

Rapport du projet d'informatique

Projet : TAKUZU

TABLE DES MATIERES :

I.	Introduction / Présentation du TAKUZU	1
II.	Objectif du projet	1
III.	Répartition du travail.....	1
IV.	Organisation du programme.....	2
V.	Rôle des fonctions.....	2
VI.	Difficultés rencontrées.....	3
VII.	Manuel utilisateur.....	4
VIII.	Conclusion.....	6
IX.	Annexes.....	6

I. Introduction / Présentation du TAKUZU :

Lorsque nous avons choisi le Takuzu comme projet pour la fin du semestre, c'était essentiellement car il nous plaisait à toutes les deux et nous avions déjà quelques pistes sur la façon dont nous pourrions le programmer.

Hélas, aucune de nous n'y avaient jamais jouées. Pour apprivoiser ce jeu inconnu, nous avons d'abord cherché comment y jouer. C'est un jeu assez simple, il se présente sous la forme d'une grille dans laquelle sont placés des 0 et des 1.

Le but étant de remplir les cases vides de la grille par des 0 et des 1 en suivant les règles. Nous avons alors regardé les règles du Takuzu, il en existe seulement trois :

- tout d'abord il faut toujours avoir autant de 0 et de 1 dans une même ligne ainsi que dans une même colonne,
- ensuite il est impossible d'avoir plus de deux 0 ou de deux 1 les uns à côté des autres,
- et pour finir il ne doit pas y avoir deux lignes identiques ou deux colonnes identiques dans le Takuzu.

Les règles du Takuzu sont très semblables à celles du Sudoku. Afin de voir si nous avons bien compris les règles du Takuzu et la façon d'y jouer, nous avons décidé de tester nos nouvelles connaissances avec des grilles qui sont sur internet. Après avoir trouvé un site qui renouvelle ses grilles de Takuzu tous les jours, l'une de nous n'a pas arrêté d'y jouer pendant assez longtemps.

Avec les connaissances que nous venions alors d'apprendre, nous avons commencé notre programmation du Takuzu.

II. Objectif du projet :

Le projet a pour objectif la création d'un programme permettant de jouer au Takuzu avec plusieurs conditions, telles que les règles du jeu et le fait de faire le programme avec des listes de listes (imposées sur le sujet).

L'objectif du projet était en soit assez simple, il fallait créer un jeu que nous ne connaissions pas à partir des compétences que nous avons acquises en cours (TD et TP) et en suivant les consignes qui étaient indiquées sur le sujet qui nous a été fourni.

III. Répartition du travail :

Lorsque nous avons découvert le sujet pour créer le Takuzu, nous n'avions pas tout de suite vu qu'il fallait le résoudre avec deux modes différents. Quand nous l'avons enfin vu, nous avons décidé de nous partager le travail de façon à ce que chacune d'entre nous fasse un mode de résolution.

Seule une fonction que nous avons faite ensemble participe aux deux modes du programme.

Néanmoins, nous n'avons pas travaillé séparément, nous nous retrouvions souvent pour voir l'avancée du programme, et pour résoudre ensemble les problèmes que nous avons pu rencontrer lorsque nous travaillions chacune de notre côté.

IV. Organisation du programme :

Notre programme est organisé en deux grandes parties :

- la première contient toutes les fonctions que nous utilisons dans le programme principal :
 - cette partie contient tous d'abord les fonctions qui servent dans le mode graphique,
 - ainsi que les fonctions qui servent aux deux modes,
 - puis les fonctions qui servent dans le mode terminal.
- et la deuxième contient le programme principal :
 - cette partie contient deux sous-parties :
 - au début l'utilisateur choisit sur quelle grille il souhaite jouer (4, 6, 8 ou 10), ensuite il choisit avec quel mode il souhaite résoudre le Takuzu (terminal ou graphique),
 - le programme principal contient les deux modes pour résoudre le Takuzu et met en place le mode que l'utilisateur a choisi à l'aide des fonctions qui sont dans la première partie.

V. Rôle des fonctions :

Cette partie va être divisée en trois avec une pour le mode terminal, une seconde pour le mode graphique et enfin une commune.

