Votre premier projet (jeu de la vie Conway) vous a permis de découvrir l'ampleur de la tâche à accomplir pour réaliser un premier jeu en programmation. Il faut, en particulier :

- être organisé ;
- se répartir le travail ;
- bien documenter ses fonctions avec notamment le docstring qui récapitule :
  - les préconditions (quels sont les paramètres : leur type, leur contenu, ...)
  - les postconditions (qu'obtient-on en sortie, sous quel type, ...)
  - une aide pour l'utilisateur de la fonction.
- tenir un planning qui permet de rendre un projet fini ;
- ...

Aujourd'hui, nous vous proposons de réaliser un projet identique où, contrairement au projet précédent, vous êtes plus libres pour modéliser, concevoir, coder le projet (en Python évidemment).
Par contre toutes les exigences listées ci-dessus (organisation, répartition, documentation, planning, ...) restent valables et seront évaluées. Il y aura également un cahier des charges à respecter, explicité ci-après.
Il est bien sûr possible, mais ce n'est pas obligatoire, de réaliser une interface graphique avec `Pygame` (voir Bonus).

1 Le takuzu : Présentation et mise en place

Source ici.
Le Takuzu ou Binairo (« cousin du sudoku ») est un jeu de réflexion consistant à remplir une grille avec les chiffres 0 et 1 par déduction logique.
Principe
Il peut s'agir de grilles allant de 4×4 à 12×12 en général. Chaque grille ne contient que des éléments d'une paire quelconque (le cas le plus courant étant des 0 et des 1), et doit être complétée en respectant trois règles suivantes :

Règle 1 : autant de 1 que de 0 sur chaque ligne et sur chaque colonne ;

Règle 2 : pas plus de 2 chiffres identiques côte à côte ;

Règle 3 : 2 lignes ou 2 colonnes ne peuvent être identiques.

Voici un exemple de grille de départ et de sa résolution :

	1		0
		0	
	0		
1	1		0

Exemple de Takuzu 4x4

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

Le même après résolution

Le but du jeu est de remplir une grille de départ donnée en respectant les 3 règles du jeu ci-dessus.

2 Cahier des charges :

- la grille de jeu sera stockée dans un tableau (liste de listes) :
  - on vous fournit dans le dossier takuzu_eleve.zip des fichiers .txt correspondant à des grilles de départ, où chaque ligne du fichier comporte des 0, 1 ou 9 (0 ou 1 si la case contient un 0 ou un 1 au départ et 9 si la case est vide au départ.
  - en vous aidant des fonctions que l'on vous a proposées dans le projet Conway, vous devrez convertir ces fichiers .txt en liste de listes que vous pourrez manipuler pour la gestion du jeu.
- à chaque tour, le programme demande une case à jouer et le joueur qui joue propose une case qui ne contenait pas d'information au départ. Ensuite, le programme :
  - redemande une case à jouer si la case n'était pas vide en début de partie, le tour recommence ;
  - met à jour le contenu de la grille ;
  - tant que la grille n'est pas complète, le tour suivant est proposé ;
- à la fin de la partie, un message indique si la grille complète proposée est valide.

Exemple d'affichage :

```
====================
Voici la grille de jeu de votre TAKUZU
--------------------
  1 2 3 4 5 6
A *|1|1|*|*|*
B *|0|0|*|1|0
C *|*|*|1|*|*
D 1|*|1|0|0|1
E 1|*|0|1|*|1
F 0|1|*|*|*|*
Dans quelle case voulez-vous jouer ? Répondez A1, A2, A3... B1, B2... A1
Quel coup voulez vous jouer ? Répondez 0 ou 1 0
====================
Grille de jeu TAKUZU mise à jour
--------------------
  1 2 3 4 5 6
A 0|1|1|*|*|*
B *|0|0|*|1|0
C *|*|*|1|*|*
D 1|*|1|0|0|1
E 1|*|0|1|*|1
F 0|1|*|*|*|*
Dans quelle case voulez-vous jouer ? Répondez A1, A2, A3... B1, B2...
```

3 Aide à la mise en oeuvre

Nous proposons un certain nombre de fonctions à écrire pour vous aider dans la conception de votre projet.
Mais vous pouvez faire le choix de partir de zéro et de répartir les tâches à réaliser pour obtenir un jeu fonctionnel comme vous le souhaitez.

