

## I - Description :

Notre programme est divisé en 3 fichiers en suivant le modèle MVC (Model-View-Controller)

- 1) `controller.py` :  
Cette partie du programme sert d'exécuteur et ne comporte que les lancements d'affichage comme par exemple la fonction "`charger_menu()`".
- 2) `ihm.py` :  
C'est la partie la plus fournie. Elle intègre quasiment tout le fonctionnement du projet. Ihm contient non seulement la fenêtre du plateau mais aussi celles du menu, des paramètres, des crédits ainsi que l'écran de fin.
- 3) `datas.py` :  
C'est la partie où l'on pose les bases. On initialise la plus grande partie des variables qui seront utilisées tout au long de notre partie. La fonction va "`verif_alentours()`" vérifier si une pièce peut être posée ou non à l'endroit où le joueur clique.

## II - Particularités :

- 1) Graphique :  
Le graphique est fait avec les bibliothèques *Tkinter*, *CustomTkinter* et *Tkextrafont*.  
  
Toutes les images se trouvent dans le dossier : `"/images/"`
- 2) Boutons :  
Les boutons sont faits avec la bibliothèque *Tkinter* et servent essentiellement à naviguer de page en page.
- 3) Police d'écriture :  
La police d'écriture utilisée est celle du jeu vidéo *Omori*. Dans le cas où elle ne fonctionne pas, c'est la police d'écriture Arial pour les labels et Comic Sans MS pour les boutons qui prennent le relais.  
  
La police d'écriture se trouve dans le dossier : `"/fonts/"`
- 4) Sons :  
Les sons sont pris en charge avec la bibliothèque Pygame. Trois sons sont présents dans le projet, la musique de démarrage lorsqu'on démarre le programme, un son de placement de pièce ainsi qu'un son de redémarrage quand on redémarre la partie.  
  
Tous les sons se trouvent dans le dossier : `"/sons/"`

### III - Stratégie de la machine :

Le robot utilise les fonctions de vérifications tel que "verif\_pieces\_possibilites()" pour récupérer une liste de possibilités de placement de pièces. Ensuite, il va tirer aléatoirement un coup dans cette liste et va jouer de la même manière que jouerai le joueur grâce à sa fonction "bot\_coup()" qui est pratiquement identique aux fonctions "clic()" et "deposer()" combinées.

### IV - Version 2 :

Pour habiller notre projet nous avons ajouté d'autres fenêtres supplémentaires.

- 1) Menu :  
Avec un logo, un titre, un son de démarrage et des boutons.
- 2) Sélection de nombre d'ordinateur / joueur :  
Pour choisir le nombre d'ordinateurs et de joueurs que l'on veut mettre dans la partie grâce à deux sliders.
- 3) Paramètres :  
Pour changer le volume des sons.
- 4) Crédits :  
Pour créditer.
- 5) Écran de fin :  
Pour afficher les scores et le gagnant.

### V - Manuel d'utilisation :

- 1) Les bibliothèques :  
Exécuter le programme *install\_bibli.bat* pour télécharger toutes les bibliothèques utilisées dans le projet automatiquement.

Les différentes bibliothèques utilisées :

- tkinter (pip install tk)
- customtkinter (pip install customtkinter)
- tkextrafont (pip install tkextrafont)
- pygame (pip install pygame)
- pillow (pip install pillow)
- time
- random
- math

- 2) Le projet :  
Ouvrir l'intégralité du dossier puis exécuter le programme : *controller.py*  
Pour plus d'informations lire le fichier : *readme.md*