➤ Pour le mode graphique, on utilise cinq fonctions :

- La fonction *creer_case_graphique()*, prend en paramètre nb, elle permet de créer la grille de Takuzu en fonction du niveau que l'utilisateur a choisi et qui est mis en paramètre dans la fonction, car plus le niveau que l'utilisateur a choisi est grand et plus il y aura de cases dans le Takuzu.
- La fonction *charger_grille_graphique()*, prend en paramètre grille, à partir d'un fichier qui a été rentré en paramètre la fonction crée une liste de liste qui contient les 0 et les 1 qui devront être placés dans la grille du Takuzu au début du jeu et les changent de type *str()* en type *int()*.
- La fonction *placer_pieces_graphique()* prend en paramètre nb, lst_lignes, en fonction du symbole que l'utilisateur a choisi pour résoudre son Takuzu(symbole) et qui est en paramètre de la fonction, la fonction affiche dans le Takuzu les 0 et les 1 qui y sont présents en distinguant ceux qui ne peuvent être changés de ceux que l'utilisateur doit trouver. Pour les placer, la fonction se sert du niveau choisi(nb) qui est placé en paramètre de la fonction ainsi que de la liste de listes(lst_lignes) qui contient les 0 et les 1 présents dans le Takuzu, ceux que l'utilisateur ne peut changer en type *int()* et ceux qu'il peut changer en *str()*.
- La fonction *changement_variable_graphique()* prend en paramètre x, y, nb, lst_lignes, elle prend en paramètre les coordonnées du « clic » de l'utilisateur ainsi que le niveau que l'utilisateur a choisi et que la liste de listes qui contient les 0 et les 1 qui apparaissent dans la grille. Grâce aux coordonnées du « clic » et au niveau que l'utilisateur a choisi, la fonction localise la case où il y a eu le « clic » dans la liste de listes et change la valeur qui se trouve dans cette case si c'est un type *str()* (et donc que l'utilisateur a le droit de la changer).

- La fonction *conditions_graphique()* qui prend en paramètre *lst_lignes* (liste de lignes), *lst_colonnes* (liste des colonnes) et vérifie que les trois conditions du jeu sont vérifiées.
- Pour la partie commune il n'y qu'une seule fonction :
- La fonction *creer_colonnes()*, celle-ci prend en paramètre la liste des lignes et crée une liste de colonne pour faciliter la vérification des conditions.
- Pour le mode terminal, on utilise cinq fonctions :
- Il y a la fonction *conditions_terminal()*, celle-ci prend en paramètre *nb* (le niveau de la grille), *liste_lignes* (liste contenant chaque ligne de la grille), *liste_colonnes* (liste contenant chaque colonne de la grille) et permet le respect des trois règles du Takuzu (autant de 1 et de 0 sur chaque lignes et sur chaque colonnes, pas plus de 2 chiffres identiques côte à côte , 2 lignes ou 2 colonnes ne peuvent être identiques), à l'aide de la méthode *.count()*, qui permet de compter les occurrences.
 - La fonction *charger_grille_terminal()*, celle-ci prend en paramètre *grille*, qui la charge à partir d'un fichier texte, à l'aide de la méthode *.readlines()* , en ouvrant le fichier en mode lecture (*read*).
 - La fonction *grille_terminal()*, celle-ci prend en paramètre *nb* (le niveau de la grille), qui permet d'afficher une grille à partir d'une liste et d'afficher cette liste sous forme de grille.
 - La fonction *changement_variable_terminal()*, celle-ci prend en paramètre le numéro de la ligne, le numéro de la colonne, le niveau de la grille (4,6,8 ou 10), *liste_lignes* (liste contenant chaque lignes de la grille), qui permet de changer les valeurs choisies par l'utilisateur dans une liste de listes et de retransformer cette liste de listes en liste de chaînes de caractères à l'aide de la méthode *.join()*, qui permet de rassembler les éléments de la liste pour former une chaîne de caractères.

VI. Difficultés rencontrées :

- Pour le mode terminal :
- La première difficulté était d'afficher les numéros dans la grille à partir d'un fichier texte et d'intégrer chaque ligne de ce texte dans une liste de listes et ensuite de l'afficher sur le terminal. (résolu)
 - La seconde difficulté à était lors de la vérification de la grille (si elle a était finie et juste), car il s'agit de liste de listes et qu'au départ on avait écrit les conditions pour une liste de chaînes de caractères. On a donc du changer la liste de listes en liste de chaînes de caractères. (résolu)
- Pour le mode graphique :
- On a rencontré une difficulté dans le changement des variables, on avait oublié que *l'attend_clic()* renvoyait les coordonnées du clic. Hélas quand nous nous en sommes souvenues, on ne savait pas comment faire pour changer la variable à l'aide des coordonnées du clic. On a alors demandé à un de nos camarades de nous aider, il nous a alors expliqué comment on pouvait changer les variables à partir des coordonnées du clic.

