Projet en python : le 2048 (version programmation objet)

Le principe du jeu

2048 est un jeu en ligne et sur application mobile, créé en mars 2014 par Gabriele Cirulli, et inspiré du jeu de taquin. Le but du jeu est de faire glisser des tuiles sur une grille carrée de taille 4×4, pour les combiner et créer ainsi une tuile portant le nombre 2048. La Fig.2(a) est une impression écran d'une partie en cours de la version en ligne.

La Fig.1 est une impression écran d'une partie en cours de la version en ligne.

Figure 1 – Capture écran de la version en ligne de 2048.

La jouabilité du jeu repose sur l'utilisation de quatre touches du clavier d'ordinateur pour déplacer les tuiles vers la gauche, la droite, le haut ou le bas. La direction choisie est appliquée à toute la grille de manière uniforme, et toutes les tuiles sont envoyées dans cette direction. Une tuile glisse d'une case vers une direction si la case adjacente correspondant à cette direction est libre. Si deux tuiles ayant la même valeur entrent en collision durant un mouvement, elles fusionnent en une nouvelle tuile de valeur double. Par exemple, deux tuiles de valeur 2 qui fusionnent donnent une tuile de valeur 4.

À chaque mouvement, une tuile de valeur 2 ou 4 apparaît dans une case vide de manière aléatoire (la valeur de la tuile aléatoire est 2 dans 80% des cas, et est 4 dans les 20% restant).La Fig.2(b) donne un exemple du résultat de l'application de la direction « *droite* » à la grille (avec fusions de tuiles et apparition d'une tuile aléatoire).

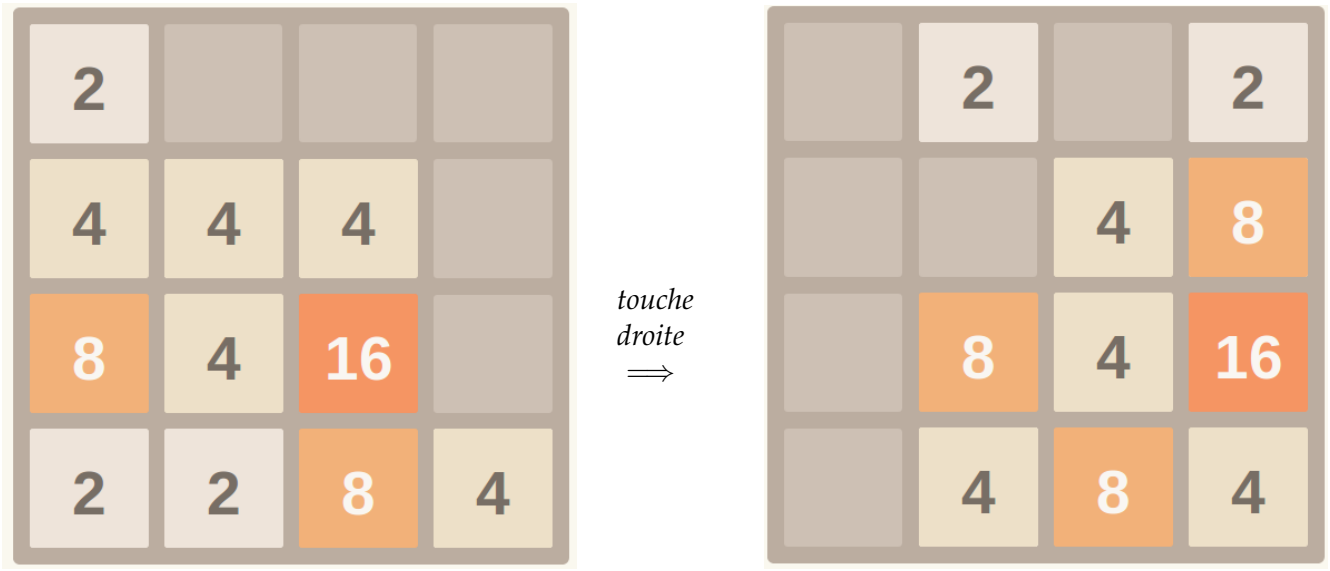
Le score final est obtenu par la somme des valeurs de toutes les tuiles fusionnées durant la partie. Par exemple, si deux tuiles 4 fusionnent en une nouvelle tuile 8, le score est augmenté de 8 points.

