

Un morpion pour sauver le monde

v 1.0



Introduction	3
Le saviez-vous ?	3
Consignes	4
Installation	5
Sous Windows	5
Sous Mac	5
Sous Linux	5
Exemples :	5
1: La logique avant tout	6
Une vue d'ensemble	6
Les jeux sont terminés	7
2: Fiat Lux ! (WIP)	8
Une fenetre	8
Des images	9
De l'interaction	9
3: Conclusion	11

Introduction

Pygame est une Bibliothèque libre multiplateforme qui a pour but de faciliter le développement de Jeux vidéos, infographie, gestion audio, avec le langage de programmation [Python](#). Nous allons l'utiliser Aujourd'hui pour Sauver le monde d'un hiver nucléaire !



Pygame a commencé son développement en 2000, de nombreux logiciels amateurs l'utilisent encore aujourd'hui. La librairie est un des piliers des jeux homebrew // indépendants open source !

[D'ailleurs voici une liste de jeux développés avec Pygames !](#)

Le saviez-vous ?

Guido van Rossum, le créateur de Python, a reçu de la communauté le titre de Dictateur bienveillant à vie (Benevolent Dictator For Life, BDFL) et a officiellement pris ses "vacances permanentes" le 12 juillet 2018, quittant le groupe de développement de Python.

Consignes

- En cas de question, pensez à demander de l'aide à votre voisin de droite. Puis de gauche. Ou inversement. Puis demandez enfin à un Cobra si vous êtes toujours bloqué(e).
- Si l'installation d'un logiciel se déroule mal, recommencez depuis le début.
- Pensez à faire valider votre projet par un Cobra lorsque vous aurez terminé.

Installation

Sous Windows

Rendez-vous sur le site de Python. En vous rendant à l'adresse python.org/downloads vous allez pouvoir télécharger votre version de Python.

Choisissez la version à installer. Nous utiliserons la version 3.x.x
Exécutez l'installateur une fois le téléchargement terminé.

Cochez la case Add Python 3.5 to PATH (Ajouter Python 3.5 au PATH). Ainsi, vous pourrez exécuter Python directement depuis l'invite de commandes.

Cliquez sur *Installer maintenant*. Python s'installe alors avec tous ses paramètres par défaut. Pour voir si Python a été correctement installé ouvrez l'interpréteur des commandes puis tapez « python » : si votre interpréteur s'ouvre tout est ok.

Vous devez maintenant installer pygame. Pour ce faire, ouvrez un invité de commande et tapez

pip install pygame

Sous Mac

Rendez-vous sur le site de Python. python.org/downloads et sélectionnez python 3.X.X

Double Cliquez sur l'archive PKG pour pouvoir installer Python.

vous pouvez ensuite lancer python en tapant python dans le terminal

Vous devez maintenant installer pygame. Pour ce faire, ouvrez un terminal et tapez

sudo pip3 install pygame

Sous Linux

Pour linux, vous pouvez utiliser votre gestionnaire de paquet et utiliser la

dernière version.

Exemples :

```
sudo yum install python  
sudo pacman -s python  
sudo apt-get install python
```

Vous devez maintenant installer pygame. Pour ce faire, ouvrez un terminal et tapez

```
sudo pip3 install pygame
```