VII. Manuel utilisateur :

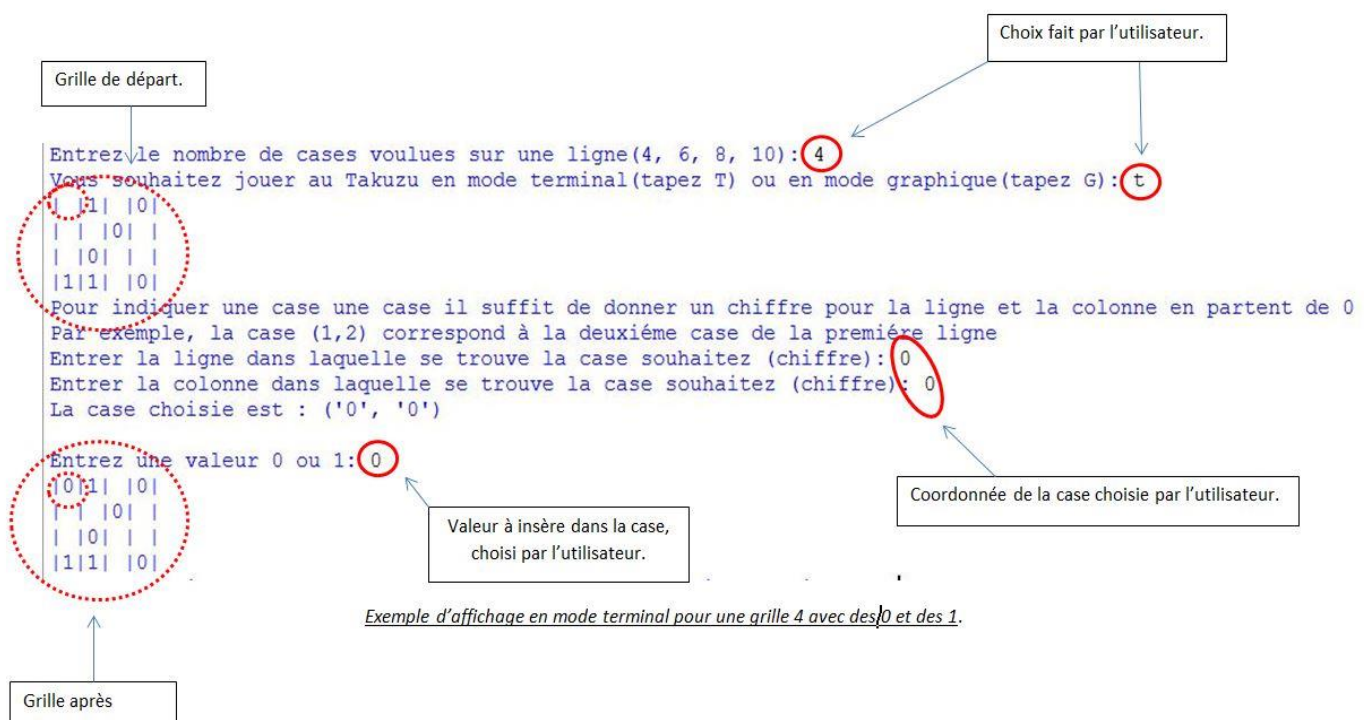
Il faut avoir dans un répertoire quatre fichiers textes, un fichier python qui contient le programme et un fichier upemtk.

Pour commencer à jouer, il suffit d'ouvrir le fichier contenant le programme et de le lancer.

Lorsqu'on le lance, une phrase apparaît pour nous demander quel type de grille on souhaite résoudre, c'est-à-dire, une grille de 4x4, 6x6, 8x8 ou bien 10x10.

Ensuite, une seconde phrase apparaît pour nous demander avec quel mode on souhaite le résoudre, c'est-à-dire, en mode terminal (t) ou en mode graphique (g) :

- Si on choisit le mode terminal, une grille apparaît, suivie d'un paragraphe qui explique comment faire pour nommer les cases, ensuite l'utilisateur doit commencer par donner une valeur pour la ligne à modifier, puis celle de la colonne et enfin il doit entrer la valeur qu'il veut ajouter à la grille et il y a l'apparition de la grille après la modification, ainsi de suite jusqu'à ce que la grille soit finie. Et si la grille est correcte on affiche « Gagné !! », sinon le jeu continu.



