COMPTE-RENDU PROJET SCRABBLE

Binôme:

- > EL BOUCHRIFI Anas
- > DOUMI Aboubakr

Pour lancer le jeu, cliquez sur le fichier main.py

Notre projet se divise principalement en 2 fichiers python, le premier c'est «scrabble.py» où toutes les fonctions sont définies et le deuxième «main.py» où est écrit le programme principal dans lequel on fait appel aux fonctions définies précédemment.

Tout d'abord, le programme demande le nombre de joueurs (Vous pouvez choisir de 2 jusqu'à 4), ensuite il demande le nom de chaque joueur et les stocks dans une liste avec chaque joueur un dictionnaire où existe leur nom, leur main qui est appelée avec la fonction piocher(x, sac) et leur score qui commence par 0 et se modifie tout au long du jeu jusqu'à la fin. Après, le programme affiche le plateau et explique comment seront affichés les bonus (Mot Double – Mot Triple – Lettre Double – Lettre Triple) et donne le premier tour à un joueur au hasard parmi ceux déclarés au début avec la fonction random.choice(joueurs), il lui demande l'action qui veut faire (passer son tour, échanger un certain nombre de jetons de sa main ou placer un mot dans le plateau),

puis le programme exécute l'action voulue; si il veut placer un mot, le mot est filtré pour voir s'il peut être placé, et quand le mot est placé le programme re-pioche les jetons manquants dans la main du joueur courant. Et après la fin de l'action, le programme passe au joueur suivant et lui redemande l'action (passer, placer ou échanger).

Que voulez vous faire? 'passer', 'échanger' ou 'placer' : placer

```
Que voulez vous faire? 'passer', 'échanger' ou 'placer' : placer
Entrez le mot à placer : DAC
Entrez la direction (horizontale - verticale) : horizontale
Entrez coordonnée ligne (0->14) : 0
Entrez coordonnée colonne (0->14) : 0
```

Si le mot ne peut pas être placé ou il est invalide il renvoie une erreur puis redemande au même joueur l'action qui veut faire:

```
.
Pendant le jeu, les bonus sont affichés par des symboles à côté des lettres posées; 'MT' = *, 'MD' = º, 'LT' = +, 'LD' = ♯
Que voulez vous faire? 'passer', 'échanger' ou 'placer' : placer
Entrez le mot à placer : SIM
Mot invalide!
Pendant le jeu, les bonus sont affichés par des symboles à côté des lettres posées; 'MT' = *, 'MD' = º, 'LT' = +, 'LD' = ¤
  voulez vous faire? 'passer', 'échanger' ou 'placer' :
```

Ensuite, quand un joueur essaie de piocher mais y a plus de jetons suffisants dans le sac, la partie se termine et le programme soustrait du score des joueurs les lettres encore en main, puis détecte le gagnant et renvoie son nom et son score final.

Enfin, on a eu de difficultés pour faire une fonction (Bonus), c'est la fonction Bonus de la partie 5 qui sert à détecter si le joueur a compléter un autre ou plusieurs

mots en plaçant le mot voulu. Mais on a réussi à faire les fonctions pour *Aide de Jeu>* dans la partie 7.4 *(dans le fichier scrabble.py);* la fonction qui suggère une liste des mots français jouables à partir de la main du joueur,

et la fonction qui donne le mot avec le plus haut score de la liste précédente, mais on a eu de difficultés pour introduire ces fonctions dans le programme principal.

Exécution des fonctions seules:

PARTIE I:

Les deux fonctions init_jetons() et init_bonus():

