

PyPong

La méthode Epitech : Learn different

- Vous allez acquérir de nouvelles compétences en pratiquant et en vous amusant.
- Votre premier réflexe lorsque vous vous posez une question doit être de demander de l'aide à vos voisins. Si vous ne trouvez pas la solution demandez de l'aide à un accompagnateur.
- Pour avancer sereinement dans la réalisation de votre projet, pensez à faire valider chaque partie au fur et à mesure par un animateur Epitech qui pourra vous donner des conseils
- Nous vous encourageons à utiliser les resources disponibles sur internet pour trouver des réponses à vos questions et pour vous renseigner.
- Vous pouvez ajouter des fonctionnalités de votre choix et personnaliser votre projet : c'est ce que nous appelons un bonus à Epitech et c'est possible pour tous les projets
    - Les IA génératives peuvent sûrement faire cet atelier, mais si vous les utilisez vous n'apprendrez rien.

{EPITECH}
LEARN DIFFERENT*

Pré-requis

TIC-80 est une fantasy retro console open source conçu pour créer, jouer et partager de petits jeux. Prenez quelques instants pour la découvrir en testant un jeu : https://tic80.com/play

Tous les outils sont intégrés pour vous permettre de créer facilement un jeu : éditeurs de code, de sprites, de cartes, de sons, ainsi qu'un terminal.

Pour réaliser cet atelier, vous pouvez utiliser directement TIC-80 dans votre navigateur : https://tic80.com/create.

Initialiser TIC-80 pour pouvoir utiliser le langage Python

il vous suffit de taper la commande suivante

```
new python
```

Reset “hello world”

Le plus simple pour bien comprendre le fonctionnement de TIC-80 consiste de partir d’un environnement vierge, nous allons donc supprimer tous les éléments de la démo.

- Utilisez la touche `ESC` de votre clavier pour vous rendre dans l’éditeur de code
- Vous pouvez sélectionner tout texte avec `CTRL` + `A` et ensuite supprimer le code existant

Créer votre mini jeu

Vous êtes prêt à créer votre premier mini jeu

Affichage du pad et de l’écran de jeu

Rendez vous dans l’éditeur, normalement vous y êtes déjà, et écrivez le code suivant :

```
# script:  python

def TIC():
 cls()
 rect(0,0,120,120,10)
 rect(45, 110, 30, 3, 12)
```

Vous pouvez maintenant retourner sur le terminal avec la touche `ESC` de votre clavier et taper la commande `run` pour lancer votre jeu

Quelques explications

Chaque environnement et chaque langage de programmation possède ses spécificités. Pour cet atelier, nous utilisons le langage Python et l'environnement TIC-80 :

- Il faut expliquer à TIC-80 que le code sera écrit en langage Python en écrivant à la première ligne de l'éditeur `# script:  python`
- La partie principale du programme se déclare de la façon suivante `def TIC():`
- Le code en dessous de la partie principale du programme doit respecter une indentation propre au python : il faut un espace au début de chaque ligne comme dans l'exemple
- Nous pouvons utiliser des instructions prédéfinies comme `cls()` pour effacer l'écran (CLear Screen) et `rect()` pour dessiner des rectangles.

Faire bouger le pad

Pour arriver à faire bouger le pad avec le clavier, nous devons :

- Déclarez une variable `padx` qui va nous permettre modifier la valeur de la position de notre pad sur l'axe des abscisses
- Utilisez une condition `if` pour modifier la valeur de `padx` lorsque l'on appuie sur la touche `←` du clavier

Retournez dans l'éditeur pour modifier le code

```
# script:  python
padx=45
padw=30
padh=3

def TIC():
 global padx

 if btn(2):
    padx = padx - 2
 cls()
```

```
rect(0,0,120,120,10)
rect(padx, 110, padw, padh, 12)
```

Vous pouvez maintenant retourner sur le terminal avec la touche `ESC` de votre clavier et taper la commande `run` pour lancer votre jeu

Exercice 1 : Faites bouger le pad dans l'autre direction

En vous inspirant du code précédent, vous devez faire en sorte que le pad puisse bouger à droite comme à gauche

▼ Indice

La fonction `btn` prend en paramètre un nombre (3 : flèche droite du clavier). Une page dans l'aide contient le tableau de correspondance avec les touches du clavier : https://github.com/nesbox/TIC-80/wiki/key-map

▼ Solution

```
# script:  python
padx=45
padw=30
padh=3

def TIC():
 global padx

 if btn(2):
    padx = padx - 2
 if btn(3):
    padx = padx + 2
 cls()
 rect(0,0,120,120,10)
 rect(padx, 110, padw, padh, 12)
```

Exercice 2 : Limitez les mouvement du pad

Vous voulez maintenant améliorer les mouvements du pad en faisant en sorte qu'il reste dans le carré du jeu

▼ Indice

Ajouter une condition supplémentaire dans votre `if` avec l'opérateur logique `and`

Créer la balle rebondissante

Exercice 1 : Dessinez la balle

Vous allez utiliser la fonction `circ` pour créer la balle au centre de l'écran, pensez à utiliser des variables `ballx` et `bally` pour pouvoir déplacer votre balle dans l'écran de jeu

▼ Indice

vous pouvez aller consulter l'aide de TIC80 https://tic80.com/learn et retrouver tous les paramètres de la fonction `circ`

▼ Solution

```
ballx=60
bally=60

def TIC():
    global ballx,bally
  circ(ballx,bally,3,5)
```