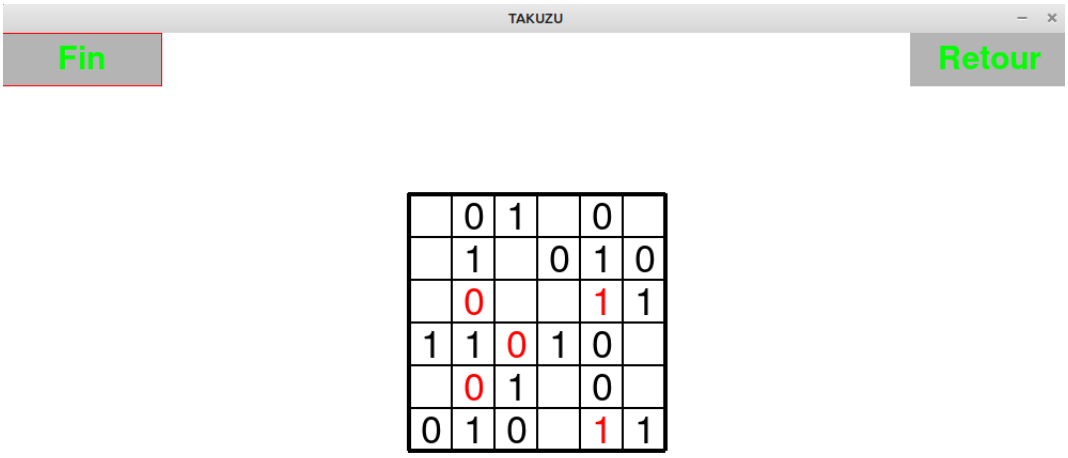
- la grille sera une « liste de listes » dont la dimension est obtenu à partir du .txt. Par exemple, la liste décrivant la situation de départ affichée dans l'exemple de la partie 1 pourrait être :
  `[[9,1,9,0],[9,9,0,9],[9,0,9,9],[1,1,9,0]]`
- vous devrez écrire le code de votre jeu en décomposant chaque opération dans une fonction séparée (par exemple, une fonction qui crée la grille de jeu à partir du .txt (voir

Conway), une pour afficher la liste, plusieurs pour tester si le joueur a gagné, ...) ;
Voici une liste d'éléments du travail à effectuer pour ceux qui ne savent pas par où commencer :

1. Écrire une fonction `lecture(nom_fic)` qui renvoie une liste de listes de la bonne dimension dont les éléments sont initialisés à partir d'un fichier .txt correspondant à une grille de départ d'un takuzu.
2. Écrire une fonction `affiche(g)` qui prend en paramètre la grille du jeu et l'affiche dans la console en remplaçant les 0, 1 et 9 par les caractères adéquats.
3. Écrire une fonction `demande_coup()` qui demande à l'utilisateur la case dans laquelle il souhaite jouer puis la valeur qu'il souhaite jouer et renvoie le couple de réponses.
4. Écrire une fonction `coord_coup_joue(case)` qui prend en paramètre la case jouée par le joueur au format texte et renvoie un couple de nombre entier correspondant à cette case dans la grille du takuzu. Par exemple, si le joueur joue 'A1', cette fonction doit renvoyer le couple $(0, 0)$. S'il joue 'C2', cette fonction doit renvoyer $(2, 1)$.
5. Écrire une fonction `joue_coup(g,l,c,v)` qui *joue* un coup dans la grille `g` à la ligne `l` colonne `c` avec la valeur `v`.
6. Écrire une fonction `grille_remplie(g)` qui prend en paramètre un grille `g` et qui renvoie un booléen indiquant si cette grille contient des cases qui sont encore vides ou non.
7. Fonctions pour le test de validité de la grille :
   - (Facultatif mais certainement utile) Écrire une fonction `rotation(g)` qui prend en paramètre une grille `g` et qui renvoie la transposée de `g` (les lignes de la grille de départ deviennent les colonnes et inversement en gardant l'ordre).
   - Écrire une fonction `verif_nb_0_nb_1(g)` qui prend en paramètre une grille `g` et qui renvoie un booléen indiquant si pour chaque ligne et chaque colonne de la grille `g` entrée en paramètre, le nombre de 0 et de 1 est cohérent (Règle 1).
   - Écrire une fonction `verif_000_111(g)` qui prend en paramètre une grille `g` et qui renvoie un booléen indiquant si pour chaque ligne et chaque colonne de la grille `g` entrée en paramètre, il n'y a jamais plus de deux 0 ou de deux 1 adjacents (Règle 2).
   - Écrire une fonction `verif_ligne_colonne(g)` qui prend en paramètre une grille `g` et qui renvoie un booléen indiquant si chaque ligne et chaque colonne de la grille g entrée en paramètre est unique (Règle 3).
8. Écrire une fonction `takuzu(nom_fic)` qui prend en paramètre un fichier `nom_fic` au format .txt et qui simule une partie de takuzu. Votre fonction devra notamment :
   - convertir le fichier `nom_fic` au format liste de listes (en utilisant la fonction `lecture(nom_fic)` ;
   - lancer le jeu et le dérouler jusqu'à la fin (grille remplie) ;
   - afficher un message indiquant si le joueur à gagner ou non ;
   - éventuellement, proposer de rejouer.

4 Bonus

Vous pourrez réaliser une interface graphique, permettant d'afficher le jeu avec `Pygame`.
Voici un exemple d'affichage graphique, mais vous êtes libre de faire autrement... Laissez place à votre imagination...



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

5 Évaluation

Pour ce projet, vous serez évalués selon les mêmes critères que précédemment :

- réalisation d'un programme fonctionnel ;
- respect des consignes ;
- bonne répartition des tâches dans le groupe ;
- présentation du code clair et soigné (avec noms et prénoms de tous les membres du groupe notamment)
- code documenté, fonctions avec docstrings. . . ;
- code commenté intelligemment(« par bloc »)
- optimisation du code ;
- un carnet de bord rempli à chaque séance (et à chaque fois que vous travaillez le projet), au format numérique, qui indique :
  - ce qui a été fait ;
  - vos choix (de modélisation notamment) ;
  - les difficultés/problèmes rencontrés ;
  - ce qui a été fait pour résoudre ces difficultés/problèmes ;
  - . . ....

Ce projet est à conserver . . .