

SAE 1.01

-

Implémentation d'un besoin client

Sommaire :

1- Introduction :	3
Contexte.....	3
Objectifs.....	3
Portée et Limites.....	3
2- Architecture du projet :	4
2.1- Organisation générale.....	4
3- Fonctionnalité implémentés :	5
3.1. Menu principal.....	5
3.2. Gestion des joueurs.....	5
3.3. Sauvegarde et chargement des scores.....	5
3.4. Les jeux.....	5
Jeu des Allumettes.....	5
Jeu de Devinette.....	6
Morpion.....	6
Puissance 4.....	6
3.5. Interaction avec l'utilisateur.....	6
4- Tests et Jeux d'Essais :	7
4.1 Test du Menu Principal.....	7
Test des choix valides.....	7
Test des choix invalides.....	9
4.2 Test de la Gestion des Joueurs.....	10
Essai Choisir joueurs :.....	10
Ajout de nouveaux joueurs.....	11
Suppression de joueurs :.....	12
4.3 Test de la Gestion Jouer :.....	14
Devinette :.....	14
Allumette :.....	19
Morpion :.....	21
Puissance 4 :.....	25
4.4 Test de la gestion de classement et de sauvegarde :.....	28
1. Affichage des Classements.....	28
2. Sauvegarde des Scores.....	28
3. Réinitialisation la sauvegarde :.....	28
4. Importation de Joueurs Fictifs.....	28
5- Problèmes rencontrés et solutions :	29
5.1 Outils utilisés régulièrement :.....	29
Couleurs.....	29
clear().....	29
attendre() et passer().....	29
5.2 Classement et affichage des joueurs.....	30
Utilisation des option de f strings.....	30
5.3 Sauvegarde :.....	30
6- Conclusion :	30

1- Introduction :

Contexte

Pour la S1.01, nous avons réalisé un projet en Python visant à concevoir un ensemble de minis-jeux. Ce projet permet de mettre en pratique nos connaissances en programmation.

Objectifs

Ce programme propose une expérience ludique pour deux joueurs. Il inclut :

1. Un **menu principal** permettant de naviguer entre les différentes sections.
2. Une **gestion des joueurs**, avec la possibilité de créer, choisir, ou supprimer des profils.
3. Quatre jeux :
 - Le jeu des **allumettes**.
 - Le jeu de **devinette**.
 - Le **morpion**.
 - Le **puissance 4**.
4. Un système de **sauvegarde et chargement** des scores pour permettre la continuité des parties.

Portée et Limites

Ce projet est conçu pour fonctionner exclusivement en mode console. Certaines fonctionnalités avancées, telles que l'interface graphique ou l'intégration en ligne, n'ont pas été implémentées dans le cadre des contraintes du projet

2- Architecture du projet :

2.1- Organisation générale

Le projet est organisé avec des fichiers et dossiers ayant des rôles spécifiques. Cette organisation facilite la lisibilité du code et sa gestion d'erreur.

- **main.py**
- **base/ :**
 - **joueurs.py** : Définit les classes Interface et Joueur ainsi que les fonctions pour manipuler les données des joueurs (ajout, suppression, recherche, etc.).
 - **outils.py**
 - **menuJoueurs.py**
 - **menuScoreSave.py**
 - **sauvegarde.py**
 - **__init__.py** : Grâce à ce fichier les imports sont plus simples à gérer car fait en sorte que ça parte toujours de la racine du projet.
- **jeux/ :**
 - **allumette.py**
 - **devinette.py**
 - **morpion.py**
 - **puissance4.py**
 - **menuJeux.py**
 - **__init__.py** : Grâce à ce fichier les imports sont plus simple à gérer car fait en sorte que ça parte toujours de la racine du projet.

3- Fonctionnalité implémentés :

3.1. Menu principal

Le **menu principal** permet d'accéder aux menus principaux du programme

- **Jouer** : Accède aux jeux disponibles.
- **Joueurs** : Gère la création, la suppression et la sélection des joueurs.
- **Classement, score et sauvegarde** : Affiche les classements des joueurs et permet de sauvegarder ou charger les scores.
- **Quitter**
-

3.2. Gestion des joueurs

Le menu **Gestion des joueurs** :

- **Sélection des joueurs actifs** : À chaque lancement du programme, l'utilisateur choisit deux joueurs actifs parmi ceux enregistrés.
- **Ajout de joueurs**
- **Suppression de joueurs**

3.3. Sauvegarde et chargement des scores

Le menu **Sauvegarde et chargement des scores** :

- **Classement des pseudos du meilleur au moins bon par jeu**
- **Sauvegarder**
- **Reset la sauvegarde**
- **Importer 10 joueurs numéroté de 1 à 10 avec des scores aléatoires de 0 à 100**

La sauvegarde est réalisée en utilisant le module **pickle**, qui permet de sauver les joueurs et leurs scores dans un fichier binaire.

3.4. Les jeux

Jeu des Allumettes

Le **jeu des allumettes** est un jeu à deux joueurs dans lequel il y a 20 allumettes au départ. À chaque tour, un joueur peut retirer 1, 2 ou 3 allumettes. Le joueur qui retire la dernière allumette perd la partie. Le gagnant remporte un point.