						C:\t	Jse	rs\e	elk	oou\	Des	sktop\	TP	INF	101\1	PRO	JET\	SCI	rabk	ole	.ру	===	====	==	=			
>>>	ını	t_](etor	ns()																								
['	''		'	' '	,		,		'		''			''		,		,		,	' '	,	' '	•	' '	′ '	'	
['	''		',	' '	,	' '	,		٠,		''	' '	•	٠,		,		,		,	' '	,	' '	•	' '	′ ']
['	٠,	•	',	' '	,	' '	,	'	٠,	•	٠,	'',	•	٠,	'	,	'	,		,	' '	,	' '	,	' '	, ']
['	٠,	•	',	' '	,	' '	,	•	٠,	•	١,	'',	'	٠,	•	,	•	,	'	'	' '	,	' '	,	' '	, '	•]
['	٠,	•	',	' '	,	' '	,	•	٠,	•	١,	'',		٠,	•	',	•	,	'	',	' '	,	' '	,	' '	, '	•]
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	٠,	'',	• •	٠,	•	',	'	',	'	',	' '	,	' '	,	' '	, ']
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	٠,	'',	• •	٠,	•	',	'	',	' '	',	' '	,	' '	,		, ']
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	٠,	'',	• •	٠,	•	',	'	', '	' '	',	' '	,	' '	,		, ']
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	٠,	'',		٠,	•	',	•	', '	' '	',	' '	,	' '	,		, ']
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	٠,	'',		٠,	•	',	•	',	' '	٠,	' '	,	' '	,		, ']
['	٠,	•	٠, ١	' '	,	' '	,	•	٠,	•	١,	'',		٠,	•	٠,	•	',	'	٠,	' '	,	' '	,	' '	, ']
['	٠,	•	٠, ١	' '	,	' '	',	•	٠,	•	٠,	'',		٠,	•	٠,	•	', '	'	٠,	' '	',	' '	,	' '	, ']
['	٠,	•	٠, ١		,		,	•	٠,	•	٠,	'''',		٠,	•	٠,	•	', '	' '	٠,	' '	,		,		, ']
['	٠,	•	٠, ١		,		,	•	٠,	•	٠,	''',		٠,	•	٠,	•	', '	' '	٠,	' '	,		,		, ']
['	٠,	•	٠, ١		,		,	•	٠,	•	٠,	٠.,		٠,	1	٠,	•	', '	' '	٠,		,		,		, ']
>>>	ini	t b	onus	s ()																								
					,	'LD'	,	•	٠,	•	٠,	1 1,	' N	т',	•	٠,	•	,		٠,	'LD'	,		,	, ,	, '	MT']
['M]	ľ',	,-	٠, ١	, ,	,	'LD'	,	•	!,	' 'LT	!, !,	! !,	' N	Т',	1	,	'	,		,	'LD'	,		,	' ' 'MD'	, ¦	MT']
['M'] ['	!', ',	'- 'МD	', '	' ' ' 'MD'	,		,		', ',	'LT	Ϊ,	' ',	;	Ϊ,	'LD	,	'LT'	,		,		,	' ' 'MD'	,	'MD'	, ¦	,]
['M'] ['	!', ',	'- 'МD	', '	' ' ' 'MD'	,		,		', ',	'LT	Ϊ,	' ',	;	Ϊ,	'LD	,	'LT'	,		,		,	' ' 'MD'	,	'MD'	, ¦	,]
['M] [' ['LI	', ', ',	'MD	', '	'MD'	,	' ' 'MD'	,		', ',	'LT	', ',	' ', 'LD',	' ' 'I	', D',	'LD	,	LT	,		,	' ' 'MD'	,	'MD'	, ,	'MD'	,	LD']
['M] [' ['LI	', ', ',	'MD' 'M' '	', ', ', ', ', ', ', ', ', ', ', ', ', '	MD'	, , , , ,	'MD'		'MD	',',',',	'LT	',,',',',	'LD',	' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	', D', ',	'LD	, , , , , ,	'LT' 'LT'	, , , , ,	MD	, , , , , ,	'MD'		MD'	,	'MD' ' ' ' ' ' ' 'LT'	, , , , , ,	LD']]]
['M] [' ['LI	', ', ',	'MD' 'M' '	', ', ', ', ', ', ', ', ', ', ', ', ', '	MD'	, , , , ,	'MD'		'MD	',',',',	'LT	',,',',',	'LD',	' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	', D', ',	'LD	, , , , , ,	'LT' 'LT'	, , , , ,	MD	, , , , , ,	'MD'		MD'	,	'MD' ' ' ' ' ' ' 'LT'	, , , , , ,	LD']]]
['M] [' ['LI	', ', ',	'MD' 'M' '	', ', ', ', ', ', ', ', ', ', ', ', ', '	MD'	, , , , ,	'MD'		'MD	',',',',	'LT	',,',',',	'LD',	' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	', D', ',	'LD	, , , , , ,	'LT' 'LT'	, , , , ,	MD	, , , , , ,	'MD'		MD'	,	'MD' ' ' ' ' ' ' 'LT'	, , , , , ,	LD']]]
['M'] [' ['LI [' [' ['M]		'MD		'MD'	, , , , , , , ,	'MD'		' MD		'LT		'LD',	'I	', D', ', D', ',	'LD		'LT' 'LT' 'LT'		'MD'		'MD'		' 'MD' ' 'LD'	, , , , , , , ,	'MD' ' ' ' ' 'LT'	,,,,,,,,,	LD']]]]
['M] [' ['L] [' [' [' ['		'MD' 'LT' 'LT'		'MD'	, , , , , , , ,	'MD'		'MD'		'LT ' 'LT ' 'LT '		'LD',	'I	', D', ', D',	'LD'		'LT' 'LT' 'LT'		MD'		'MD'		'MD'		'MD' ' ' ' ' 'LT' ' ' 'LT'	,,,,,,,,,,	LD']]]]]
['M] [' ['L] [' [' [' ['		'MD' 'LT' 'LT'		'MD'	, , , , , , , ,	'MD'		'MD'		'LT ' 'LT ' 'LT '		'LD',	'I	', D', ', D',	'LD'		'LT' 'LT' 'LT'		MD'		'MD'		'MD'		'MD' ' ' ' ' 'LT' ' ' 'LT'	,,,,,,,,,,	LD']]]]]
['M] [' ['L] [' [' [' ['M] [' [' [' L] ['		'MD'		'MD'		'MD'		'MD'		'LT		'LD', 'LD', 'LD', 'LD',	'I	', D', ', D', D', ', D',	'LD'		'LT'		MD MD		'MD'		'MD'		'MD' 'LT' 'LT'	,,,,,,,,,,,,,	LD']]]]]]]
['M] [' ['L] [' [' [' ['M] [' [' [' L] ['		'MD'		'MD'		'MD'		'MD'		'LT		'LD', 'LD', 'LD', 'LD',	'I	', D', ', D', D', ', D',	'LD'		'LT'		MD MD		'MD'		'MD'		'MD' 'LT' 'LT'	,,,,,,,,,,,,,	LD']]]]]]]
['M] [' ['L] [' [' [' ['M] [' [' [' L] ['		'MD'		'MD'		'MD'		'MD'		'LT		'LD', 'LD', 'LD', 'LD',	'I	', D', ', D', D', ', D',	'LD'		'LT'		MD MD		'MD'		'MD'		'MD' 'LT' 'LT'	,,,,,,,,,,,,,	LD']]]]]]]]
['M] [' ['L] [' [' ['M] [' ['M] [' [' [' [' [' [' [' [' [' [' [' [' ['		'MD' 'LT' 'LT'		'MD'		'MD'		'MD		'LT 'LT 'LT 'LT		'LD', 'LD', 'LD', 'LD', 'LD',	'I	', D', ', D', D', ', ', ', ', ', ', ', ', ', ', ', ', '	'LD'		'LT'		MD MD		'MD'		MD'LD'LD'		'MD' 'LT' 'LT' 'LT'	,,,,,,,,,,,,,,,,	LD']]]]]]]]]
['M] [' ['L] [' [' ['M] [' ['M] [' [' [' [' [' [' [' [' [' [' [' [' ['		'MD' 'LT' 'LT'		'MD'		'MD'		'MD		'LT 'LT 'LT 'LT		'LD', 'LD', 'LD', 'LD',	'I	', D', ', D', D', ', ', ', ', ', ', ', ', ', ', ', ', '	'LD'		'LT'		MD MD		'MD'		MD'LD'LD'		'MD' 'LT' 'LT' 'LT'	,,,,,,,,,,,,,,,,	LD']]]]]]]]]