Vous pouvez, et c'est même conseillé, essayer (de manière raisonnable et avec modération) la version en ligne et gratuite du jeu à l'adresse `http://jeu2048.fr/`.

Déroulement d'une partie

Le but du jeu est d'obtenir la tuile 2048. La grille initiale est vide, à l'exception d'une case aléatoire occupée par une tuile de valeur aléatoire 2 ou 4. À chaque mouvement, la direction est appliquée à toute la grille et les tuiles de même valeur et rentrant en collision fusionnent. La partie se termine lorsque :

- soit la tuile 2048 apparaît (*victoire*),
- soit la grille est pleine et plus aucune fusion de tuiles n'est possible, peu importe la direction choisie (*défaite*).



(a) Une configuration.

(b) Configuration après mouvement.

FIGURE 2 – Application de la direction *droite* et apparition d'une tuile 2 sur une configuration de la grille.

Travail demandé

Vous devez, en respectant les consignes, en utilisant les formats et en implantant les fonctions imposés par les instructions ci-dessous, réaliser un programme Python, suivant un paradigme de programmation orientée objet, qui permet de jouer une ou plusieurs parties de 2048 à un joueur selon les règles définies ci-dessus. Au début de chaque partie, il sera proposé à l'utilisateur de reprendre une partie en cours, le cas échéant. Après chaque partie, il sera proposé à l'utilisateur de rejouer. Si l'objectif 2048 est atteint, le joueur doit pouvoir continuer la partie. De plus, le joueur peut interrompre et sauvegarder sa partie en cours à tout moment. Après la dernière partie, le programme affichera le score du joueur et lui proposera de le sauvegarder. Le cas échéant, le score du joueur et son nom seront insérés dans un classement contenant les 10 meilleurs scores et stocké sur un fichier externe. De plus, votre code devra permettre, après des modifications rapides de quelques variables facilement paramétrables dans le code, de jouer des parties avec une grille et un objectif différents (par exemple, atteindre la tuile 8192 sur une grille 6×6, où des tuiles 4 apparaissent dans 50 % des cas).

Instruction 0. La grille du jeu devra nécessairement être représentée en objet par une classe `Grille`, qui sera implantée dans un fichier `grille.py`.

Instruction 1. Implanter la fonction / méthode permettant d'afficher une grille de jeu.

Instruction 2. Implanter, dans un fichier `saisie.py`, le code des fonctions permettant toutes les saisies clavier nécessaires au déroulement du jeu (saisie des touches de jeu, saisie du choix du joueur pour rejouer ou sauvegarder son score). La saisie au clavier devra être *protégée* : le programme ne devra pas en aucun cas s'interrompre si le joueur ne saisit pas les touches prédéfinies et nécessaires à la jouabilité du jeu.

Instruction 3. Implanter les fonctions / méthodes permettant d'appliquer les mouvements directionnels sur la grille.

Instruction 4. Implanter les fonctions / méthodes permettant de tester la fin d'une partie.

Instruction 5. Implanter une classe `Pile`, permettant de manipuler des structures de type *LIFO*, qui devra être utilisée pour sauvegarder l'historique de la partie en cours. Celui-ci sera stocké dans un fichier externe `last_games.bk`.

Instruction 6. Implanter une classe `Partie`, permettant de représenter une partie complète de 2048 en objet. La classe `Partie` sera implantée dans un fichier `partie.py`, et devra, en plus de la grille de jeu, contenir le score et l'historique de la partie en cours.

Instruction 7. Créer le fichier principal nommé `2048.py`. C'est dans ce fichier que seront importés les autres fichiers Python à l'aide d'instructions de la forme `from nom_module import *`, que le menu de choix sera affiché, et qu'une partie sera lancée.

Instruction 8. Implanter, dans un fichier `classement.py`, le code des fonctions permettant de consulter un fichier externe `highscores.txt`, créé lors de la première partie, et contenant les dix meilleurs scores, ainsi que le nom correspondant des joueurs ayant réalisé ces scores. Les scores devront être triés en ordre décroissant, à l'aide d'un tri vu en cours (les méthodes et fonctions de tri natives `Python` sont prohibées).