Jeu de Devinette

Le **jeu de devinette** est un jeu où un joueur choisit un nombre secret dans une plage donnée et modifiable, et l'autre joueur doit deviner ce nombre en un nombre limité d'essais. Si le joueur devine le bon chiffre, il gagne un point. Sinon, l'autre joueur gagne le point.

Morpion

Le **jeu du morpion** se joue sur une grille de 3x3. Les joueurs alternent pour placer leur symbole (X ou O) et cherchent à aligner trois symboles sur une ligne, une colonne ou une diagonale pour gagner. Si la grille est remplie sans qu'il y ait de gagnant, la partie est déclarée nulle.

Puissance 4

Le **jeu de puissance 4** se joue sur une grille de 6 lignes et 7 colonnes. Les joueurs placent leurs jetons dans la colonne de leur choix, et le premier joueur à aligner quatre jetons de sa couleur (horizontale, verticale ou diagonale) gagne la partie.

3.5. Interaction avec l'utilisateur

L'ensemble des jeux et fonctionnalités est accessible via des menus interactifs. Les utilisateurs interagissent principalement à travers des choix numériques (1, 2, 3, etc.) pour naviguer dans les menus et prendre des décisions pendant les jeux.

4- Tests et Jeux d'Essais :

4.1 Test du Menu Principal

Test des choix valides

1. Lancer le programme :

Le programme affiche le menu principal. Les options suivantes sont présentées :

1. Jouer
2. Joueurs
3. Classement, score et sauvegarde
4. Quitter

```
=====
                          Menu Principal
=====

Les joueurs actifs sont : - et -

-----

      1. Jouer
      2. Joueurs
      3. Classement, score et sauvegarde
      4. Quitter

=====
Veuillez entrer 1, 2, 3 ou 4 : |
```

2. Choix 1 : Jouer

L'utilisateur choisit l'option "1" pour commencer à jouer. Le programme redirige l'utilisateur vers le menu des jeux disponibles (Allumette, Devinette, Morpion, Puissance 4).

```
=====
                          Menu des Jeux
=====

Les joueurs actifs sont :

      Allumette   Devinette   Morpion   Puissance 4
toto :          0pts          0pts          0pts          0pts
titi :          0pts          0pts          0pts          0pts

-----

      1. Devinette
      2. Allumette
      3. Morpion
      4. Puissance 4
      5. Retour

=====
Veuillez entrer 1, 2, 3, 4 ou 5 : |
```

Si aucun joueurs n'est préalablement choisis, le programme nous demande d'en choisir :

```
=====
                          Menu des Jeux
=====

Veillez d'abord choisir les joueurs actif.
|
```

3. Choix 2 : Joueurs

L'utilisateur choisit l'option "2" pour gérer les joueurs. Le programme redirige l'utilisateur vers le menu de gestion des joueurs (ajout, suppression, choix des joueurs actifs).

```
=====
                          Menu des Joueurs
=====

1. Choisir les joueurs
2. Créer un joueur
3. Supprimer un joueur
4. Retour

=====
Veillez entrer 1, 2, 3 ou 4 : |
```

4. Choix 3 : Classement, score et sauvegarde

L'utilisateur choisit l'option "3" pour consulter les classements et sauvegarder/charger les scores. Le programme redirige l'utilisateur vers le menu de gestion des scores et sauvegardes.

```
=====
                          Menu des Scores et Sauvegardes
=====

1. Classement des pseudos du meilleur au moins bon par jeu
2. Sauvegarder
3. Reset la sauvegarde
4. Quitter

=====
Veillez entrer 1, 2, 3 ou 4 : |
```


5. Choix 4 : Quitter

L'utilisateur choisit l'option "4" pour quitter l'application. Le programme ferme l'application et sauvegarde les données

```
Sauvegarde de l'interface...  
bisous <3  
PS G:\Mon Drive\cours\S1.01> |
```

Test des choix invalides

Saisie d'une valeur invalide :

L'utilisateur entre un choix qui n'est pas valide, un nombre hors de la plage, comme 5 ou -1 (la rentrée d'un texte ou d'un nombre flottant ne sera pas traité ici).

```
=====
                          Menu Principal
=====

Les joueurs actifs sont : - et -

-----

      1. Jouer
      2. Joueurs
      3. Classement, score et sauvegarde
      4. Quitter

=====
Veuillez entrer 1, 2, 3 ou 4 : 5
Choix invalide.
|
```

Choix invalide est noté durant 1,2 seconde.

4.2 Test de la Gestion des Joueurs

Essai Choisir joueurs :

1. Première essai

L'utilisateur choisit l'option "1" pour entrer dans le menu de sélection de joueur. Le programme indique qu'aucun joueur n'est disponible pour l'instant.

```
=====
Choix des Joueurs Actifs
=====

Veillez créer au minimum 2 joueurs avant de pouvoir en choisir.
Appuyez sur entrée pour continuer...
```

En appuyant sur la touche entrée on revient en arrière.