➤ Si on choisit le mode graphique, un menu s'affiche, avec trois possibilités :

- Avec des 0 et des 1, on obtient :

Entrez le nombre de cases voulues sur une ligne(4, 6, 8, 10): **6** ←

Vous souhaitez jouer au Takuzu en mode terminal(tapez T) ou en mode graphique(tapez G): **g** ←

Voulez-vous résoudre le takuzu avec:

1:des 0 et des 1

2:des jetons noirs et blancs

3:des cercle et des croix

Entrez votre choix: **1** ←

Choix fait par l'utilisateur.

1				1	
1			0		
			1		0
1	0	0	1		
		1	0	0	1
			0		0

Exemple d'affichage en mode graphique pour une grille 6 avec des 0 et des 1.

- Avec des jetons, on obtient :

Entrez le nombre de cases voulues sur une ligne(4, 6, 8, 10): **8** ←

Vous souhaitez jouer au Takuzu en mode terminal(tapez T) ou en mode graphique(tapez G): **g** ←

Voulez-vous résoudre le takuzu avec:

1:des 0 et des 1

2:des jetons noirs et blancs

3:des cercle et des croix

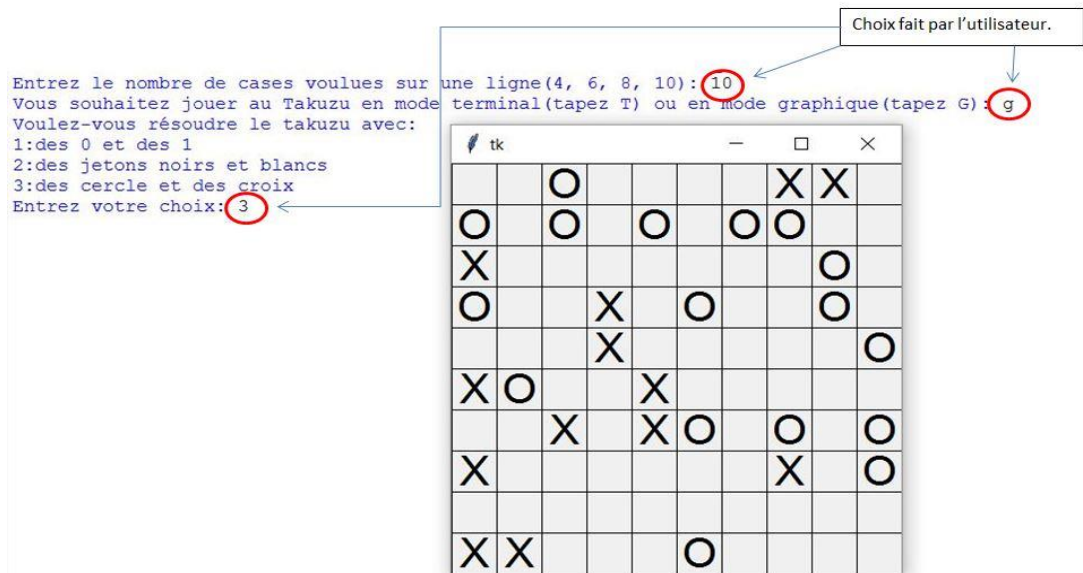
Entrez votre choix: **2** ←

Choix fait par l'utilisateur.

	○						○
						○	
	●		●			●	
	○					○	
	○						●
○	●	●	○	○	●	○	
				●			
				○		○	○

Exemple d'affichage en mode graphique pour une grille 8 avec des jetons.

- Avec des croix et des cercles, on obtient :



Exemple d'affichage en mode graphique pour une grille 10 avec des croix et des cercles.

Pour les trois cas une fenêtre s'ouvre et il suffit de cliquer sur la case dont on veut mettre la valeur souhaitée qui apparaît avec les cliques. Enfin lorsque la grille est correcte tout s'efface et « Bravo ! » s'affiche. Il suffit ensuite d'appuyer sur la fenêtre pour fermer le programme.

VIII. Conclusion :

Comme vous avez pu le constater tout au long de ce rapport, les nombreuses difficultés rencontrées pendant la réalisation de ce projet, ont fait plus au moins l'intérêt de ce projet. En effet, on a pu apprendre de nombreuses notions, et appréhender des techniques vues en cours.

Néanmoins, nous ne sommes pas parvenues à permettre à l'utilisateur d'annuler, un par un, les coups précédents sur le mode terminal.

IX. Annexe :

Liens utilisé pour comprendre :

- ❖ <http://www.20minutes.fr/services/takuzu>
- ❖ <http://www.binairepuzzel.net/>

Fichiers à trouver dans le répertoire :

- ❖ grille4.txt
- ❖ grille6.txt
- ❖ grille8.txt
- ❖ grille10.txt
- ❖ upemtk.py
- ❖ TAKUZU.py