qui sont utiles pour générer des nombres aléatoires. Une fois importée, elle permet créer la matrice où des éléments aléatoires sont placés

* **Math**: un module standard qui fournit des fonctions mathématiques avancées. Il permet aux programmeurs Python de réaliser des calculs mathématiques plus complexes que ceux possibles avec les fonctions mathématiques de base.

1. **LISTE DE FONCTIONS PREVOIR :**

class **Fenetre\_jeu**():

    def **init**(self):

        """initialisation des variables"""

    def **creer\_widgets**(self):

        """créer le premier interface avec Tkinter"""

    def **ouvrir\_fenetre\_Canvas**(self, *event*):

        """créer un TopLevel pour dessiner un canvas du jeu"""

    def **lire\_fichier**(self, *level*):

        """lire le fichier csv qui contient des informations du niveau *level* """

    def **dessine\_labyrinthe**(self, data):

        """créer un Canvas pour dessiner une zone de labyrinthe"""

    def **run\_timer**(self):

        """être utilisée avec after() pour faire le chronomètre """

    def **afficher\_résultat**(self):

        """affichier le chemin plus court sur le Canvas"""

    def **sauvergarder**(self):

        """sauvergarder les scores du joueur"""

    def **creer\_matrice\_aléatoire**(self):

        """créer une matrice aléatoire du labyrinthe """

def **calcul\_chemin\_plus\_court**(self, map):

"""trouver le plus court chemin avec le labyrinthe entré """

    def **quitter**(self):

        """quitter le jeux"""

def **jouer\_musique**(self, song):

        """jouer la musique"""

    def **pause\_musique**(self, id):

        """faire une pause de la musique"""