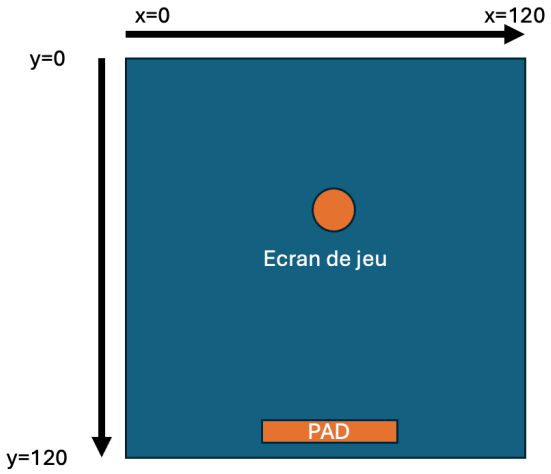
Exercice 2 : Faire bouger la balle

Pour faire bouger la balle, vous allez modifier les coordonnées du centre : `ballx` et `bally`

Pour obtenir une trajectoire en diagonal (comme sur un billard), il faut modifier à chaque fois `ballx` et `bally`

Tout d'abord essayez de faire bouger la balle en diagonal vers le haut et vers la droite en utilisant deux nouvelles variables `ballspeedx` et `ballspeedy`

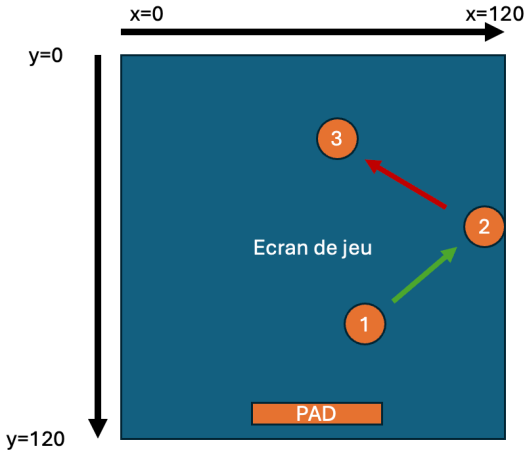
▼ Indice



Exercice 3 : Faire rebondir la balle

Pour faire rebondir la balle, vous devez inverser la direction de la balle en fonction de sa position à l'écran.

si la balle se retrouve en limite avec la zone de jeu alors on simule une collision en inversant la vitesse de la balle sur l'axe sur lequel se trouve la collision



La balle est dans la position 1 et se dirige vers la position 2 en suivant la trajectoire verte dans ce cas la `ballspeedx = 2`

Une fois arrivé au niveau de la bordure de la zone de jeu, sur notre schéma à la position `x=120` alors on inverse la vitesse de la balle et `ballspeedx = -2`

▼ Indice

Il faut penser à prendre en compte le rayon de la balle.

Vous avez 3 cas à gérer, la bordure du haut, la bordure de droite et la bordure de gauche.

Gérer le respawn et la collision avec le pad

Exercice 1 : Respawn

Lorsque la balle dépasse la limite de la bordure du bas de l'espace de jeu, elle continue à descendre pour ensuite disparaître.

Vous allez devoir faire en sorte de la balle "respawn" au point de départ dès que la balle sort de l'espace de jeu par le bas

▼ solution

```
def TIC():




 if bally+3 > 120 :
  ballx = 60
  bally = 60
```

Exercice 3 : gérer la collision avec le pad

Lorsque la balle se retrouve en collision avec le pad, vous allez devoir faire en sorte que la balle rebondisse sur le pad. Cette étape est importante car c'est à partir de ce moment la que votre jeu sera vraiment jouable

▼ Indice

Utilisez une triple condition qui utilise les variables `ballx` `bally` `padx` `pady` et `padw` pour gérer la collision

Ajouter une interface

Exercice 1 : le score

Avec la fonction `print` et la fonction `str` , vous allez afficher le score à l'écran, la méthode de calcul est simple : plus 10 points à chaque fois que la balle rebondit sur le pad

▼ Indice

Positionnez la fonction print à la fin de votre fonction `TIC()` pour que le score s'affiche au dessus de tous les autres élements

Créez une nouvelle variable `score` que vous allez incrémenter dans la condition qui permet de faire rebondir la balle

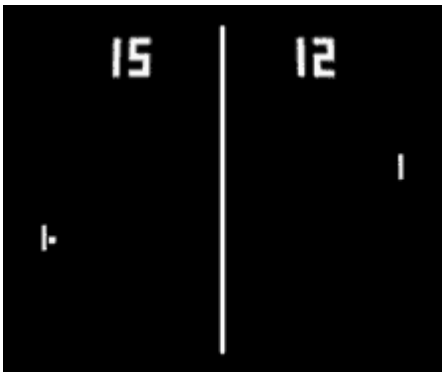
Exercice 2 : Life & game over

De la même manière que dans l'exercice précédent, créez un système de vie, avec 3 vies et un affichage dans l'interface en dessous du score. Lorsque le nombre de vie est égale à zéro alors replacez la balle à son point de départ, faites en sorte que la balle ne bouge plus et affichez "game over" au centre de l'écran

Pour aller plus loin

Vous pouvez ajouter des nouvelles fonctionnalités de votre jeu comme un écran de "high scores" qui s'affiche lors du game over comme dans les jeux retro et une option pour relancer le jeu pour tenter de battre "high scores"

Vous pouvez aussi créer un nouveau jeu en vous appuyant sur tout ce que vous avez appris dans ce coding club comme le célèbre jeu PONG à deux joueurs et pourquoi pas implémenter une IA pour jouer contre vous



Crédits

Cet atelier a été écrit et testé à Epitech Montpellier 💙 pour le Coding Club

Merci à Pico-8 Fanzine #1 pour l'inspiration grâce à ses tutos de création de jeux sur PICO-8 (la grande soeur de TIC-80)