Instruction 9. Améliorer la jouabilité grâce à l'utilisation d'une fonction permettant une *saisie en ligne* : celle-ci permettra de fluidifier le jeu car elle ne nécessite pas que le joueur appuie sur la touche `Entrée` pour valider le choix de sa direction. Demander le code de cette fonction à votre enseignant, puis l'adapter à votre programme.

Consignes et évaluation

Le code source de votre projet est à envoyer par mail intitulé `[NSI] Prénoms - projet 2048` à l'adresse mail `senotprof@gmail.com` **avant le lundi 09/10/2023 à 23h59**. Le code source sera joint au mail dans un format de dossier compressé de type `.zip`, ou mieux, d'archive compressée au format `.tar.gz` (aucun format `.rar` ne sera accepté). Aucun délai supplémentaire ne sera accordé à aucun moment, la date butoir est immuable, et tout retard sera pénalisé : il convient donc de commencer le projet au plus tôt.

Le projet est **à coder obligatoirement en binôme**. La note sera commune au binôme, sauf cas de travail déséquilibré flagrant. Il n'est pas interdit (et c'est même conseillé) de discuter du projet entre les différents groupes, d'envisager à plusieurs les différentes manières d'implanter les fonctions, de s'échanger des idées ou de demander un peu d'aide (à commencer par celle de votre enseignant), mais **la rédaction et le style doivent être propres au groupe et le code doit être intégralement produit par le groupe** (une fonction ou une portion de code bêtement recopiée ne sera pas utile dans un autre contexte que le programme initial, et surtout ne sera ni comprise, ni assimilée, et sera, en plus des risques de sanctions pour fraude et plagiat, une perte de temps pour tout le monde, élève et enseignant). **Tout plagiat fera l'objet de sanctions avec une tolérance strictement négative**.

L'évaluation prendra en compte :

- les erreurs de syntaxe : les fichiers `.py` doivent être directement interprétés sans erreur par la console `Python`
- l'**originalité du code** (dans le sens de « code personnel et individuel »)
- l'implantation des fonctions et structures imposées et le **respect des consignes** (règles du jeu, cahier des charges, consignes sur le code, les structures de données et les spécifications des fonctions, mais aussi les consignes annexes, comme l'intitulé du mail et le format de fichier, ainsi que bien évidemment le respect des délais)
- la correction et la performance du code
- la propreté et la lisibilité du code ainsi que tout ce qui facilitera sa compréhension par le correcteur (commentaires, choix des noms de fonctions et variables, commentaires, « aération » du code et sauts de ligne, commentaires, *etc*)
- la facilité d'utilisation et la jouabilité

Annexe : un exemple

Un exemple d'interface et de déroulement d'un début de partie est donné ci-dessous. Cet exemple a pour vocation à donner des idées et servir d'inspiration, et non à être recopié à l'identique. Les trois grille correspondent respectivement à un enchaînement de touches `bas` et `droite`.

```
***** Bienvenue au 2048 *****

Le jeu 2048 est suffisamment célèbre pour ne pas avoir à présenter les règles et le but du jeu

Rappels des touches de direction :
- z : haut
- q : gauche
- s : bas
- d : droite
- m : quitter la partie en cours
- 9 : annuler le dernier coup

Appuyez sur n'importe quelle touche pour continuer.

Il existe une partie en cours. Reprendre ? (oui/non) non
```

```
 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |   2  |      |   2  |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------

 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |   2  |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |   2  |      |   2  |
|      |      |      |      |
 ----------------------------

 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |   2  |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
 ----------------------------
|      |      |      |      |
|   2  |      |      |   4  |
|      |      |      |      |
 ----------------------------

Vous voulez quitter. Souhaitez-vous reprendre la partie plus tard ? (oui/non)
non

Votre score est de 4
Bravo, vous rentrez dans le classement des meilleurs scores. Saisissez votre pseudo (12 caractères au maximum) : Nash

  *** MEILLEURS SCORES ***
 Rang   Nom             Score
 ----------------------------
   1    Von Neumann    122136
   2    Nash            59020
   3    Turing          53028
   4    Eleve lambda        4

Que voulez-vous faire ?
1) rejouer
2) afficher le classement des meilleurs scores
3) afficher l'historique de la partie ayant eu le meilleur score
4) quitter
Saisir une option parmi 1/2/3/4 : 4
```