La fonction affiche_jetons(j,x):

PARTIE II:

La fonction init_pioche() qui renvoie le sac:

Les fonctions *piocher(x,sac)*, *completer_main(main, sac)* et la fonction échanger(jetons, main, sac) qui piochent un nombre x de jetons du sac, complètent la main d'un joueur si elle contient moins de 7 jetons, et la dernière fonction échange des jetons de la main contre des jetons du sac:

```
>>> piocher(7,sac)
['E', 'U', 'E', 'I', 'A', 'N', 'E']
>>> complete_hand(['A','B','C'],sac)
['A', 'B', 'C', 'N', 'Z', 'R', 'E']
>>> complete_hand(['E', 'U', 'E', 'I', 'A', 'N', 'E'],sac)
Hand Full
['E', 'U', 'E', 'I', 'A', 'N', 'E']
>>> exchange(['E','U','E'], ['E', 'U', 'E', 'I', 'A', 'N', 'E'], sac)
['I', 'A', 'N', 'E', 'A', 'N', 'T']
>>>
```

PARTIE III:

Les fonctions *mot_jouable* et *mots_jouables*, la première retourne un booléen True si le mot est jouable avec les lettres de la main et la deuxième retourne les mots français qu'on peut jouer avec les lettres en main:

PARTIE IV:

Les fonctions qui calculent les valeurs des mots en dépendant de leurs positions et des bonus:

```
>>> valeur_mot('BEBE', dic, 0, 0, 'horizontale')
24
>>> valeur_mot('BEBE', dic, 5, 8, 'horizontale')
8
```

PARTIE V et PARTIE VI:

Les fonctions de ces deux parties sont testés dans le programme principal *main.py* en jouant au jeu et ils marchent parfaitement.