2. Deuxième essai

On verra juste après comment créer des joueurs

```
=====
Choix des Joueurs Actifs
=====

  Nom du joueur | Allumette | Devinette | Morpion | Puissance 4
1-      Alice |      0pts |      0pts |      0pts |      0pts
2-      Bob  |      0pts |      0pts |      0pts |      0pts

-----

Veillez entrer l'indice du joueur 1 : 1

Joueur 1 trouvé

-----

Veillez entrer l'indice du joueur 2 : 2

Joueur 2 trouvé

=====
```

Après quelques secondes on revient au menu précédent. Il faut les redéfinir à chaque lancement du programme.

Ajout de nouveaux joueurs

1. Action :

L'utilisateur choisit l'option "2" pour entrer dans le menu de création de joueur. Nous ajoutons 3 joueurs, "Alice", "Bob" et "Tracey" puis nous essayons de rentrer un nom de plus de 15 caractères.

```
=====
                          Ajout de Joueurs
=====

Veuillez entrer le nom du joueur a créer (15 caractères max) : Alice

      Joueur ajouté.

Voulez vous ajouter un autre joueur ? (o/n) : o
-----

Veuillez entrer le nom du joueur a créer (15 caractères max) : Bob

      Joueur ajouté.

Voulez vous ajouter un autre joueur ? (o/n) : o
-----

Veuillez entrer le nom du joueur a créer (15 caractères max) : plus de 15 caracteres

      Nom trop long.

Veuillez entrer le nom du joueur a créer (15 caractères max) : Tracey

      Joueur ajouté.

Voulez vous ajouter un autre joueur ? (o/n) :
```

Nom trop long, affiché quelques secondes puis on repose la question.

2. Vérifications :

On le verra dans les menus suivants.

Suppression de joueurs :

1. Suppression

L'utilisateur choisit l'option "3" pour entrer dans le menu de suppression de joueurs et supprime Tracey.

```
=====
Suppression de Joueurs
=====
Nom du joueur | Allumette | Devinette | Morpion | Puissance 4
1- Alice | 0pts | 0pts | 0pts | 0pts
2- Bob | 0pts | 0pts | 0pts | 0pts
3- Tracey | 0pts | 0pts | 0pts | 0pts
-----
Veuillez entrer l'indice du joueur a supprimer : 3
      Êtes-vous sûr de vouloir supprimer ce joueur ? (o/n) : o
Joueur supprimé.
```

Le message s'affiche quelques secondes et on reviens au menu principal.

2. Choix invalides

L'utilisateur entre un choix qui n'est pas valide, un nombre hors de la plage, comme 3, 0 ou -1 (la rentrée d'un texte ou d'un nombre flottant ne sera pas traité ici).

```
=====
Suppression de Joueurs
=====
Nom du joueur | Allumette | Devinette | Morpion | Puissance 4
1- Alice | 0pts | 0pts | 0pts | 0pts
2- Bob | 0pts | 0pts | 0pts | 0pts
-----
Veuillez entrer l'indice du joueur a supprimer : 0
      Indice invalide, aucun joueur supprimé.
```

```
=====
Suppression de Joueurs
=====

  Nom du joueur | Allumette | Devinette | Morpion | Puissance 4
1-      Alice |      0pts |      0pts |      0pts |      0pts
2-       Bob |      0pts |      0pts |      0pts |      0pts

-----

Veuillez entrer l'indice du joueur a supprimer : 3

Indice invalide, aucun joueur supprimé.

=====
Suppression de Joueurs
=====

  Nom du joueur | Allumette | Devinette | Morpion | Puissance 4
1-      Alice |      0pts |      0pts |      0pts |      0pts
2-       Bob |      0pts |      0pts |      0pts |      0pts

-----

Veuillez entrer l'indice du joueur a supprimer : -1

Indice invalide, aucun joueur supprimé.
```

Chaque message est affiché plusieurs secondes puis on revient en arrière.

4.3 Test de la Gestion Jouer :

Devinette :

1. Début de partie.

L'utilisateur doit donner au programme le joueur qui commence.

```
=====
                        Jeu de Devinette
=====

Qui choisi le nombre a deviner ? (l'autre devras le trouver, donnez l'indice)
  1- Alice
  2- Bob
>>> |
```

2. Partie standard avec une réponse correcte

- Plage des nombres : 1 à 50.
- Nombre d'essais : 4.

Alice choisit le nombre secret : **25**.

```
=====
                        Jeu de Devinette
=====

      Alice, veuillez entrer un nombre entre 1 et 50 compris : 25

-----

Vous avez choisi le nombre 25 (Message affiché 5s).

=====
```

Bob propose successivement :

- **10** → Indice : "Le nombre est plus grand."
- **30** → Indice : "Le nombre est plus petit."
- **25** → Réponse correcte.
- Bob gagne.

```
=====
                          Jeu de Devinette
=====

      Bob, c'est à vous de jouer !
      Vous avez 4 essais pour trouver le nombre compris entre 1 et 50 compris.

-----

      Veuillez entrer un nombre : 10

Le nombre est plus grand que 10.
Il vous reste 3 essais.

-----

      Veuillez entrer un nombre : 30

Le nombre est plus petit que 30.
Il vous reste 2 essais.

-----

      Veuillez entrer un nombre : 25
ah ?!
Il vous reste 1 essais.

-----

Bravo Bob ! Vous avez trouvé le nombre 25 en 3 essais.

=====
Appuyez sur entrée pour continuer...|
```

3. Partie perdue sans trouver le nombre

- Plage des nombres : 1 à 50.
- Nombre d'essais : 4.