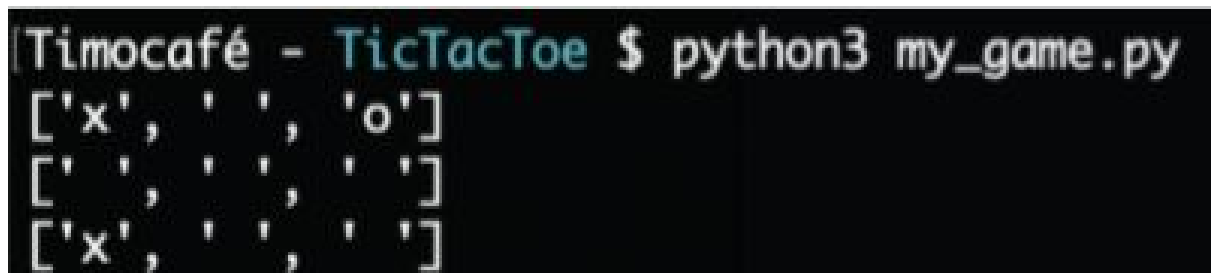
1: La logique avant tout

Une vue d'ensemble

Au début de la décennie, en 1980, le monde vit avec la peur constante d'un conflit nucléaire. Pour être sûr d'être prêt à riposter, une puissance occidentale a développé un ordinateur capable de simuler le conflit en continu; en cas d'attaque, il serait alors capable de riposter de la meilleure façon. Le problème, c'est que cet ordinateur est devenu aussi paranoïaque qu'un être humain, et a donc décidé de déclencher sans provocation un conflit nucléaire. Il a toutefois donné une condition pour qu'il arrête la procédure avant qu'il ne soit trop tard: il faut lui démontrer que les êtres humains sont plus qu'une créature qui agit selon son instinct. Pour ce faire, il vous défie au morpion.

Votre mission, si vous décidez de l'accepter, est de **créer un jeu du morpion en Python**.

La première étape est celle d'expérimentation, il faut créer et afficher un tableau 3x3 sur un terminal. Vous pouvez tester que le tableau fonctionne correctement en ajoutant des croix ou des cercles dans des cases. A la fin, vous devriez avoir quelque chose dans ce style:



```
Timocafé - TicTacToe $ python3 my_game.py
[['x', ' ', 'o'],
 [' ', ' ', ' '],
 ['x', ' ', ' ']]
```

L'affichage sur terminal du morpion

Les jeux sont lancées

Vous avez réussi à afficher le tableau de jeu, mais pour le moment il n'y a pas de jeu à proprement parler. Votre prochain objectif est d'implémenter une boucle de jeu, c'est-à-dire une liste d'instructions qui vont se répéter jusqu'à ce que le jeu se termine.

A ce stade, la boucle de jeu devra contenir :

- tenir compte du tour ; Joueur 1 ou 2
- demander où le joueur souhaite placer une croix ou un cercle (donc récupérer de l'**input** utilisateur)
- placer une croix ou un cercle selon le joueur
- Ecrire sur terminal le tableau à la fin de chaque tour

```
x?1
y?1
[' ', ' ', ' ', ' ']
[' ', 'x', ' ', ' ']
[' ', ' ', ' ', ' ']
x?0
y?0
['o', ' ', ' ', ' ']
[' ', 'x', ' ', ' ']
[' ', ' ', ' ', ' ']
x?
```


Les jeux sont terminés

Votre jeux est presque complet, il doit seulement pouvoir se terminer. Vous pouvez regarder [ici](#) pour vous rafraîchir la mémoire sur les règles du jeu.

En somme, vous devez être capable de:

- détecter la fin du jeu et déclarer un vainqueur
- éviter que l'on puisse sélectionner une case déjà prise



Congratulations, vous avez atteint le niveau 1 !

Vous avez prouvé à l'ordinateur que vous êtes capable de raisonnement logique

2: Une bonne impression

Une fenetre

Vous avez réussi à impressionner l'ordinateur. Il a accepté de mettre en pause son attaque, mais vous donne un deuxième défis pour lui démontrer que vous n'avez plus besoin de lui, que les humains sont capables de penser avant d'agir.

Vous devez, en reprenant le morpion sur terminal, le transformer en un vrai jeux, avec des graphismes grâce à [Pygame](#).

Pour ce faire, la première chose à faire est d'ouvrir une fenêtre qui devra:

- avoir une taille de 600 par 700
- un titre de "Morpion"

Elle devra aussi répondre à certains événements utilisateur, comme le fait de fermer la fenêtre quand l'utilisateur clique sur la croix de la fenêtre.

