Carnet de bord

Nous avons travaillé en équipe pour développer un jeu de Takuzu. Ce projet nous a posé plusieurs problèmes, que nous avons dû résoudre ensemble en planifiant notre travail et en trouvant des solutions créatives. Dans ce carnet de bord, nous allons vous présenter notre organisation, les difficultés que nous avons rencontrées, mais aussi les solutions que nous avons trouvées pour les surmonter.

Début du projet *(Mardi 2 Mars)*

Ce jour nous avons dans un premier temps analysé le projet, cilbé les différantes taches à faire et nous nous les sommes réparties.

Timothé: responsable de la structure de la grille

- *lecture(nom_fic)*
- *affiche(g)*
- *demande_coup()*
- *coord_coup_joue(case)*
- *joue_coup(g,l,c,v)*

Louis et Lucas: logique du jeu vérification de la validé de la grille

- *grille_remplie(g)*
- *verif_nb_0_nb_1(g)*
- *verif_000_111(g)*
- *verif_ligne_colonne(g)*

Tous ensemble: mis en commun

- *takuzu(nom_fic)*

Pour représenter la grille du Takuzu dans notre programme en python. Nous avons choisi d'utiliser une liste de listes, où chaque sous-liste représente une ligne de la grille. Cette structure nous à permis de manipuler facilement les cases individuelles de la grille.
Pour faire en sort que l'on puisse différancier les cases non modifiables et les cases modifiables, Nous avons crée une liste avec 5 caractères différants.

- Les chiffres 0 et 1 pour les cases non modifiables
- Les chiffres 2 et 3 pour les cases modifiables
- Le chiffre 9 pour les cases vides

Pour codé la fonction *affiche(g)* :

- Nous avons eu besoins utlisé une variable de type "str" nommé alpha (pour alphabet). Cette variable permet d'associer à une lettre un numéro d'une colonne dans la Takuzu
- Nous avons aussi eu besoin de créer un fonction annexe, que nous avons nommé replaceCaractere. Cette fonction prend en paramètre un caractères (un chiffre) de la grille et renvoie le caractères graphique associer. ("*", "0", "1"). Elle est appelé pour chaque chiffre de la grille

Pour codé la fonction *demande_coup()* :

- Utilisation de la fonction input() pour demandé une valeur à l'utilisateur

Pour codé la fonction *coord_coup_joue()* :

- Une fois la fonction affiche codé, cette fonction était simple car nous avons a nouveau utilisé la variable alpha. Pour associer à chaque lettre une colonne

Pour codé la fonction *joue_coup(g,l,c,v)* :

- Dans cette fonction, le paramètre v est l'entier correspondant a la valeur joué. Dans ce cas comme les cases sont modifiables. Nous ajoutons dans la liste le chiffre 2 et 3, le 2 correspondant au 0 et le 3 correspondant au 1.

Avant de codé codé la fonction *takuzu(grille)* qui gère le déroulement de la partie
Nous avons codé une fonction verification qui a pour paramètre (grille, ligne, colonne, valeur)
Cette fonction renvoie 2 elemenst différants sous la forme d'une liste. [boolean, str]. Le premier element correspond à validié de l'action faite par la joueur. Le deuzième element correspond au message d'erreur si validié fausse il y a.
Dans cette fonction 3 verifications sont effectuées:

- Test si la case est présante dans la grille: (A7)
- Test si la case peut être modifié
- Test si la valeur saisi est correct: (0 ou 1)

Cette fonction permet d'évité les principales erreurs, et dans un même temps d'avertir le joueur dans le cas ou l'action qu'il shouaite effectué soit impossible a réaliser.

Pour codé la fonction *verif_nb_0_nb_1(g)* :
Au début nous avons utilisé deux boucle, une pour les lignes et une autre pour les colonnes.
Les deux fonctions était presque identique donc pour optimisé le code nous avons décidé de crée une fonction en lien avec la première *verif_0_1_boucle(g).* Une fonction qui contient la boucle verification des lignes. Puis pour vérifier les colonnes nous avons simplement utilisé la fonction rotation qui permet de passé les colonnes en ligne.

Pour chaque fonction de vérif nous avons crée une fonction boucle associé qui nous permet de ne pas écrie 2 fois la même boucle for dans une fonction. Cela permet une optimisation au niveau du nombre de lignes utilisé or au niveau des ressource utilisé par le programme cela est identique.

Au moment de codé la fonction: *verif_ligne_colonne(g)*, J'ai hésité a vous envoyer ce mail:

> Bonjour,
> Nous arrivons a la fin du projet,
> Mais je ne comprends pas le but de la fonction: *verif_ligne_colonne(g)* En quoi est t'elle utile ?
> Car selon moi les fonctions *verif_000_111(g)* et *verif_nb_0_nb_1(g)* sont suffisantes pour vérifier la validité d'une grille.
> Je ne comprend donc pas l'utilité que de chaque ligne et chaque colonne soit différénte.
> Avez vous un exemple de table qui répondrait True au deux première fonctions et False a la troisième ?

Seulement en relisant la consigne j'ai compris que le jeu avait 3 régles.
J'ai d'ailleurs répondu moi même a ma question en trouvant un grille qui répondait au deux première régles mais pas a la dernière.

Pour codé la fonction takuzu(grille):

- La fonction qui gère le déroulement d'une partie de TAKUZU. Une fois toute les fonctions précédante, la fonction takuzu n'a pas posé de problème pour être réalisé

A la fin le jeu est 100% Fonctionne, aussi bien sur la partie Console que la partie Graphique.
Et pour l'essayer nous avons réussi a complèter la Grille10x10_1

Fin du projet *(Lundi 3 Avril)*