Joueur 1 choisit le nombre secret : **15**.

```
=====
                          Jeu de Devinette
=====

      Alice, veuillez entrer un nombre entre 1 et 50 compris : 15

-----

Vous avez choisi le nombre 15 (Message affiché 5s).

=====
```

Joueur 2 propose successivement :

- **5** → Indice : "Le nombre est plus grand."
- **18** → Indice : "Le nombre est plus petit."
- **12** → Indice : "Le nombre est plus grand."
- **14** → Indice : "Le nombre est plus grand."
- Après 4 essais incorrects, le jeu déclare Joueur 2 perdant et attribue 1 point au joueur 1. Le jeu révèle le nombre secret (**15**).

```
=====
                          Jeu de Devinette
=====

      Bob, c'est à vous de jouer !
      Vous avez 4 essais pour trouver le nombre compris entre 1 et 50 compris.

-----

      Veuillez entrer un nombre : 5

      Le nombre est plus grand que 5.
      Il vous reste 3 essais.

-----

      Veuillez entrer un nombre : 18

      Le nombre est plus petit que 18.
      Il vous reste 2 essais.

-----

      Veuillez entrer un nombre : 12

      Le nombre est plus grand que 12.
      Il vous reste 1 essais.

-----

      Veuillez entrer un nombre : 14

      Le nombre est plus grand que 14.
      Il vous reste 0 essais.

-----

      Dommage Bob ! le nombre était 15.

=====
      Appuyez sur entrée pour continuer...|
```


4. Modification des paramètres du jeu

L'utilisateur augmente le nombre d'essais à 10.

```
=====
Options du jeu de Devinette
=====

Le nombre de tentatives actuel est de 4.
Veuillez entrer le nombre de tentative maximum : 10

Le nombre de tentatives a été changé à 10.
=====
Appuyez sur entrée pour continuer...|
```

L'utilisateur modifie la plage des nombres pour qu'elle soit de 1 à 100.

```
=====
Options du jeu de Devinette
=====

Le maximum de la plage actuel est de 50.
Veuillez entrer le maximum de la plage a deviner : 100

Le maximum de la plage a été changé à 100.
=====
Appuyez sur entrée pour continuer...|
```

5. Entrées utilisateur invalides

Joueur 1 entre un nombre hors de la plage autorisée (par exemple : **120** alors que la plage est de 1 à 100)

```
=====
Jeu de Devinette
=====

Nombre invalide.

Alice, veuillez entrer un nombre entre 1 et 100 compris : |
```

Joueur 2 entre une valeur invalide comme un nombre hors de la plage (-5).

```
=====
Jeu de Devinette
=====

Bob, c'est à vous de jouer !
Vous avez 10 essais pour trouver le nombre compris entre 1 et 100 compris.

-----

Veuillez entrer un nombre : -5
Nombre invalide.
Il vous reste 9 essais.

-----
```

6. Victoire dès le premier essai

- Plage des nombres : 1 à 100.
- Nombre d'essais : 10.

Joueur 1 choisit le nombre secret : 7.

Joueur 2 propose immédiatement : 7.

```
=====
Jeu de Devinette
=====

Bob, c'est à vous de jouer !
Vous avez 10 essais pour trouver le nombre compris entre 1 et 100 compris.

-----

Veuillez entrer un nombre : 7
ah ?!
Il vous reste 9 essais.

-----

Bravo Bob ! Vous avez trouvé le nombre 7 en 1 essais.

=====
Appuyez sur entrée pour continuer...|
```

Allumette :

1. Début de Partie

```
=====
                          Jeu de l'allumette
=====

  Alice commence la partie !
=====
```

2. Partie standard avec une victoire

1. Alice retire **3 allumettes** → 17 restantes.
2. Bob retire **2 allumettes** → 15 restantes.
3. Alice retire **1 allumette** → 14 restantes.
4. Bob retire **3 allumettes** → 11 restantes.
5. Alice retire **2 allumettes** → 9 restantes.
6. Bob retire **3 allumettes** → 6 restantes.
7. Alice retire **3 allumettes** → 3 restantes.
8. Bob retire **2 allumettes** → 1 restante.
9. Alice retire la dernière allumette

```
=====
                          Jeu de l'allumette
=====

  Il reste 1 allumettes.
  Alice, vous devez retirer la dernière allumette.
  Alice retire 1 allumettes.

=====

  Alice vous avez perdu !
  Bob a gagné la partie !

=====
Appuyez sur entrée pour continuer...
```