Des images

Vous avez affiché une fenêtre, très bien ! Mais l'ordinateur n'est pas tellement impressionné. En effet, une fenêtre vide est beaucoup moins utile que ce que vous lui aviez présenté auparavant. Il est donc venu le moment d'afficher des images dans votre nouvelle fenêtre. Il vous faut donc :

- charger l'image d'un rond
- charger l'image d'une croix
- charger l'image d'un fond

Une fois les images chargées, vous devez ensuite les afficher ([blit](#)) sur l'écran.



Congratulations, vous avez atteint le niveau 2 !

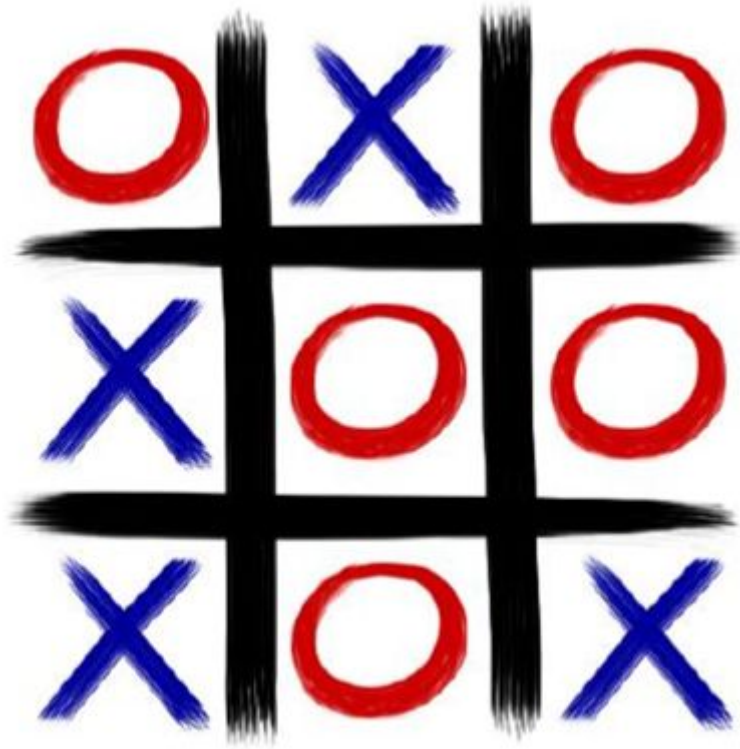
Vous avez prouvé à l'ordinateur que vous avez un peu de goût esthétique

De l'interaction

L'ordinateur est assez content ; vous affichez des images. Cependant, le plus dur reste à venir, car afficher des images c'est bien, mais quand l'utilisateur peut se servir de l'application, c'est mieux. Il vous faut maintenant pouvoir jouer à cette version du morpion. Pour ce faire, vous devez:

- récupérer la position de la souris lors d'un click
- vérifier si la position de la souris est dans une case du morpion
- selon le tour du joueur, afficher une croix ou un rond sur la case

Le résultat pourrait être quelque chose dans le style:



Un jeu de morpion fini



Congratulations, vous avez atteint le niveau 3 !

Vous avez prouvé à l'ordinateur que vous avez ce qu'il faut pour discuter entre vous sans avoir besoin de se menacer à coup de bombes.

3: Conclusion

Bravo, vous avez sauvé le monde de l'hiver nucléaire ! Mais la menace est toujours là : vous n'avez pas réussi à éteindre l'ordinateur qui vous menaçait. Pour ce faire, vous pouvez essayer de le faire effectivement jouer au morpion qui vous avez fini de programmer.

Vous pouvez donc :

- ajouter un algorithme qui a pour but de gagner la partie
- ajouter un menu pour sélectionner le mode de jeu : joueur contre joueur ou intelligence artificielle contre joueur
- ajouter un mode intelligence artificielle contre intelligence artificielle