3. Partie avec choix invalides

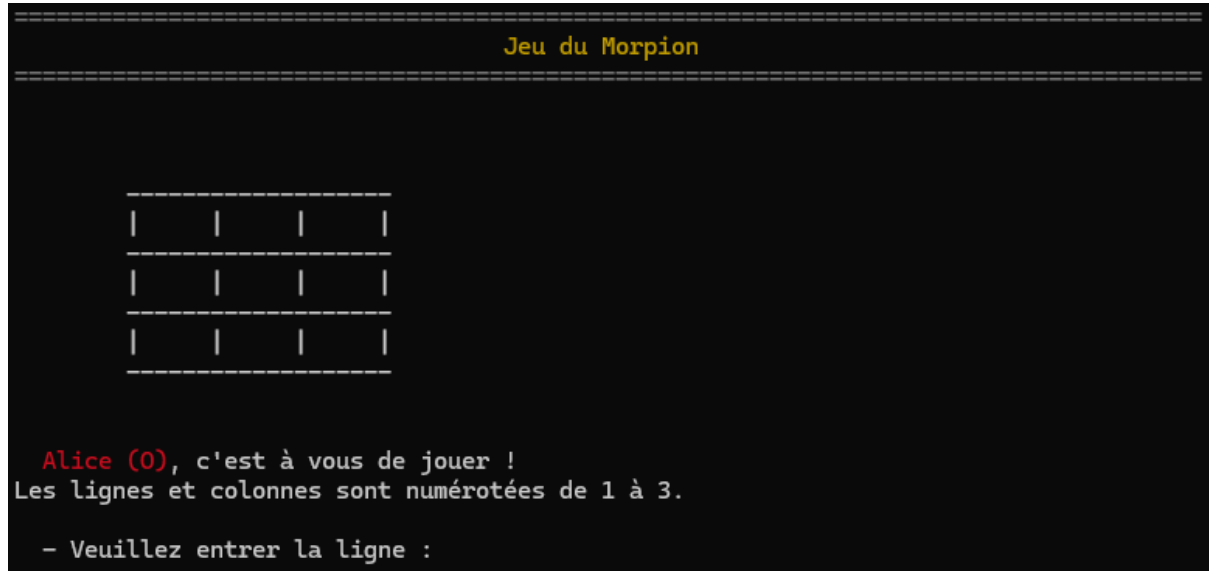
- Alice tente de retirer **4, -1 ou 0 allumettes** (choix invalide)

```
Il reste 20 allumettes.  
Bob, combien d'allumettes voulez-vous retirer ? (1, 2 ou 3) : 4  
Choix invalide.  
Bob, combien d'allumettes voulez-vous retirer ? (1, 2 ou 3) : -1  
Choix invalide.  
Bob, combien d'allumettes voulez-vous retirer ? (1, 2 ou 3) : 0  
Choix invalide.  
Bob, combien d'allumettes voulez-vous retirer ? (1, 2 ou 3) :
```

Morpion :

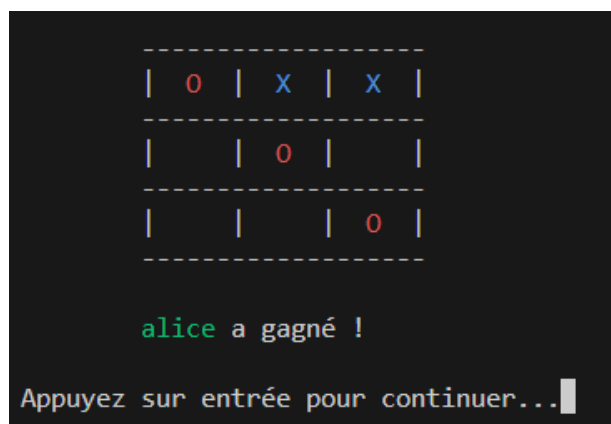
1. Début de Partie

La grille initiale est vide :



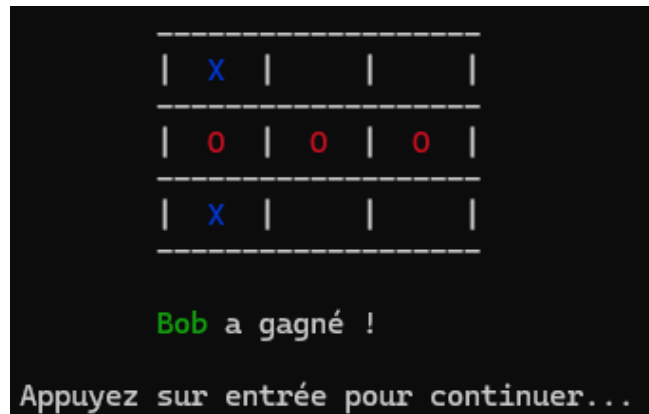
2. Partie standard avec victoire

- **Diagonale :**
 Alice place **O** (ligne 1, colonne 1).
 Bob place **X** (ligne 1, colonne 2).
 Alice place **O** (ligne 2, colonne 2).
 Bob place **X** (ligne 1, colonne 3).
 Alice place **O** (ligne 3, colonne 3),
 Ce qui aligne trois **X** en diagonale.



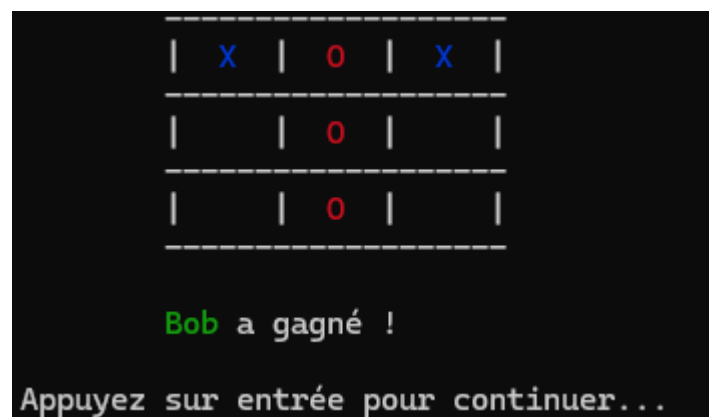
- **Horizontale :**

Bob place **O** (ligne 2, colonne 1).
 Alice place **X** (ligne 1, colonne 1).
 Bob place **O** (ligne 2, colonne 2).
 Alice place **X** (ligne 3, colonne 1).
 Bob place **O** (ligne 2, colonne 3),
 Ce qui aligne trois **O** horizontalement.



- **Verticale :**

Bob place **O** (ligne 1, colonne 2).
 Alice place **X** (ligne 1, colonne 1).
 Bob place **O** (ligne 2, colonne 2).
 Alice place **X** (ligne 1, colonne 3).
 Bob place **O** (ligne 3, colonne 2),
 Ce qui aligne trois **O** dans la colonne centrale.



3. Partie nulle

Alice place **O** (ligne 1, colonne 1).

Bob place **X** (ligne 2, colonne 2).

Alice place **O** (ligne 1, colonne 2).

Bob place **X** (ligne 1, colonne 3).

Alice place **O** (ligne 2, colonne 3).

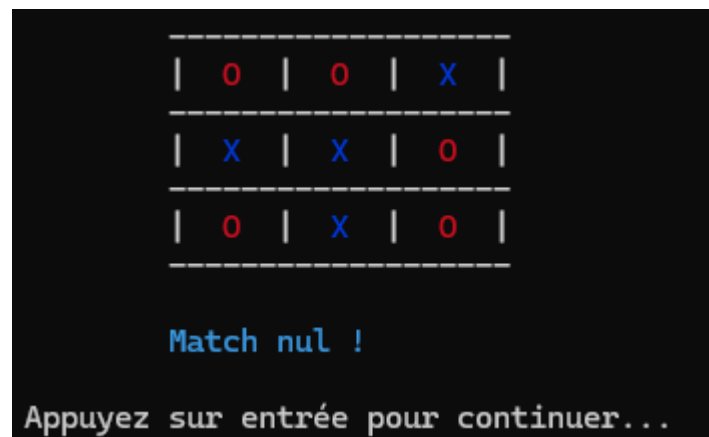
Bob place **X** (ligne 3, colonne 2).

Alice place **O** (ligne 3, colonne 1).

Bob place **X** (ligne 2, colonne 1).

Alice place **O** (ligne 3, colonne 3).

Ceci créer un match nul.



4. Partie avec choix invalide

Si **Alice** met -1, 0 ou 4 :

```

-----
|   |   |   |
-----
|   |   |   |
-----
|   |   |   |
-----

Alice (0), c'est à vous de jouer !
Les lignes et colonnes sont numérotées de 1 à 3.

- Veuillez entrer la ligne : -1

    Ligne invalide.

- Veuillez entrer la ligne : 0

    Ligne invalide.

- Veuillez entrer la ligne : 4

    Ligne invalide.

- Veuillez entrer la ligne :

```

5. Case déjà occupé

```

-----
| 0 | X | 0 |
-----
|   | X |   |
-----
| X | 0 | 0 |
-----

Alice (X), c'est à vous de jouer !
Les lignes et colonnes sont numérotées de 1 à 3.

- Veuillez entrer la ligne : 3
- Veuillez entrer la colonne : 3

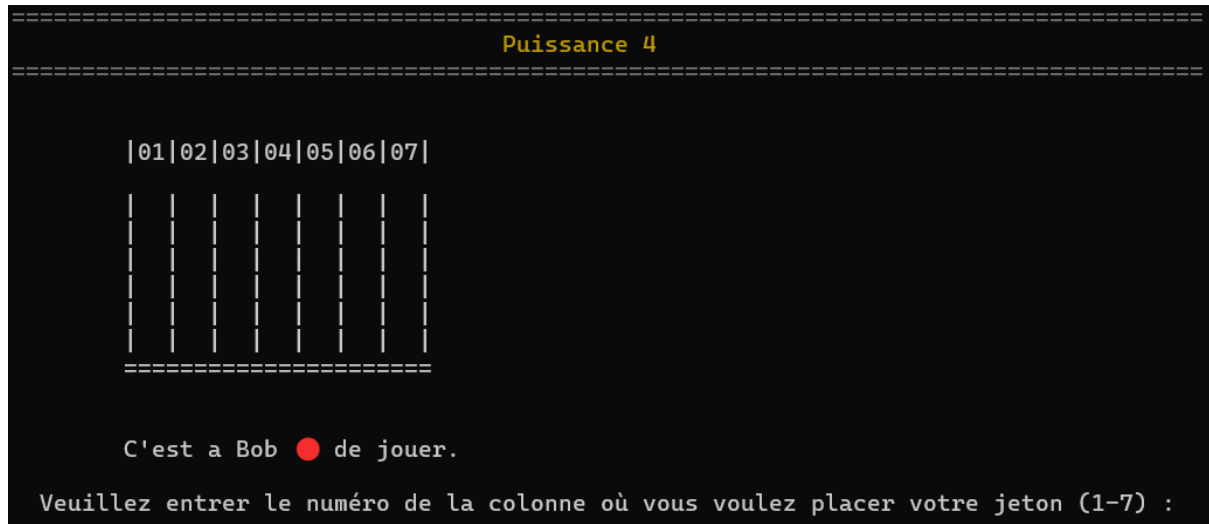
    La case est déjà occupée.

```


Puissance 4 :

1. Début de Partie

La grille initiale est vide :

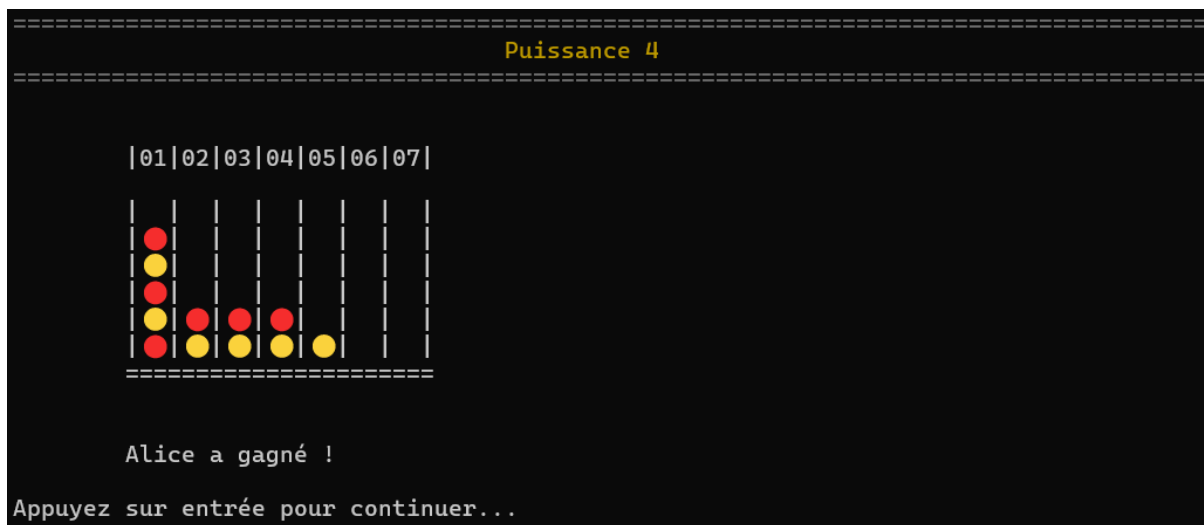


2. L'ajout de jetons :

L'ajout des jeton, sont bien alignés sur leur ligne :



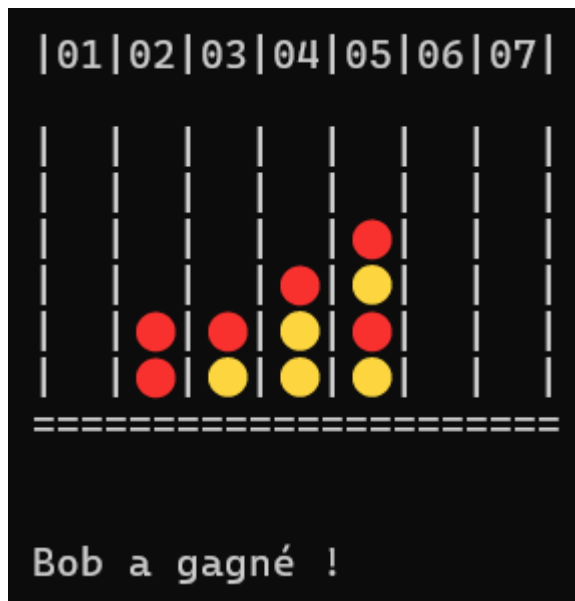
3. La détection de victoire horizontale :



4. La détection de victoire verticale :



5. La détection de victoire diagonale (montante et descendante) :



7. La détection de match nul :



8. Des entrées invalides :

Veillez entrer le numéro de la colonne où vous voulez placer votre jeton (1-7) : -1

Colonne invalide.

Veillez entrer le numéro de la colonne où vous voulez placer votre jeton (1-7) : 0

Colonne invalide.

Veillez entrer le numéro de la colonne où vous voulez placer votre jeton (1-7) : 12

Colonne invalide.

4.4 Test de la gestion de classement et de sauvegarde :

1. Affichage des Classements

Classement des pseudos du meilleur au moins bon par jeu			
Devinette	Allumettes	Morpion	Puissance 4
Joueur 8 : 85	Joueur 3 : 97	Joueur 7 : 99	Joueur 4 : 86
Joueur 2 : 83	Joueur 4 : 91	Joueur 3 : 96	Joueur 5 : 76
Joueur 3 : 76	Joueur 2 : 90	Joueur 8 : 90	Joueur 8 : 63
Joueur 9 : 61	Joueur 8 : 61	Joueur 10 : 71	Joueur 7 : 56
Joueur 4 : 58	Joueur 9 : 59	Joueur 6 : 60	Joueur 9 : 53
Joueur 5 : 45	Joueur 6 : 34	Joueur 2 : 44	Joueur 3 : 35
Joueur 6 : 45	Joueur 5 : 30	Joueur 4 : 29	Joueur 2 : 14
Joueur 10 : 36	Joueur 7 : 30	Joueur 9 : 5	Joueur 6 : 12
Joueur 7 : 13	Joueur 10 : 3	Joueur 5 : 2	Joueur 1 : 0
Joueur 1 : 0	Joueur 1 : 0	Joueur 1 : 0	Joueur 10 : 0

Appuyez sur entrée pour continuer...|

2. Sauvegarde des Scores

Lorsque l'on choisit Sauvegarder, tous les joueurs et leurs scores sont importés dans interface.dat ce qui permet de les récupérer au redémarrage

3. Réinitialisation la sauvegarde :

Supprime tous les joueurs.

Menu des Scores et Sauvegardes	
1. Classement pseudos du meilleur au moins bon par jeu 2. Sauvegarder 3. Reset la sauvegarde 4. Importer 10 Joueurs avec scores aléatoires de 0 à 100 5. Retour	
Veuillez entrer 1, 2, 3 ou 4 : 3 ATTENTION Voulez-vous vraiment supprimer la sauvegarde ? (o/n) : o Sauvegarde supprimée.	

4. Importation de Joueurs Fictifs

Lorsque l'on exécute, un message nous en informe. Cette option a été utilisée au 4.4.1.

5- Problèmes rencontrés et solutions :

5.1 Outils utilisés régulièrement :

Couleurs

Extrait du module outils.py:

```
jaune = "\033[33m"  
gris = "\033[90m"  
magenta = "\033[35m"  
bleu = "\033[34m"  
vert = "\033[32m"  
rouge = "\033[31m"  
blanc = "\033[37m"  
cyan = "\033[36m"  
reset = "\033[0m"
```

Pour avoir des couleurs dans le texte, nous avons juste à dans un f string mettre la couleur au début de la zone que l'on veut colorer et un reset à la fin.

clear()

Extrait du module outils.py:

```
def clear() -> None:  
    #windows  
    if name == 'nt':  
        _ = system('cls')  
    #linux  
    else:  
        _ = system('clear')  
    print()
```

Sert à nettoyer la console.

attendre() et passer()

Servent à afficher des messages attendre() fait une pause de 1,2 secondes et passer() affiche un message demandant si d'appuyer sur Entrée pour continuer.

5.2 Classement et affichage des joueurs

Utilisation des option de f strings

Pour un bel affichage des scores

```
# Affichage des scores
for i in range(interface.longueur):
    print(f"{interface.joueurs[liste_indices_devinette[i]].nom:>15s} : {interface.joueurs[liste_indices_devinette[i]].score_devinette:<8d} "
          f"| {interface.joueurs[liste_indices_allumette[i]].nom:>15s} : {interface.joueurs[liste_indices_allumette[i]].score_allumette:<8d} "
          f"| {interface.joueurs[liste_indices_morpion[i]].nom:>15s} : {interface.joueurs[liste_indices_morpion[i]].score_morpion:<8d} "
          f"| {interface.joueurs[liste_indices_puissance4[i]].nom:>15s} : {interface.joueurs[liste_indices_puissance4[i]].score_puissance4:<8d}*")
```

Ajout de `>15s` et `<8d` ce qui permet d'aligner tous les joueurs.

5.3 Sauvegarde :

Pour une raison logique la sauvegarde devait se faire à chaque fois que l'on ferme le programme mais au final pour des raison pratiques le jeu se sauvegarde a chaque fois qu'on revient sur le menu Joueur, Jeux ou Principal.

6- Conclusion :

Nous avons pu appliquer nos connaissances en Python en faisant le programme de mini-jeu. En travaillant sur ce projet, nous avons amélioré nos compétences telles que la gestion de fichiers et la manipulation de données.

Les tests de jeux ont montré que toutes les fonctionnalités ont été mises en œuvre correctement.

Le fait que les modules soit correctement organisés nous a aidés avec l'entretien et la lecture.. Pour l'avenir, nous pourrions ajouter quelques fonctionnalités, telles que des machines contre qui jouer 😊

En somme, cette SAE nous a permis non seulement d'approfondir nos compétences techniques, mais aussi de développer une méthodologie de travail collaborative et structurée, essentielle dans tout projet informatique.